NASA/CP-1999-209339

Advanced Training Technologies and Learning Environments

Compiled by
Ahmed K. Noor
University of Virginia
Center for Advanced Computational Technology, Hampton, Virginia

John B. Malone
Langley Research Center, Hampton, Virginia

Proceedings of a workshop sponsored by the National Aeronautics and Space Administration and the University of Virginia, Center for Advanced Computational Technology, Hampton, Virginia, and held at NASA Langley Research Center, Hampton, Virginia
March 9-10, 1999

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

July 1999

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 605-6000

Preface

This document contains the proceedings of the Workshop on Advanced Training Technologies and Learning Environments, held at NASA Langley Research Center, Hampton, Virginia, March 9-10, 1999. The workshop was jointly sponsored by the University of Virginia's Center for Advanced Computational Technology and NASA. Workshop attendees came from NASA, other government agencies, industry, and universities. The objective of the workshop was to assess the status and effectiveness of different advanced training technologies and learning environments.

Certain materials and products are identified in this publication in order to specify adequately the materials and products that were investigated in the research effort. In no case does such identification imply recommendation or endorsement of products by NASA, nor does it imply that the materials and products are the only ones or the best ones available for this purpose. In many cases equivalent materials and products are available and would probably produce equivalent results.

Ahmed K. Noor
Center for Advanced Computational Technology
University of Virginia
Hampton, Virginia

John B. Malone
NASA Langley Research Center
Hampton, Virginia

Contents

Attendees

1. Mr. Paul R. Aaron
U.S. Department of Energy
EM-14, 1E-278 #6
1000 Independence Avenue, S.W.
Washington, D.C. 20585
(202) 586-0847; Fax (202) 586-8862
Email: paul.aaron@em.doe.gov

2. Mr. Salvatore M. Abate
Pathlore Software Corporation
7965 North High Street, Suite 300
Columbus, OH 43235
(614) 781-7223; Fax (614) 781-7200
Email: sabate@pathlore.com

3. Mr. Colin H. Albaugh
Betac Corporation
HQ ACC/DIFM
129 Andrews St., Ste 150
Langley AFB, VA 23665
(757) 764-5031; Fax (757) 764-8516
Email: colin.albaugh@langley.af.mil

4. Mr. George D. Allison
Mail Stop 309
NASA Langley Research Center
Hampton, VA 23681
(757) 864-2594; Fax (757) 864-8543
Email: g.d.allison@larc.nasa.gov

5. Dr. Scott R. Angster
Analytical Mechanics Associates
17 Research Drive
Hampton, VA 23666
(757) 865-0944
Email: angster@ama-inc.com

6. Mr. Robert P. Armstrong
Syscom, Inc.
400 East Pratt Street, 3rd Floor
Baltimore, MD 21202
(410) 828-0947; Fax (410) 539-7774
Email: rarmstrong@syscom-inc.com

7. Mr. David S. Bell
CBT Systems
16100 N. Greenway Hayden Loop
Suite 800
Scottsdale, AZ 85260
(602) 609-2622; Fax (888) 395-0015
Email: david_bell@cbtsys.com

8. Mr. Richard L. Bulle
Mail Stop 157D
NASA Langley Research Center
Hampton, VA 23681
(757) 864-6838
Email: r.l.bulle@larc.nasa.gov

9. Ms. Susan Davis
Regional Manager
ComputerPREP, Inc.
2121 Eisenhower Avenue
Alexandria, VA 22314
(703) 548-3063; Fax (703) 548-3093
Email: sdavis@worldnet.att.net

10. Mr. Gary D. DeLaney
FTSCLANT, Code 4313
9727 Avionics Loop
Norfolk, VA 23511
(757) 444-3872, ext. 1214
Fax (757) 444-2799
Email:
Doug_DeLaney@ftsclant.navy.mil

11. Mr. LeRoy Dennison
Mail Stop 157B
NASA Langley Research Center
Hampton, VA 23681
(757) 864-7432; Fax (757) 864-8888
Email: d.l.dennison@larc.nasa.gov

12. Mr. Charles D. Dietrich
Betac Corporation
129 Andrews Street
Langley AFB, VA 23185
(757) 764-8619; Fax (757) 764-8516
Email: charles.dietrich@langley.af.mil

13. Dr. Haldun Direskeneli
Mail Stop 157D
NASA Langley Research Center
Hampton, VA 23681
(757) 864-8890; Fax (757) 864-8807
Email: h.direskeneli@larc.nasa.gov

14. Mr. Philip V. W. Dodds
Randall House Associates, Inc.
9 Randall Court
Anapolis, MD 21401
(410) 626-1623
Email: pdodds@rhassociates.com

15. Mr. Bryan E. Downs
Betac Corporation
2101 Executive Drive
7th Floor South, Tower Box 69
Hampton, VA 23666
(757) 826-9033; Fax (757) 827-0031
Email: downs@betac.com

16. Mr. Kimball Fisher
The Fisher Group, Inc.
12550 S.W. Butner Road
Beaverton, OR 97005
(503) 641-9316; Fax (503) 641-5235
Email: fshgrp@transport.com

17. Dr. J. Dexter Fletcher
Institute for Defense Analyses
1801 North Beauregard Street
Alexandria, VA 22311-1772
(703) 578-2837; Fax (703) 931-7792
Email: fletcher@ida.org

18. Mr. Peter G. Gammon
Lotus Development
230 Lexington Green Circle, Suite 600
Lexington, KY 40503
(606) 425-3524; Fax (606) 425-3528
Email: pete_gammon@lotus.com

19. Ms. Thea M. Ganoe
Center for Advanced Computational Technology/Univ. of Virginia
Mail Stop 201
NASA Langley Research Center
Hampton, VA 23681
(757) 864-8391; Fax (757) 864-8089
Email: t.m.ganoe@larc.nasa.gov

20. Mr. Geoffrey A. Gates
ExecuTrain
2101 Executive Drive
Tower Box 62
Hampton, VA 23666
(757) 825-1144; Fax (757) 825-1155
Email: geoffg@eva.executrain.com

21. Dr. Andrew S. Gibbons
Utah State University
2830 Old Main Hill
Logan, UT 84322-2830
(435) 797-2393; Fax (435) 797-2693
Email: gibbons@cc.usu.edu

22. Mr. Charles C. Gilbert
United States Air Force
HQ ACC/DIFM
129 Andrews St., Ste 150
Langley AFB, VA 23665
(757) 764-5031; Fax (757) 764-8516
Email: charles.gilbert@langley.af.mil

23. Dr. Stephen L. Goldberg
Chief, Simulator Systems Res. Unit
U.S. Army Research Institute
12350 Research Parkway
Orlando, FL 32826
(407) 384-3980; Fax (407) 384-3999
Email:
stephen_goldberg@stricom.army.mil

24. Mr. Joseph P. Hale
NASA Headquarters
Code AF
300 E Street, S.W.
Washington, D.C. 20546
(202) 358-4529; Fax (202) 358-2920
Email: jhale@hq.nasa.gov

25. Ms. Nancy Harvin
Virginia Quality Institute
3221A North Armistead Avenue
Hampton, VA 23666
(757) 865-4894; Fax (757) 865-4893
Email: tqinet@aol.com

26. Dr. Michael L. Heck
Research Triangle Institute
One Enterprise Parkway, Suite 310
Hampton, VA 23666
(757) 827-8450; Fax (757) 827-3273
Email: mlheck@rti.org

27. Ms. Molly Herceg
Pathlore Software Corporation
7965 North High Street, Suite 300
Columbus, OH 43235
(614) 781-0036; Fax (614) 781-7200
Email: mherceg@pathlore.com

28. Dr. Robert F. Hodson
JDH Technologies
11834 Canon Blvd.
Newport News, VA 23606
(757) 873-4747
Email: rhodson@jdhtech.com

29. Ms. Carol D. Horwitz
Air Force Research Laboratory
Information Training Branch
2509 Kennedy Circle
Brooks AFB, TX 78235
(210) 536-2764; (210) 536-2426
Email: cdh@bigbird.brooks.af.mil

30. Dr. W. Lewis Johnson
Director, Center for Advanced Research in Technology for Education
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292
(310) 822-1511; Fax (310) 822-0751
Email: johnson@isi.edu

31. Mr. Donald A. Jones
U.S. Navy
Fleet Technical Support Ctr, Atlantic
9727 Avionics Loop
Norfolk, VA 23511
(757) 444-3872, ext. 1200
Fax (757) 444-2799
Email:
donald_jones@ftsclant.navy.mil

32. Mr. Tom Kahl
CBT Systems USA
8201 Greensboro Drive, Ste 1000
McLean, VA 22102
(703) 610-6877; Fax (703) 714-6926
Email: tkahl@cbtsys.com

33. Ms. Thomasine J. Kashin
James City County Government
Information Center
101B Mounts Bay Road
Williamsburg, VA 23185
(757) 253-6659; Fax (757) 253-6663
Email: tjkashin@james-city.va.us

34. Mr. Alex Kass
The Institute for the Learning Sciences
1890 Maple Avenue
Evanston, IL 60201
(847) 467-1003; Fax (847) 491-5258
Email: a-kass@nwu.edu

35. Ms. Kris Kirkwood
Betac Corporation
2101 Executive Drive, Tower Box 69
Hampton, VA 23666
(757) 826-9033, ext. 364
Fax (757) 827-0031

36. Mr. Robert L. Kudlinski
Mail Stop 157
NASA Langley Research Center
Hampton, VA 23681
(757) 864-6707; Fax (757) 864-7635
Email: r.a.kudlinski@larc.nasa.gov

37. Prof. Alan M. Lesgold
Executive Associate Director
Learning Research & Development Ctr
University of Pittsburgh
3939 O'Hara Street, Room 559
Pittsburgh, PA 15260
(412) 624-7045; Fax (412) 624-9149
Email: al@pitt.edu

38. Mr. Harris L. Liebergot
National Institute of Standards and
Technology
Admin. Bldg., Rm A321
Gaithersburg, MD 20899
(301) 975-5196; Fax (301) 926-9524
Email: liebergot@mailserver3.nist.gov

39. Prof. R. Bowen Loftin
Department of Computer Science
Mail Stop CSC-3475
University of Houston
Houston, TX 77204-3475
(713) 743-1238; Fax (713) 743-1198
Email: bowen@uh.edu

40. Dr. James M. Luckring
Mail Stop 280
NASA Langley Research Center
Hampton, VA 23681
(757) 864-2869
Email: j.m.luckring@larc.nasa.gov

41. Mr. Alan R. McCoy
Center for Advanced Computational
Technology/Univ. of Virginia
Mail Stop 201
NASA Langley Research Center
Hampton, VA 23681
(757) 864-8518; Fax (757) 864-8089
Email: a.r.mccoy@larc.nasa.gov

42. Dr. Janet M. McKenzie
Mail Stop 309
NASA Langley Research Center
Hampton, VA 23681
(757) 864-2585; Fax (757) 864-8543
Email: j.m.mckenzie@larc.nasa.gov

43. Ms. Fran D. McLemore
LaRC DAAC User Service/Outreach
NASA Langley Research Center
Mail Stop 157A
Hampton, VA 23681
(757) 864-7430; Fax (757) 864-8807
Email: f.d.mclemore@larc.nasa.gov

44. Ms. Mala Mehrotra
Pragati Synergetic Research, Inc.
145 Stonelake Court
Yorktown, VA 23693
(757) 865-7080; Fax (757) 766-8892
Email: mm@infi.net

45. Mr. Raul E. Mejia
Mail Stop 158
NASA Langley Research Center
Hampton, VA 23681
(757) 864-3485; Fax (757) 864-8342
Email: r.e.mejia@larc.nasa.gov

46. Dr. Allen Munro
Director, Behavioral Technology Lab.
University of Southern California
250 North Harbor Drive, Suite 309
Redondo Beach, CA 90277
(310) 379-0844; Fax (310) 372-7788
Email: munro@usc.edu

47. Ms. E. Rose W. Mustain
Mail Stop 309
NASA Langley Research Center
Hampton, VA 23681
(757) 864-2571; Fax (757) 864-8543
Email: e.r.mustain@larc.nasa.gov

48. Prof. Ahmed K. Noor
Director, Center for Advanced Computational Technology
University of Virginia
Mail Stop 201
NASA Langley Research Center
Hampton, VA 23681
(757) 864-1978; Fax (757) 864-8089
Email: a.k.noor@larc.nasa.gov

49. Mr. James C. Ong
Stottler Henke Associates, Inc.
1660 South Amphlett Blvd., Suite 350
San Mateo, CA 94402
(650) 655-7242; Fax (650) 655-7243
Email: ong@shai.com

50. Ms. Juliet Z. Pao
Mail Stop 124
NASA Langley Research Center
Hampton, VA 23681
(757) 864-7328; Fax (757) 864-7605
Email: j.z.pao@larc.nasa.gov

51. Dr. F. G. Patterson, Jr.
NASA Headquarters
Code FT
300 E Street, S.W.
Washington, D.C. 20546
(202) 358-2171; Fax (757) 358-4164
Email: pat.patterson@hq.nasa.gov

52. Mr. Rob Perry
Vice President of Federal Sales
Pathlore Software Corporation
7965 North High Street, Suite 300
Columbus, OH 43235
(614) 781-0036; Fax (614) 781-7245
Email: rperry@pathlore.com

53. Ms. Jeanne M. Peters
Center for Advanced Computational Technology/Univ. of Virginia
Mail Stop 201
NASA Langley Research Center
Hampton, VA 23681
(757) 864-1989; Fax (757) 864-8089
Email: j.m.peters@larc.nasa.gov

54. Mr. Raymond S. Pettit
Analytical Mechanics Associates
Mail Stop 328
NASA Langley Research Center
Hampton, VA 23681
(757) 864-9748; Fax (757) 864-1975
Email: r.s.pettit@larc.nasa.gov

55. Mr. Joseph R. Pilkington
NASA Code FT/Tony Maturo
7700 Leesburg Pike, Suite 108
Falls Church, VA 22043
(703) 749-3040; Fax (703) 749-3046
Email: jpilkington@sri-hq.com

56. Mr. Gary Randall
Virginia Quality Institute
3221A North Armistead Avenue
Hampton, VA 23666
(757) 865-4894; Fax (757) 865-4893
Email: tqinet@aol.com

57. Dr. J. Wesley Regian
Air Force Research Laboratory/HEJT
7909 Lindbergh Drive
Brooks AFB, TX 78235-5352
(210) 536-3908; Fax (210) 536-2426
Email: regian@alhrt.brooks.af.mil

58. Prof. Jeff W. Rickel
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292
(310) 822-1511, ext. 124
Fax (310) 822-0751
Email: rickel@isi.edu

59. Ms. Lillian C. Ross
NASA Langley Research Center
Mail Stop 309
Hampton, VA 23681
(757) 864-2657; Fax (757) 864-8543
Email: l.c.ross@larc.nasa.gov

60. Mr. Paul Russo
Syscom, Inc.
400 East Pratt Street, 3rd Floor
Baltimore, MD 21202
(410) 539-3737; Fax (410) 539-7774

61. Mr. Paul C. Schutte
Mail Stop 152
NASA Langley Research Center
Hampton, VA 23681
(757) 864-2019; Fax (757) 864-7793
Email: p.c.schutte@larc.nasa.gov

62. Mr. Willard J. Scott
CBT Systems USA
8201 Greensboro Drive, Ste 1000
McLean, VA 22102
(703) 758-8166; Fax (703) 758-9891
Email: wscott@cbtsys.com

63. Ms. Nichole R. Silvers
LaRC DAAC User Service/Outreach
NASA Langley Research Center
Mail Stop 157D
Hampton, VA 23681
(757) 864-9128; Fax (757) 864-8807
Email: n.r.silvers@larc.nasa.gov

64. Mr. Patrick A. Simpkins
NASA
John F. Kennedy Space Center
HM0A
Kennedy Space Center, FL 32899
(407) 867-3365; Fax (407) 867-8259
Email:
patrick.simpkins-1@ksc.nasa.gov

65. Mr. Randy J. Stiles
Lockheed Martin Advanced
Technology Center
O/H143, B/255
3251 Hanover Street
Palo Alto, CA 94304-1191
(650) 354-5256; Fax (650) 354-5235
Email: randy.stiles@lmco.com

66. Dr. Douglas M. Towne
Behavioral Technology Laboratories
University of Southern California
1120 Pope Street, Suite 201C
St. Helena, CA 94574
(707) 963-3060; Fax (707) 963-3976
Email: dtowne@usc.edu

67. Dr. Jay Vaughan
Carley Corporation
3203 Lawton Road
Orlando, FL 32808
(407) 647-6194; Fax (407) 895-3865
Email: von@innocent.com

68. Ms. Charlene Welch
Mail Stop 157D
NASA Langley Research Center
Hampton, VA 23681
(757) 864-7431; Fax (757) 864-8807
Email: charlene.welch@larc.nasa.gov

69. Mr. Jeffrey B. Welch
Naval Aviation Department
MCAS Cherry Point
Naval Aviation Depot
H43ISST.3
Cherry Point, NC 28533
(252) 464-5607
Email: jbwelch@navair.navy.mil

70. Mr. Bill L. Williams
Virginia Quality Institute
3221A North Armistead Avenue
Hampton, VA 23666
(757) 865-4894; Fax (757) 865-4893
Email: tqinet@aol.com

71. Mr. Dale R. Winchell
Betac Corporation
129 Andrews Street
Langley Air Force Base, VA 23665
(757) 764-8667; Fax (757) 764-2462
Email: dale.winchell@langley.af.mil

72. Mr. P. G. Wist
GP Learning Technologies
1221 Woodland Court
Hampstead, MD 21074
(410) 239-9794; Fax (410) 239-9795
Email: pwist@genphysics.com

386718

Pathway to the Future of Learning

34 p.

Ahmed K. Noor
Center for Advanced Computational Technology
University of Virginia
Hampton, VA

Pathway to the Future of Learning

Ahmed K. Noor
Center for Advanced Computational Technology
University of Virginia
NASA Langley Research Center
Hampton, VA 23681

Outline

The convergence of computing, communication and information technologies is reshaping relationships among researchers and organizations and dramatically changing the way we work and the way we learn. This presentation provides an overview of training technologies and learning environments. The outline is given in Fig. 1. First, the goals of education and training are described. The instructional models and learning technologies appropriate for achieving different learning goals are identified. Second, a brief description is given of the evolution of learning technologies. Third, the forces driving a paradigm change in learning are identified, and the technologies that can lead to a revolutionary change in learning are listed. Fourth, different learning strategies are described along with three advanced learning environments. Finally, current government and non-government activities on advanced learning technologies and environments are listed.

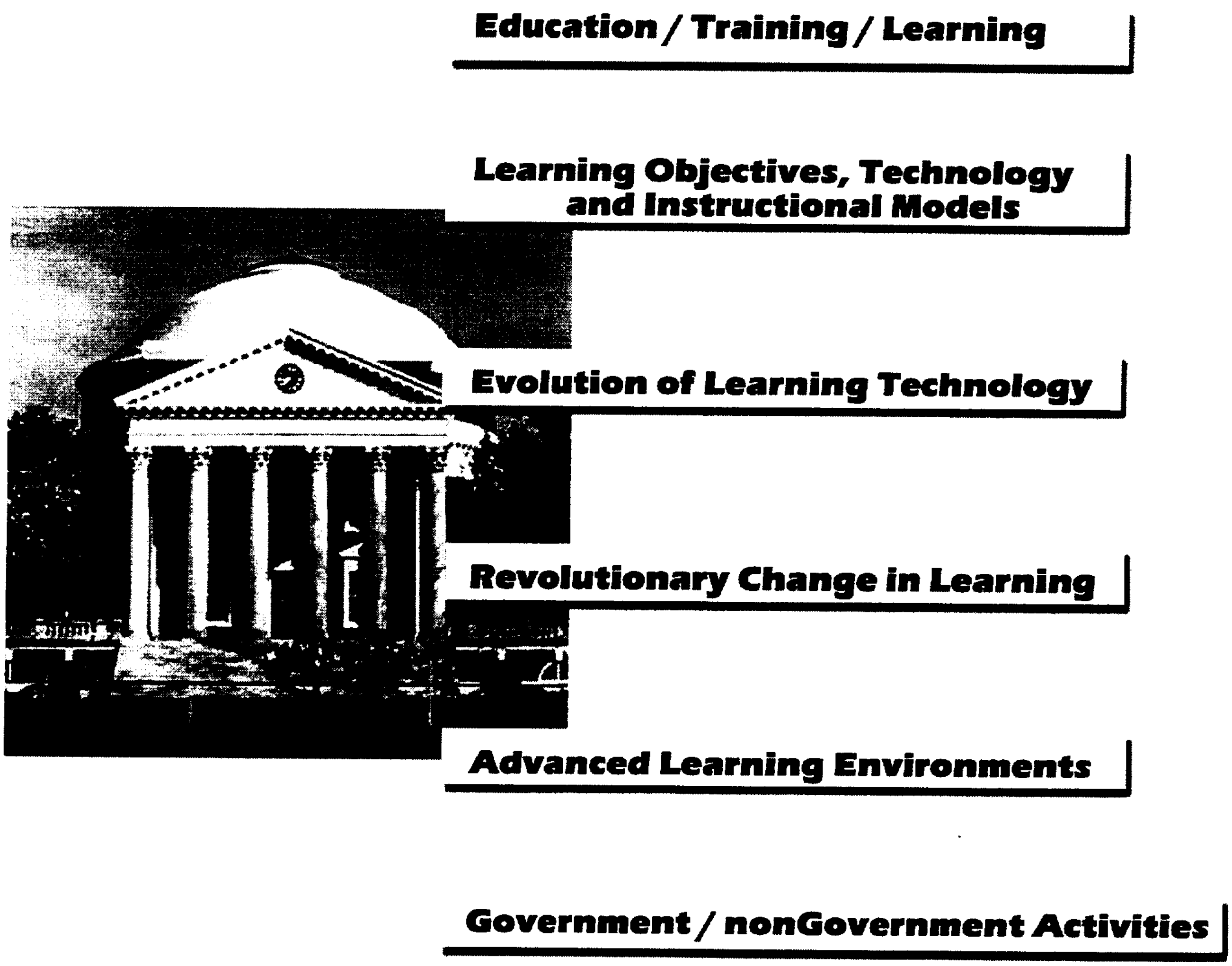


Figure 1

Education, Training and Learning

There has long been a philosophical gap between education and training. The goal of education was to impart high-level cognitive skills that would underpin lifelong learning. The goal of training was to bring performance up to a level that would let people successfully achieve tasks. Recently, however, we have seen training begin to emphasize the skills involved in lifelong learning, as evidenced by continual-growth workshops and online training facilities on the internet. In a sense, both education and training objectives fit in the larger classification of learning objectives (see Fig. 2).

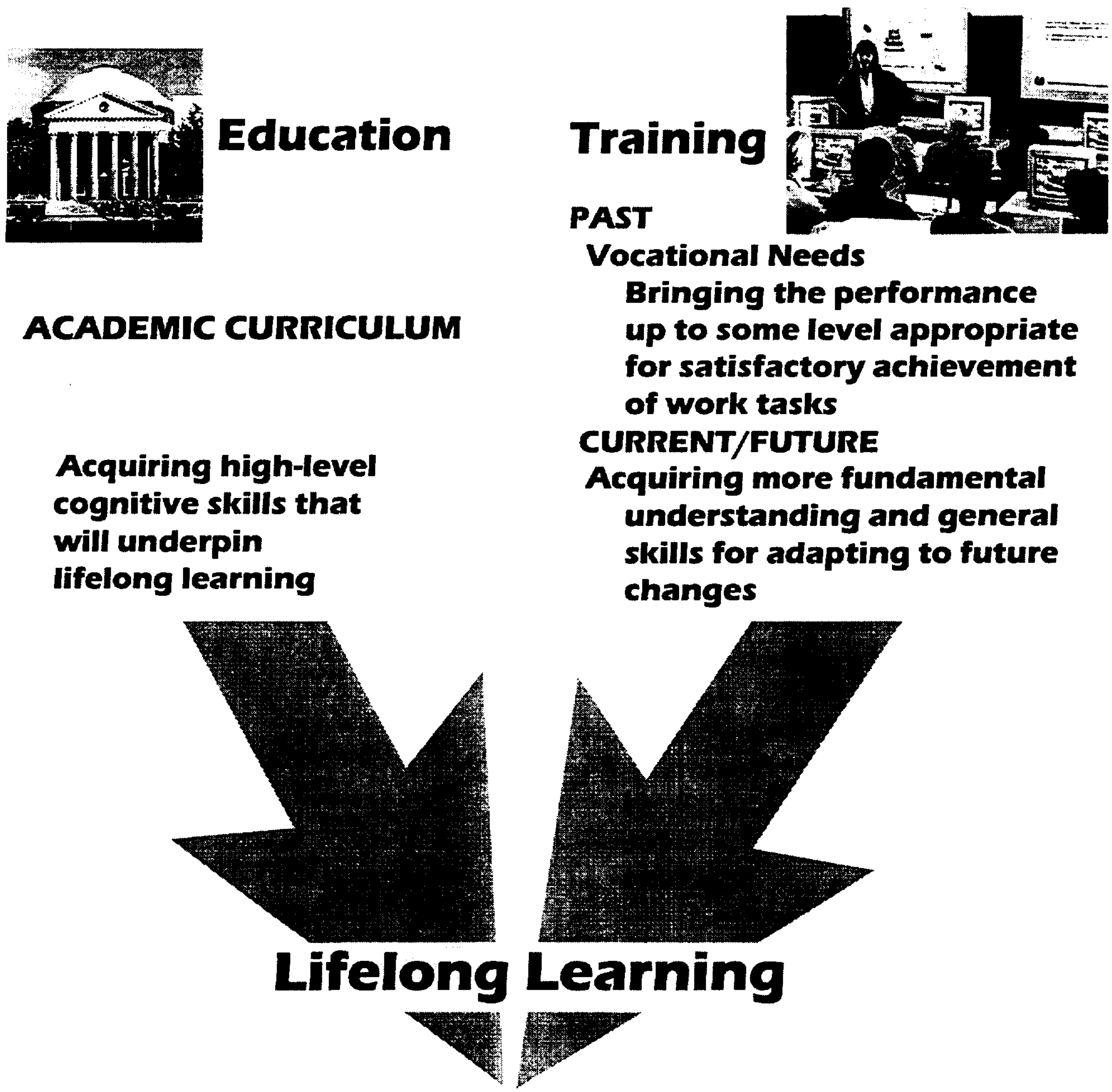


Figure 2

Learning Objectives, Instructional Models and Technologies

The desired outcome of learning can range from information transfer to skill and knowledge acquisition to the more ambitious goal of development of critical thinking and creativity skills. The instructional model and method used for accomplishing these goals vary from instructor-centered, learner-centered to learning-team centered. In the learner-centered model the learner is at the center of the learning process, and calls on many information sources. Learning-team centered models include virtual campuses and web-based distance learning models. The technologies employed in the three models are distribution, interactive and collaborative technologies, respectively (Fig. 3).

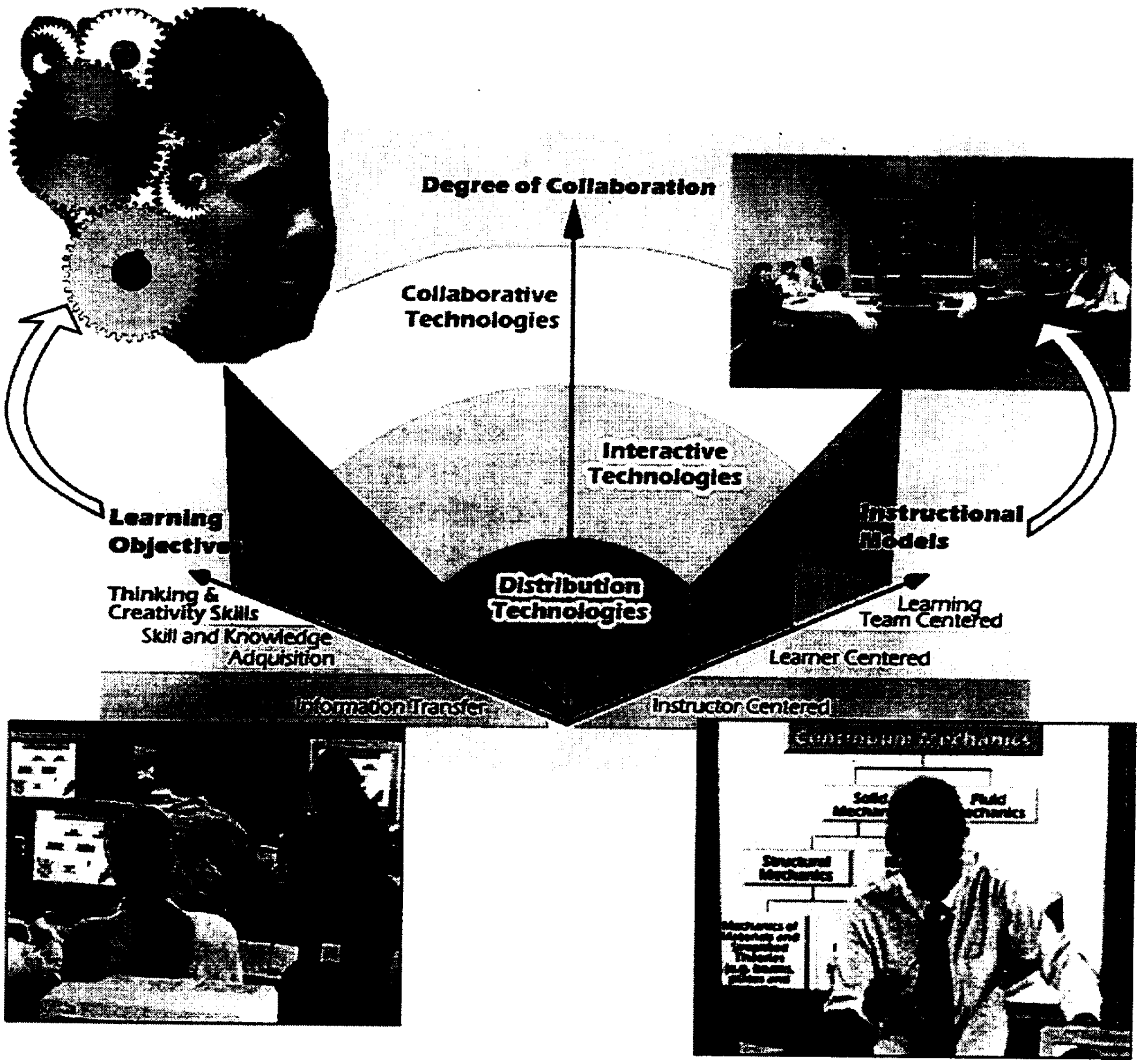


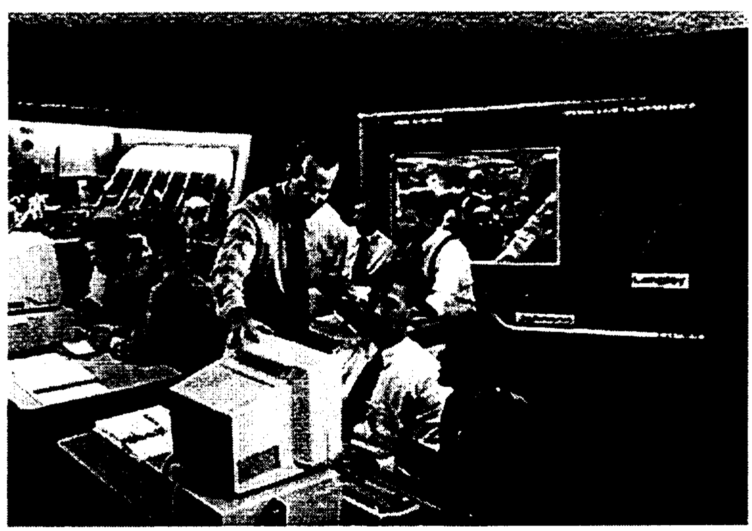
Figure 3

Role of Instructor

The instructors have traditionally been information transmitters. Their role in the future will change to that of course managers, coaches, facilitators and guides. They will inspire and motivate the learners, and not merely transmit information (Fig. 4).

A course manager

A coach, facilitator, guide



Inspires, motivates ...not merely an information transmitter

Figure 4

Evolution of Learning Technology

Figure 5 shows the evolution of learning technology, the level of sophistication of different systems and the extent of using intelligent agents in them. Computer-based technology (CBT) systems of the 1960's and 1970's were initially passive. Later developments in that period included learner modeling and more elaborate computer-learner interfaces. The addition of expert systems to CBT resulted in Intelligent Tutoring Systems (ITS) of the 1980's. ITS had explicit models of tutoring and domain knowledge, and were more flexible in their response than CBT systems. However, they were developed for "information transfer" and were not change tolerant. The advent of intelligent agents, which enabled the learner to manipulate cognitive artifacts from several perspectives or viewpoints, led to the interactive learning systems (ILS) of the 1990's. Examples of these systems include online courses, interactive learning systems and intelligent evaluation facilities. The current trend is towards collaborative distributed learning systems extensively using intelligent agents in the learning environment. Such dynamic environments will provide the learners access to other ideas and concepts, allow them to express their viewpoints and incrementally adapt initial viewpoints to more informed and mastered concepts.

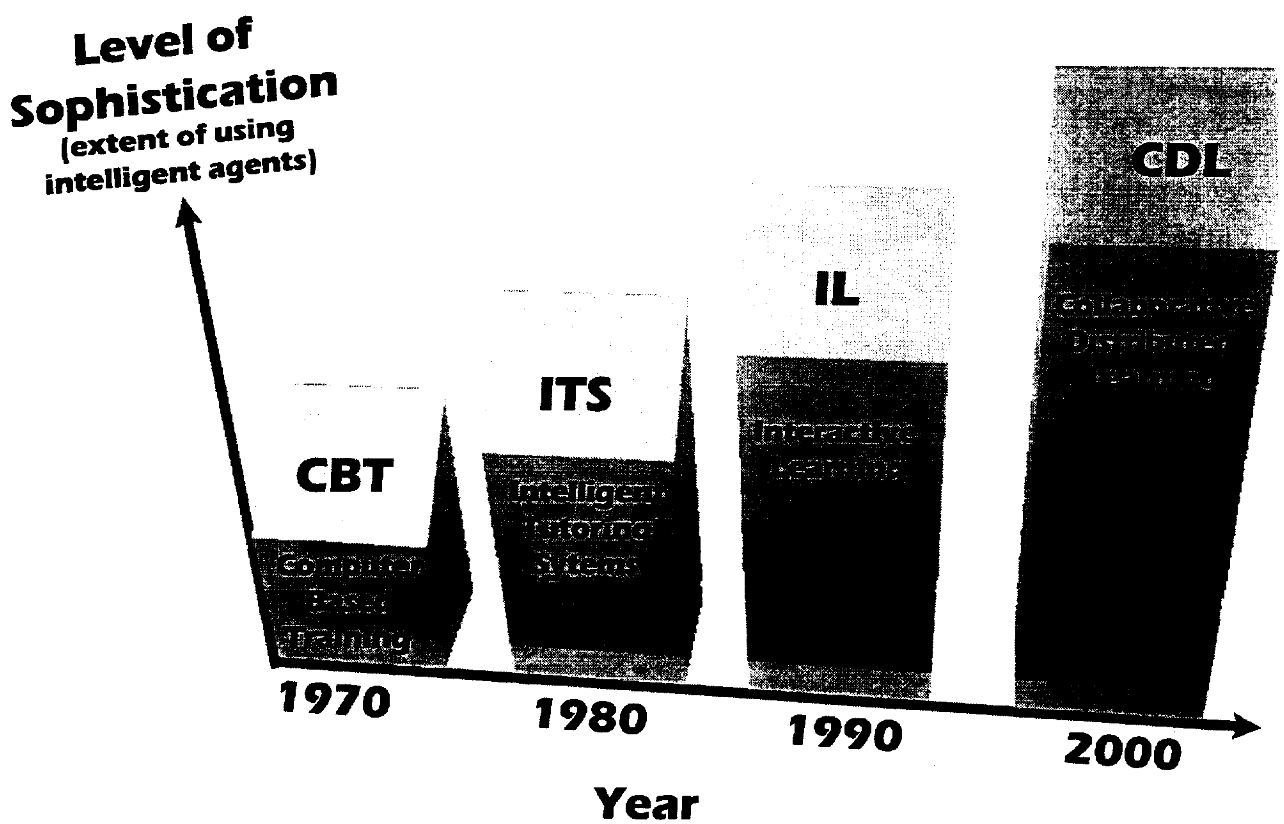


Figure 5

Electronic Learning Environment

Figure 6 shows the range of possibilities for the electronic environment generated by varying displacement in time and place.

The traditional electronic classroom corresponds to the "same time, same place" model. One may argue that it is the most effective environment for learning and communication because the instructor and learners have all mentally and physically committed themselves to a common window in time and space. The "different time, same place" model corresponds to electronic laboratories, which are useful when the lab resources are limited. Synchronous (real time) distance learning is associated with the "same time, different place" model. The learners can be distributed across the country, or around the world, but they must all tune in to the class at the same time. This model is not new. Instructional television has been used for some time. The richness of distance learning has increased dramatically over time to include two-way and multi-way videos so that instructors and learners at different locations can see each other.

Finally, the "different time, different place" model corresponds to a very versatile asynchronous distributed learning environment. It provides facilities for delivering instruction any time and anywhere including online courses, virtual classrooms, individualized and computer-assisted instruction, and intelligent tutoring systems.

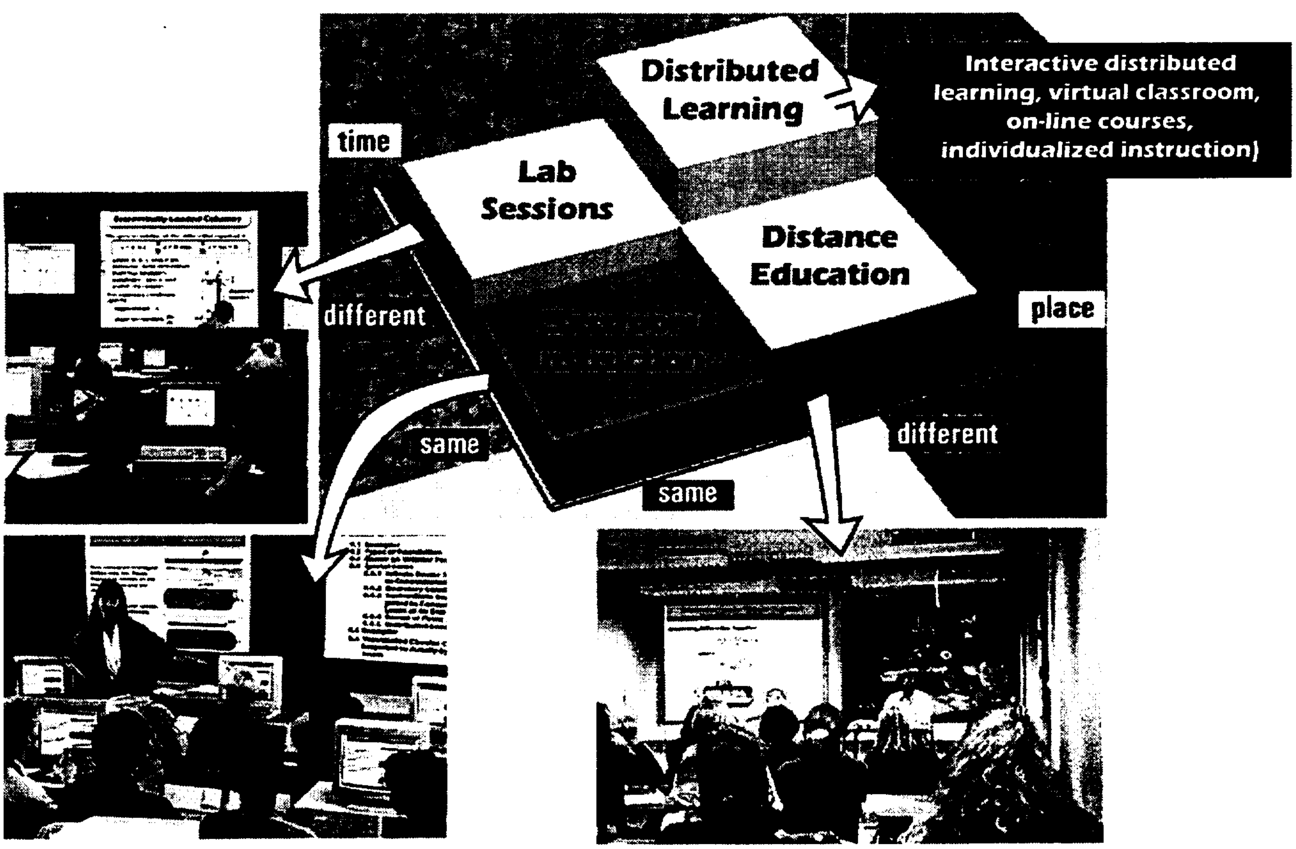


Figure 6

Online Courses and Virtual Classrooms

A number of innovative learning environments are in use today, including online courses, virtual classrooms, and even virtual universities that provide distributed networked learning environments with custom self-instructional courses, flexible tutorial support and control over both the place and time of learning (Fig. 7). Some of the new environments may be viewed as attempts to reinventing the university. The level of sophistication of the facilities used in these models is rapidly increasing.

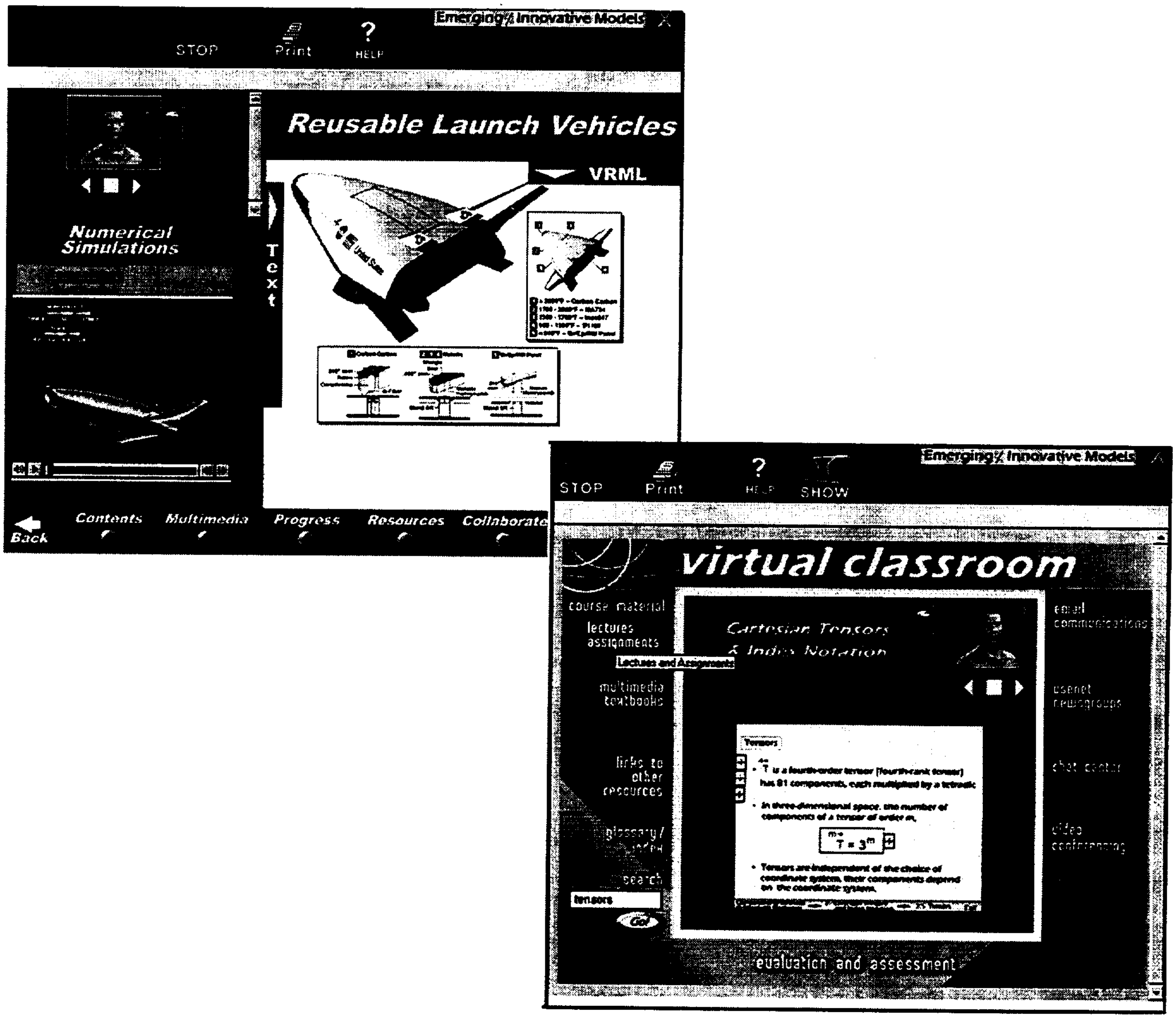


Figure 7

Forces Driving a Paradigm Change in Learning

There are four categories of forces that are driving a paradigm change in learning, namely (Fig. 8):

- *Developments in organizations and workplaces.* In the 1980's the focus of engineering organizations was on quality through reduction of defects and use of TQM models. In the 1990's the focus shifted to reengineering and streamlining the processes through the use of virtual product development (VPD) and enterprise resource planning (ERP) systems. As we move from the industrial to the knowledge era, engineering organizations will make radical changes in the workplaces, use electronic performance support systems (EPSS), and create virtual organizations with the overall goal being to achieve very high performance workplaces. The workplace will be transformed from the stationary offices centered around desktop computers and workstations into an intelligent networked environment that enable diverse geographically dispersed teams to collaborate in real time.
- *Economic pressures.* To remain competitive, engineering organizations will move from mass production to mass customization. In addition, the rapid changes in technology will result in continuously changing knowledge requirements of workers (continuous need for learning).
- *Learner demographics.* Because of the increasing number of older learners and the need for just-in-time training, flexible delivery systems have to be used.
- *Growing interdependence among technology, workplace and learning.* Technology will have a significant impact on the workers by providing just-in-time information and skills. It will change the workplace by making learning and work synonymous, and it will help in transforming engineering organizations into learning organizations.

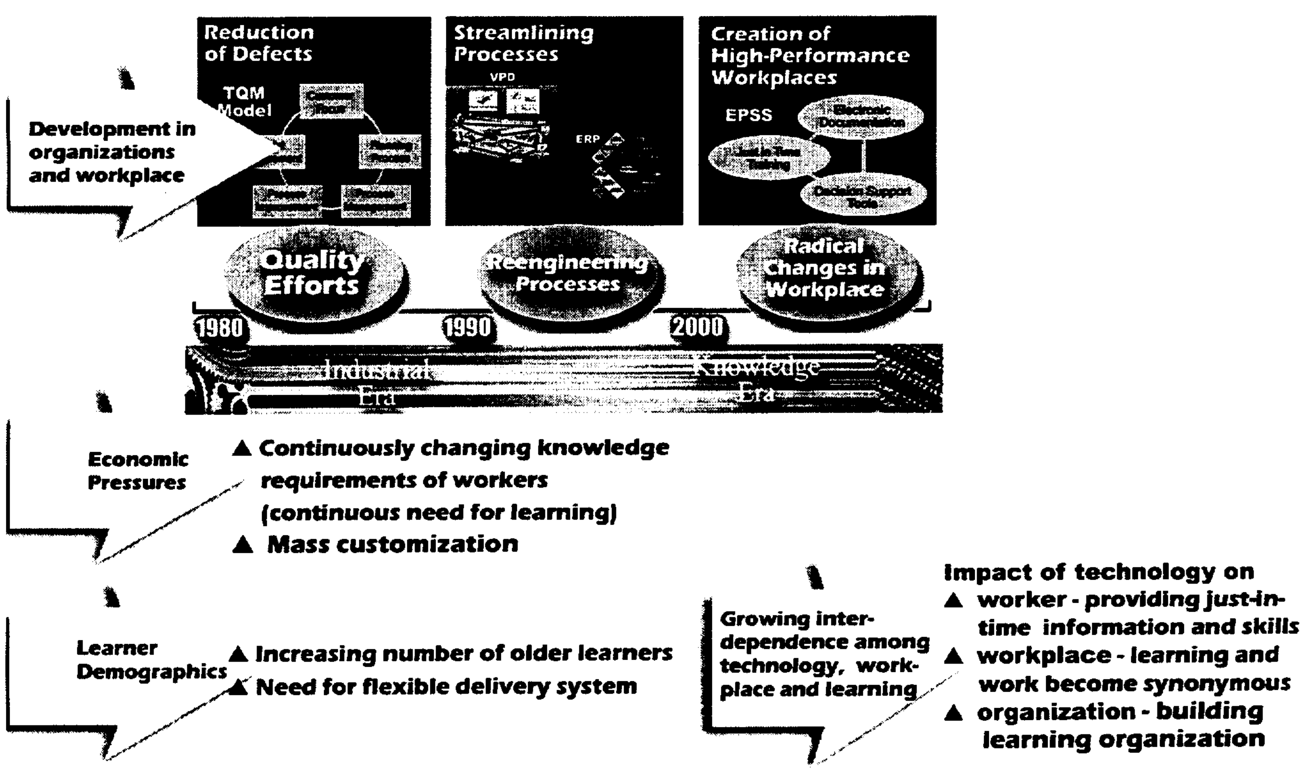


Figure 8

Curriculum Reform

Many of the current university programs and delivery systems are not likely to meet the needs of engineering organizations and workforces in the next century. The focus of the teaching mission of the university should be on the quests for meaning and relevance. The following changes should be made in the curriculum (Fig. 9):

- Emphasizing the distinction between knowledge and information (knowing how and knowing about).
- Providing flexible curricula (offering broader perspectives in dealing with complexity and uncertainty).
- Alignment of graduate educational offerings with the needs and interests of working professionals (special emphasis on certifying competence in selected areas).
- Transforming sequential classroom curriculum to nonlinear hyper-learning environment with flexible delivery system. In nonlinear instruction the learner selects his/her path to achieve the learning goal. Branching programs with remedial material are provided for self-instruction.

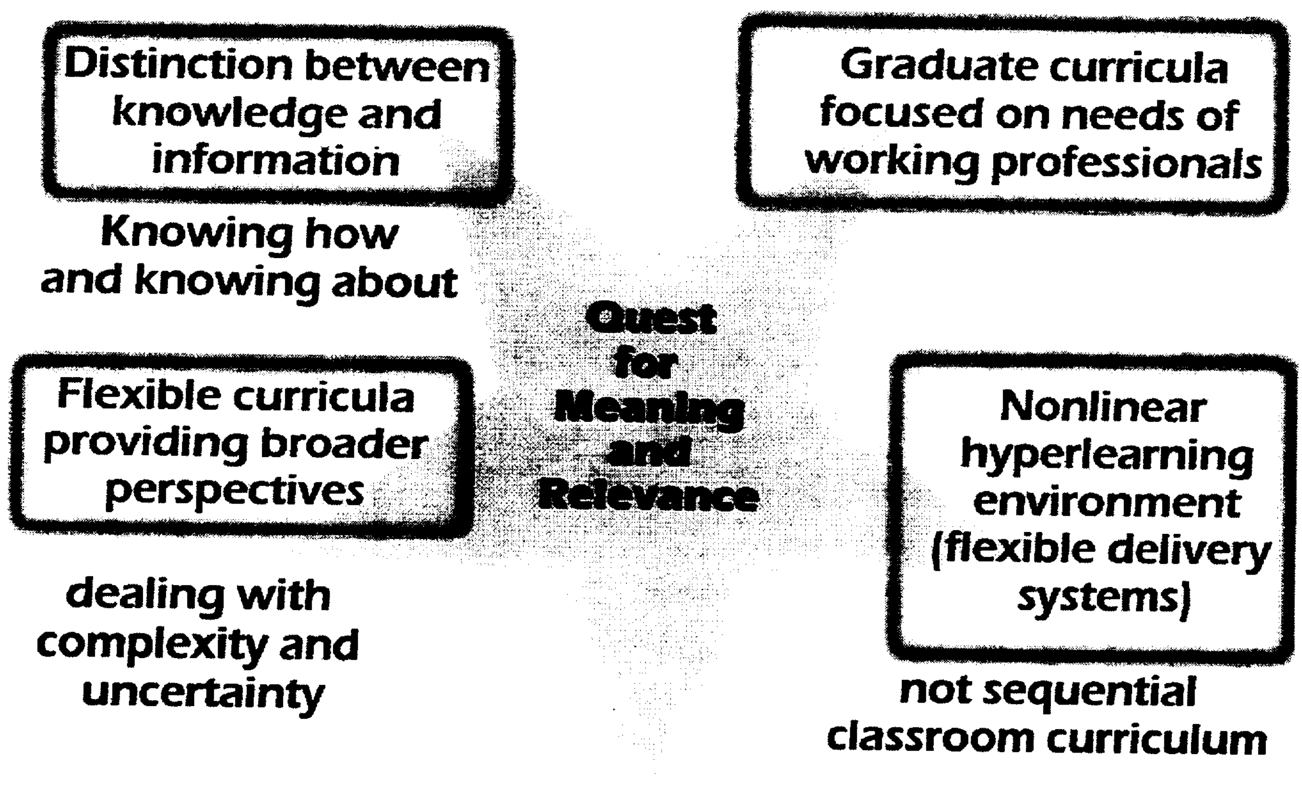


Figure 9

Knowledge Update Facilities

For universities to attract good students and compete with private for-profit businesses offering learning facilities, they need to think about novel concepts and arrangements in education. For example, providing knowledge update facilities for their graduates, free of charge, ten years after graduation (Fig.10).

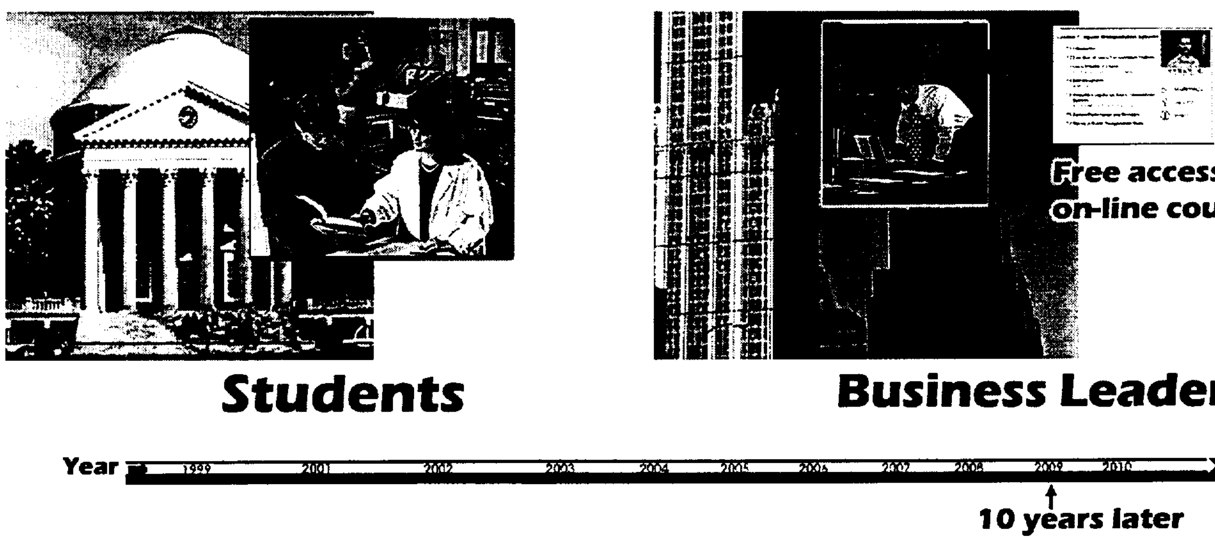


Figure 10

Relevant Technologies

The relevant technologies that can significantly impact learning include the following seven categories: high-performance computing; high-capacity communication and networking; information/knowledge; instructional; modeling, simulation and visualization; human performance; and human-computer interaction/communication (Fig. 11). Some of these technologies are described in the succeeding figures. The synergistic combination of these technologies can dramatically increase the effectiveness of learning. Learning can be independent of time and place, and available at all stages of a person's life. The learning context will be technologically rich. Learners will have access not only to a wide range of media, but also to a wide range of resources of education and training.

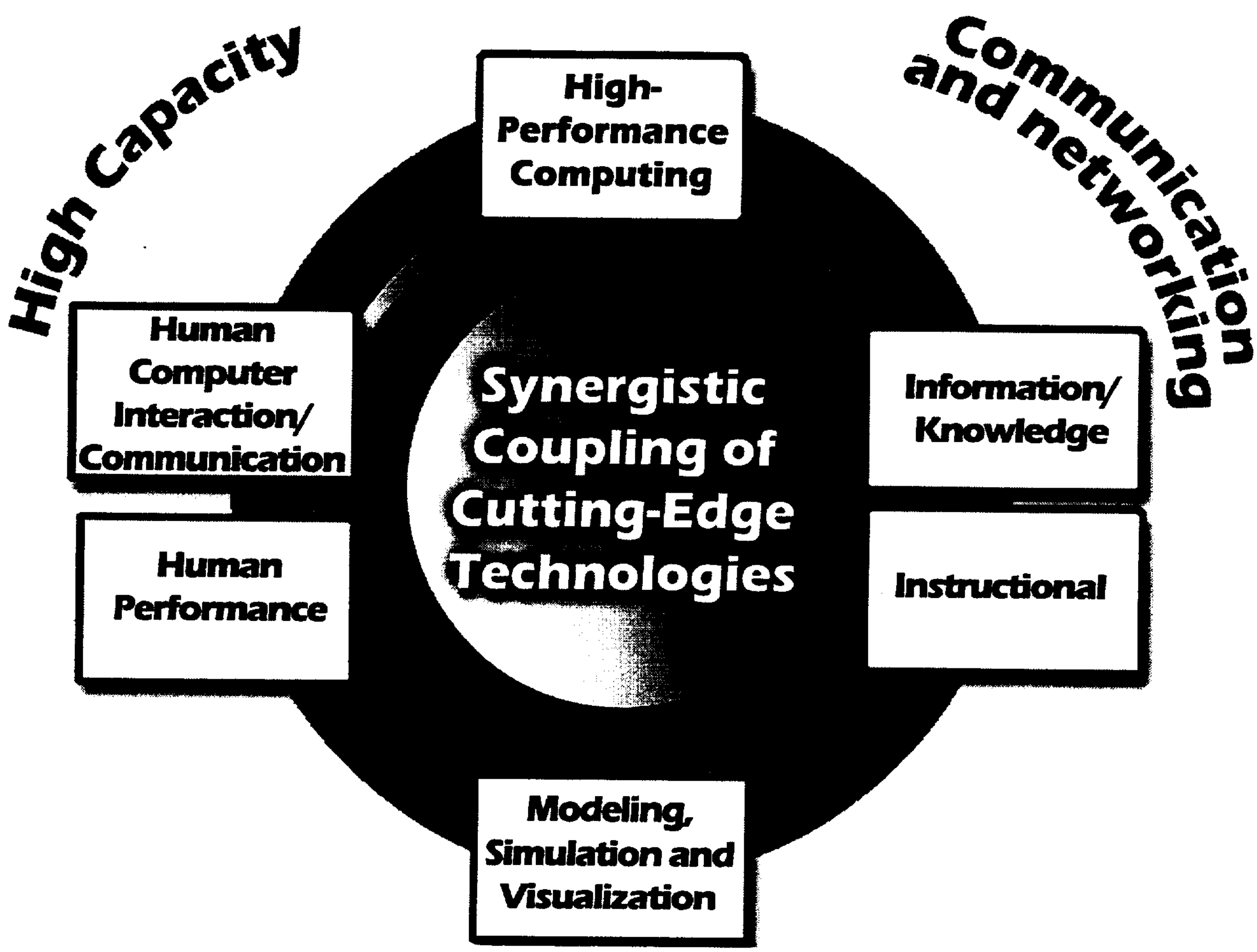


Figure 11

Ultrafast Computers, Powerful Microprocessors, Laptops And Wearable Computers

Computing technology covers a broad spectrum of technologies (Fig. 12) ranging from the ultrafast computers such as the three teraflop machines of the Department of Energy to the powerful microprocessors and visual computers, with motion video instruction (MVI) facilities such as those based on the new Intel and alpha processors. This superset of desktop multimedia PCs is based on Windows NT. They will have motion-video instruction, especially suited for image processing, three-dimensional animations, motion video, and visual communications. By the year 2001, the speed of the Intel and alpha processors will be over 1 GHz.

Laptops are becoming powerful (e.g., 400 MHz laptops are now available), so are wearable voice-activated computers that are especially suited for virtual environments.

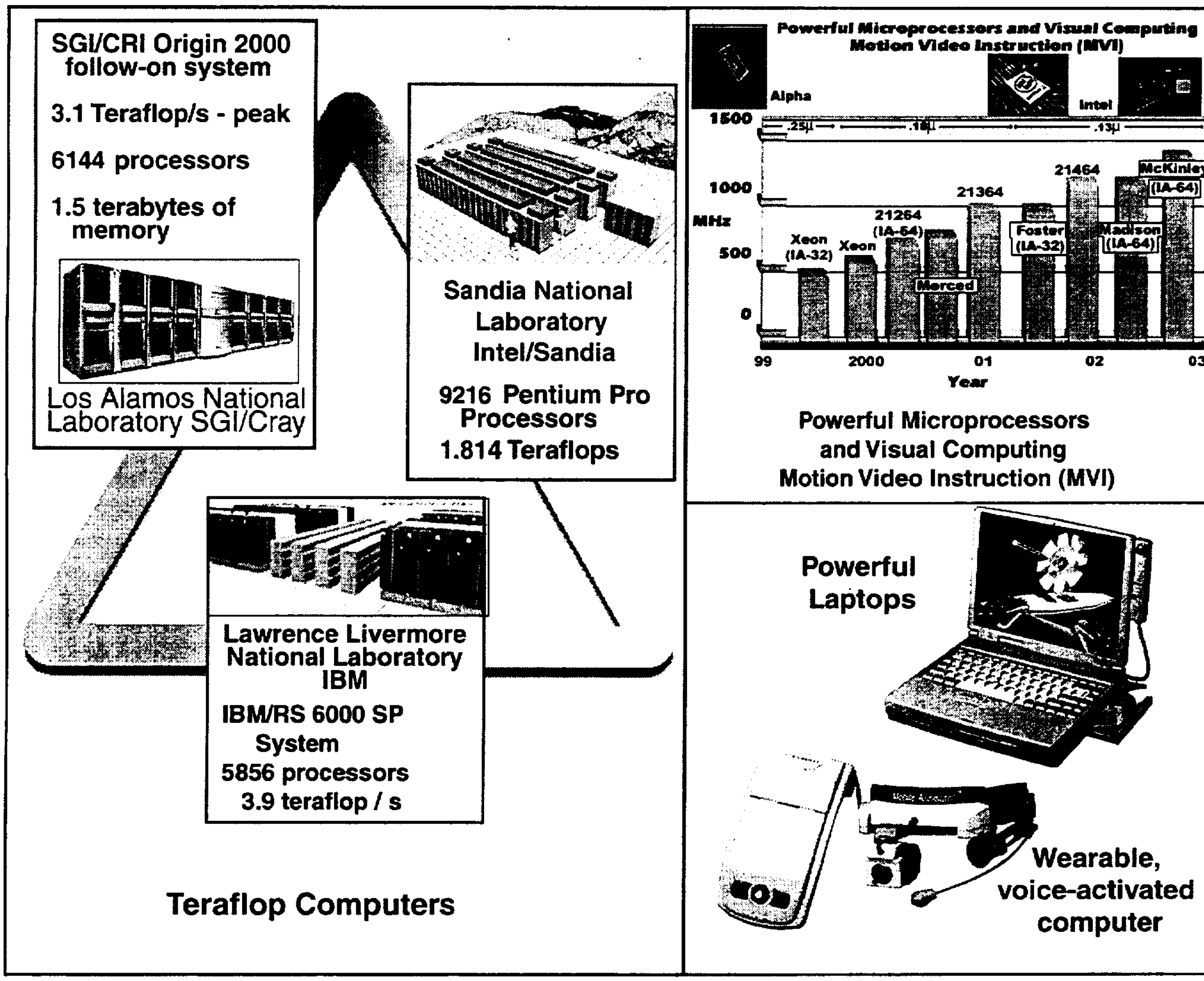


Figure 12

Future High Performance Networks

Future high-performance networks include Next-Generation Internet (NGI) and Internet 2 (Fig. 13). By leaping ahead with a new generation of network technologies, both networks significantly increase the capacity, cost and performance, and reduce the cost of communications. NGI is a multi-agency project sponsored by DARPA, SF, DOE, NASA and NIST. It is expected to increase the transmission capacity to ten terabits per second by the year 2005. As of May 1999, a consortium of 154 universities, 27 affiliated nonprofit organizations, and 45 corporate members supports Internet 2.

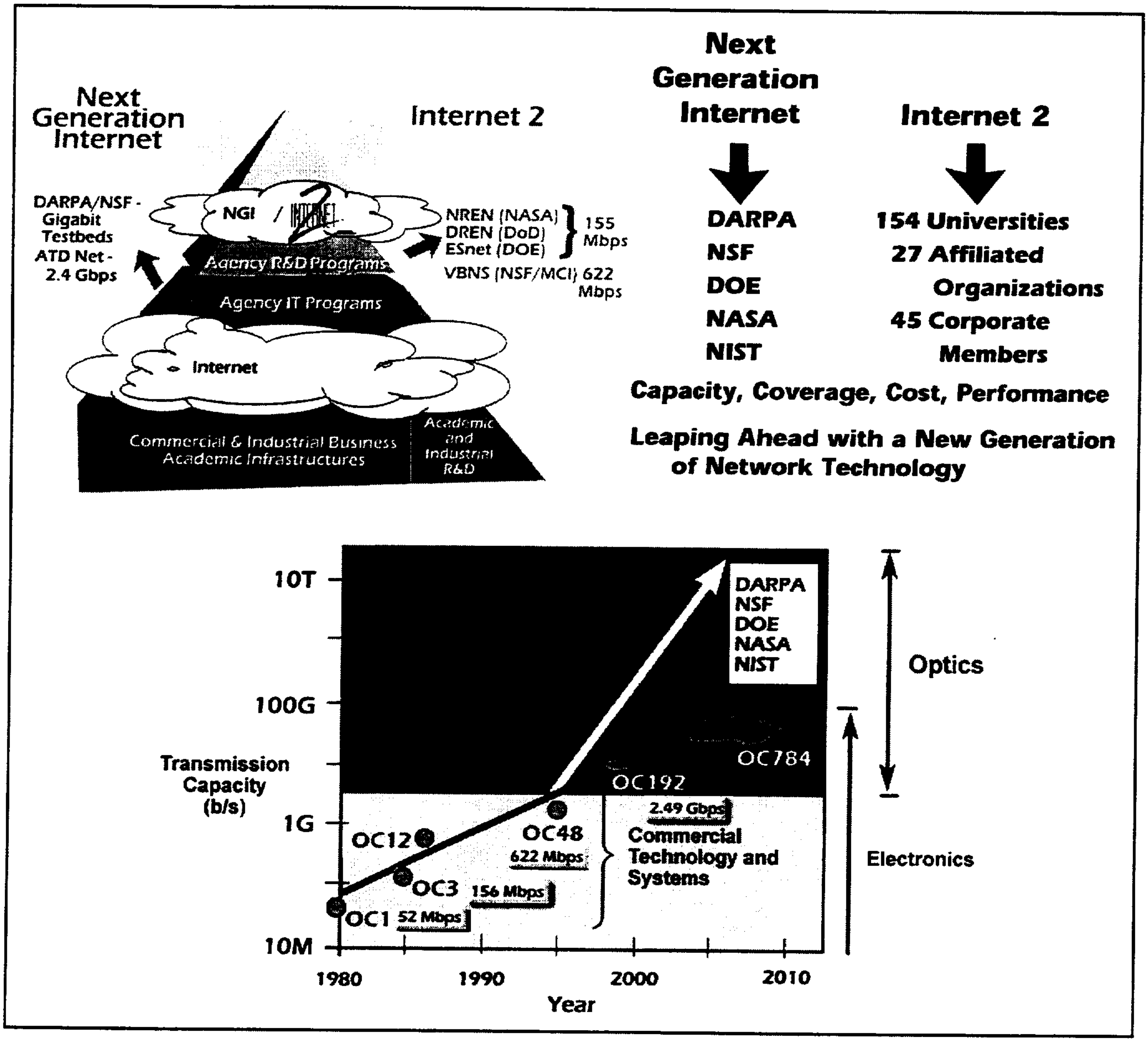


Figure 13

Information Technology

Information technology deals with dissemination, processing, storage, retrieval and use of information. Information science deals with processes of storing and transmitting information. Information theory deals with the mathematical representation of the conditions and parameters affecting the transmission or processing of information. The hierarchy of knowledge is shown in Fig. 14. Processing small, unstructured data we get information; processing information, we get knowledge; then expertise and unique capability, respectively.

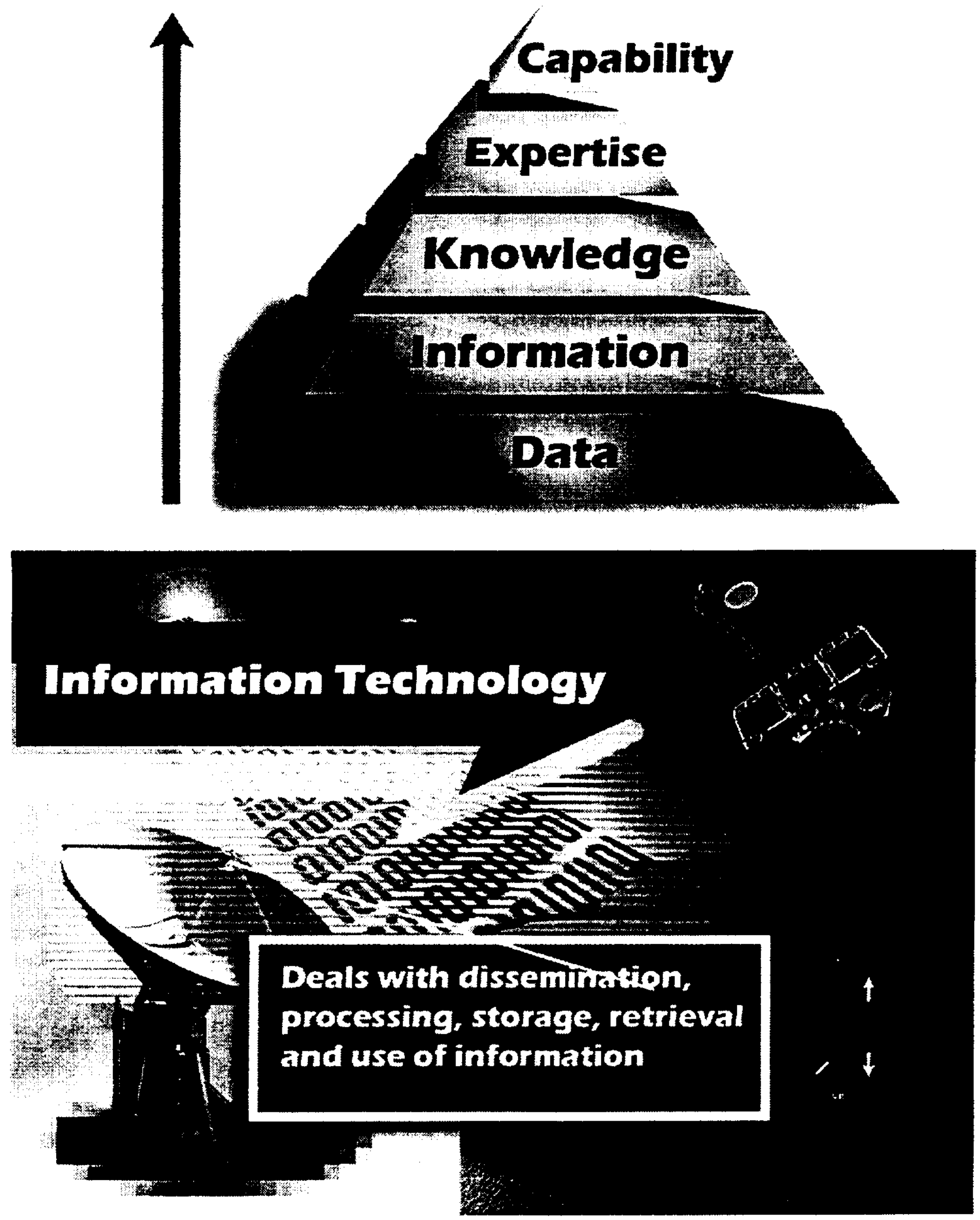
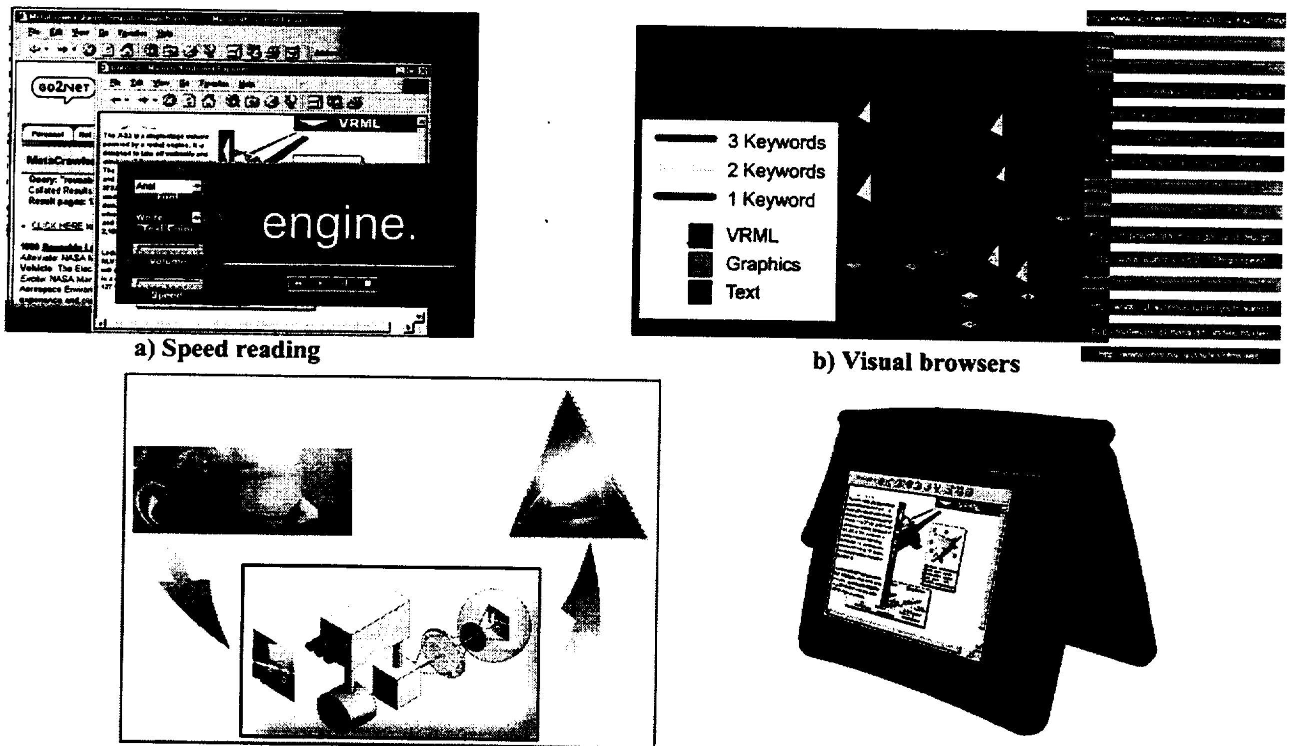


Figure 14

Knowledge-User Interfaces

Knowledge-user interfaces (KUI) can go far beyond the traditional graphical user interfaces (GUI) in speeding up the rate at which users can understand information, not just see it. Some of the interfaces that have the potential of performing this task are speed reading facilities, visual browsers, virtual retinal displays (VRD), and personal access devices (PAD), as shown in the graphic depiction below (Fig. 15).



a) Speed reading

b) Visual browsers

c) Virtual Retinal Display (VRD)

d) Personal Access Device (PAD)

Figure 15

Knowledge Integration Environment and Creativity Machine

By using a combination of knowledge-user interfaces, knowledge coming from various sources can be integrated and managed. A step beyond that is to help users make inventions and discovery. The creativity machine paradigm is an attempt towards mimicking the process of invention of the brain. It consists of two artificial neural networks. The two ANN are simple feed forward networks, trained by standard back-propagation technique. The first ANN is referred to as the imagination engine. With the application of chaotic perturbation (noise) to its architecture, the output is monitored by the second ANN, which is referred to as the Alert Associative Center, to identify any useful concepts that emerge (Fig. 16).

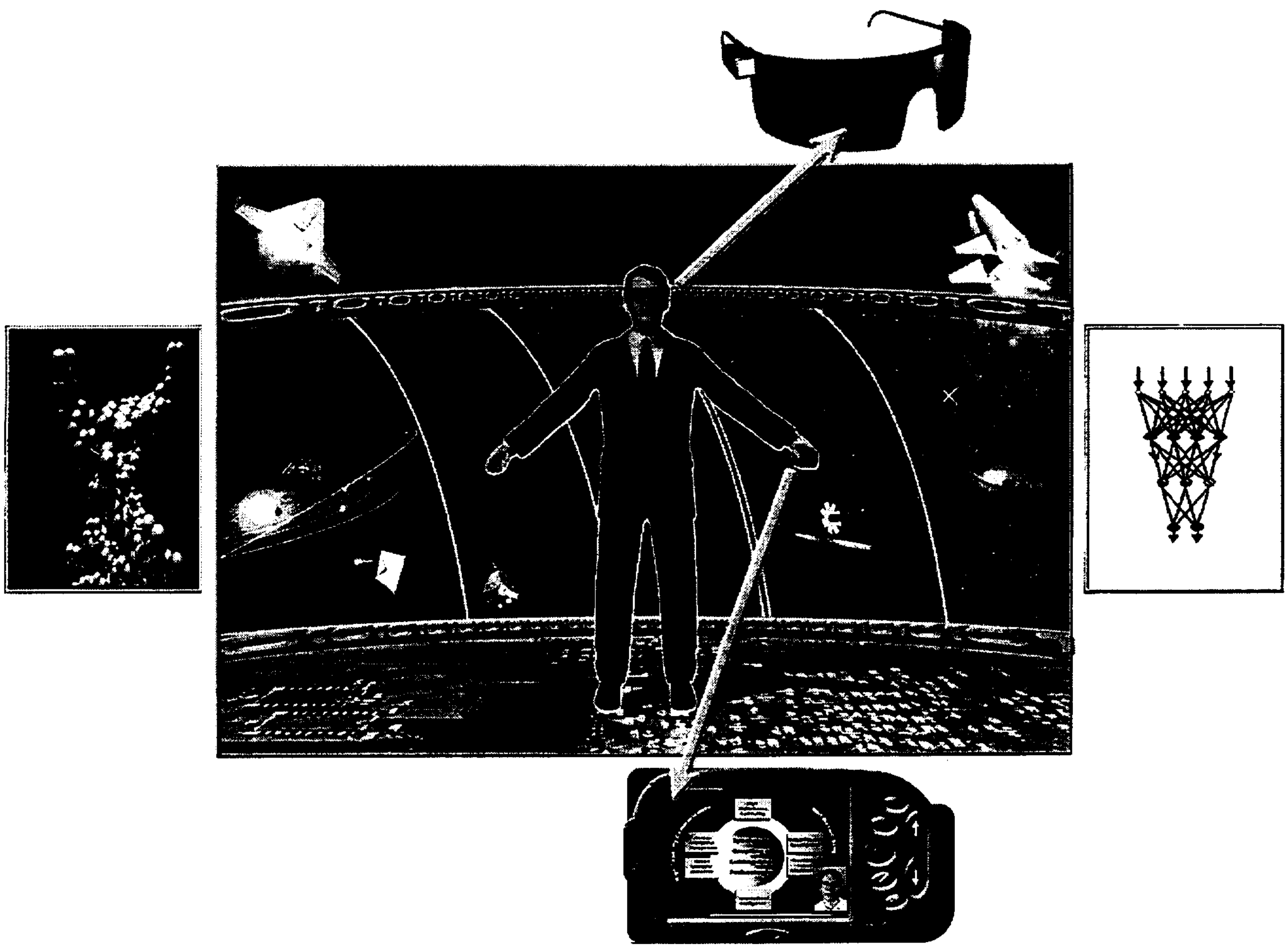


Figure 16

Modeling and Visualization Technologies

A broad spectrum of component technologies, tools and facilities have been developed for modeling and visualization, including (Fig. 17):

- Real-time model generation
- Multimedia workstations that can significantly reduce the time for post-processing and understanding the data
- Sonification facilities for mapping data into the sound domain; and
- Advanced visualization engines and novel visualization paradigms. The visualization engines include virtual reality and augmented reality facilities. The novel visualization paradigms include interactive visualization (real time simulation and visualization), and computational steering (real time simulation, visualization and control).

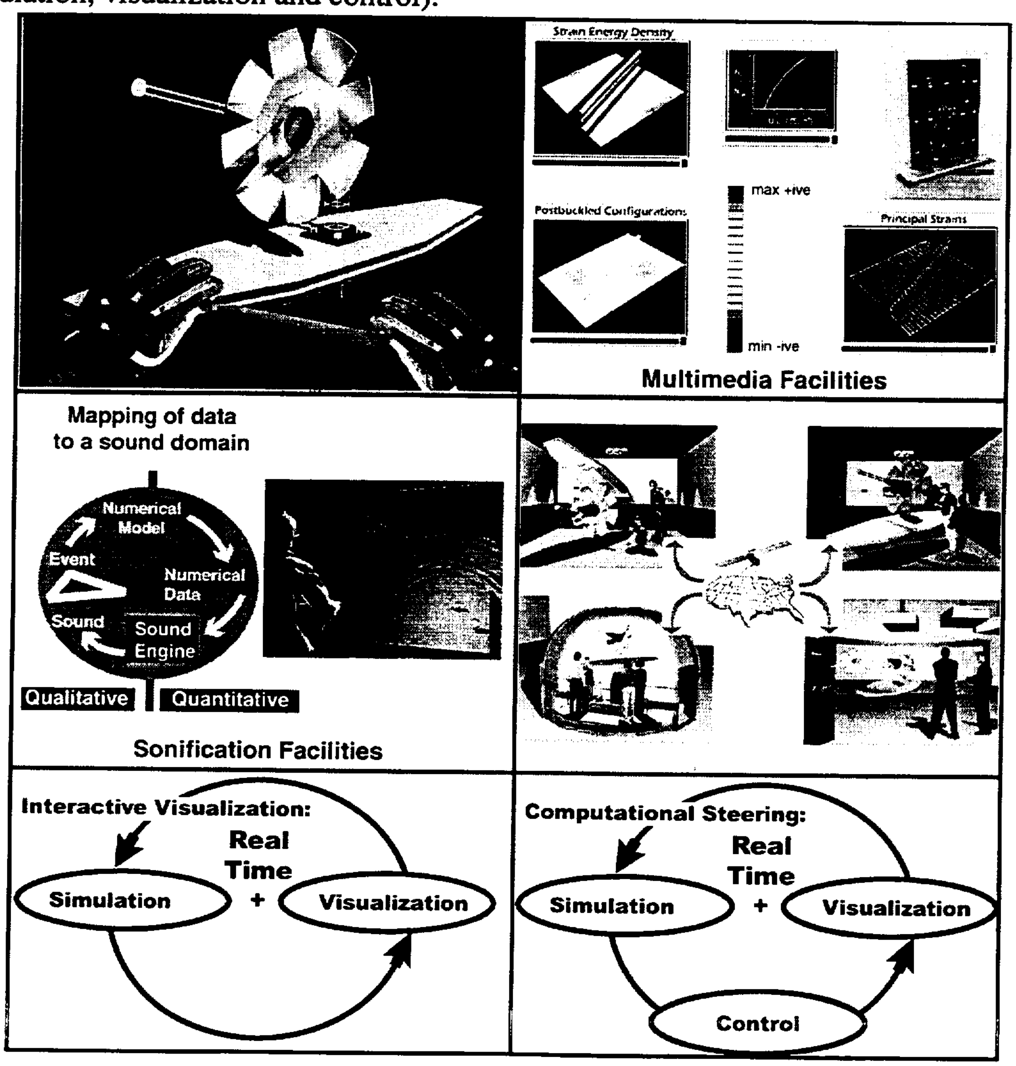


Figure 17

Simulation Technology

Simulation tools include traditional deterministic disciplinary and multidisciplinary methods. There are also tools for life-cycle simulations, including virtual manufacturing and prototyping, maintenance, operations and training. Nontraditional simulation methods address multi-scale phenomena (such as computationally driven material development), highly coupled problems (e.g., smart materials and structures with strong couplings between mechanical, thermal, electrical and magnetic fields), and complex problems with uncertainties (Fig. 18).

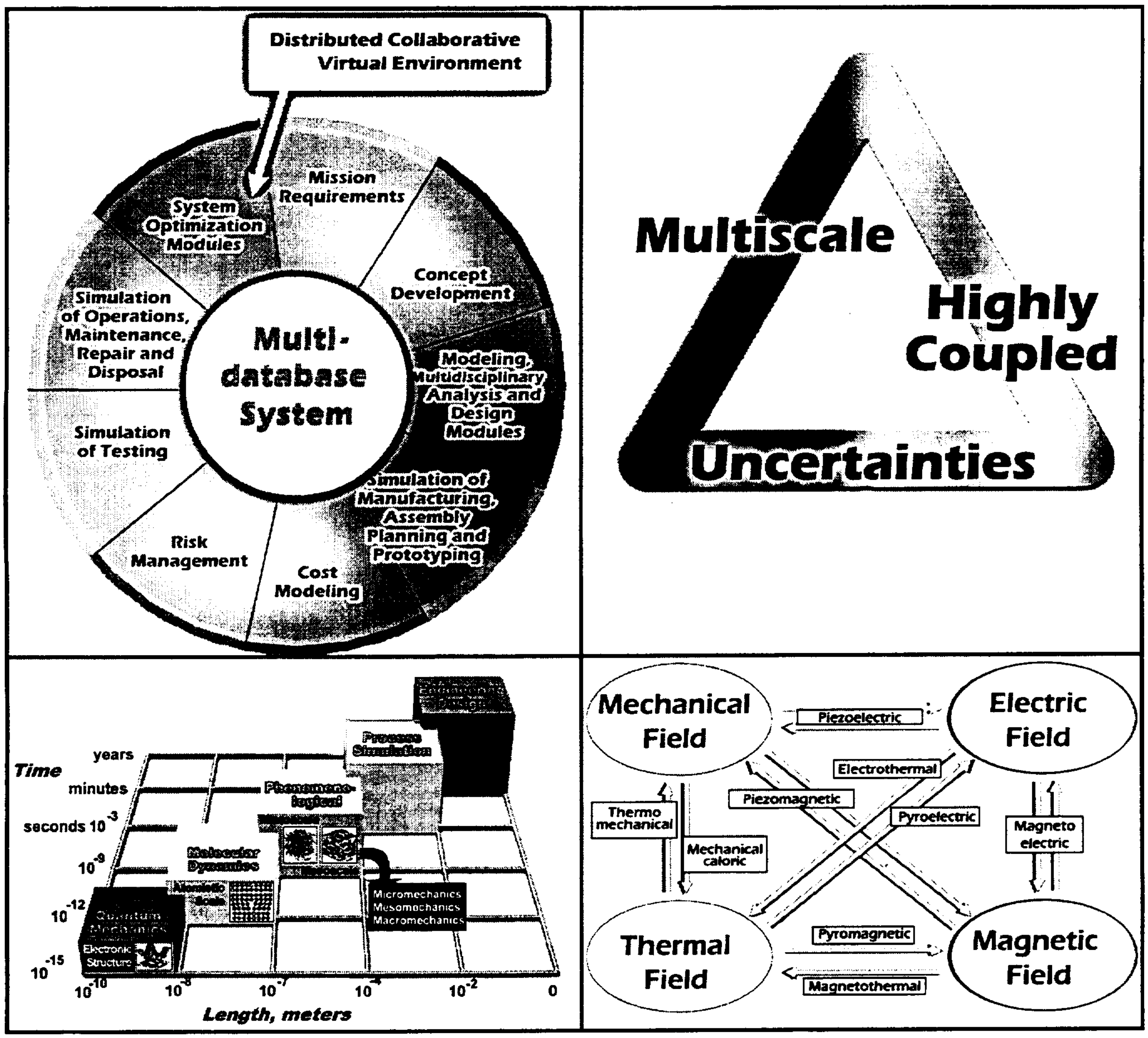


Figure 18

Evolution of Human-Computer Interfaces

Figure 19 shows the evolution of human-computer interfaces. During the period of the 1950's through the 1970's, static interfaces were used in the form of teletype style and full-screen text and light pen. The system designer built the interface and the user had to learn how to use it. This was followed in the 1980's and early 1990's by more flexible interfaces – windows, mouse and graphical tablet. The flexibility was restricted to simple changes (colors, size or positions and animation). In the 1990's, windows, mouse, graphical tablets, adaptive multimedia (audio, video and animation) interfaces were introduced. The adaptation covered both the communication and functionality and included: user-initiated self adaptation, user-controlled self-adaptation, computer-aided adaptation and system initiated adaptation. The trend is now moving towards intelligent interfaces, which integrates adaptive interfaces with intelligent agents for making intelligent help and tutoring available to the user.

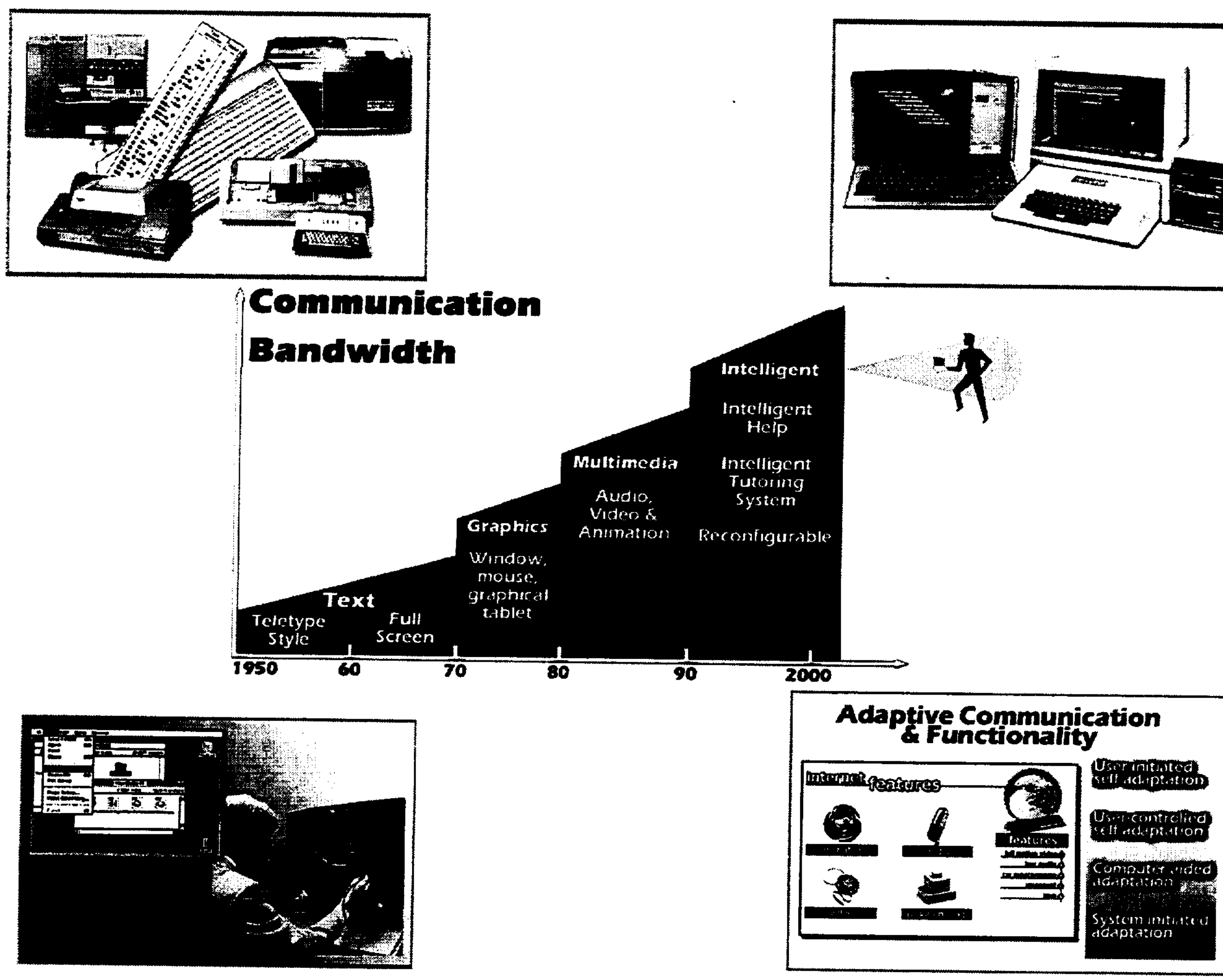


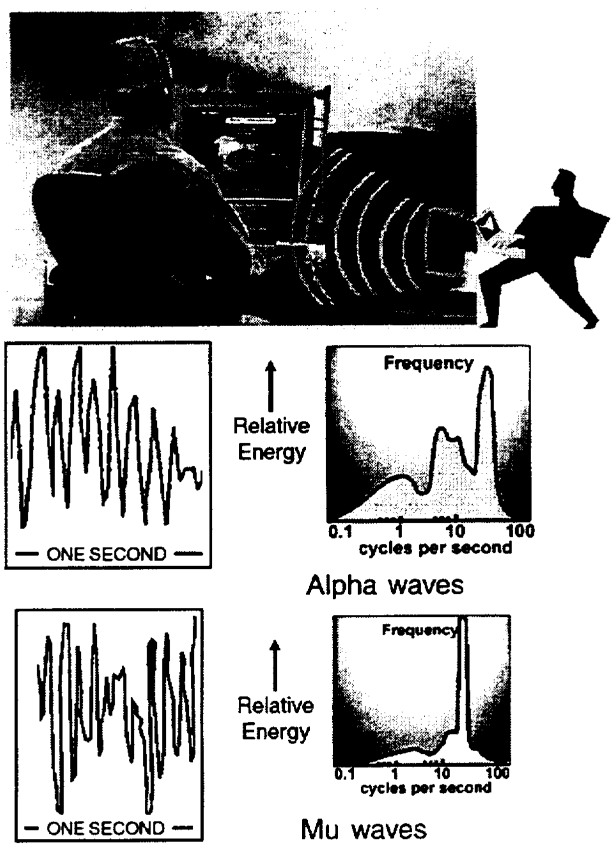
Figure 19

Adaptive/Reconfigurable Human-Computer Interfaces

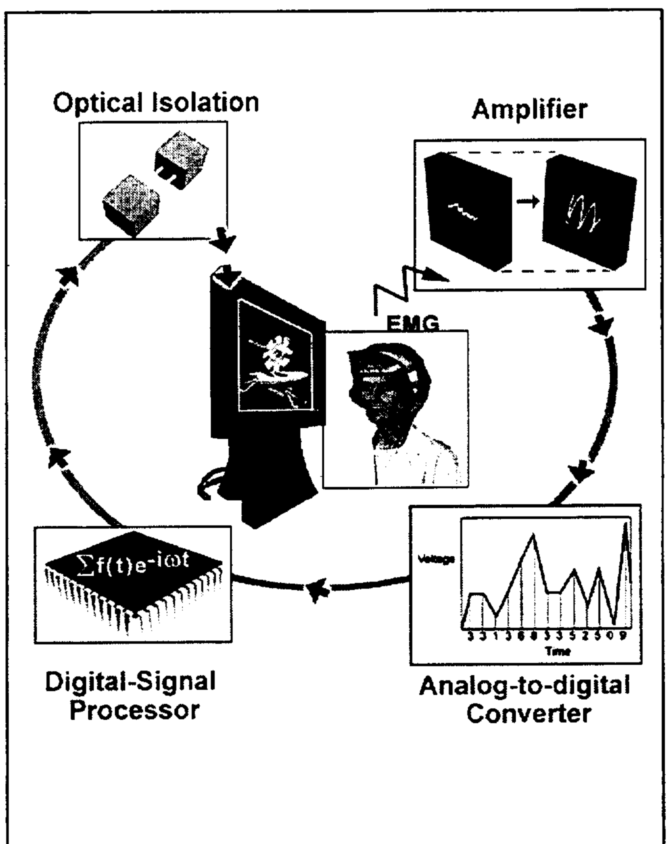
Future HCI will use cognitive neuroscience to couple humans with computers and maximize the performance of both. Natural languages will provide effortless communication with the computer, and the interfaces will be adaptive and reconfigurable.

Two categories of interfaces that can be used to identify different states of mental alertness are shown in Fig. 20: neural interfaces, based on brainwaves (e.g., alpha waves and mu waves); and biological interfaces using EMGs (electro-myographic signals). The user wears a specially designed headband with electrodes that can detect electrical signals from the muscles when they pass through the skin. An amplifier is used to amplify the tiny signals from the muscles. An analog-to-digital converter translates the voltage to the computer, and a digital signal processor extracts important features of the signal (reflecting the mental alertness of the user) and the computer responds by selecting a suitable interface. An optical isolation device is used to avoid electric shock of the user.

Ultimately, it is desirable to have computers with thought recognition (minds coupled to computers - a person thinks of a command and the computer immediately responds), a scheme analogous to voice recognition in use today.



neural interface



biological interface

Figure 20

Intelligent Software Agent/Virtual Instructor

The coupling of intelligent agents with feature recognition can produce virtual instructors that recognize the facial expressions of the learner and provide an appropriate response. The learner can ask the virtual instructor to give him/her an example, give details or provide background for the material covered (see Fig. 21).

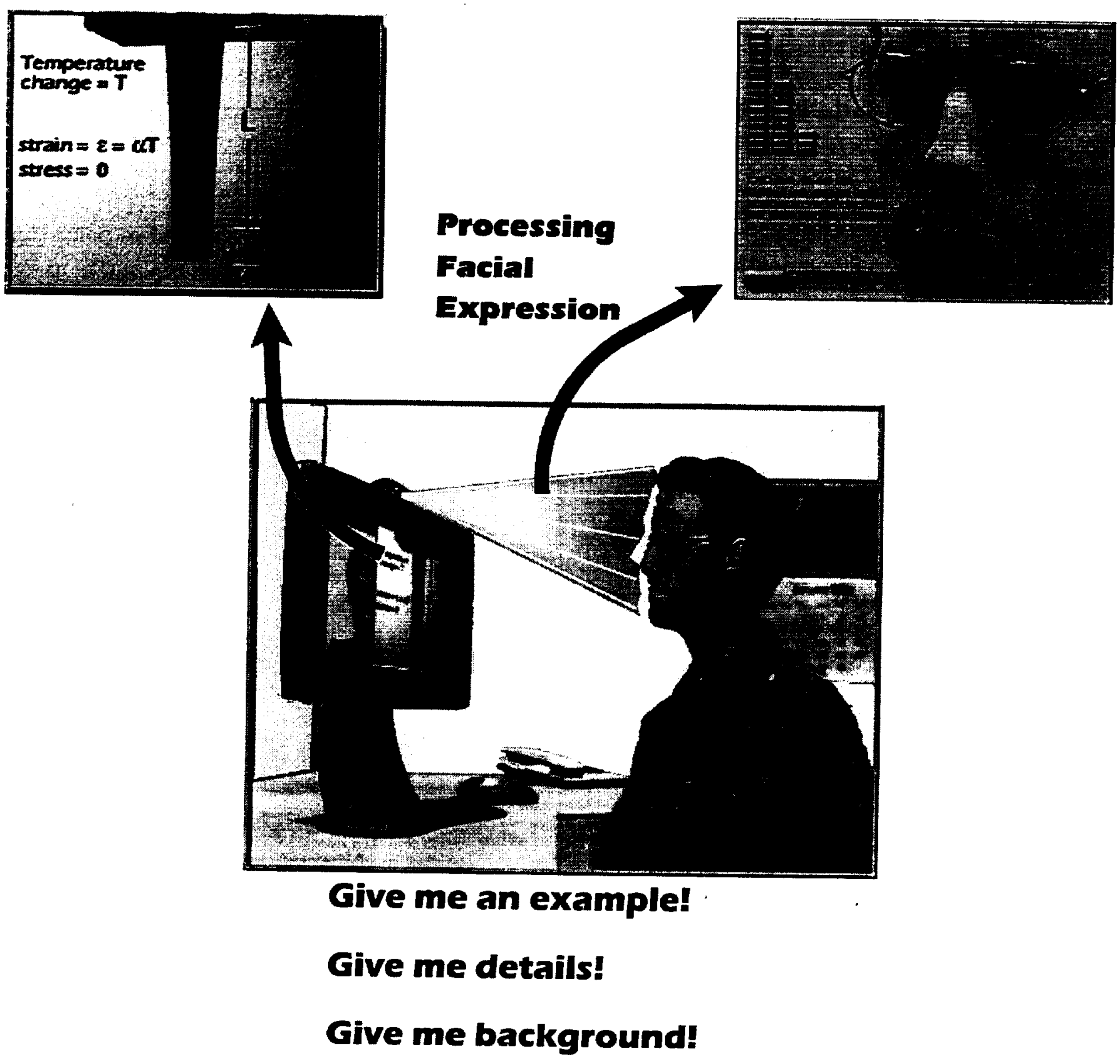


Figure 21

Learning Strategies and Advanced Learning Environments

A number of strategies have been developed for maximizing learning in organizations. Seven of these strategies and their best use are listed subsequently (Fig. 22):

- Cognitive – presentations and explanations
- Inquiry – critical, creative and dialogical thinking
- Mental models – problem solving and decision making
- Behavioral – advanced skill development
- Virtual/synthetic environment – development of competence in simulated environment
- Holistic – personal learning through experience
- Group dynamics – collaboration and working in teams.

Three categories of learning environments are needed for implementing the aforementioned learning strategies, namely: expert-led, self-paced, and collaborative. The three environments are described subsequently.

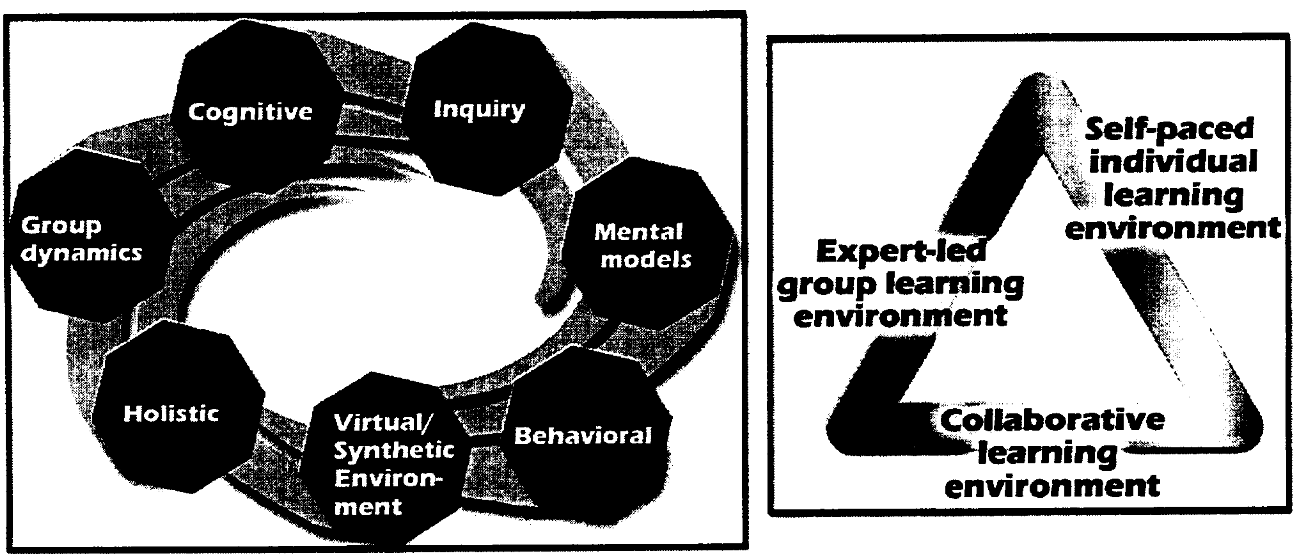


Figure 22

Expert-Led Group Learning Environment

The instructor in the expert-led learning environment focuses his/her presentation on giving a broad overview of the topic and its diverse applications, and ends the presentation with penetrating, what-if questions to help the learners in developing critical thinking skills. Elaborate visualization and multimedia facilities are used in the presentation (Fig. 23). Mundane instructional and training tasks are relegated to computer laboratory sessions.

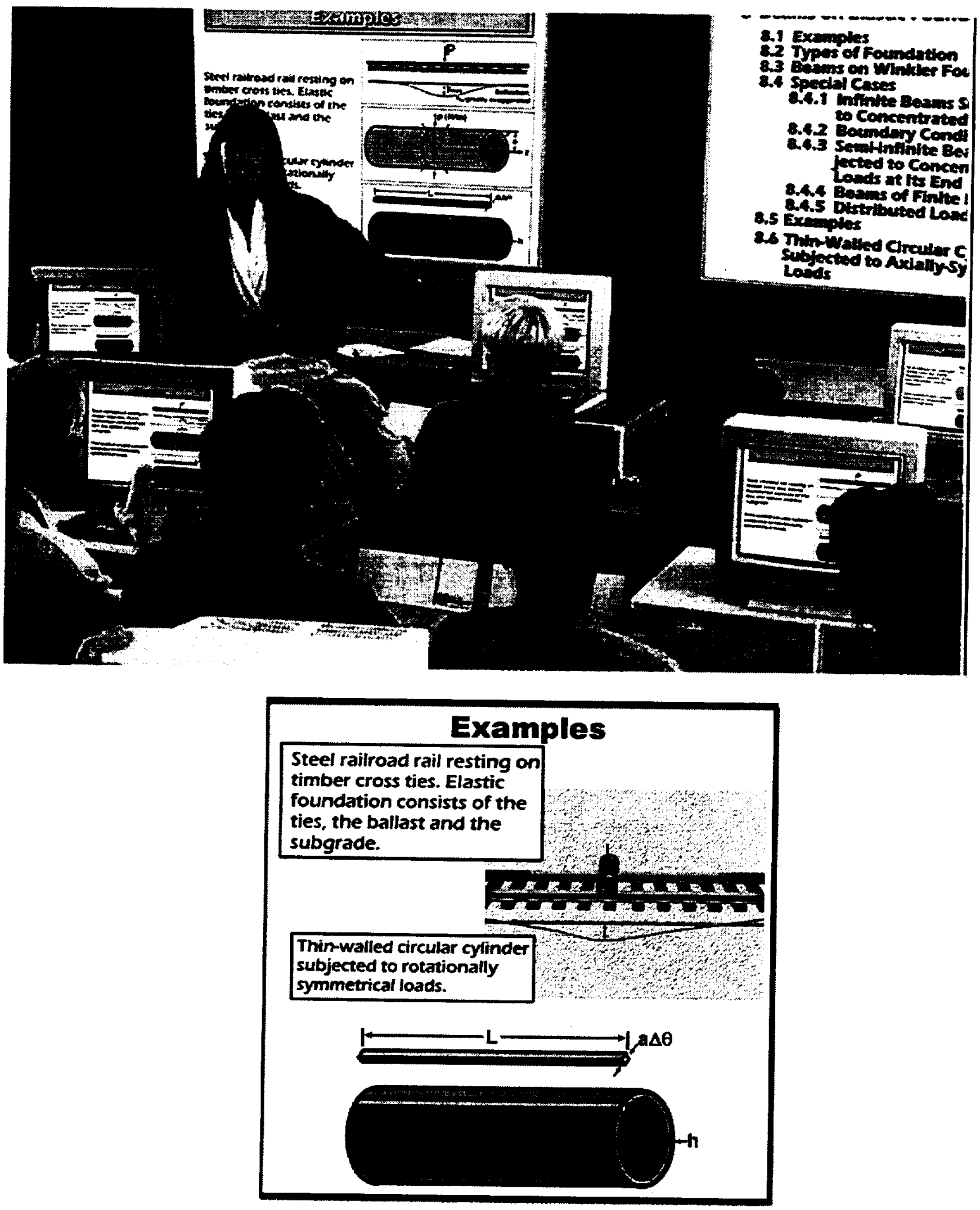


Figure 23

Self-Paced Individual Learning Environment

The individual learning environment engages the learner and provides a high degree of interactivity with the environment. It can be used for (Fig. 24): a) self-paced instruction of the routine tasks not covered in the lecture; b) under of physical phenomena using advanced visualization (e.g., animation), multimedia and immersive virtual reality facilities; c) virtual experiments (computer-simulation of physical experiments).

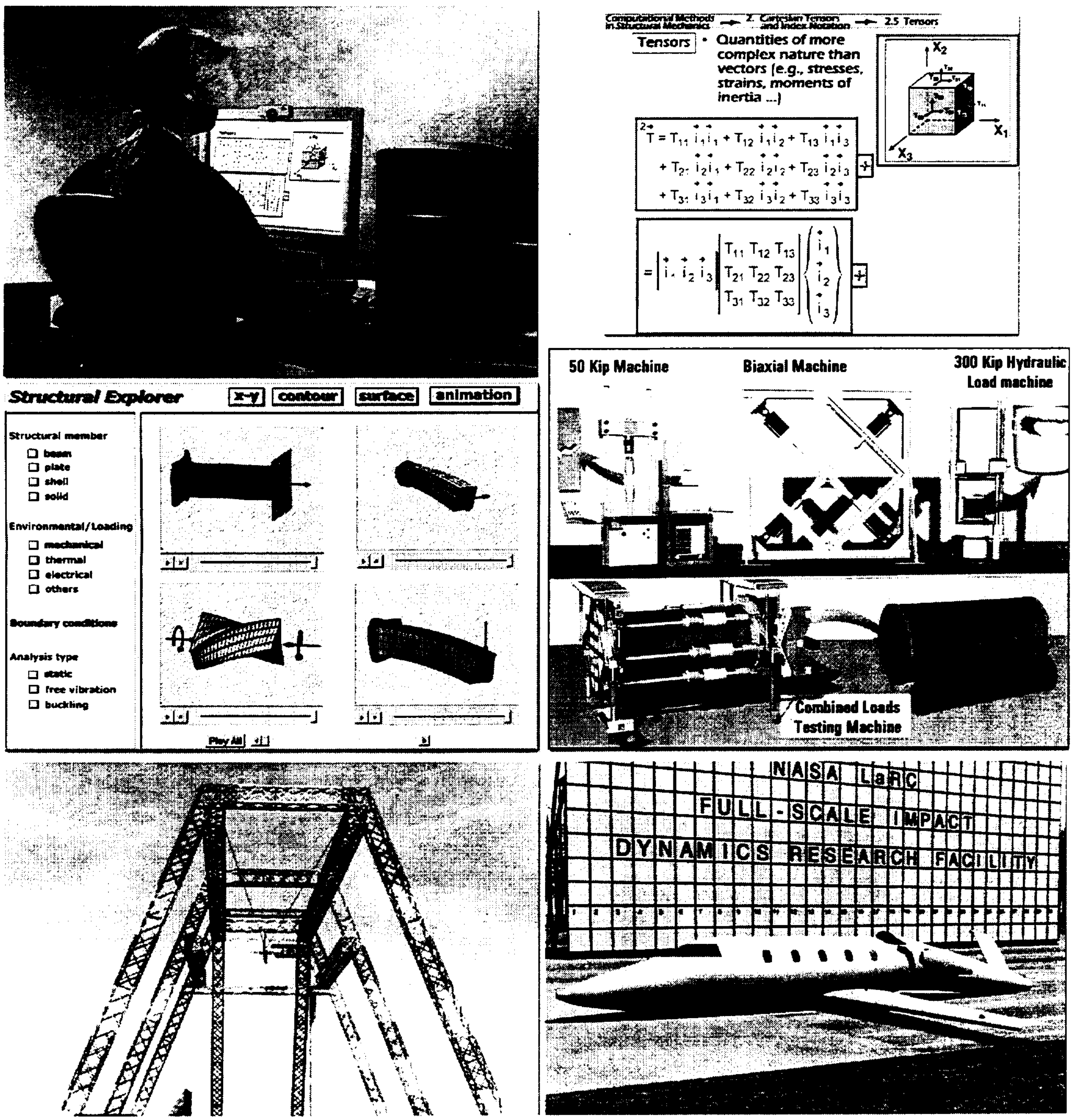


Figure 24

Collaborative Learning Environment

Collaborative learning environments train the learners in teamwork. The learners can be geographically dispersed and can be brought together through immersive telepresence facilities to work on joint projects, or to share their experiences in highly heterogeneous environments. They can be virtually collocated in industry labs (Fig. 25).

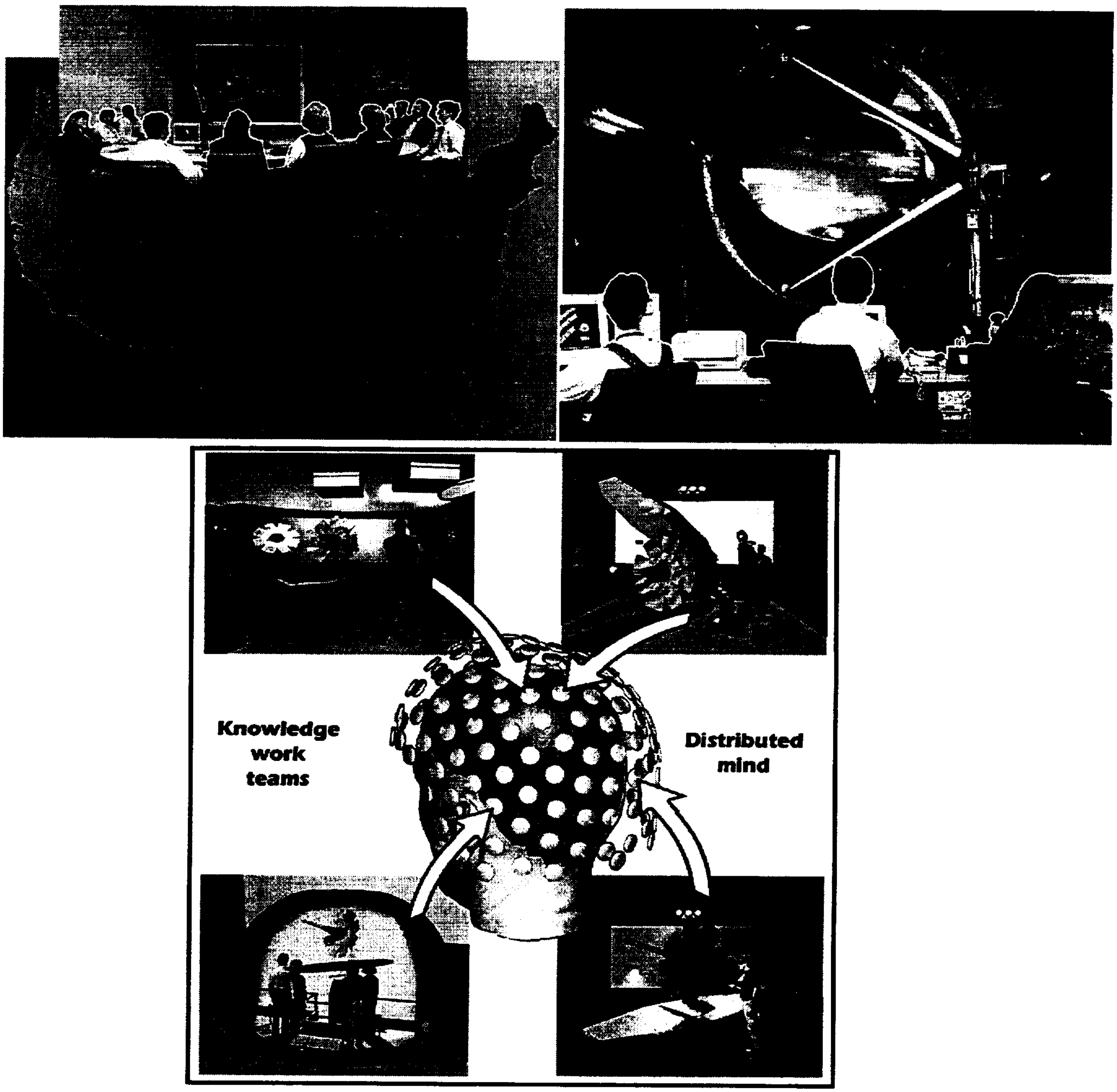


Figure 25

Distributed Learning Environment (DLE)

Collaborative learning environments can evolve into active distributed learning environments for constructive engagement of the learners. It should incorporate state-of-the-art multimedia, immersive facilities and multi-sensory interfaces, and be tailored to each individual learner's needs. The environment enables learning anywhere and at any time.

Dynamic process improvement can be performed by: a) simulation of the procedures, processes and coordination/integration mechanisms involved in the distributed interactions between instructors and learners as well as the information flow during these interactions; and b) development of metrics for assessing the effectiveness of learning (Fig. 26).

The effective deployment of DLE can result in a skilled workforce able to sustain technological growth.

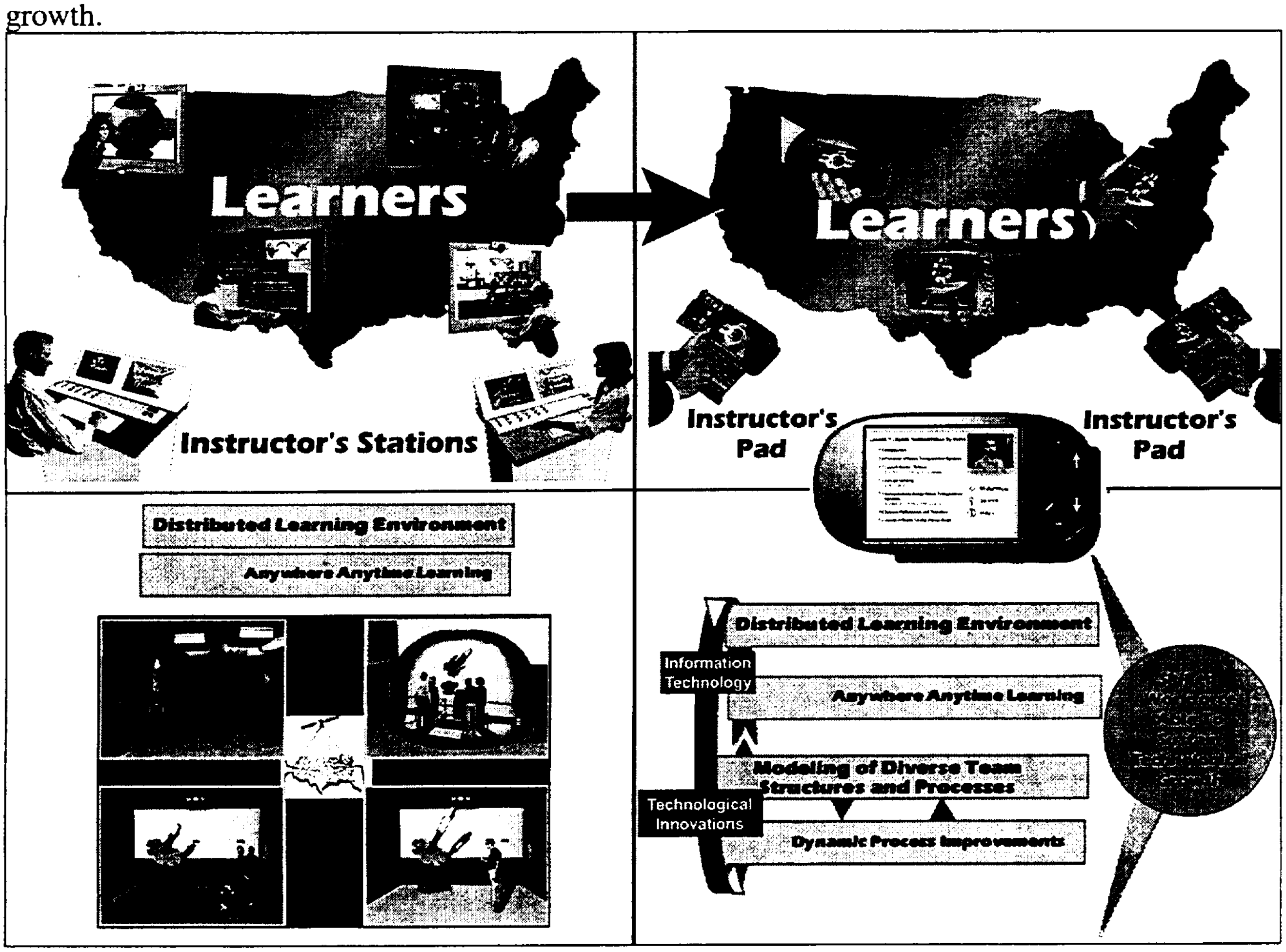


Figure 26

Government Activities

A number of government activities are currently underway in the areas of advanced learning technologies and learning environments. These include (Fig. 27):

- Advanced Distributed Learning (ADL) of DoD/OSTP
- Advanced Technology Program (ATP) of NIST
- Technology-Supported Learning of the Department of Energy, and
- Digital libraries initiative, which is now in its second phase.

Figure 27

Non-government Activities

A number of industries and professional societies are developing training technologies. A partial list is given in Fig. 28. The professional societies include:

- Global Alliance for International Education
- American Society for Training and Development
- U.S. Distance Learning Association.

Figure 28

Learning Technology Vendors

The percentage of training time using learning technology has grown from 10% in 1996 to 17% in 1997 and is expected to reach 35% by 2000. Consequently, the number of learning technology vendors is increasing. A partial list of these vendors is given in Fig. 29.

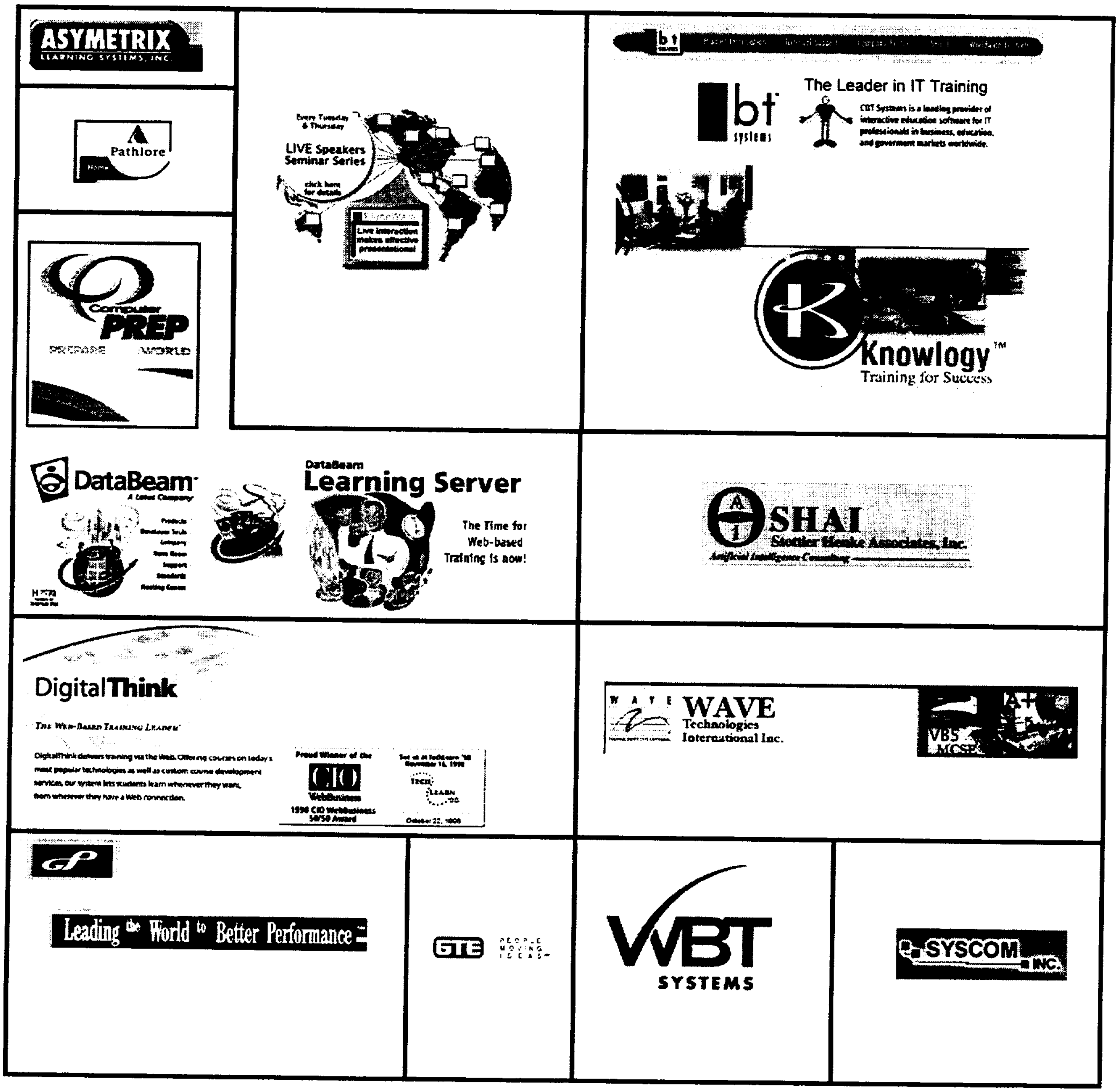


Figure 29

Authoring Tools and Other Resources

A number of general-purpose and web-based authoring tools are now available for developing multimedia and online courses. In addition, magazines and internet resources are also available. Some of these are listed in Fig. 30.

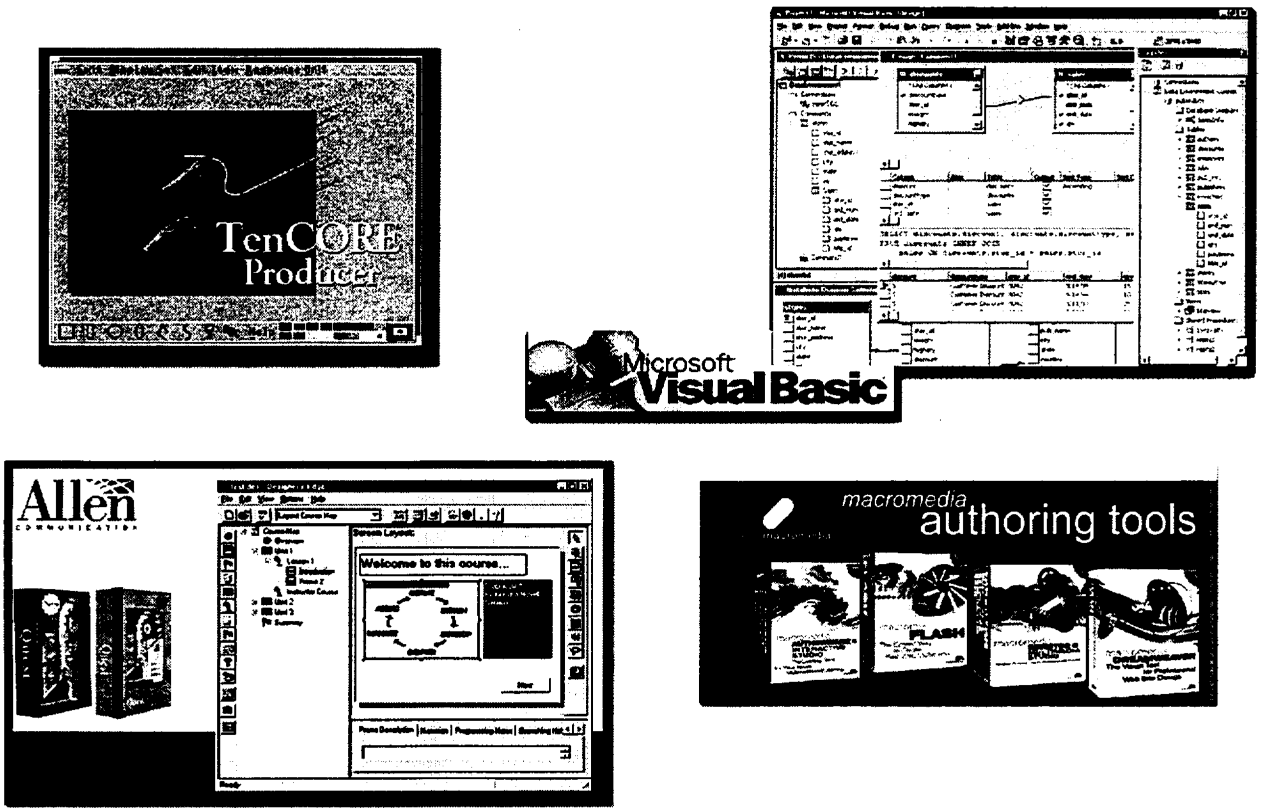


Figure 30

Advanced Technologies for Training and Education

R. Bowen Loftin
Director, NASA/University of Houston
Virtual Environment Technology Laboratory
Houston, TX

Advanced Technologies for Training and Education

R. Bowen Loftin, Director
NASA/UH Virtual Environment Technology Laboratory
Department of Computer Science
University of Houston
Houston, TX 77204

The NASA/University of Houston Virtual Environment Technology Laboratory (VETL) was born at the NASA/Johnson Space Center in late 1990 when Robert T. Savely (Johnson Space Center's Chief Scientist for Advanced Software Technology) and I reached the conclusion that commercial off-the-shelf hardware needed to create immersive, interactive virtual environments and had reached sufficient maturity to justify an investment. It was our judgment that virtual environment technology (otherwise known as virtual reality) had the potential to address significant NASA training needs, perhaps more cheaply than conventional techniques.

Over the next two years, the development of a skilled group of software engineers and students (as applications were developed and refined) culminated, in 1993, with the creation of a series of virtual environments that were used to train over 100 members of the ground-based flight team for the Hubble Space Telescope Servicing Mission.

In early 1995 the magnitude and diversity of VETL's work grew to the point where it could not be sustained in the original lab site at the Johnson Space Center. Jane Stearns (Chief of the Johnson Space Center Business and Information Systems Directorate) elected to pursue a formal relationship with the University of Houston to form a joint laboratory. In May 1995, the Director of the Johnson Space Center executed a Space Act Agreement with the University of Houston that created the current VETL. The laboratory's objectives now include research and development activities in training, education and scientific/engineering data visualization.

Advanced Technologies for Training and Education

R. Bowen Loftin, Director
NASA/UH Virtual Environment
Technology Laboratory
Professor of Computer Science
University of Houston

March 9, 1999

Outline

For more than a decade, the Advanced Training Technologies project at the NASA/Johnson Space Center and the University of Houston has implemented new technologies, from intelligent training and tutoring systems to virtual reality, in an effort to reduce the cost and enhance the distribution and effectiveness of training within NASA and other government agencies. In keeping with NASA's mandate to transfer its technology to the public sector, this effort is also producing exciting "spinoffs" for education. In recent years the development of virtual environment technologies has offered another avenue to both address unique training challenges within NASA and to explore the application of these technologies in education at all levels.

Further information, including papers and images, can be obtained from our home page at http://www.vetl.uh.edu

Outline

- A Little History . . .
  - NASA Missions
  - NASA Training
- Applications of VEs
  - Training
  - Education
- Issues
- What's Next?
- Contact Information

NASA HEDS Missions

NASA's "manned" missions include the ongoing Space Shuttle program as well as the newly-initiated International Space Station program. In addition, NASA personnel are planning for future lunar and Mars missions. Advanced technologies such as virtual environments and intelligent systems will play an important role in preparing crews and support personnel for these missions and in providing for "just-in-time" training during missions.

NASA HEDS Missions

- Space Shuttle
  - Payloads, Maintenance, Science . . .
- International Space Station (ISS)
  - Assembly, Maintenance, Science . . .
- Beyond LEO
  - Lunar Missions
  - Mars Missions

NASA Training

NASA training has, from the days of the Apollo program, relied heavily on neutral buoyancy environments where astronauts and payloads are immersed in water. The problems with such training include the cost of the facilities and the personnel required to operate them, the negative training that can be introduced due to water's viscosity, and the total time required to train (including suiting, unsuiting and decompression). Virtual environments have been shown to be effective in training for a number of tasks and are now in routine use for extravehicular activity training. Over time, this technology will take on an even greater role in training, leading to its delivery in space.

NASA Training

- The Past
  - Neutral Buoyancy
  - Part-Task Trainers
  - Full-Scale Simulators
- The Present
  - All of the Past + Some Virtual Environment Technology (VET)
- The Future
  - Less of the Past + More VET + "Intelligent" VET + Just-in-Time Training

Training and Education Examples

The following eleven pages contain details and images for four applications that have been developed during the past six years.

Training and Education Examples

- Hubble Space Telescope Mission
- Shared Virtual Environments for Training
- Peacekeeping Operations
- Project ScienceSpace

Hubble Space Telescope

See the following paper for details of this application:

R. B. Loftin and P. Kenney, "Training the Hubble Space Telescope Flight Team," *IEEE Computer Graphics and Applications,* Vol. 15, No. 5, Sept. 1995, pp. 31-37.

Hubble Space Telescope

- Prepared Flight Team for December 1993 Mission
- 105 Trained
- Focus on Procedures and Visualizing Work Environment
- Contributed to Mission Success
- Detected Operational Errors before Flight Largest Use of VEs for Actual Mission Training to Date

Shared Virtual Environments

See the following papers for details of this work:

R. B. Loftin, "Virtual Reality Links Astronaut Training," *Real Time Graphics,* Vol.4, No. 4, Oct./Nov. 1995.

R. B. Loftin, "Hands Across the Atlantic," *Virtual Reality Special Report,* Vol. 3, No. 2, Mar.-Apr. 1996, pp. 39-41.

R. B. Loftin, "Aerospace Applications of Virtual Reality," *Computer Graphics,* Vol. 30, No. 4, 1996, pp. 33-35.

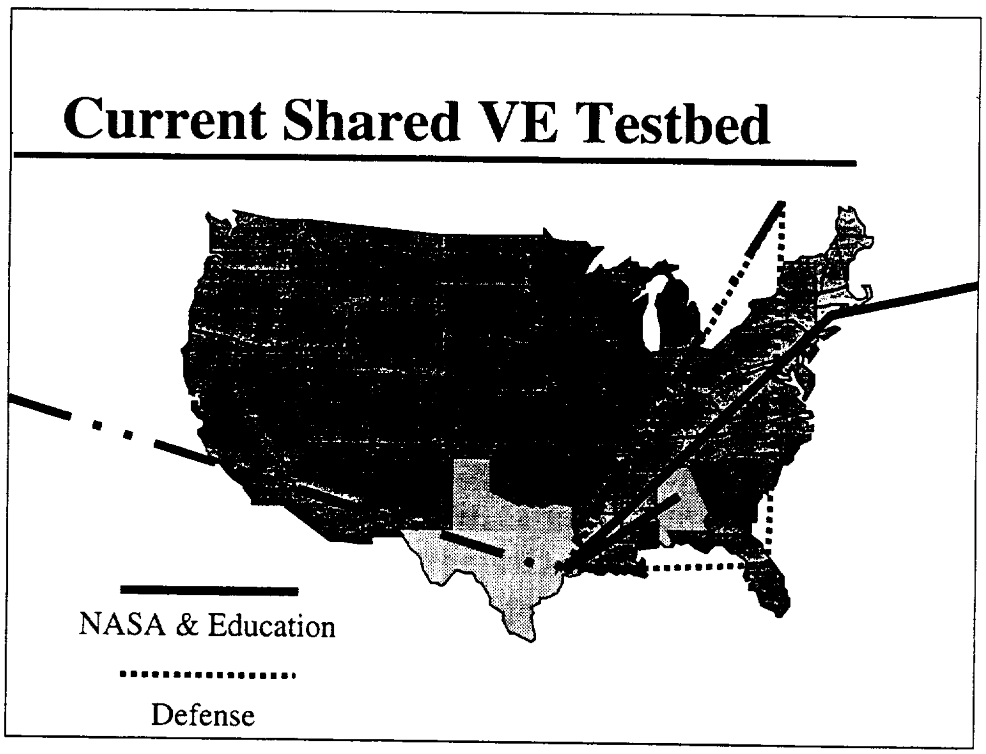
R. B. Loftin, "Distributed Virtual Environments for Collective Training," in *Proceedings of the 1998 IMAGE Conference*, Scottsdale, Arizona, Aug. 7, 1998.

Shared Virtual Environments

- Cooperative Efforts with
  - Fraunhofer Institute for Computer Graphics (Germany)
  - Raytheon and Hughes Research Laboratories (Texas and California)
  - NASA/Marshall Space Flight Center (Alabama)
- Approach
  - ISS Intravehicular Activities Testbed
  - Render Separate Databases
  - Communicate State Change Data

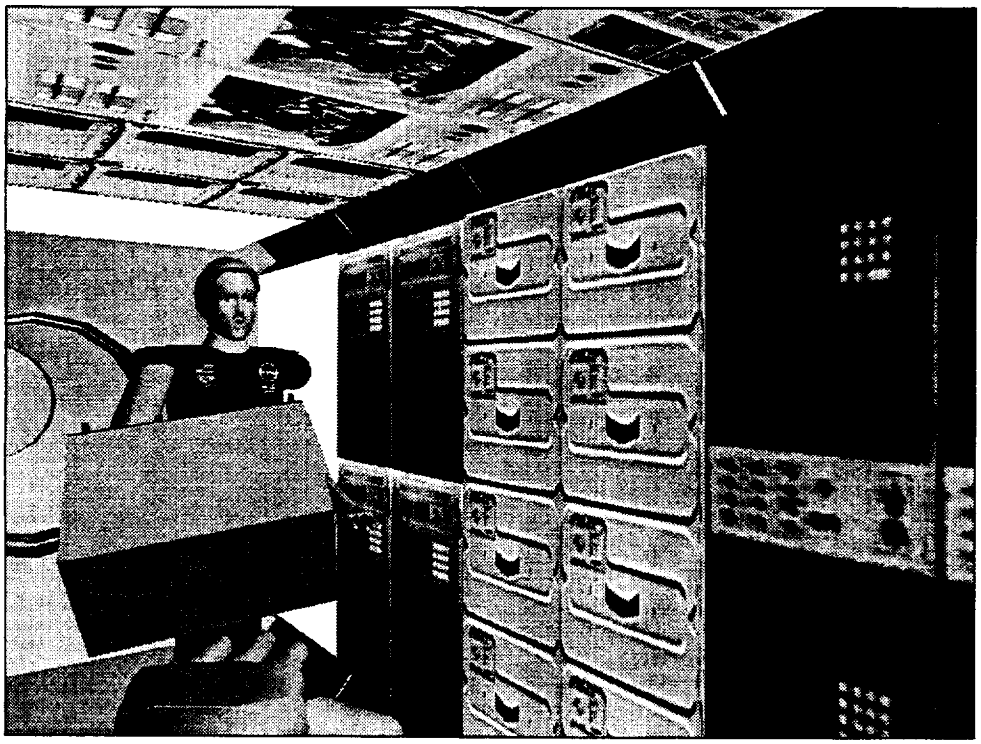
Current Shared VE Testbed

This map illustrates both the current shared virtual environments testbed and its future extensions. The solid line shows the testbed between Houston and the Marshall Space Flight Center in Huntsville, Alabama as well as the testbed between Houston and the Fraunhofer Institute for Computer Graphics in Darmstadt, Germany. The dotted line shows the testbed between Houston, a U.S. Army laboratory in Orlando, Florida, and a Canadian Defense Ministry laboratory in Toronto, Canada. The broken lines show future testbeds between Houston, Canada and Japan.



International Space Station

This image shows the avatar of one astronaut as seen through the eyes of another astronaut working in one of the science modules of the International Space Station. The two astronauts are in different locations and "see" each other as human figure representations (or avatars) that are animated by their movements. In this case a storage box containing supplies for a biotechnology experiment is being handed from one astronaut to the other.

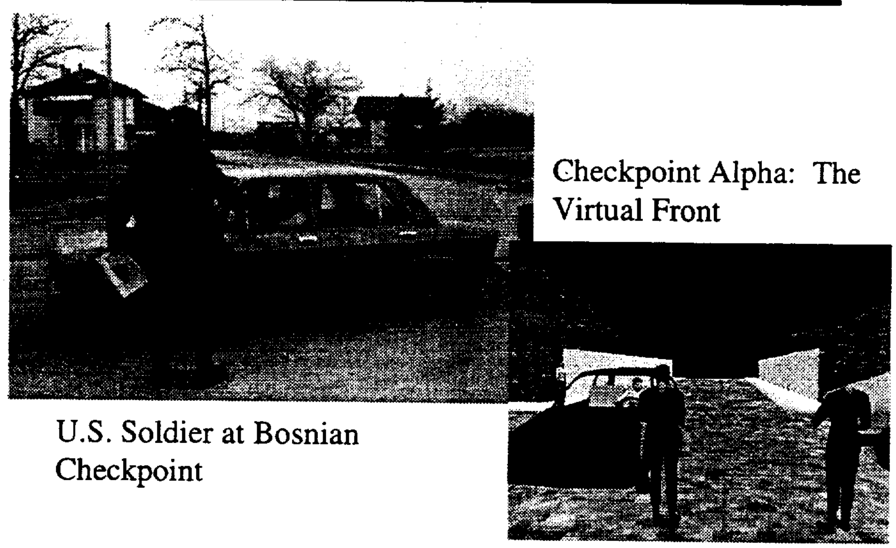


VEs for Peacekeeping Operations

See the following paper for details of this work:

R. B. Loftin, "Distributed Virtual Environments for Collective Training," in *Proceedings of the 1998 IMAGE Conference*, Scottsdale, Arizona, Aug. 7, 1998.

VEs for Peacekeeping Operations



Checkpoint Alpha: The Virtual Front

U.S. Soldier at Bosnian Checkpoint

VEs for Peacekeeping Operations

See the following paper for details of this work:

R. B. Loftin, "Distributed Virtual Environments for Collective Training," in *Proceedings of the 1998 IMAGE Conference*, Scottsdale, Arizona, Aug. 7, 1998.

VEs for Peacekeeping Operations

- Focus on Peacekeeping Operations (as in Bosnia)
- Designed for Collective/Team Training and Shared VEs
- Address
  - Cultural Issues
  - Nonverbal Communication
- Support for Rapid Development and "Anywhere" Deployment

Project ScienceSpace

With the support of the National Science Foundation (Grants RED-9353320 and REC-9555682), we are exploring the potential utility of physical immersion and multisensory perception to enhance science education through the design of ScienceSpace, a series of virtual realities for teaching science. One objective of this project is to investigate whether sensorially immersive learning can remediate typical misconceptions in the mental models of reality held by many students. Another is to study whether mastery of traditionally difficult and abstract subjects is enhanced by immersive, learning-by-doing. ScienceSpace enables learners to experience these phenomena (e.g., electrostatics and molecular docking) and may inculcate an instinctive, qualitative understanding as a motivation and basis for future study. ScienceSpace now consists of three virtual worlds.

Project ScienceSpace

- Genesis: Virtual Physics Laboratory (1992)
- Supported by the National Science Foundation and the Shell Oil Company Foundation
- Audience: Junior High School through Introductory College Science Students
- Objectives
  - Remedy Common Misconceptions in Science
  - Aid Students in Building Mental Models of Abstract Concepts
  - Explore Collaboration via Shared Virtual Environments

The Worlds of ScienceSpace

Many publications on Project ScienceSpace can be found at the following web sites:

www.vetl.uh.edu
www.virtual.gmu.edu

Images can be downloaded as well.

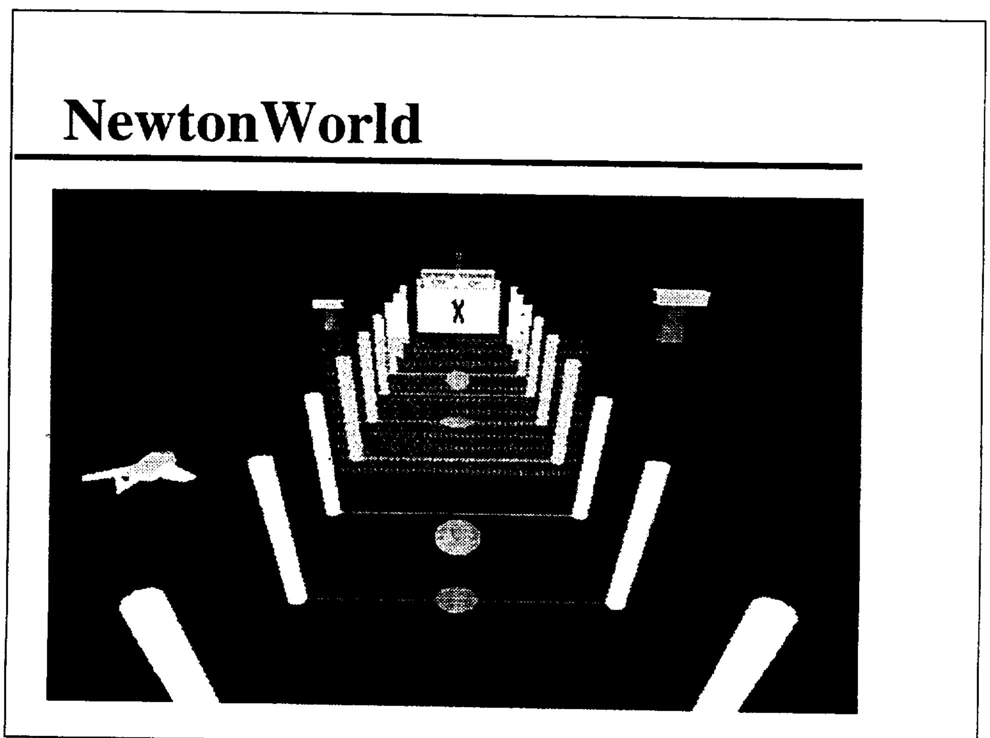
The Worlds of ScienceSpace

- NewtonWorld
  One-Dimensional Kinematics and Collision Physics
- MaxwellWorld
  Electrostatics
- PaulingWorld
  Molecular Structure and Collaborative Learning

NewtonWorld

NewtonWorld is intended for the exploration of Newton's laws of motion as well as the conservation of both kinetic energy and linear momentum. In NewtonWorld, students spend time in and around an open "corridor" created with a colonnade on each side and a wall at each end (see figure). Students interact with NewtonWorld using a "virtual hand" and a menu system. Students can launch and catch balls of various masses and "beam" to cameras strategically placed around the corridor. The balls move in one dimension along the corridor, rebounding when they collide with each other or the walls.

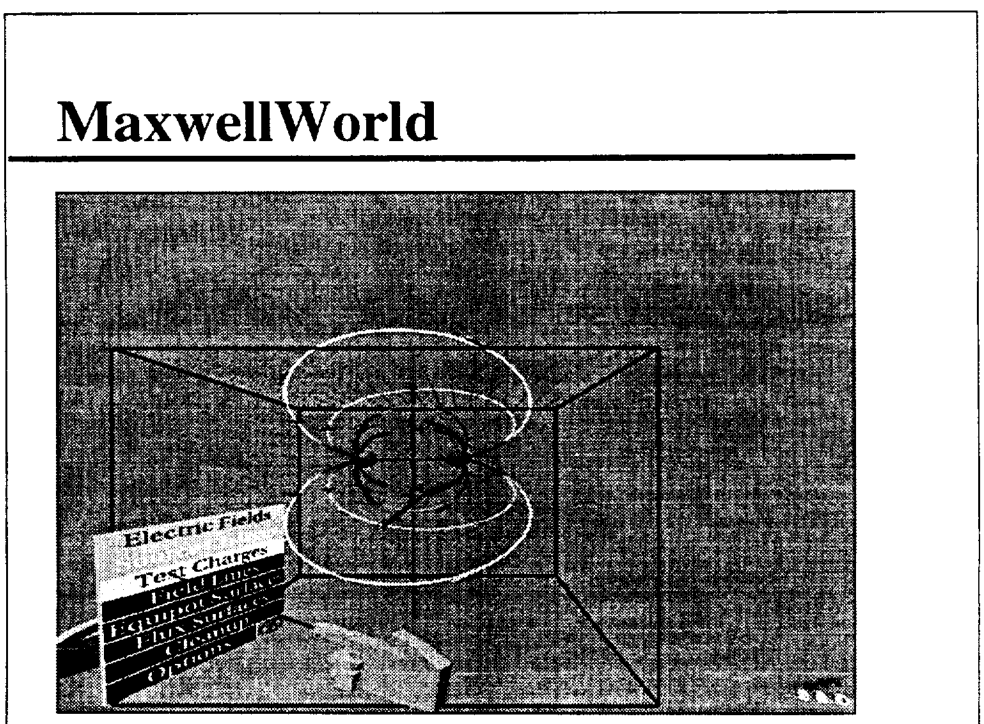
Multisensory cues help students experience phenomena and direct their attention to important factors such as mass, velocity and energy. For example, potential energy is made salient through haptic and visual cues, and velocity through auditory and visual cues. Currently, the presence of potential energy before launch is represented by a tightly coiled spring as well as by vibrations in a vest worn by the user. As the ball is launched and potential energy becomes kinetic energy, the spring uncoils and the vibrations cease. The balls now begin to cast shadows whose areas are directly proportional to the amount of kinetic energy associated with each ball. On impact, when kinetic energy is instantly changed to potential energy and then back to kinetic energy, the shadows disappear and the vest briefly vibrates. To aid students in judging the velocities of the balls relative to one another, the columns light and chime as the balls pass. Additionally, students can become one of the balls in the corridor, a camera attached to the center-of-mass of the bouncing balls, or a movable camera hovering above the corridor.



MaxwellWorld

This world has been designed to enable the examination of the nature of electrostatic forces and fields. MaxwellWorld occupies a cubical workspace with Cartesian axes displayed for convenient reference. Menus and a virtual hand are used for interaction in this world. Unlike NewtonWorld's menus, the menus in MaxwellWorld are attached to the left wrist just as a stopwatch (for left-handed users, the menu location can be on the right hand). This allows the menus to be removed from the field of view but keeps them immediately accessible since the user always "knows" where his or her hands are located. Executions of menu commands are confirmed by audible chimes.

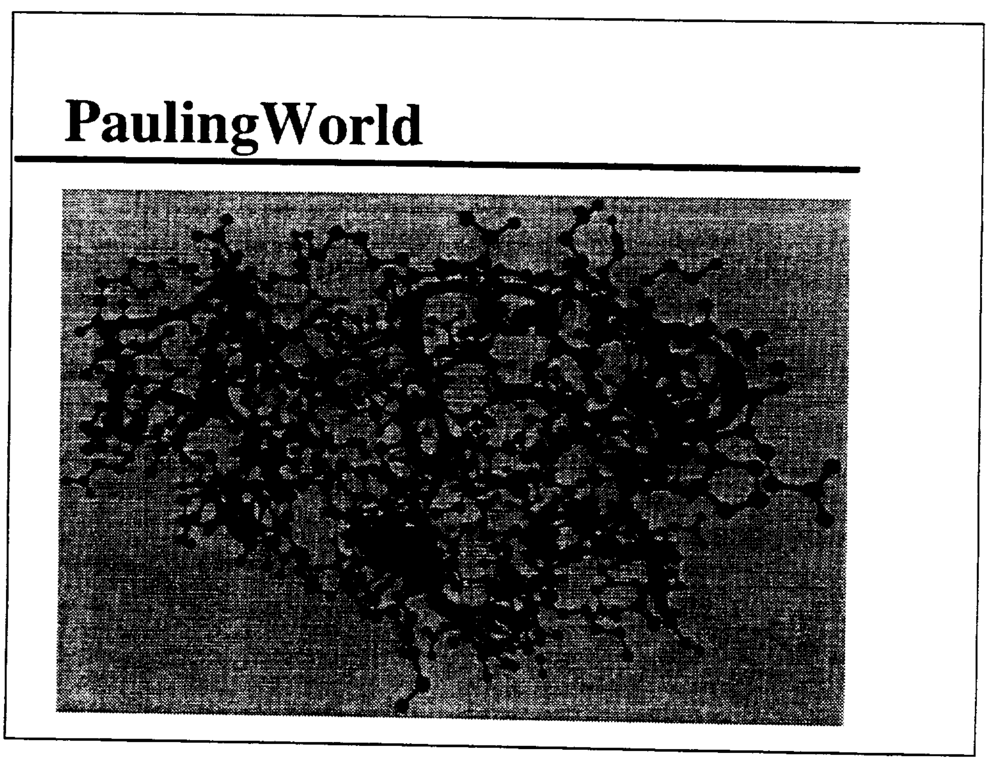
By using a ray extending from the index finger of the virtual hand students can place both positive and negative charges of various relative magnitudes into the world. Once a charge configuration is established, the force on a positive test charge, electric field lines, potentials, surfaces of equipotential, and lines of electric flux through surfaces, can all be instantiated, easily observed, and controlled interactively. For example, the tip of the ray can be attached to a small, positive test charge and a force vector associated with the charge depicts both the magnitude and direction of the force of the test charge (and, hence, the electric field) at any point in the workspace. The test charges can be "dropped" and used to visualize the nature of the electric field throughout a region or the entire world. Navigation in MaxwellWorld is accomplished by selecting the navigation mode, pointing the index finger in the desired direction of travel, and depressing the mouse button (see figure).



PaulingWorld

PaulingWorld is the most recently developed world and has been created to serve as both a teaching and "research" tool. PaulingWorld allows one to examine the structure of both small and large molecules from any viewpoint and in a number of single or mixed representations. One moves between representations by using the same menu approach that MaxwellWorld provides. Molecules can be represented in the familiar ball-and-stick form, as van der Waals' spheres, as a ribbon or "wireframe" backbone, and as icons that replace repetitive structures. In the latter case, the icons can be interrogated by selecting them with the index finger and depressing the mouse button. The icon is then replaced by a complete representation. Thus, the macrostructure of the molecule remains "iconic" while the region of interest is depicted in a standard representation of choice. Navigation is carried out as in MaxwellWorld (see figure).

To support the rapid examination of various molecules, structural data can be read in directly from pdb (protein database) files that are widely available on the world wide web, allowing a new molecule of interest to be built in a few minutes. Future extensions planned for PaulingWorld include the display of equipotential surfaces (defined as in MaxwellWorld) and the provision for interactively exploring the effects of atom removal and substitution through direct links to molecular modeling applications.



Issues...

- Hardware for virtual environments is still relatively costly and lacks robustness. Recent decisions by Disney (and others) to widely deploy these technologies for entertainment (Disney Quest) promise to reduce the cost and improve the reliability of the hardware.
- Connectivity will happen!
- Our group and others are working to improve the usability and reliability of the requisite software. We are confident that we will soon see commercially available software that is accessible to the educator/trainer and that is reasonably stable.
- Work goes on to determine the degree of fidelity needed to achieve specific educational or training outcomes.
- A major barrier is the resistance of the training and educational establishments to the introduction of new technologies.
- Integration of these new technologies with those current ones that have proven worthy should be done.
- Finally, we need more examples of good experiments that demonstrate objectively that these technologies are effective.

Issues . . .

- Hardware (Cost and Reliability)
- Connectivity (Bandwidth and Latency)
- Software (Usability and Reliability)
- Fidelity . . . How Much Is Enough?
- Organizational Culture
- Integration
- Documented Effectiveness

What's Next

Work that is underway, principally in the development of virtual environments for peacekeeping operations, is seeking to integrate intelligent agents into the virtual world. Such agents can serve as coaches, surrogates for missing team members, and as the drivers of avatars that play the role of the training "context." The approach taken to high level control is described in:

Jianping Shi, Thomas J. Smith, John P. Granieri, and Norman I. Badler, "Smart Avatars in JackMOO," *Proceedings of the 1999 IEEE Virtual Reality Conference*, Houston, Texas, Mar. 13-17, 1999.

Additional work is underway to improve both the appearance and the control that can be exercises (autonomously or by humans) of the avatars.
Some experiments on fidelity have been carried out and future experiments on training transfer are planned.

Finally, we look forward to new and improved hardware (at lower cost) that will make the deployment of these technologies feasible and cost effective.

What's Next?

- Adding "Intelligence" to VEs
  - Intelligent Agents
  - High-Level Control Mechanisms
- Improved Avatar Control
  - Non-Contact Tracking
  - Facial Expressions/Eyepoint
- Research Results on Fidelity, Training Transfer, . . .
- New Hardware (Autostereoscopic Displays, Wearable Computers, True 3D Interfaces, . . .)

Please Visit

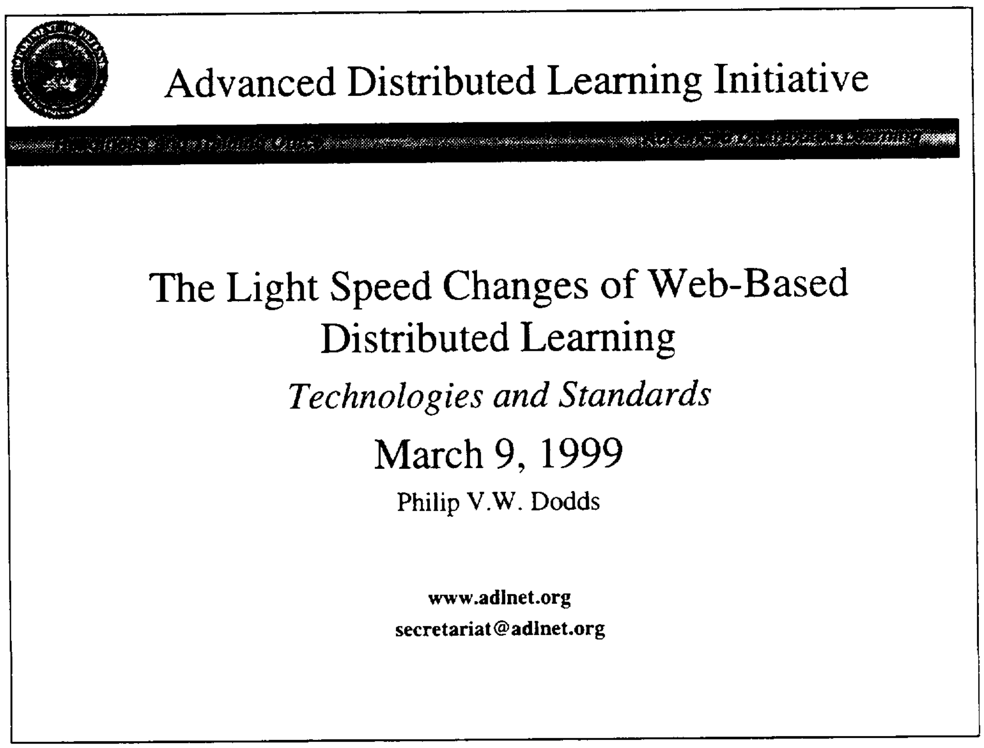
- http://www.vetl.uh.edu
- bowen@uh.edu
- 713-743-1238
  713-743-1198 (fax)
- R. Bowen Loftin
  Department of Computer Science
  Mail Stop CSC-3475
  University of Houston
  Houston, TX 77204-3475

The Light Speed Changes of Web-Based Distributed Learning Technologies and Standards

Philip V. W. Dodds
Randall House Associates, Inc.
Annapolis, MD

The Light Speed Changes of Web-Based Distributed Learning
Technologies and Standards

Philip V. W. Dodds
Randall House Associates, Inc.
9 Randall Court
Annapolis, MD 21401



Overview

Overview

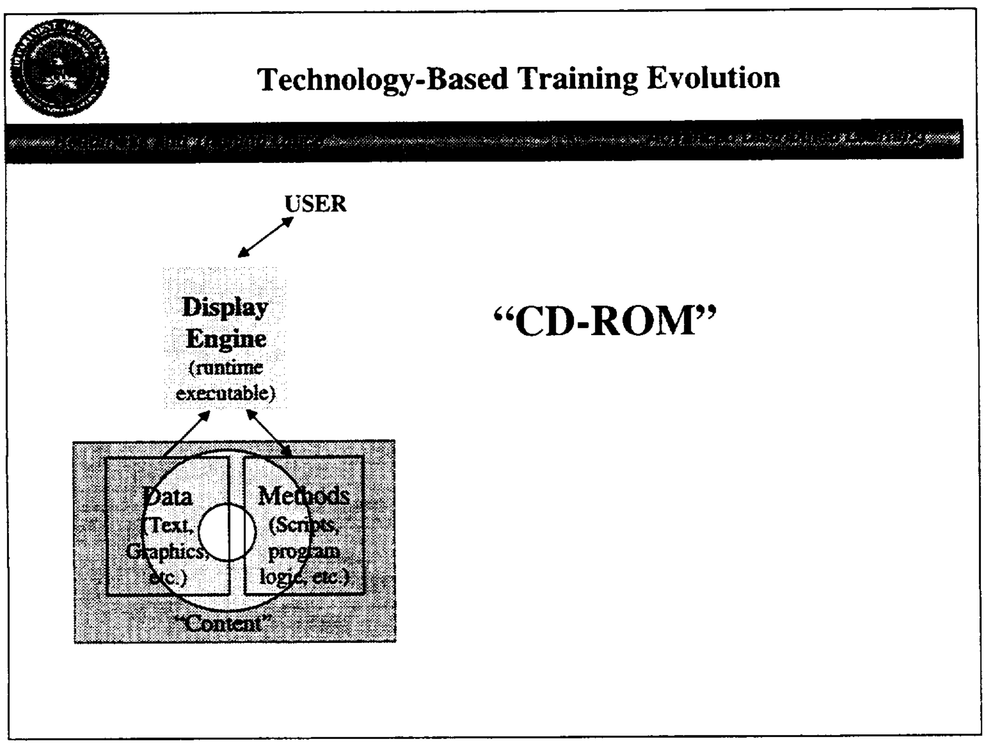
- Transition to web-based learning systems
  - *a very new way of building and delivering learning content*
- So-called "learning objects"
  - *"I don't really know what they are but I want them."*
- CMI/LMS revolution in progress
  - *"Oh boy! Look what I can do now!"*
- "Learning model" chaos
  - *torturous terminology and multifarious pedagogies*
- Proposed reference model
  - *a possible common approach?*
- Big open issues
  - *More work, thinking and discussion needed here...*
- Dodds' Crystal Ball

CM-ROM

As the CD-ROM technology evolved, multimedia content became all digital, and digital audio, video and pictures replaced the analog versions on a laser disc. All the same kinds of data were stored on a CD-ROM instead of floppy diskettes and laser discs.

Figuring out what the content was, and the model for delivery, remained essentially unchanged, however. You still had a local (proprietary) engine on each machine, and you moved the content around on CD.

Note: Many CD-ROM's also had copies of the run time engine on the CD, but that was to make it simpler to distribute and install. The engine still resided on the local machine and was installed only once as needed.

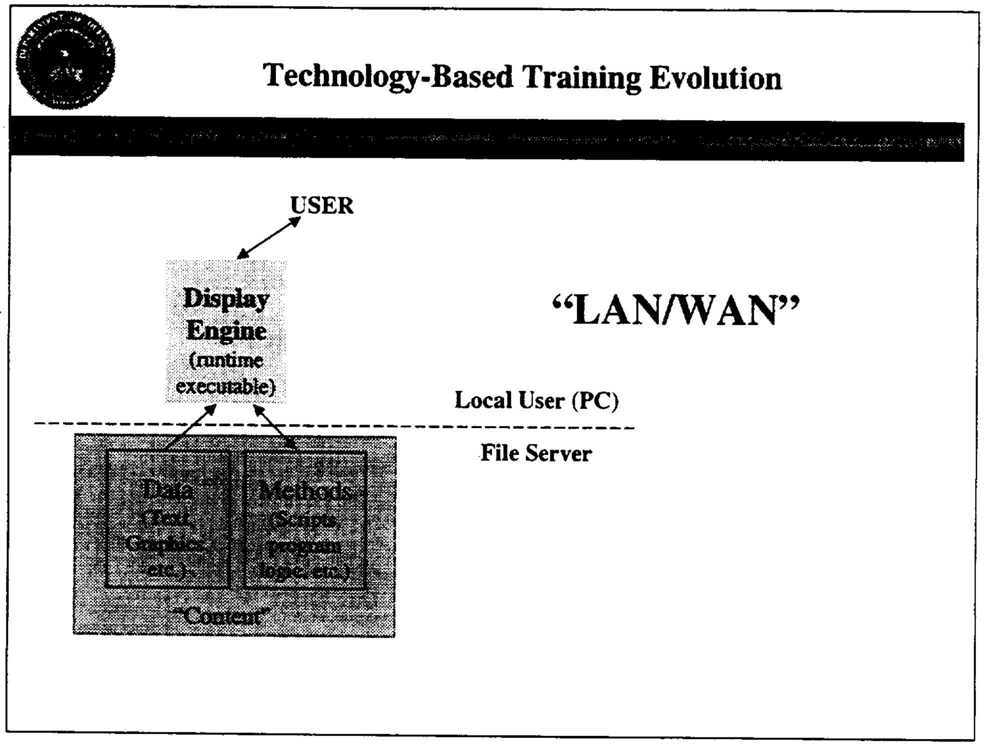


LAN/WAN

Organizations that already had installed local area networks quickly realized that they could locate the very same content that was on CD-ROM's onto a file server connected to many workstations. This eased administration and distribution headaches, but was limited to local networks with fairly high bandwidth and not too many users.

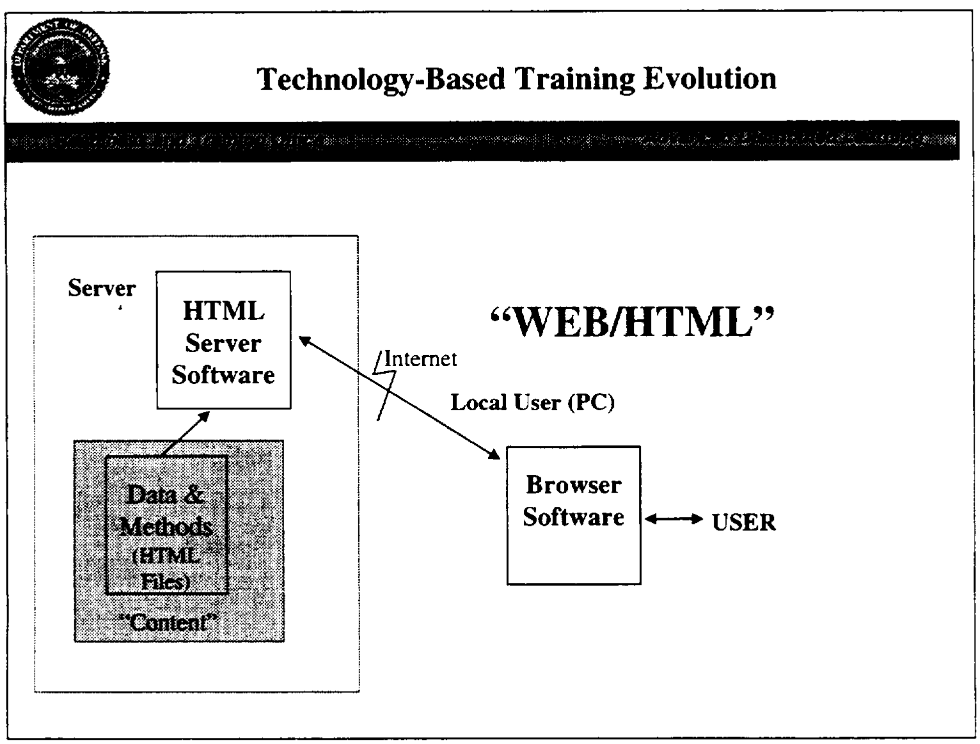
Once again, the model for what the content is and how it runs is no different than in the past; the run time engine had to be installed on the local machine, and files that were stored on the file server were simply downloaded to the local machine for rendering.

So far the content model remains the same.



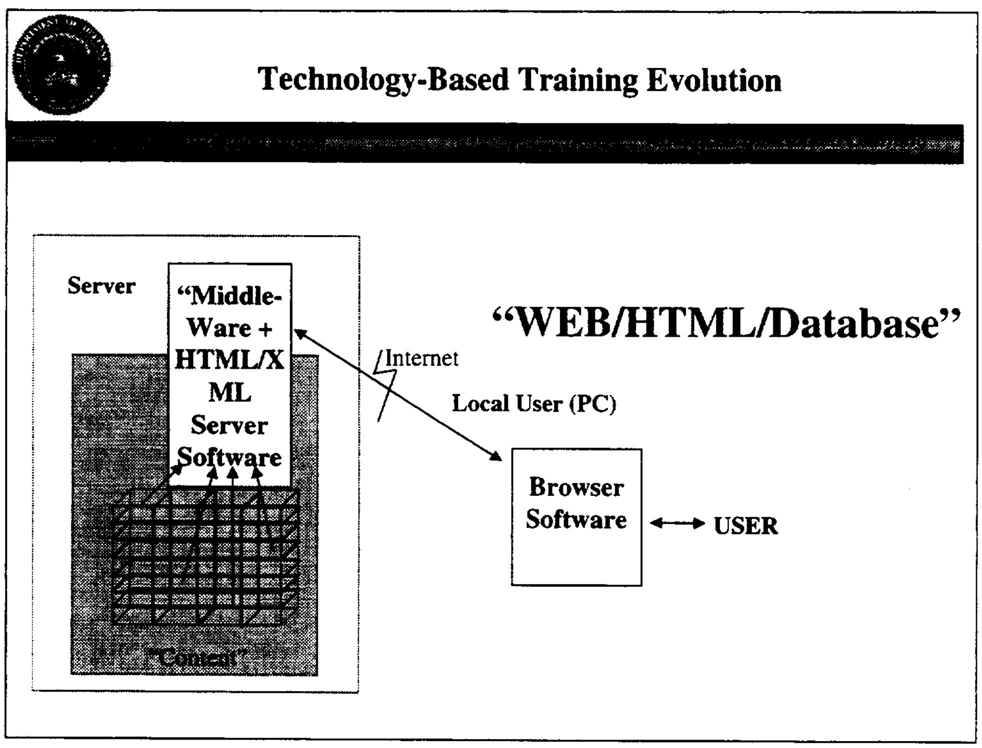
WEB/HTML

- When the Internet wave hit, there was a rush to convert content to the web. Initially content took the form of HTML.
- HTML, a very, very simple subset of SGML, essentially combines the logic of the content with the data. In this form, logic (presentation) is expressed as tags and data is the content between the tags.
- This is the first fundamental change of content form.
- In a web-based system a user has a generic, non-proprietary (in theory) browser installed on the local machine, replacing the proprietary run time engine of the past. This browser uses Internet protocols to request HTML files (using the hyper text transfer protocol, HTTP).
- A web server processes these requests from a browser and "serves" out the files.
- Here content is very, very simple (and not very robust): the content is a collection of HTML files. Period.



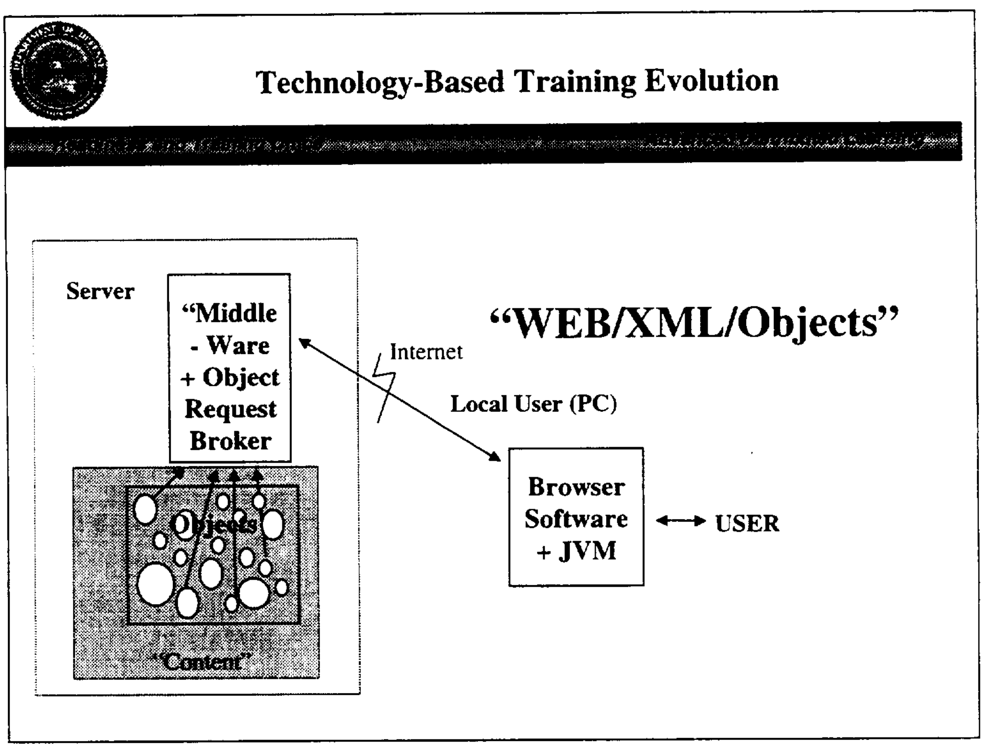
WEB/HTML/Database

- As web-based technology has continued to evolve, the management of content has become more and more complex. Increasingly there is a desire to separate the "raw" content from the display format through the use of (sometimes) complex middleware and databases.
- Middleware is software running on a server that knows where components of content are stored in a database and can retrieve these chunks, format them and send them to a browser on the fly. This allows the database to be created, modified and monitored in a semi-automated way, and opens the opportunity for data collection and management never before possible.
- For training, however, such an approach muddies the definition of content significantly. Now "content' consists of a particular database, it's data structure, the contents of the database, and the logic contained in the middleware on the server.
- For the first time we no longer can easily "move" content from one place to another (without replicating the entire server environment). On the other hand, we can manage data as never before. A trade-off.



WEB/XML/Objects

- In the future we look forward to the time when content is expressed in "tidy chunks" that encapsulate all the "data/resources and methods." Such chunks will be self-contained learning objects that can be stored, retrieved (by object request brokers) and executed remotely.
- This model, more than any of the others, most closely meets the high level requirements (the "-ilities" listed earlier). Many companies are working toward this long term goal and have made significant progress.
- However, for a true object-based learning environment to exist, a host of standards must be established and adopted on a wide-spread basis. This has not yet happened, but is underway.
- Since object technology, especially as applied to learning technology, is still in the early stages, we can expect that today's view of how standards and technology will evolve will change with time.
- So, object technology is a worthy goal, but we will have to migrate to this model step-by-step.



Example Definitions for "Learning Objects"

Example Definitions for "Learning Objects"

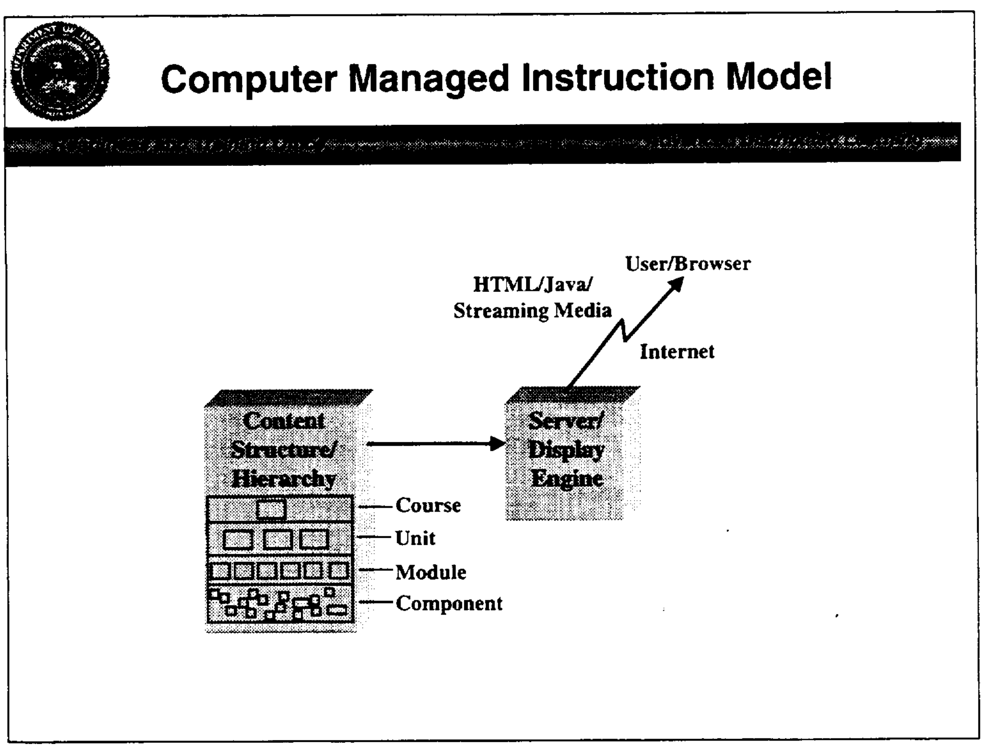
A Learning Object:

- "Has a specific instructional intent; are web delivered; are reusable; and have metadata."
- "Is any entity, digital or non-digital, that can be used, reused or referenced during technology-supported learning."
- "Is a 'chunk' of learning delivered on the web with learning objectives."

There is no useful common agreement among vendors; "Learning Objects" today are rarely based on software object technologies.

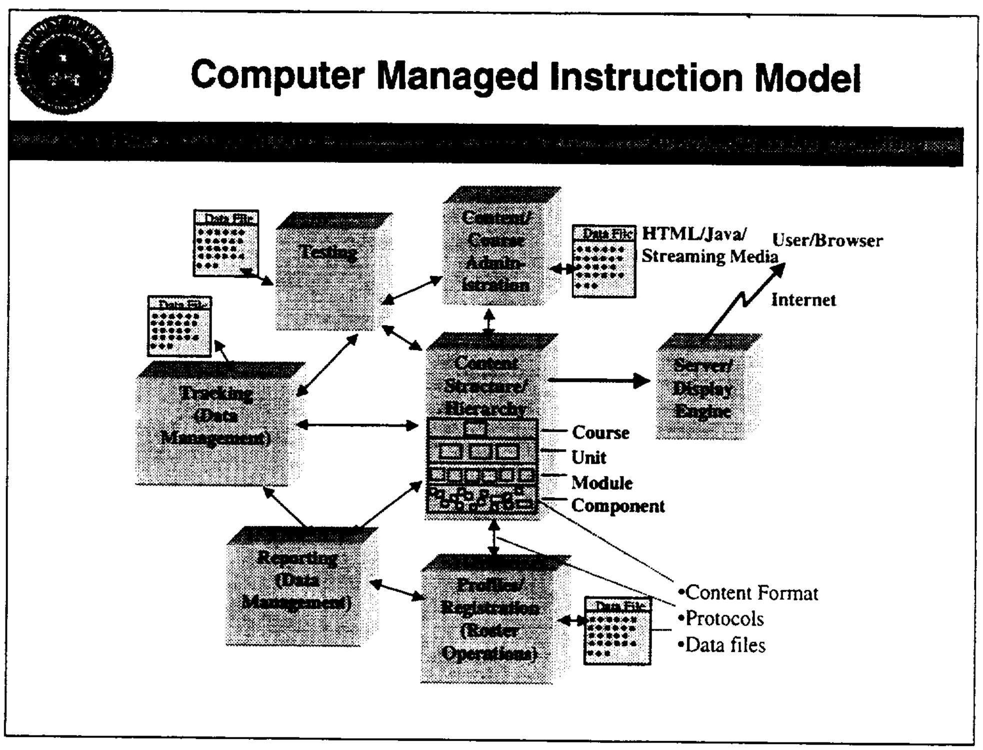
Computer Managed Instruction Model

- Returning to the discussion of "content," we see here a slightly different model for content, this time broken down in a hierarchy that suggests how the content might be assembled.
- In this illustration, the content might be residing in a kind of "repository" where very "granular" components might be added together to form modules, units, and then courses (many people have variations on this sort of hierarchy). The point is that there is an interest in very "granular" content that can be reused and reassembled.
- The problem is that today's technology is not up to the task of managing (or creating) small components or modules. When true object-based learning technology evolves, it is hoped that this model will be realized.
- Right now, only the top two layers are typically created using COTs authoring tools.
- Now let's take a look at what we need to add to this content model to create a "Computer Managed Instruction" (CMI) system.



Computer Managed Instruction Model

- There are three key elements in this model that affect the robustness of the environment:
  - **the format of the content**
  - **protocols for exchanging data among the modules**
  - **the exact data that is gathered and exchanged in the system.**
- The content format defines what can and cannot be displayed. If it is only HTML, there is little interactivity; for example, Java is more robust. The form of the content must be supported by the client's browser or workstation.
- The protocols for exchanging content and data define the underlying architecture that is used. A LAN-based CMI system might use files, for example (e.g., AICC), whereas an Internet-based approach might use XML/RDF (e.g., IMS). For systems to inter-operate, there must be compatible protocols.
- Data that is gathered must likewise be compatible and consistent. The tracking module, for example, must know what data the registration/profiles module gathers and sends.



Web Technology - Realities

- It appears that 1998 was a transition year from relatively limited web capabilities to a more capable interactive form. The standards needed for "next generation" content are still evolving. It seems likely that systems and content developed this year will probably need to migrate to new forms over the next few years.
- XML is expected to play a central role in stabilizing web formats, but many other aspects of content design are still to be worked out.
- Meanwhile, limited bandwidth will continue to restrict the complexity of content. Even as bandwidth increases locally or from a particular server, Internet loads and bottlenecks into and out of local area networks will persist for a considerable time. Even as bandwidth increases, its availability will remain limited as content creators add back multimedia content eating up available bandwidth as fast as it is created.

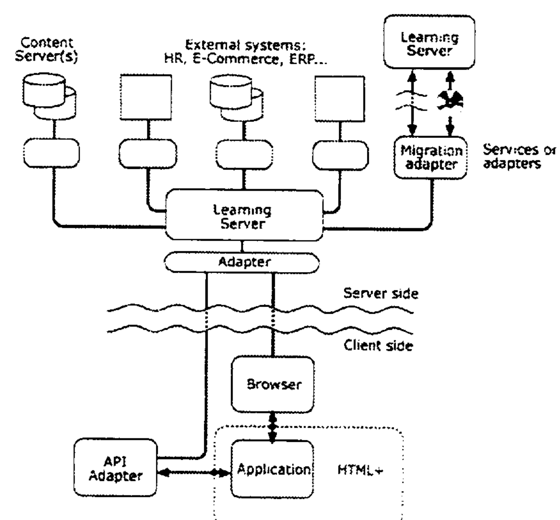
Web Technology - Realities

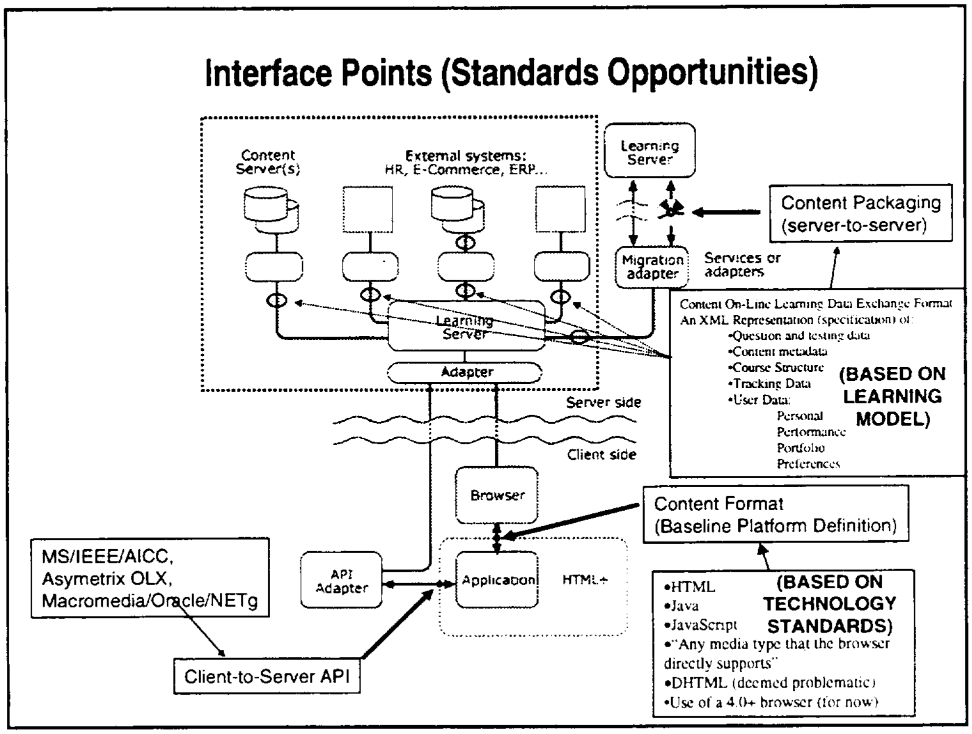
- "True" object technology is further away than originally hoped
- Content format is in flux
- XML related standards are still emerging and may help stabilize content and data formats -- eventually...
- There is no "common courseware object model"

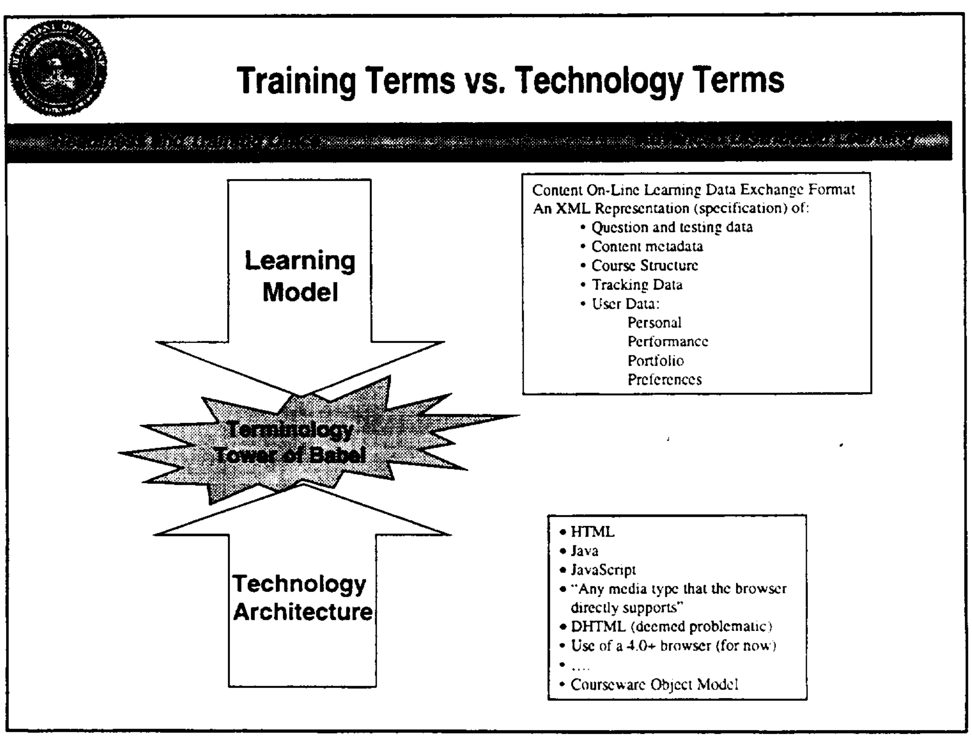
Standards Efforts

- IEEE 1484
  - *de jure* standards body
- IMS
  - Closer to "Consortium" model
- AICC
  - Airline industry-based
- Macromedia/Oracle/Netg + Asymetrix
  - *de facto* standards
- ADL Technical Work Group
  - Catalyst

An Architectural Model for Distance Learning




Interface Points (Standards Opportunities)
Content Server(s)
External systems: HR, E-Commerce, ERP...
Learning Server
Content Packaging (server-to-server)
Migration adapter
Services or adapters
Learning Server
Adapter
Content On-Line Learning Data Exchange Format
An XML Representation (specification) of:
•Question and testing data
•Content metadata
•Course Structure
•Tracking Data
•User Data:
Personal
Performance
Portfolio
Preferences
(BASED ON LEARNING MODEL)
Server side
Client side
Browser
Content Format (Baseline Platform Definition)
MS/IEEE/AICC, Asymetrix OLX, Macromedia/Oracle/NETg
API Adapter
Application
HTML+
•HTML
•Java
•JavaScript
(BASED ON TECHNOLOGY STANDARDS)
•"Any media type that the browser directly supports"
•DHTML (deemed problematic)
•Use of a 4.0+ browser (for now)
Client-to-Server API


Training Terms vs. Technology Terms
Learning Model
Terminology Tower of Babel
Technology Architecture
Content On-Line Learning Data Exchange Format
An XML Representation (specification) of:
• Question and testing data
• Content metadata
• Course Structure
• Tracking Data
• User Data:
Personal
Performance
Portfolio
Preferences
• HTML
• Java
• JavaScript
• "Any media type that the browser directly supports"
• DHTML (deemed problematic)
• Use of a 4.0+ browser (for now)
• ....
• Courseware Object Model

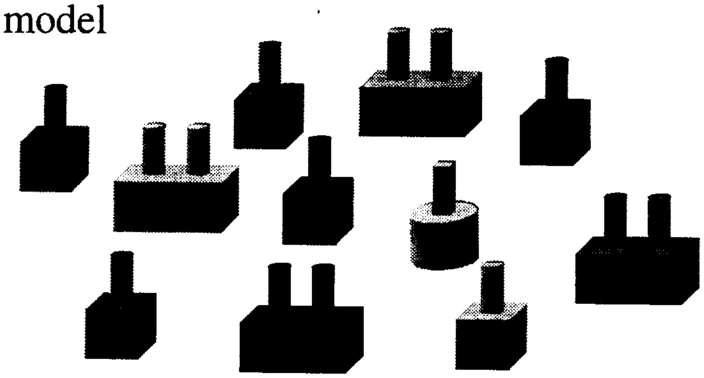
Multiple Learning Models and Terms

There are many so-called "Learning Models"

- AICC (Course, Block, AU)
- Oracle (Course, Topic Group, Topic, Activity)
- Macromedia (Course, Block, AU, Knowledge Object)
- Asymetrix (Lesson, Block, Module)
- DoD Enterprise Model (Course Module, Lesson, Learning Objective)
- Army (Course, Module, Lesson, Learning Objective, Learning Step)
- Canadian DoD (Course, Performance Objective, Enabling Objective, Teaching Point)

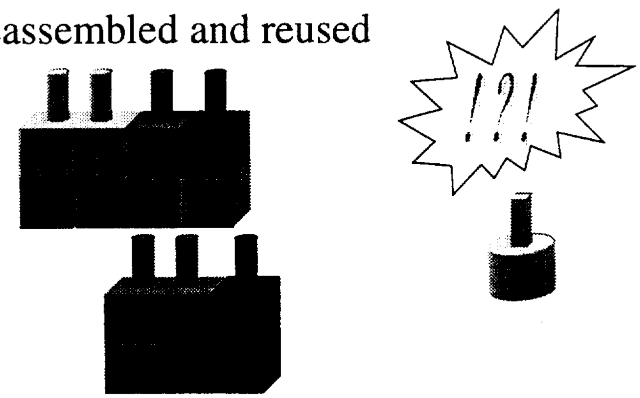
Shareable Courseware Object Model

For reuse of component pieces to be possible, they must be built to a common "object" model



Shareable Courseware Object Model

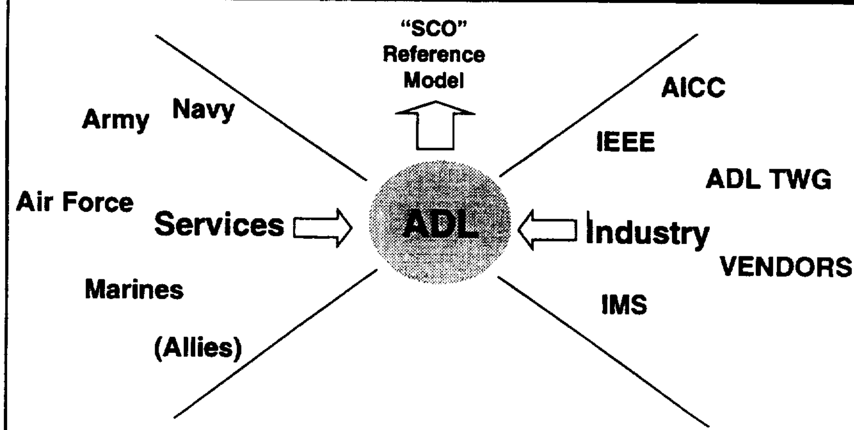
Components that share a common model can be reassembled and reused



ADL Approach

- Examine military training learning models
- Develop a common "Shareable Courseware Object Reference Model"
- Map learning models to SCO reference model (to determine standards requirements)
- Submit requirements to appropriate groups

ADL Process



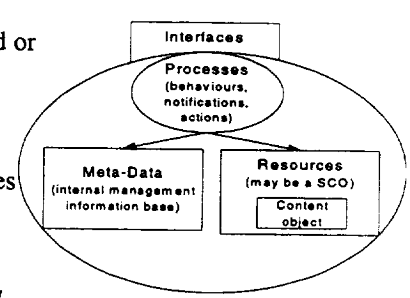
ADL Proposed "Learning Object" Definition

Suggest we use "Shareable Courseware Object" (SCO) instead of "Learning Object" (LO)

Two definitions taken together:

"An SCO is an entity, that can be used, reused or referenced during technology-supported learning."

"An SCO is a software "package" that includes metadata, processes and resources."

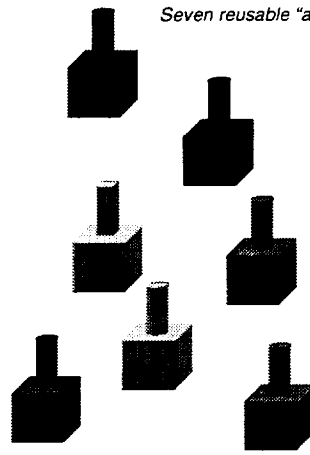


Combines learning and technical definitions

SCO Reference Model

- Tier 0 ("Atomic")

  Reusable Interaction

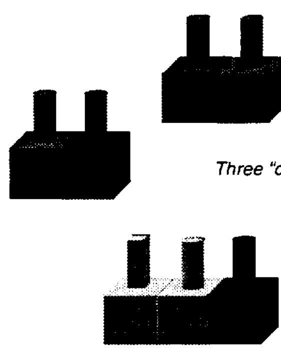


<u>Some Example Terms:</u>
Interaction
Activity
Knowledge Object
Bean/component
Learning Step
Teaching Point

SCO Reference Model

- Tier 1 ("A collection of atoms")

  Nesting Container



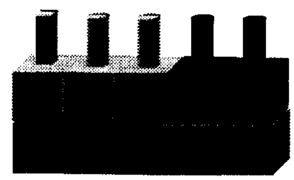
<u>Some Example Terms:</u>
Assignable Unit
Topic
Module
Learning Objective
Enabling Objective

SCO Reference Model

- Tier 2 ("Multiple collections of atoms")

  Nesting Container

A multiple collection of atoms



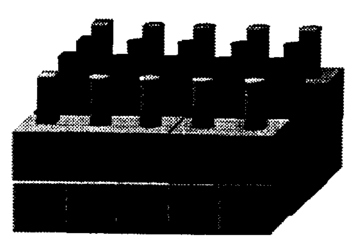
Some Example Terms:
Block
Topic Group
Unit
Lesson
Performance Objective

SCO Reference Model

- Tier 3 ("Packaged Collection")

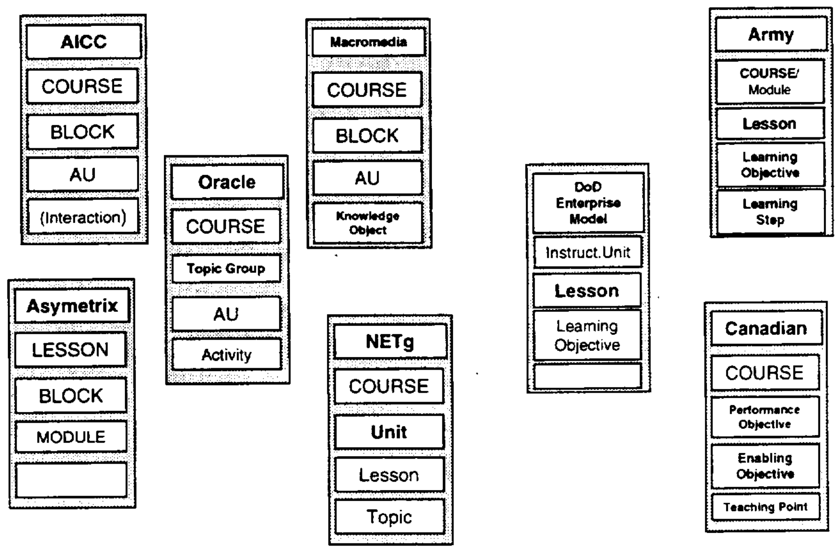
  Outer Container

One packaged collection



Some Example Terms:
Course
Module
Instructional Unit

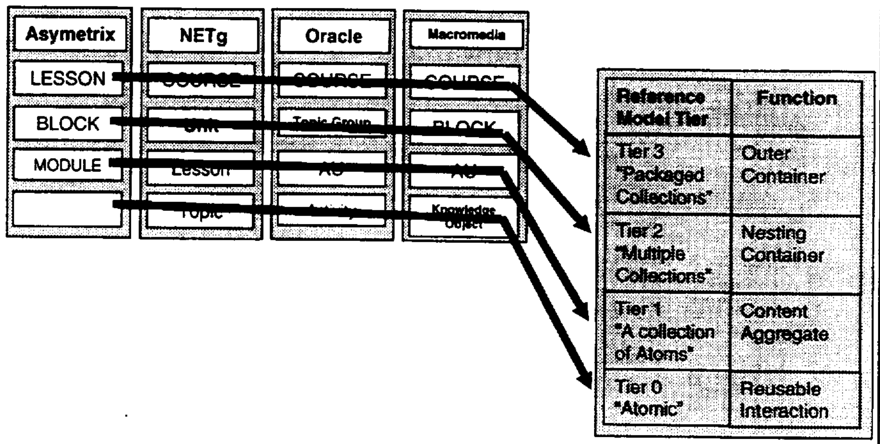
"Learning Models"



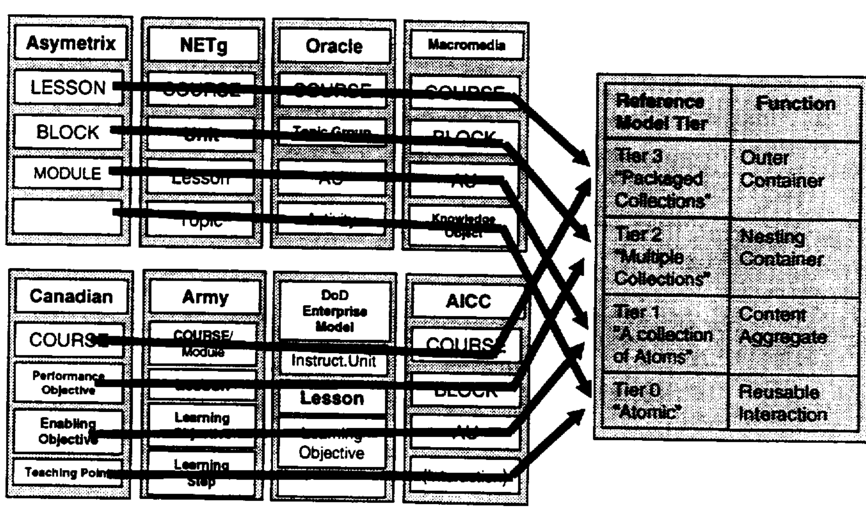
ADL Common SCO Reference Model

Reference Model Tier	Function
Tier 3 "Packaged Collections"	Outer Container
Tier 2 "Multiple Collections"	Nesting Container
Tier 1 "A collection of Atoms"	Content Aggregate
Tier 0 "Atomic"	Reusable Interaction

Mapping to the Common SCO Reference Model

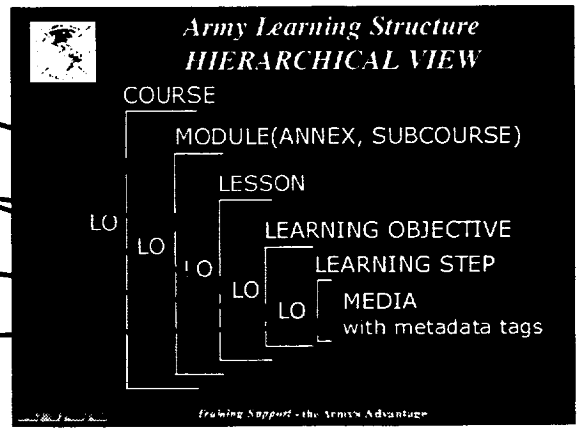


Mapping to the Common SCO Reference Model



Mapping the Army Learning Structure to the Reference Model

Reference Model Tier	Function
Tier 3 "Packaged Collections"	Outer Container
Tier 2 "Multiple Collections"	Nesting Container
Tier 1 "A collection of Atoms"	Content Aggregate
Tier 0 "Atomic"	Reusable Interaction



Open Issues

- Definition of reference model is not complete
- Mapping of existing models to reference model may be imperfect or may not work as expected

Open Issues

"If it were easy, everyone would be doing it..."

- BIG questions about how to represent:
  - assignment (course structure)
  - objectives
  - content

The SCO reference model is focused on content; we also need a CMI model & standards

- BIG debates over terminology
- BIG transition from the old stand-alone courseware (internalized structure and data), and web-based content delivery (server-managed course management)
  - a lot of data must now be exposed (and standardized)
  - new features/functions are being thought about faster than they can be standardized (e.g., on-line collaboration, back office interfaces, and more)

Even Bigger Issues Loom...

What about:

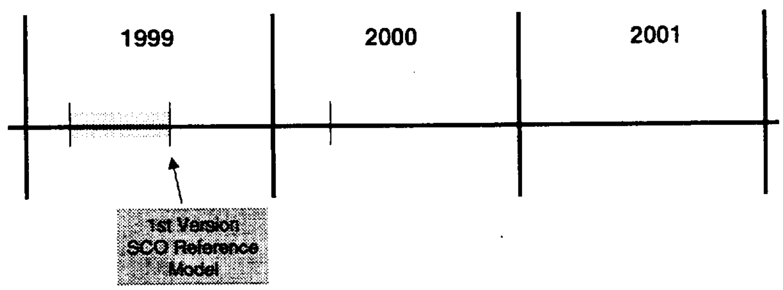
- Embedded simulation
- Intelligent tutoring systems
- Performance support

"Oh, that's no problem; the model accommodates all of these..."

Nope! Not really...

These are <u>not</u> addressed very well (or at all) in most standards efforts or products

Dodds' Crystal Ball



pdodds@rhassociates.com

Intelligent Tutoring Systems: Then and Now

J. Dexter Fletcher
Institute for Defense Analyses
Alexandria, VA

Intelligent Tutoring Systems: Then and Now

J. Dexter Fletcher
Institute for Defense Analyses
1801 North Beauregard Street
Alexandria, VA 22311

This talk was presented on March 10, 1999 by Dr. Dexter Fletcher who is a research staff member at the Institute for Defense Analyses (IDA) in Alexandria, Virginia. IDA's sole function is to perform studies and analyses of scientific and technical matters for the Office of the Secretary of Defense (OSD). As usual, this presentation represents the views of the presenter and does not represent official policies or positions of either IDA or OSD.

The phrase Intelligent Tutoring Systems (ITS) covers a form of computer-based instruction (CBI) that has also been called intelligent computer-assisted instruction (ICAI). ITSs may be as intelligently or un-intelligently designed as any other form of CBI. 'Intelligent' in this case refers to a particular functionality that is the goal of these systems and is further defined in this discussion.

Although much in this presentation may be relevant to private and public sector education, it is focused on applications of ITS to military training. The presentation is also fairly compressed. More could be said about all the issues it raises.

Intelligent Tutoring Systems: Then and Now

Workshop on Advanced Training Technologies and Learning Environments

9-10 March 1999

J. D. Fletcher
Institute for Defense Analyses

Summary

The object of this presentation is to suggest that:

Substantial improvements in instructional effectiveness may be obtained through increased tailoring of military training to the needs and capabilities of individual learners. Routine provision of a single human instructor for every student is, with a few significant exceptions, prohibitively expensive.

Individualization of military training and its substantial benefits can be significantly increased through the use of ITSs which generate instructional interactions tailored to the needs of individual students. They substitute technology for human labor, thereby making increased individualization economically feasible.

ITSs can thereby make high-quality training available any time, any place and tailored to the needs of any student.

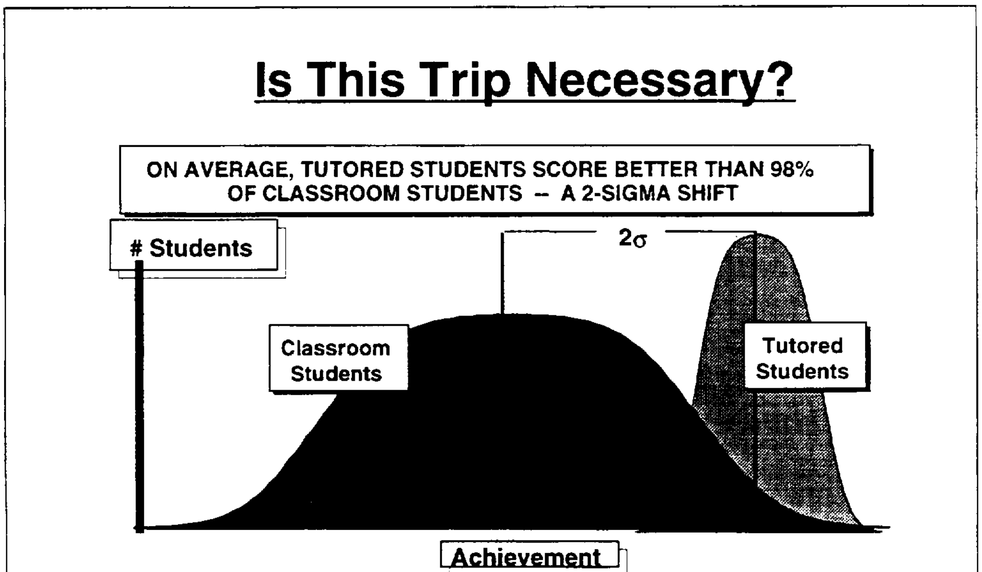
There are no unmanned systems, nor automatic operations. By increasing the accessibility of high-quality, individualized training whenever, wherever, and to whomever it is needed, ITSs can improve readiness and operational effectiveness.

In Short ...

- **Individualization is a good thing, but expensive.**
- **ITSs increase individualization through real-time, on-demand generation of relevant instructional interactions.**
- **ITSs thereby make DoD training more accessible and relevant than is otherwise possible.**
- **ITSs will improve operational effectiveness.**

Individualized Instruction

Why should we care about tailoring training to individual students? Why should we care about individualized instruction? Benjamin Bloom and his students attempted to answer these questions by comparing the achievement of individually tutored students (one (human) teacher per one student) with that of classroom students (one (human) teacher per 28-32 students) (Bloom 1984). They found the difference in achievement to be about two standard deviations, which means, roughly, that tutoring improved the achievement of 50th percentile students to that of 98th percentile students.



Adapted From: Bloom, B. S. The Two Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. Educational Researcher 13, 4-16 (1984)

Technology-Based Training May Answer Bloom's Challenge

Comments on Bloom's research might include the following:

Intuitively we would expect tutoring to effect some improvement in achievement or learning. What is remarkable about Bloom's finding is the size of the difference. Two standard deviations is a very large difference.

Education is a worthy end in its own right. It is preparation for life and its learning objectives can be more flexible and negotiable than they are in training. Educators are likely to emphasize the maximization of learning assuming fixed costs. Training is a means to an end -- operational effectiveness or readiness in the military and perhaps productivity in the private sector. Its learning objectives are determined by the needs of jobs and careers and not by the variable needs of students. Trainers in both the public and private sectors, therefore, are likely to emphasize the minimization of costs to accomplish fixed levels of learning. The ultimate goal of DoD training is improved readiness and operational effectiveness.

Bloom's research indicates an instructional imperative and an economic impossibility. We need to break out of this dilemma. Technology and specifically ITSs may provide that.

Applications of computer technology are in their infancy. ITSs most probably apply an old metaphor (one-on-one tutoring) to a new technology just as we originally attempted to make carriages run without horses or telegraphs operate without wires. The technology-based instruction analog to the automobile, radio and television has yet to emerge and may not even have been imagined, but it may well be in the direction of intelligent tutoring systems.

Some Comments on Bloom's Theme

- **It is a big difference.**
- **We (DoD) are in this business to increase accessibility and operational effectiveness.**
- **We face a training imperative and an economic impossibility.**
- **What we seek will not be what we get.**

Question: What do ITSs have to offer that we can't get otherwise?

The Challenges of Classroom Instruction: Pace

At this point it seems reasonable to ask what ITSs offer that might solve Bloom's two standard deviation challenge, or as Bloom calls it, the two sigma problem - sigma being a common symbol for standard deviation.

The variability that classroom teachers must accommodate is daunting:

The ratio of time needed by the slowest Kindergarten students to build words from letters to the fastest Kindergarten students was found by Suppes (1964) to be 13:1.

Similar ratios for grade 5 students to master a unit of social studies were found to be 3:1 and 5:1 (Gettinger and White 1980).

In two studies of rates of learning by hearing impaired and native American students the ratios were found to be 4:1 (Suppes, Fletcher and Zanotti 1975 and 1976).

Based on a range of research findings, Carroll (1970) estimated the overall ratio for elementary school students to be 5:1.

Even among highly selected students at a major research university the ratio needed by undergraduates has been found to be 7:1 (Corbett 1998, personal communication).

NB: Individual Differences in Pace

- **Ratio of time needed to build words from letters in grade K -- 13:1**
- **Ratios of time needed to learn in grade 5 -- 3:1 and 5:1**
- **Ratios of time needed by hearing impaired and native American students to reach mathematics objectives -- 4:1**
- **Overall ratio of time needed to learn, K-8 -- 5:1**
- **Ratio of time needed by college undergraduates to learn LISP -- 7:1**

The Challenges of Classroom Instruction: Interactivity

Those who study classroom interactions have found that groups of students ask about three questions an hour and that any single student in a class asks about 0.11 questions an hour. By contrast, students in individual tutorial sessions have been found to ask 20-30 questions an hour and have been required to answer 117-146 questions an hour. Finally, students taking drill and practice computer-based instruction answer 8-12 questions a minute - questions that have been especially selected to meet the needs of the individual student and that are immediately graded.

It is not difficult to see why the intensity of tutorial settings, either based on human tutoring or computer tutoring might produce large differences in achievement, or reach instructional objectives substantially more efficiently.

NB: Classroom and Tutorial Interactivity

- **Average number of questions asked by a class during a classroom hour: 3**
- **Average number of questions asked by any student during a classroom hour -- 0.11**
- **Average number of questions asked by a student during a tutorial hour -- 21.1 (Research methods); 32.2 (Algebra)**
- **Average number of questions students answered during a tutorial hour -- 117.2 (Research Methods); 146.4 (Algebra)**
- **Average number of questions answered per minute in mathematics drill and practice CBI -- 10**

What Individualization Has Technology Provided in the Past?

What about drill and practice? What about this technique of computer assisted instruction that has been used from the late 1950s? What level of tutorial instruction was it able to provide?

Notably, it could (a) accommodate each student's rate of progress, allowing as much or as little time as each individual student needed to reach instructional objectives; (b) adjust the sequence of content to each student's needs; (c) adjust the content itself - different students would receive different content; (d) make the instruction as easy or as difficult as necessary; (e) adjust to the learning style (e.g., verbal versus visual) that was most appropriate for each student. All of these capabilities have been available and used in computer-based instruction since its inception in the 1950s (e.g., Atkinson and Fletcher 1972; Suppes and Morningstar 1972; and Fletcher and Rockway 1986).

What About That Drill and Practice?

It tailored instruction to individuals by adjusting:

- **Rate of progress**
- **Sequence of content**
- **Content itself**
- **Difficulty**
- **Style**

What Are the Unique Contributions of Intelligent Tutoring Systems?

What then is left for ITSs to provide? What can we get from them that is not otherwise available? Three candidate functionalities may deserve mention:

1) The ability to capture the interactions of one-on-one tutoring.

2) The ability to generate instructional material and interactions as needed rather than foresee and pre-store all such materials and interactions needed to meet all possible eventualities.

3) Related to the above, the ability to allow either the computer or the student to ask open-ended questions and initiate instructional dialogue as needed or desired.

The key defining characteristic of ITSs then, is not application of computer techniques from artificial intelligence or knowledge representation, although these may be essential, but rather the functional capability to generate in real-time and on-demand instructional interactions that are tailored to student requests and/or needs. It was this generative capability that motivated DoD to invest in ITSs in the first place (Fletcher 1988). The motivation was to reduce or eliminate the high costs of foreseeing or predicting all possibly needed materials and interactions and then programming them into computer-based instruction.

What, Then, Do We Get From ICAI/ITS?
What's New?

Perhaps three things:

- **Individualized Tutoring (versus Automated Teaching)**
- **Generative Capability (versus Pre-stored Material)**
- **Mixed-Initiative Dialogue (versus Computer Initiated Dialogue)**

The key may be real-time, on-demand generation of instructional interactions tailored to student requests and/or needs.

An Example of An Intelligent Tutoring System At Work

An example of these capabilities at work may be found in a student-computer mixed-initiative dialogue taken from the SOPHIE (Sophisticated Instructional Environment) system (Brown, Burton and DeKleer 1982). SOPHIE used a general-purpose electronic simulator to represent an electronic power supply and create a reactive, tutorial environment for training electronic troubleshooting. SOPHIE employed a semantic network of information about the power supply, one of the most sophisticated natural language understanders ever built into an ITS, and an articulate, expert troubleshooting capability that could not only solve arbitrary faults introduced by students or instructors, but also explain to the student how it solved them.

The reactive environment in SOPHIE responded in a veridical fashion to the actions and decisions of a student engaged in practical problem solving. It employed strategies akin to those used by the student to make inferences about the circuit based on the student's measurements. Because the inferencing strategies were like those of the student, SOPHIE could determine which deductions the student was using, construct a model of his/her state of knowledge, and use this model to generate explanations in terms that the student was prepared to understand. SOPHIE highlighted the need for intelligent training systems not only to solve arbitrary problems, but to do so in a way that can be explained to students and be understood by them.

The first student-SOPHIE interaction is a straight-forward question and answer. SOPHIE has introduced a fault into the simulated power supply and the student is trying to locate and repair it.

ITS Example
Troubleshooting Tutorial Dialogue From SOPHIE

•
•
•

Student: What is the base emitter voltage of the voltage limiting transistor?

SOPHIE: The base emitter voltage of Q5 is -.037 volts.

Student: What about for the current limiting transistor?

SOPHIE: The base emitter voltage of Q8 is .349 volts.

An Example of An Intelligent Tutoring System At Work (Cont'd.)

The second student-SOPHIE interaction is a little less straight-forward because, in addition to simulating the faulted power supply, SOPHIE must be able to parse the dialogue sufficiently to know that the student is asking for the base emitter voltage of transistor Q8. The meaning of the student's question depends heavily on the context established by his prior question.

The third student-SOPHIE interaction continues its dependence on SOPHIE's understanding of context and additionally requires SOPHIE to extract the answer by simulating both the faulted power supply and one that is operating correctly.

The fourth student-SOPHIE interaction indicates a clear step beyond what Brown et al. described as a knowledgeable system to an intelligent system. SOPHIE has progressed from a knowledgeable parsing of its dialogue with the student and simulation of various states of the power supply to a system exercising tutorial intelligence. It shadowed the student's solution path, modeled the student's troubleshooting hypotheses, determined that they are incorrect, elected to capture the dialogue initiative back from the student, and is undertaking a series of tutorial interactions intended to lead the student back to a more correct approach to the problem.

It is difficult to imagine any way to achieve this level of instructional functionality without a generative capability, which is, then, a defining characteristic of ITSs, distinguishing them from other forms of computer-based instruction.

ITS Example (Cont'd.)
Troubleshooting Tutorial Dialogue From SOPHIE

Student: What should it be?

SOPHIE: In a working circuit, the base emitter voltage of Q8 is .593 volts.

Student: Replace Q8.

SOPHIE: I am going to ask you some questions about how Q8 is faulted. Are any junctions shorted?

•
•
•

What's the Underlying Technical Idea?

Achieving these instructional functionalities requires a technical idea. This idea was first articulated by Jaime Carbonell in 1970. It avoids ad-hoc frame-oriented instructional approaches such as those seen in programmed texts and is best described as "intrinsic programming" (Crowder 1959). This frame-oriented approach is still found ubiquitously in computer-based instruction. In its place, Carbonell recommended an information systems orientation approach or, as we might say today, an approach based on knowledge representation.

What Was the Underlying Technical Idea?

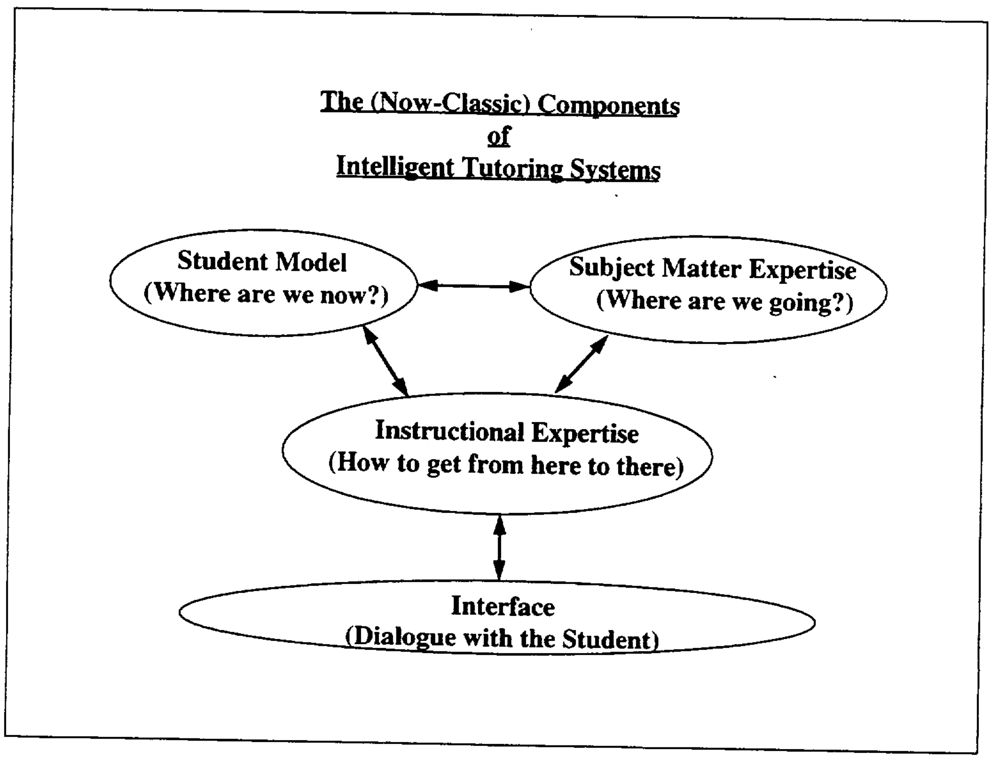
Carbonell, J. R., "AI in CAI: An Artificial Intelligence Approach to Computer-Assisted Instruction," *IEEE Transactions on Man-Machine Systems*, Vol. 11, 1970, pp. 190-202.

The key idea was:

Information Systems Oriented (ISO) Systems
in place of
Ad-hoc Frame Oriented (AFO) Systems

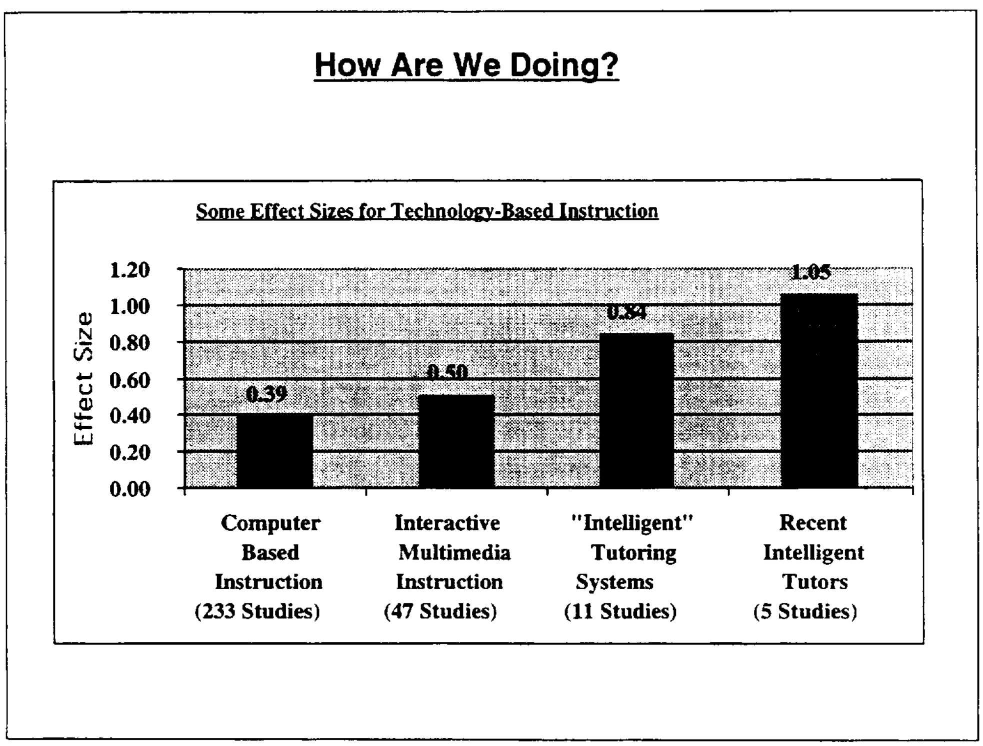
The Structure of Intelligent Tutoring Systems

An approach such as that defined by the functionalities possessed by SOPHIE and other ITSs and described by Carbonell involves computer representation or modeling of the student, the subject matter, and expert tutoring (Fletcher 1975). This approach, involving information systems and knowledge representation, is now commonly understood and found to a significant degree in ITSs. As Regian et al. (1996) have emphasized, our ability to represent human cognition has gained considerable potency with advances made in cognitive science over the last ten years, thereby enhancing substantially the promise and capabilities of ITSs.



Are We Getting Anywhere? Effectiveness.

After more than twenty, or perhaps thirty years (Feurzeig, et al. 1964) of developing ITSs, it may be reasonable to ask how they, or we, are doing. Are they instructionally more effective than what we have? Are they getting better? What data do we have to support the intuitive appeal of ITSs? In a review of 233 empirical evaluations almost entirely made up of standard (non-ITS) computer-based instruction compared with conventional classroom approaches, Kulik (1994) reported an effect size advantage for computer-based approaches of about 0.39 standard deviations, or roughly an improvement of 50th percentile students to about the 65th percentile of achievement. In an attempt to determine the advantages to instruction added by multimedia capabilities, Fletcher (1990) reviewed 47 evaluations of interactive multimedia instruction compared to conventional classroom approaches, and found an effect size advantage for these approaches of about 0.50 standard deviations, or roughly, an improvement of 50th percentile students to about the 69th percentile of achievement. A review performed for this workshop involving 11 ITS evaluations found an effect size advantage for ITSs of about 0.84 standard deviations, or roughly, an improvement of 50th percentile students to about the 79th percentile of achievement. In a recent review covering five evaluations of the SHERLOCK ITS, Gott, Kane and Lesgold (1995), show an overall effect size advantage for SHERLOCK of about 1.05 standard deviations, or roughly, an improvement of 50th percentile students to about the 85th percentile of achievement. We cannot say if we will achieve Bloom's target improvement of two standard deviations, or even exceed it, but the available evidence suggests that we are progressing in the right direction.

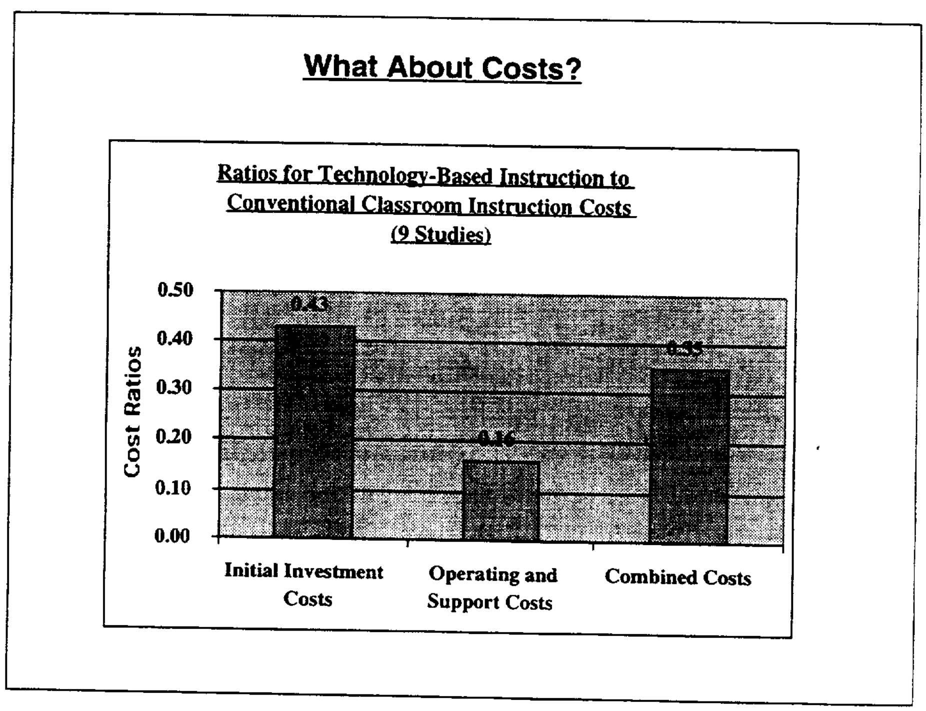


Are We Getting Anywhere? Costs.

Nearly all administrative decision making must consider what is to be gained in the light of what must be given up to get it. Generally, these decisions involve tradeoffs between costs and effectiveness. Decision making in training is no different. Costs must be considered as well as effectiveness.

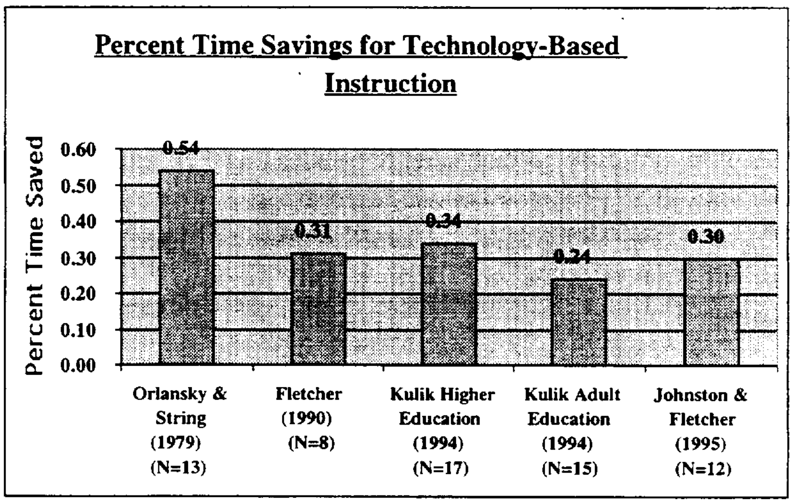
Most ITSs intended for DoD training simulate devices that students must learn to operate or maintain. These devices may cost 1-2 orders of magnitude more than a desktop computer system used to host an ITS. This consideration suggests that favorable cost arguments can be made for ITSs. Unfortunately, little such data exists. However, cost data can be extracted from evaluations that used adequate models of costs to assess the benefits of computer-based instruction using simulated equipment.

A review performed for this presentation found nine evaluations in which the performance of computer-based instruction students trained on simulated equipment and tested on real equipment was found to be at least as good as that of students trained using only real equipment. In fact, the performance of the computer-based instruction students was superior in every case, but the requirement for inclusion in this review was that it only be as good. The ratios of costs for the computer-based instruction to the more conventional instruction using real equipment are shown in three categories; initial investment costs to develop and implement both types of training; operating and support costs to support both types of training once it is in place; and these two cost categories combined. The smaller the ratio then, the better the news for the technology-based training. The results are as shown: 0.43 for initial investment; 0.16 for operating and support; and 0.35 overall.



Are We Getting Anywhere? Time to Learn.

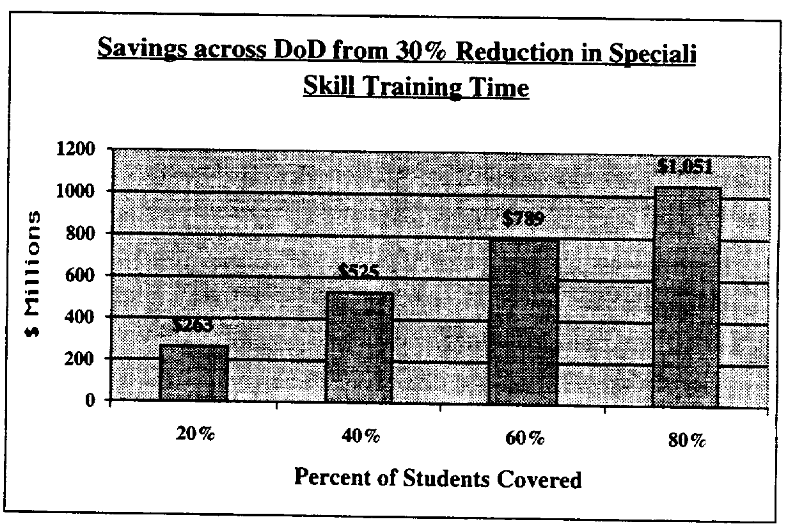
Technology-based instruction should reduce time to reach instructional objectives. By avoiding material the student already knows, concentrating on material yet to be learned, and otherwise keying on each student's pace and style, technology should allow learning to occur more quickly. The reviews of evaluation studies shown in this slide (where the "Ns" refer to numbers of studies reviewed) suggest that these time savings can be obtained and that they average about 30 percent. This is a robust finding in that it appears almost inevitably across many independently performed reviews.



Some Cost Implications of Savings in Time to Learn

To take an example, what would be the cost benefit of reducing time to perform specialized skill training for all DoD students? Specialized skill training is a Congressionally defined category that provides officer and enlisted personnel with skills and knowledge needed to perform specific jobs. It is performed in formally convened military schools. It provides initial training and certification for military occupational specialties such as vehicle mechanics, electronics repair and radar operation. It does not include recruit training, flight training, training performed by operational units or field exercises. In FY 1999 the DoD will provide specialized skill training for about 900,000 students at a cost of about $4.4 billion.

Suppose we were to save 30 percent of the time to provide this training to about 20 percent of the specialized skill students. How much would the DoD save? There is a modestly-careful cost analysis behind this finding, but the end result suggests, as the chart shows, that the DoD would save about $263 million each year. Suppose we were to save 30 percent of the time to provide this training to about 60 percent of the specialized skill students. How much would the DoD save? As the chart shows, the savings would be about $789 million each year. A time savings of 30 percent is not at all unreasonable, as the reviews of time savings show. When a concentrated effort is made to reduce time to train, as for instance, Noja (1987) has done, it is not unreasonable to expect reliable time savings of 50 percent.



The "Thirds"

In summary, all of the assessments of technology-based instruction leave us with "the thirds." Use of technology reduces the cost of instruction by about one-third and, additionally, it either reduces time to reach given instructional objectives by one-third or it increases the achievement of its students by about one-third. The primary payoff for the DoD is, of course, the more rapid preparation of military personnel for operational duties and the increased accessibility of training for sustaining and increasing personal competencies within operational units.

The "Thirds"

Use of technology-based instruction reduces cost of instruction by about 1/3

And

Either ...
- **Reduces time of instruction by about 1/3**

Or ...
- **Increases effectiveness of instruction by about 1/3**

But the REAL Payoff is in Improved Readiness

Some Conclusions

More extensive tailoring of instruction to the needs of individual students obtained through the use of ITSs can only be expected to increase. ITSs may raise the bar for the ultimate effectiveness of technology-based instruction. They may make available far greater efficiencies than we can now obtain from other approaches using technology in instruction. More concentrated attention to the systematic and empirical assessment of ITSs should improve their capabilities and promise. In a formative sense these assessments should improve the design of environments in which individuals learn effectively and efficiently. In a summative sense, they may demonstrate to all stakeholders that the results are worth the effort.

In Conclusion ...

- **ITSs must assert their unique capabilities to raise the ceiling on training effectiveness and efficiency.**
- **Individualization and generative capacity may be key**
  - **We could use more data:**
    - **Formative**
    - **Summative (costs)**

References

1. Atkinson, R. C. and Fletcher, J. D., "Teaching Children to Read with a Computer," *The Reading Teacher*, Vol. 25, 1972, pp. 319-327.
2. Bloom, B. S., "The 2-Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring" *Educational Researcher*, Vol. 13, No. 6, 1984, pp. 4-16.
3. Brown, J. S., Burton, R. R. and DeKleer, J., "Pedagogical, Natural Language and Knowledge Engineering in SOPHIE I, II and III," in *Intelligent Tutoring Systems*, D. Sleeman and J. S. Brown (eds.), Academic Press, New York, NY, 1982.
4. Carbonell, J. R., "AI in CAI: An Artificial Intelligence Approach to Computer-Assisted Instruction," *IEEE Transactions on Man-Machine Systems*, Vol. 11, 1970, pp. 190-202.
5. Carroll, J. B., "Problems of Measurement Related to the Concept of Learning for Mastery," *Educational Horizons*, Vol. 48, 1970, pp. 71-80.
6. Crowder, N. W., "Automatic Tutoring by Means of Intrinsic Programming," in *Automatic Teaching: The State of the Art*, E. H. Galanter (ed.), John Wiley, New York, NY, 1959.
7. Feurzeig, W., Munter, P., Swets, J. and Breen, M., "Computer-Aided Teaching in Medical Diagnosis," *The Journal of Medical Education*, Vol. 39, 1964, pp. 746-754.
8. Fletcher, J. D., "Artificial Intelligence Applications in Computer-Based Instruction," in *Proc. of ADCIS Summer Meeting*, A. G. Smith (ed.), University of Maine, Portland, ME, 1975.
9. Fletcher, J. D., "Intelligent Training Systems in the Military," in *Defense Applications of Artificial Intelligence: Progress and Prospects*, S. J. Andriole and G. W. Hopple (eds.), Lexington Books, Lexington, MA, 1988.
10. Fletcher, J. D., *The Effectiveness of Interactive Videodisc Instruction in Defense Training and Education,* Institute for Defense Analyses, Alexandria, VA, IDA Paper P-2372 (DTIC No. ADA 228 387), 1990.
11. Fletcher, J. D. and Rockway, M. R., "Computer-Based Training in the Military," in *Military Contributions to Instructional Technology*, J. A. Ellis (ed.) Praeger Publishers, New York, NY, 1986.
12. Gettinger, M., and White, M. A., "Evaluating Curriculum Fit with Class Ability," *Journal of Educational Psychology*, Vol. 72, 1980, pp. 338-344.
13. Gott, S. P., Kane, R. S. and Lesgold, A., *Tutoring for Transfer of Technical Competence,* Armstrong Laboratory, Human Resources Directorate, Brooks AFB, TX (AL/HR-TP-1995-0002), 1995.
14. Kulik, C-L. C., Kulik, J. A. and Shwalb, B., "Effectiveness of Computer-Based Instruction in Adult Education," *Journal of Educational Computing Research*, Vol. 2, 1986, pp. 235-252.
15. Kulik, J. A., "Meta-Analytic Studies of Findings on Computer-Based Instruction," in *Technology Assessment in Education and Training*, E. L. Baker and H. F. O'Neil, Jr. (eds.), Lawrence Erlbaum Associates, Hillsdale, NJ, 1994.
16. Kulik, J .A. and Kulik, C-L. C., "Effectiveness of Computer-Based Education in Colleges," *AEDS Journal,* Vol. 19, 1986, 81-108.
17. Noja, G. P., "DVI and System Integration: A Further Step in ICAI/IMS Technology," in *Advanced Technologies Applied to Training Design*, R. J. Seidel and P.R. Chatelier (eds.), Plenum Press, New York, 1991.

References (Cont'd.)

18. Orlansky, J. and String, J., *Cost-Effectiveness of Computer Based Instruction in Military Training,* Institute for Defense Analyses, Alexandria, Virginia, IDA Paper P-1375, 1979.
19. Regian, W., Seidel, R., Schuler, J. and Radke, P., *Functional Area Analysis of Intelligent Computer-Assisted Instruction*, Report to the Director of Defense Research and Engineering, Training and Personnel Systems Science and Technology Evaluation and Management Committee, Washington, D.C., 1996.
20. Suppes, P., "Modern Learning Theory and the Elementary-School Curriculum," *American Educational Research Journal*, Vol. 1, 1964, pp. 79-93.
21. Suppes, P., Fletcher, J. D., and Zanotti, M., "Performance Models of American Indian Students on Computer-Assisted Instruction in Elementary Mathematics," *Instructional Science*, Vol. 4, 1975, pp. 303-313.
22. Suppes, P., Fletcher, J. D. and Zanotti, M., "Models of Individual Trajectories in Computer-Assisted Instruction for Deaf Students," *Journal of Educational Psychology,* Vol. 68, 1976, pp. 117-127.
23. Suppes, P. and Morningstar, M., *Computer-Assisted Instruction at Stanford 1966-68: Data, Models, and Evaluation of the Arithmetic Programs,* Academic Press, New York, NY, 1972.

Intelligent Learning Environments for Technical Training: Lessons Learned

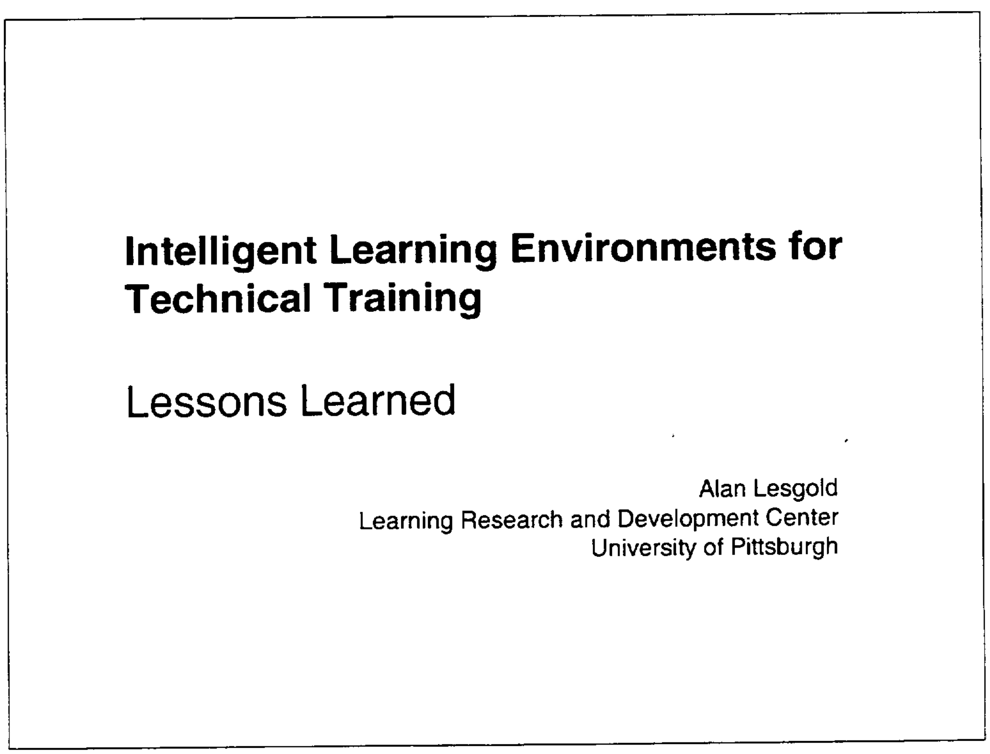
Alan M. Lesgold
Executive Associate Director
Learning Research and Development Center
University of Pittsburgh
Pittsburgh, PA

Intelligent Learning Environments for Technical Training
Lessons Learned

Alan M. Lesgold
Learning Research and Development Center
University of Pittsburgh
Pittsburgh, PA 15260

The Intelligent Coached Apprenticeship Approach

The approach that my colleagues and I have developed over the past dozen years [1,2] derives from several basic facts about learning. We see these facts in everyday life, though they are also confirmed in the laboratory. Real high-stakes learning involves doing. For example, when a person learns to play football, this is done not by attending a sequence of classes on each aspect of football, but rather by playing, with the help of a coach, peer critique, and reflection (thinking afterwards about how a particular situation could have been handled better). Traditionally, one learned a trade as an apprentice, actually doing progressively more complex work with coaching from a master. Today, as jobs change so quickly, one apprenticeship is not enough preparation for a lifetime of productive labor, and for some new jobs, there really are few if any masters available yet. It seems appropriate, therefore, to design computer systems that allow apprenticeship to happen in a more virtual environment.



Overview

At the Learning Research and Development Center, we have been at work simultaneously on both high-end intelligent training systems and on non-technological approaches to training that depend on substantial instructor time but with only a blackboard. We have begun now to bridge the gap between these technology extremes in our efforts to improve the professional skills of teachers. In addition, in work we performed a few years ago for the World Bank, we began development of "just-in-time" training and knowledge enhancement opportunities that can be triggered by clicking on icons embedded in the web-based forms that are becoming the basis for office activity in the information age. A worker using a form such as a budget sheet or a project plan template can click on the training icon to gain access to either focused coaching or an on-line library of useful training documents and records of "lessons learned"by other workers. Below, I describe how simple and relatively low cost training technology can be informed by the more expensive earlier efforts on intelligent systems for training that have been made in the U.S. and Western Europe. I also discuss the specific skills needed by software developers to produce useful training tools and the ways in which tools can be developed incrementally, permitting government and industry sponsors to test the value of this kind of approach.

Overview

- An example of Learning by Doing
- Problems of Diagnosis
- Task Analysis to Support Intelligent Learning Environments
- Supporting "Just-in-Time" Learning Opportunities

Learning by Doing with Reflection

Our systems provide the opportunity to learn while doing hard work. To make this possible, we need to be able to simulate the work environment; to provide a collection of difficult tasks with supporting simulation of the task situations; to provide coaching on demand so that a trainee is never completely at an impasse in doing this simulated work; and to support post-problem reflection. This is similar to the learning paradigm followed in training football players, for example, except that no simulation is needed in that case. A coach provides advice, usually pretty terse, to players during games. After completing a game, players often review game films and attempt to refine principles to guide future play.

The Domain. We recently built a training system jointly with Intel Corporation. The system taught technicians how to determine what was failing in a complex semiconductor manufacturing device called an ion deposition system. This device is used to put layers onto silicon wafers as part of the process of making chips. The cost of downtime for such machines is very high. To address this training problem, we worked with Intel to build an intelligent, coached apprenticeship system.

Learning by Doing with Reflection

❶ Realistic simulated work environment in which hard problems (at the upper edge for the course, or beyond) can be confronted

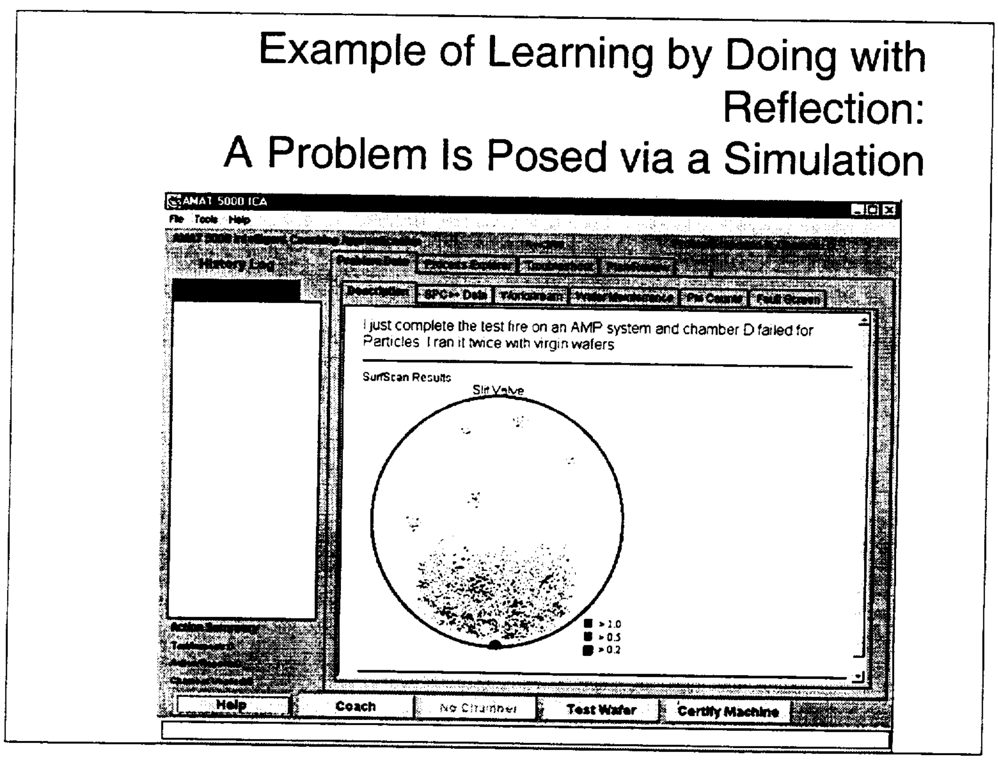
❷ Intelligent coaching to assure that with effort every student can do every problem

❸ Tools to support reflection after solving a problem

An Overview of the Training System

The system poses problems to technicians the same way they appear in a plant. A technician in a plant would be called to a machine that is malfunctioning and would have access to a brief description of the problem by the operator, recent statistical process control (SPC) data, the logbook of recent activity with the machine (called Workstream at Intel), and various console displays showing the age of various parts and the expected number of cycles before they need to be replaced for preventive maintenance.

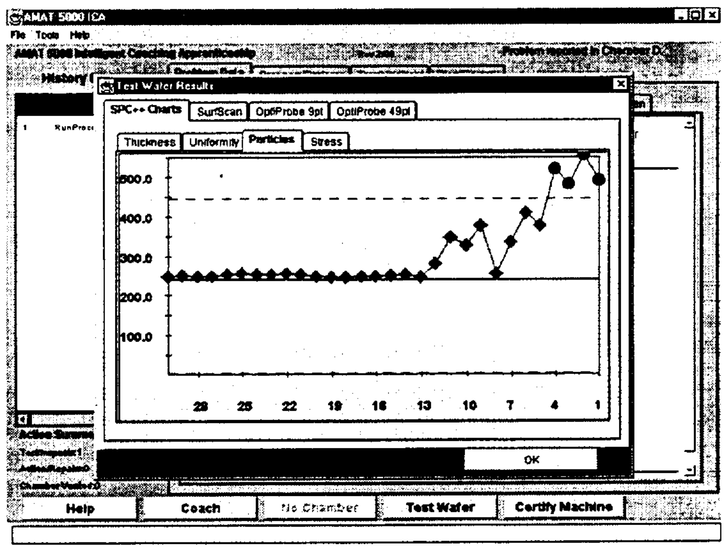
The first figure shows the computer screen for the beginning of one of the training problems in our system. The symptom being displayed is that wafers are coming out of the system with grit on them, primarily around one point on the wafer's edge. The technician would be expected to click on tabs (the lower row of tabs at the top of the screen) to access SPC, Workstream and other data. This provides an initial grounding for the problem.



Problem Staging

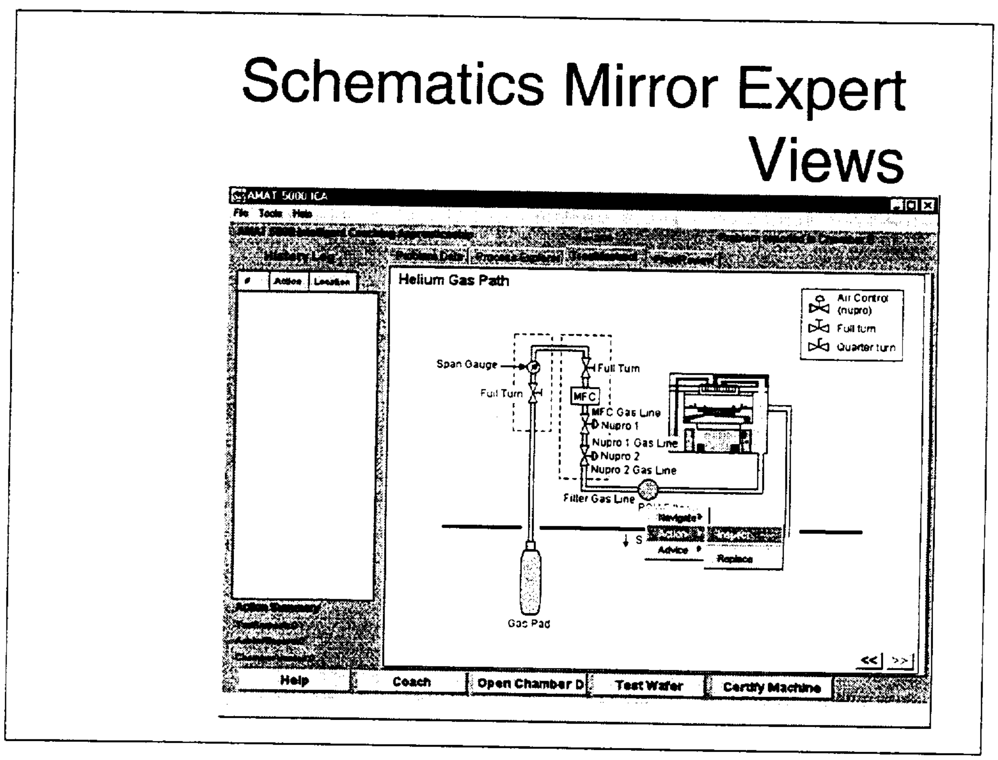
For example, the figure below shows a statistical process control chart that signals the problem of particles appearing on the wafer in greater than acceptable numbers. These charts are dynamic so if the technician continues to operate the machine after making adjustments, he can quickly see if the adjustments did any good.

Example of Problem-Staging Data From the Simulation



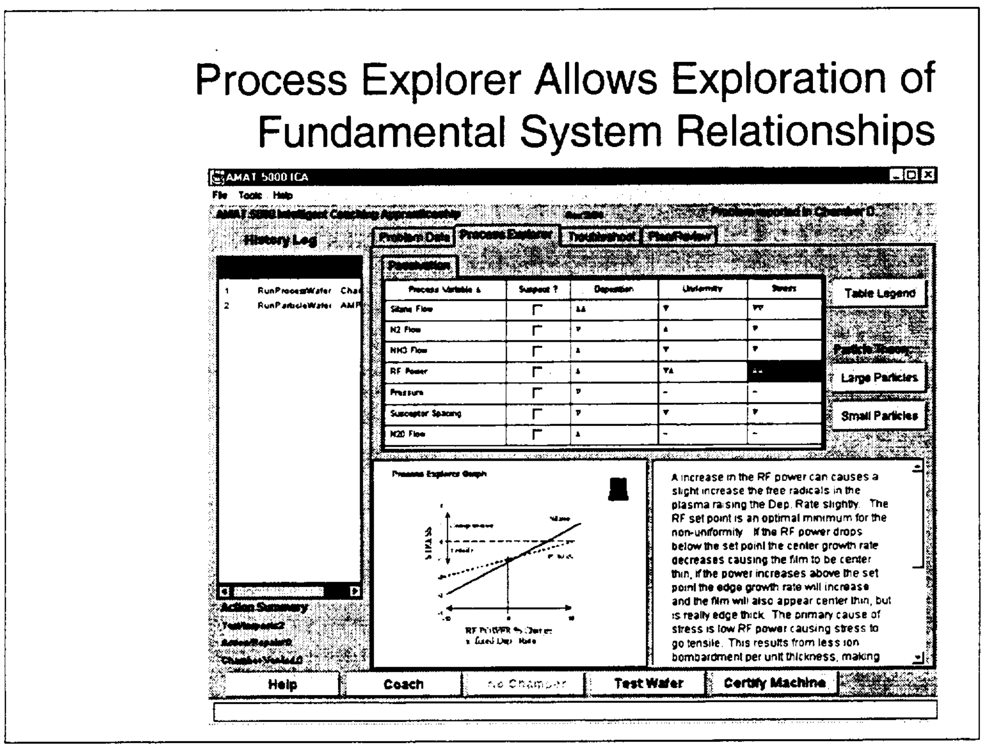
Schematics Provide Access to the Simulation and Emulate Expert Thinking

Once the technician is ready to begin troubleshooting activity, the interface permits him to measure, adjust, or replace any part that a real technician could access. This is done in two ways. First, the technician can access system components via schematic diagrams of the system. For example, in the figure on the left, a schematic diagram is presented of the path from helium tanks into the chamber in which the processing of the wafer occurs. By clicking on any component, the technician can open it (if it is decomposable), inspect it, make measurements on it, adjust it, or replace it. An important detail is that our schematic diagrams are designed to reflect how expert technicians think about the system. The grouping of components in the diagrams matches expert groupings. Hence, the interface itself teaches about troubleshooting.



The Process Explorer

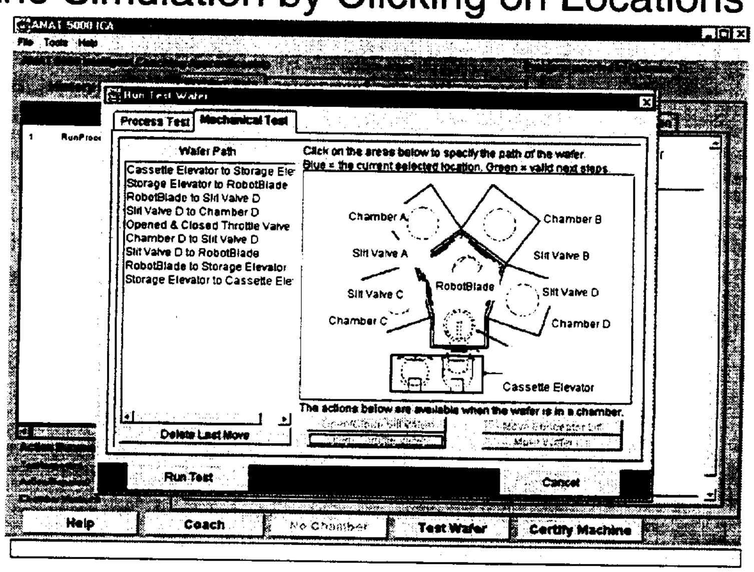
The next step is for the technician to develop a preliminary plan for troubleshooting. This planning process is an aspect of technical expertise that we want to encourage; that is, we want technicians to consider what information is available, try to match this to their knowledge of the relationships among variables in the machine, and evolve a plan for narrowing the range of possible sources of the machine's failure. To help technicians develop this capability, we provide a tool called the *process explorer* (illustrated below). This tool constructs, for a given problem, a matrix showing the relationships between classes of failures for the particular machine process that failed and potential failure sources, such as deviations in electrical power or in one of the gases that must be delivered to the chamber for the layer to be deposited. Up and down arrows in the cells of the matrix capture the kind of relationship that exists. Clicking on a cell produces a graph of the relationship caricatured by that cell along with an explanation of the relationship – why it happens.



Simulating Robotics

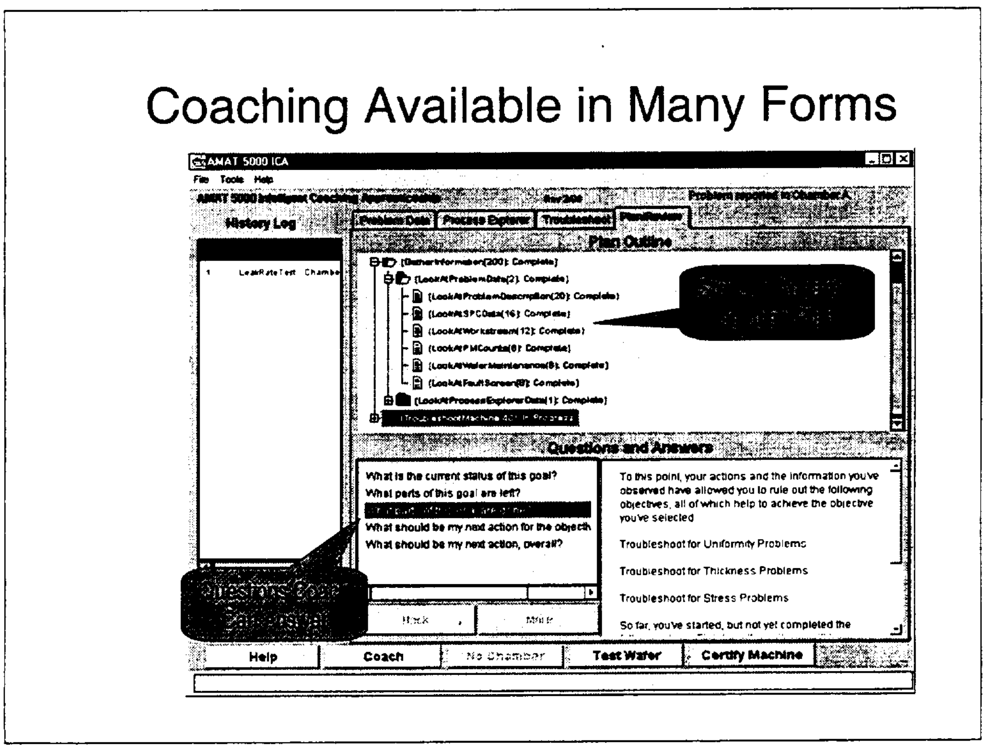
Since robotics are involved in this machine (a robot takes each wafer from a cassette, through an airlock, and then into each of several chambers in turn for various layering processes), we needed a way for technicians to explore the robotic activity to see if it was causing the problem (for example, sometimes a robotic failure in one chamber will cause the wafer to be grabbed incorrectly and then placed incorrectly in the next chamber). To deal with this situation, we created a simple programming capability. The technician can operate the robot simply by clicking on the consecutive destinations to which the wafer should be transported; these are then listed and become a program for the simulation (see middle figure). By doing inspections before and after these transport steps, the technician can do troubleshooting of the robotic process. Note that none of this requires any animation, video, or virtual reality – but it works quite adequately.

Programming a Sequence of Robot Actions for the Simulation by Clicking on Locations



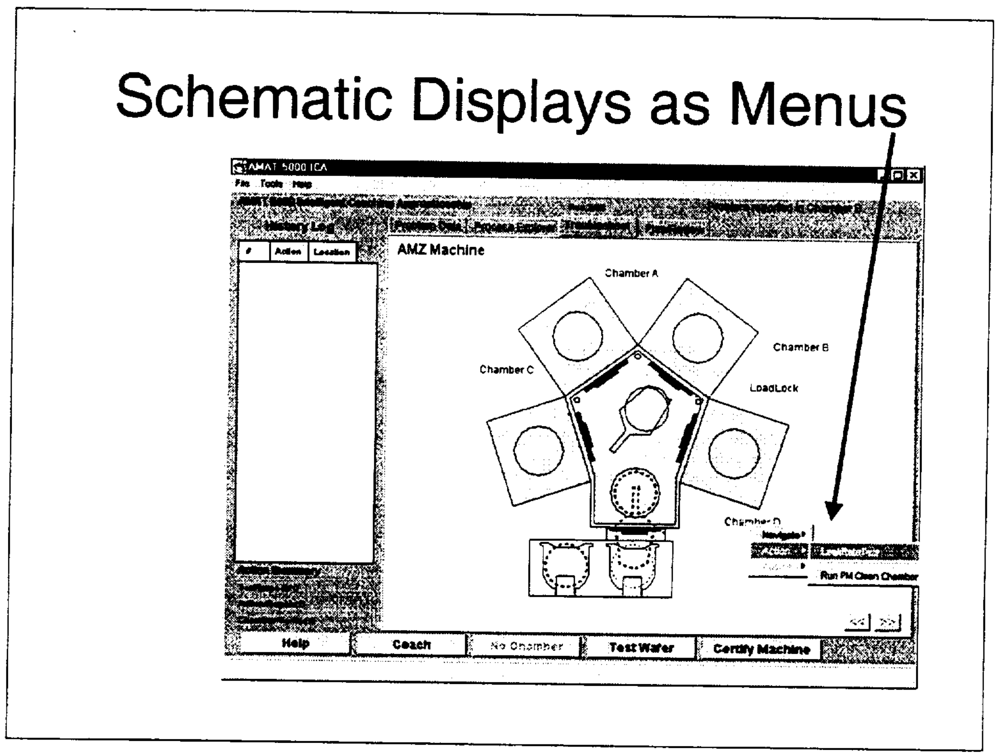
Coaching

We provide coaching support for technicians as they work on the problems our system presents (see figure on previous page). Just as with the action interface, the coaching interface is meant to be a learning tool. We ask technicians to look through a rough plan for solving the problem at hand. This plan is based upon the actions they have taken so far. Once the technician points to a particular element in the plan, a menu of questions appears. Clicking on a question produces the answer to that question. At least one of the questions asks what to do next, so the technician can never be completely stuck – the system will always be able to tell him what to try next. The interface allows technicians to test their solutions either by running "test wafers" (wafers inserted to test a particular process) or by asking the system to "certify" the machine (basically, to have the system state whether the machine now works or not).



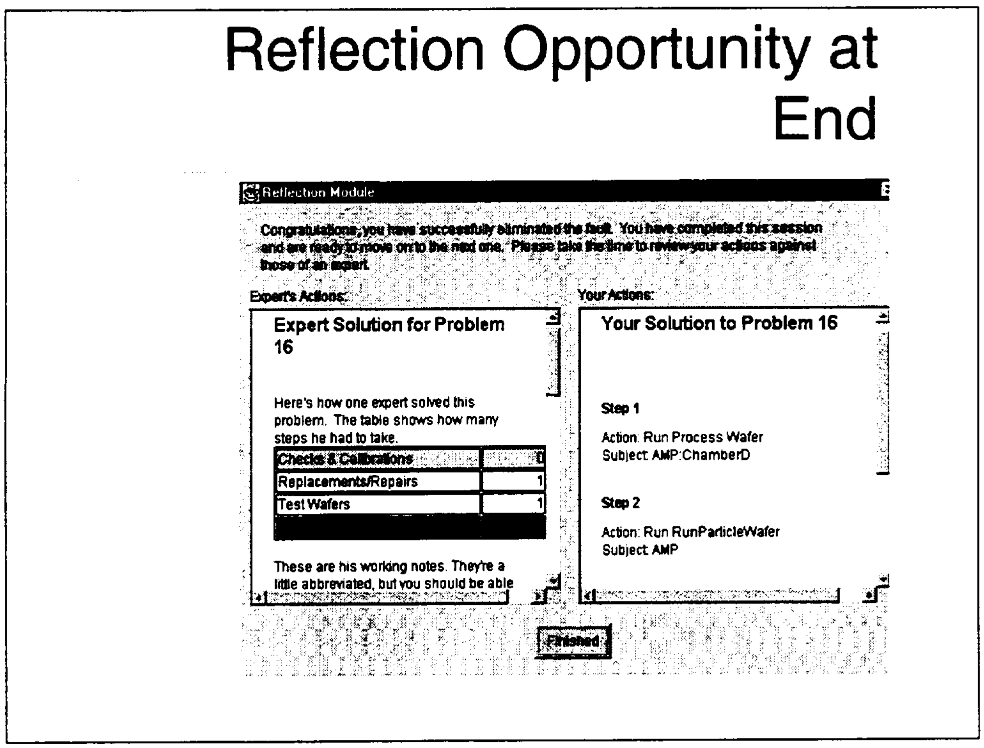
Menus

In order to provide the remaining capabilities needed in the interface, the system puts a label on each component of each schematic. Clicking on this label produces a hierarchical menu of options for inspecting, altering and replacing the labeled component and also of opening it, if the simulation contains detail on its internal construction.



Reflection

Once that certification is completed successfully, the program goes into a final reflection mode (shown in figure on the right). Here, the technician can see the trace of activities that he engaged in while trying to solve the problem. This trace includes a discussion of the goals that drove the expert solution and a comparison of the costs of the two solutions (for each part that can be manipulated, we know the time it takes to deal with it and the cost of a replacement). This cost information allows the technician to focus attention on expert actions that get things done in less time, which is where we want him to be focused.



Diagnosis of Student/Trainee Needs

One important characteristic of our approach is that it puts minimal software effort into student modeling (diagnosing what the student does or does not know). Because of the realistic nature of our tasks, students generally know when they have knowledge gaps and can thus decide for themselves what advice they need. Further, there are several problems that make precise diagnosis very difficult in cases of technical skills, the skills of real work.

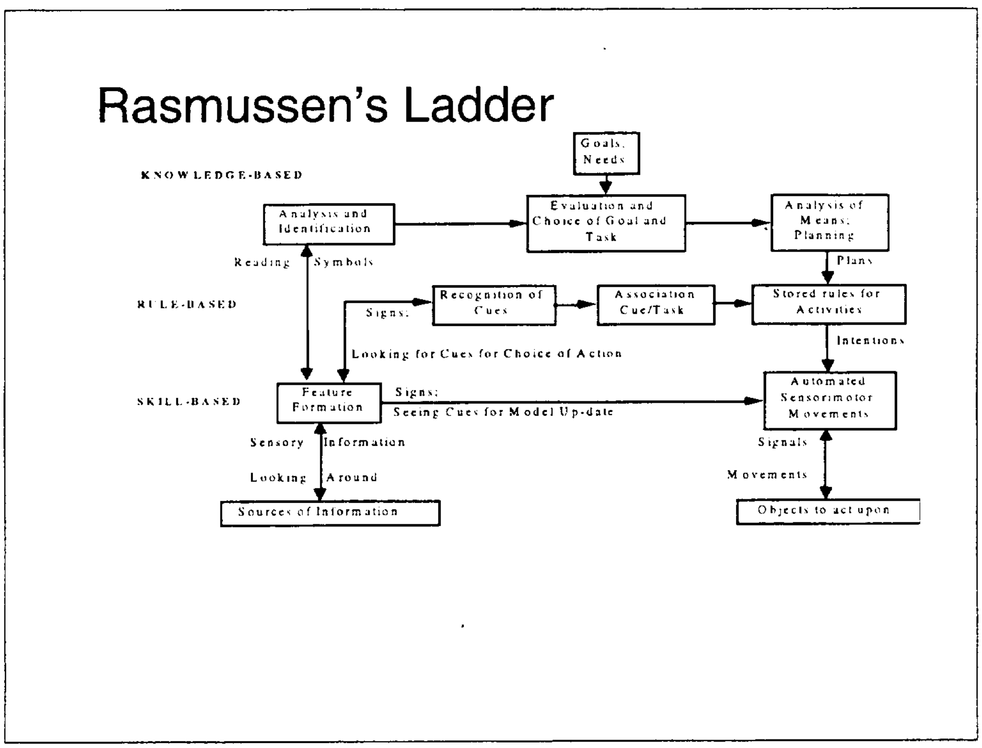
Problems of Diagnosis

- Multiple Levels of Processing: Skills, Rules, Knowledge (Rasmussen)

Rasmussen's Three Levels of Competence

We must bear in mind that a correct action by a technician can come about in multiple ways. The technician might know exactly what to do from extensive practice, or he might possess rule-based knowledge in strong form that is sufficient to decide what to do (near transfer). In the most difficult kinds of situations, a technician may possess conceptual understanding of the domain sufficient to drive weak problem solving methods that will leverage this conceptual knowledge to reach a problem solution (far transfer). Rasmussen made these distinctions clear.

Because of the multiple sources of competence, it is dangerous to read too much into a successful performance of a piece of work by a technician. The ultimate determiner of technical competence is the ability to function effectively in a mix of situations with varying degrees of remoteness to the situations already practiced.



Contextual Factors

There are value issues that arise in tutoring. For example, the tradeoffs between speed of solution and cost of solution differ from one work site to another. These economic issues should stand outside of the basic knowledge in the tutor; that is, a problem solution that optimizes time instead of money is less acceptable in a low budget environment, but it should not be flagged as wrong, in the sense of indicating lack of knowledge in the trainee. Rather, the available coaching should respond to expected value differences and comment on them, e.g., "here are two solutions, one better suited to low-budget environments and the other to time-sensitive mission-critical environments." Fortunately, estimating the value of a performance need not require extensive expert knowledge – just a credible pricing scheme.

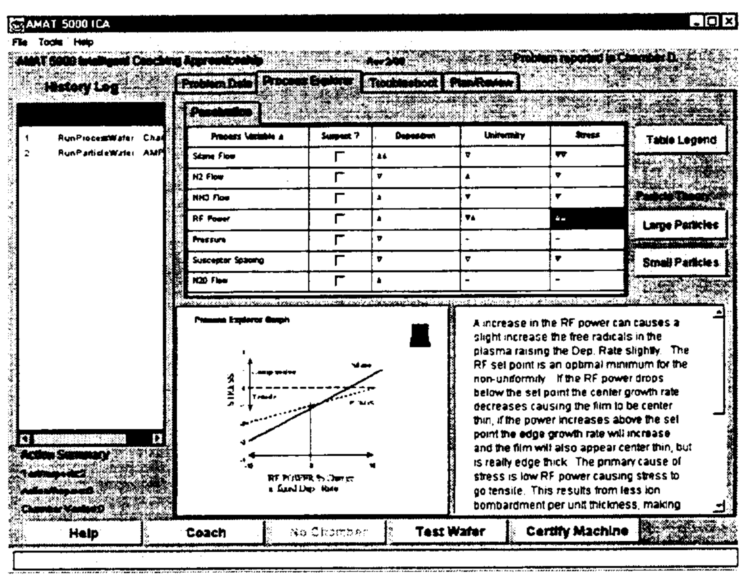
Contextual Factors

- Different local economics and different local equipment and work practice
  - Spare parts availability
  - What is included in replacing a component or calibrating a component
  - Field engineering changes aren't made all at once

An Expert Intermediate Representation of the Domain

The expert's intermediate representation of the domain is the most difficult part of building an ICA system. In our own work, we needed to enlist the aid of Martin Nahemow, a plasma physicist with a cognitive science background who held patents on related ion deposition processes. Nahemow was able to distill the needed background knowledge into the process explorer scheme discussed above. That scheme captures important relationships and presents them in relatively simple form, even though the ultimate underlying physical chemistry is extremely complex. For example, a change in the frequency of the RF power supplied to the chamber can change the resulting chemical reactions and result in a different material being deposited on the wafer than was planned. Few technicians understand chemistry well enough to understand the details of why this is so, but with the intermediate, process-explorer level of representation, they do not have to. The intermediate level knowledge acquisition task is perhaps the one tough task that cannot be skipped or minimized to cut costs.

How Does One Decide that This Representation is the Right Kind? Where Is the Data to Support Building It?



Weak Methods for Generating Expert Solutions From the Simulation

Once the relevant parts of the system are simulated, it is possible to generate expert solution rules more easily. The key to doing this is to represent the hierarchical decomposition of the system in the way experts do. Once the object structure of the simulation is defined at multiple levels of hierarchy (actually, a lattice), that match expert thinking, then one can use weak solution rules to generate an expert solution, such as space splitting strategies and the like. A small amount of specialization is required, but not a lot.

On the other hand, it is possible even more quickly to build specific rule sets for the individual problems; that is, by interviewing experts, one can produce a rule set that can solve a problem. By continuing this process with each subsequent problem and elaborating rule conditions when a rule overgeneralizes, it is possible to produce a rule set that handles all the problems. This is not as extensible to new domains, but it can be done and sometimes is helpful in building organizational confidence that the ICA approach is worthwhile.

The same knowledge gathered to support the expert hierarchical structure for the simulation and the process explorer can be used to develop generic goal structures for problem solutions. These goal structures then become a scaffold within which the specific actions of the user can be stored. Then, when the user asks for advice, the system will know which subgoals of these generic goal structures have been achieved and can advise on what else needs to be considered.

Generating Intermediate Representations

1. Intelligent analyses of formal representations of knowledge
2. Task analysis steps focused on intermediate representation
3. Diagnostic strategies designed into artifacts and protocols

Coaching Expertise

It is possible to develop intelligent schemes for deciding exactly what coaching to provide. Indeed, in the earlier versions of Sherlock, we did this. We used a student modeling scheme to decide how specific the advice should be, hoping to stimulate technicians to push their thinking and not ask for help prematurely. There is no evidence that this was particularly beneficial, so we no longer invest time and effort in making the coaching intelligent with respect to the user's current knowledge. The resources just described are sufficient to make the coaching intelligent with respect to the specific problem-solving context in which the request for advice arises.

No Matter How Hard We Try, We Can't Be Completely Ready to Provide Perfect Coaching

1. But, we can be ready to provide a range of potentially useful information: explanations of function, alternate possible next steps, etc.
2. And, there's another intelligence in the room: the student!

References

1. Gott, S. P., Lesgold, A. M. and Kane, R. S., "Promoting the Transfer of Technical Competency," in *Instructional Design: International Perspectives*, S. Dijkstra, et al. (eds.), Vol. 2, Lawrence Erlbaum Associates, Hillsdale, NJ, 1997.

2. Lesgold, A. M., Lajoie, S. P., Bunzo, M. and Eggan, G., "SHERLOCK: A Coached Practice Environment for an Electronics Troubleshooting Job," in *Computer Assisted Instruction and Intelligent Tutoring Systems: Shared Issues and Complementary Approaches*, J. Larkin and R. Chabay (eds.), Lawrence Erlbaum Associates, Hillsdale, NJ, 1992, pp. 201-238.

DATAGUIDES (Deployable Agents for Training, Aiding and Guidance)

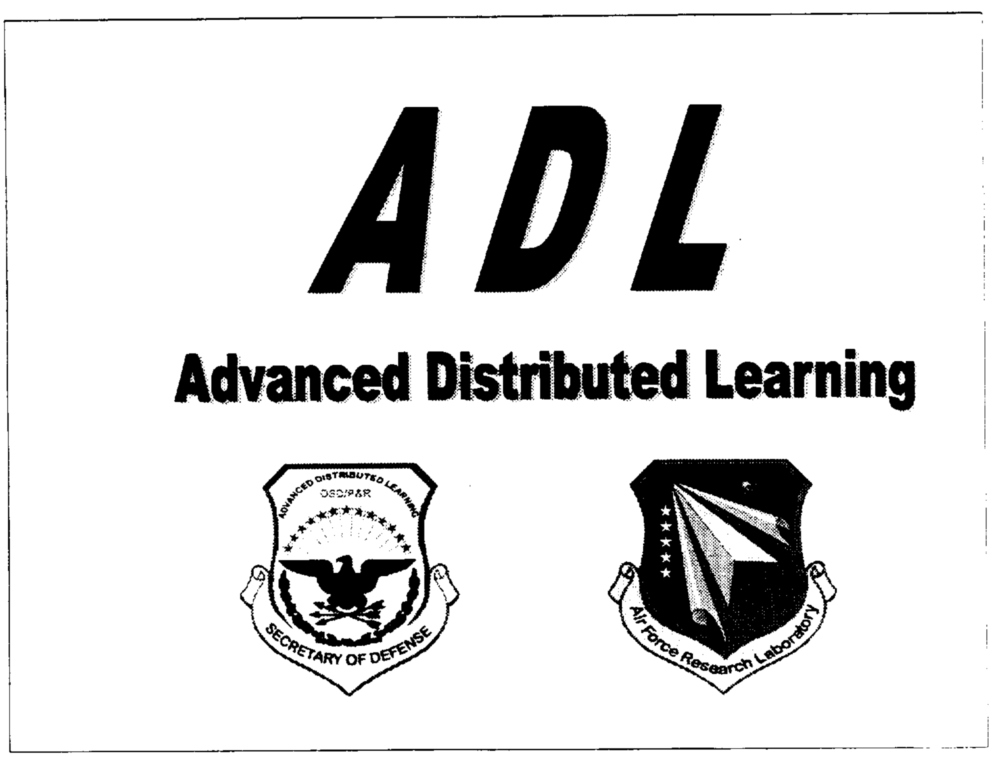
J. Wesley Regian
and
Jim Fleming

Air Force Research Laboratory
Brooks Air Force Base, TX

DATAGUIDES (Deployable Agents for Training, Aiding and Guidance)

J. Wesley Regian and Jim Fleming
Air Force Research Laboratory
Brooks Air Force Base, Texas

This presentation utilizes Microsoft ActiveX technologies to demonstrate a new class of internet-deployable intelligent agents - called Dataguides. Dataguides are knowledge-based interactive personalities that can support users in both training and operational environments. In operational settings, Dataguides will serve as intelligent assistants that continuously learn about their performance domains as well as their primary user. This evolving user model will also serve to support refresher training and subsequent new training for users. Dataguides will support skill maintenance by alerting the users, or supervisors, when refresher training or update training is required, and then provide tailored training scenarios. This as-needed training capability will be especially useful in maintaining readiness for critical but rare situations. The combined contents of many student models within a career field will serve to drive new training and support enhancements for that career field. When plugged into an embedded training environment, Dataguides will provide the student's history to the embedded training agent, and provide training, job aiding, tailored display, and decision support for the student.



Intelligent Computer Assisted Instruction

A substantial empirical base supports the claim that we are in possession of a workable engineering discipline for the principled and reliable implementation of consistently effective courseware. In fact, we are well into a second generation courseware engineering technology. That is, a documented approach exists for engineering courseware that roughly doubles the already impressive effectiveness of ISD (Instructional Systems Design) based courseware. Courseware systems that apply this advanced approach are variously referred to as ICAI, Intelligent Tutoring Systems, or Cognitive Tutoring Systems. The primary common feature of these advanced courseware systems is that they are implemented around knowledge models of human learning and performance. The pedagogy that they apply, the principles around which instructional interactions revolve, are derived very directly from contemporary cognitive science. If ICAI can be routinely engineered, why is it so rare? Why are our schools increasingly festooned with cutting-edge computers that are loaded with outdated and pedagogically unprincipled courseware?

Intelligent Computer Assisted Instruction

Ready for prime time, yet rarely applied.

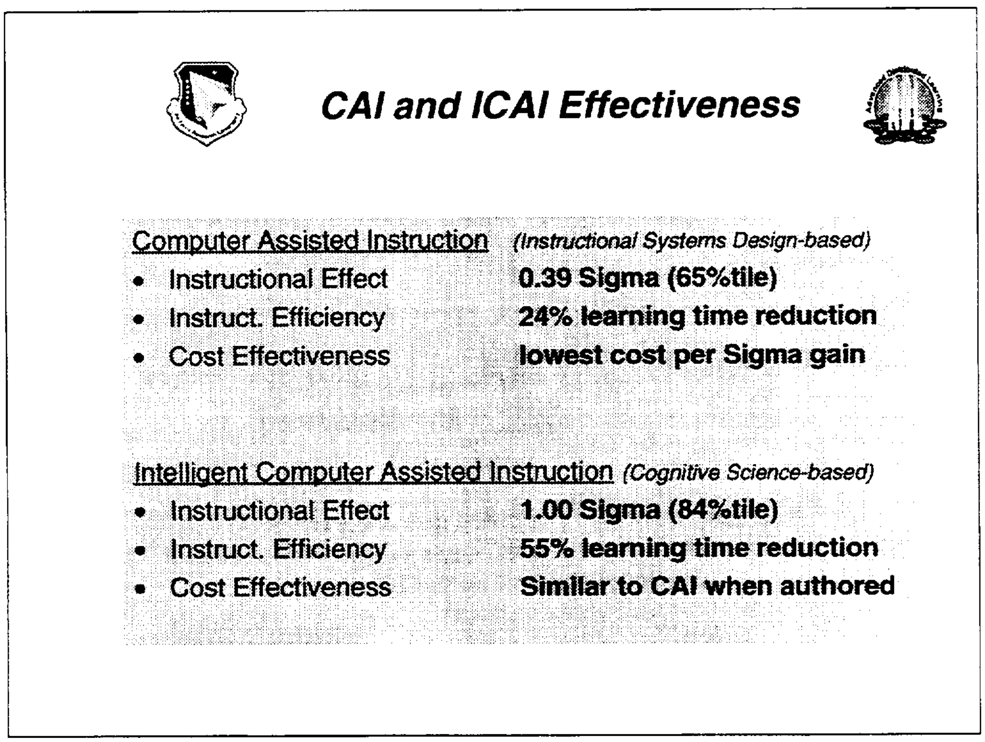
Why so rare?

CAI and ICAI Effectiveness

We have the capability to consistently produce effective courseware.

Using established methods, CAI routinely enhances learning by 0.3 to 0.4 Sigma. This translates to an achievable 15% increase in learning over traditional methods. Alternatively, CAI can produce student performance comparable to traditional instruction, but in 24% less time. There are a variety of ways to enhance learning or reduce learning time, but CAI is the cheapest.

Studies of intelligent CAI are even more promising, with an average effect of 1.00 Sigma, reflecting a 34% increase in learning; or a 55% reduction in learning time. Efficiency studies of ICAI authoring tools suggest that cost will be similar to CAI, but this has only been demonstrated for simulation-based ICAI.



Why is Automated Instruction Often Ineffective?

Why is fielded courseware sometimes ineffective?

Good pedagogically-sound courseware can fail in deployment because of technology integration pitfalls. These pitfalls are avoidable, but useful guidelines have not been published (to my knowledge). We find that we can provide guidelines and training to our field implementation personnel, and that we are able to consistently achieve successful integration. Technology integration is not difficult but should not be taken for granted. Without access to appropriately experienced personnel, those who attempt to deploy ICAI into new settings are likely to experience initial technology integration setbacks.

Another reason fielded courseware is sometimes ineffective is that it is not good instruction. Pedagogically unsound courseware is often developed by groups who simply do not have the skills (are not qualified) to build current-generation courseware. Much of what we know about pedagogy and effective courseware implementation was developed in the last twenty years. There are no simple recipes here, and the relevant literature is huge, but broad knowledge of appropriate literature yields many useful guidelines.

Bad courseware can also result from uninformed procurement specifications. Even qualified groups can build ineffective instruction when they are directed to do so, when sufficient funds are not provided, or when critical formative evaluations are disallowed.

Why is Automated Instruction Often Ineffective?

Failed deployment of automated instruction can result from ...
intrinsic failures of courseware implementation, or
extrinsic failures of technology integration.

Extrinsic Failures
Hardware inadequacies or failures
Software aging or incompatibility
Poor integration with curricula
Poor integration with setting
Lack of support or expertise
Negative bias in key personnel
Avoidable
Careful planning
Integration guidelines

Intrinsic Failures
Over-use of page-turners
Simulation without guidance
Ad-hoc selection of features
Ad-hoc instructional design
No formative evaluation
Poor formative evaluation
Avoidable
Systematic development
Instructional guidelines

Ineffective Automated Instruction

Three explanations account for the majority of failed CAI deployments due to intrinsic shortcomings. First, much of the CAI that is deployed today is of the old frame-based variety (sometimes called page-turners). We now know that frame-based CAI is ineffective for many instructional domains and for many students. Second, while some of the CAI that is deployed today uses advanced features (e.g., simulations for enhanced interactivity), these features are implemented in an ad-hoc fashion rather than in accordance with any principled approach to implementation. Third, most of the CAI that is developed outside of the research community, and unfortunately, some of that which is developed within the research community, is not subjected to the kind of rigorous formative evaluation that is currently required to build good CAI. Given our current state of knowledge with regard to cognition and instruction, even world-class leaders in the field do not develop effective CAI in a single iteration of design and implementation. Rather, effective systems are implemented according to general guidelines derived from a variety of intellectual traditions, and then iteratively improved by a series of empirical evaluations and consequent refinements in the system. These general guidelines are the knowledge-base underlying Instructional Systems Design.

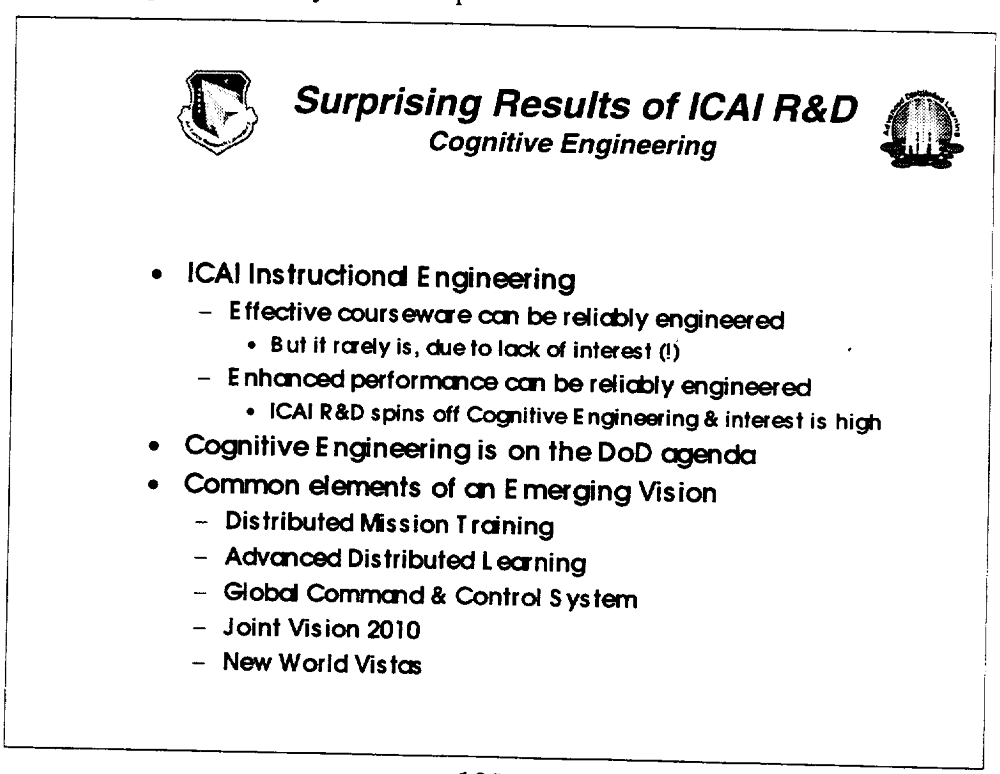
Ineffective Automated Instruction

- **Poor specification of the targeted task.**
- **Ad-hoc instructional manipulations.**
- **Weak (or no!) empirical examination of instructional effectiveness.**

Surprising Results of ICAI R&D
Cognitive Engineering

During the past twenty years, the emergence of Artificial Intelligence and Cognitive Science have driven R&D on Intelligent CAI systems. Like effective CAI systems, ICAI development includes careful task analysis, theory-based instructional manipulations, and empirical system refinements. The ICAI approach, however, uses information processing models of cognition that are more precisely specified than earlier theories of learning. The rigorous empirical attack on cognitive modeling that was required to enable ICAI has also enabled a more general technical discipline, usually referred to as Cognitive Engineering or Cognitive Ergonomics. Cognitive Engineering involves designing systems, procedures, and technologies that capitalize on human information processing strengths and remediate against human IP weaknesses.

Engineering has quickly proven to be remarkably effective at enhancing final performance in virtually any field of complex human endeavor. In retrospect, it is not surprising that effects of CE are so readily obtained. Until quite recently almost no attention was paid to the human user as an information processing system component having inherent performance characteristics and robust interactions with other system components. Whatever the cause of our tardiness, there is an urgency to re-engineer technical and large-scale human enterprises and capitalize on Cognitive Ergonomics. The Department of Defense is no exception. A variety of DoD modernization programs share common elements of an envisioned future that includes global connectivity and information processing humans way in the loop.



2000+ Challenge

The common elements of those various DoD visions that I wish to focus on can be summed up as Global Performance Engineering. I believe that this concept is achievable with current and near term technical capabilities and that the payoff will be huge. The payoff should be measurable in terms of enhanced readiness, increased adaptivity and decisive lethality when required.

My confidence in the achievability of this concept derives from two primary tenets, neither of which are really open to much debate. The first tenet is that, given the right conditions, humans can be selected and trained to attain performance levels that are remarkable in comparison even to the performance of well-trained individuals, and seem absolutely impossible to beginners. A great deal is known about bringing humans to this high level of performance, which we will refer to simply as the "expert" level of performance. The second tenet is that, given the right conditions, human-machine systems can be developed that are remarkable in the sense that the systems are much more efficient, more effective, and less error prone than other systems designed with similar goals.

2000+ Challenge

"Given the right conditions, humans can attain performance levels that are remarkable, even in comparison to well-trained individuals, and seem absolutely impossible to beginners.

Human-machine systems can be developed that are much more efficient, more effective, and less error prone than other systems designed with similar goals.

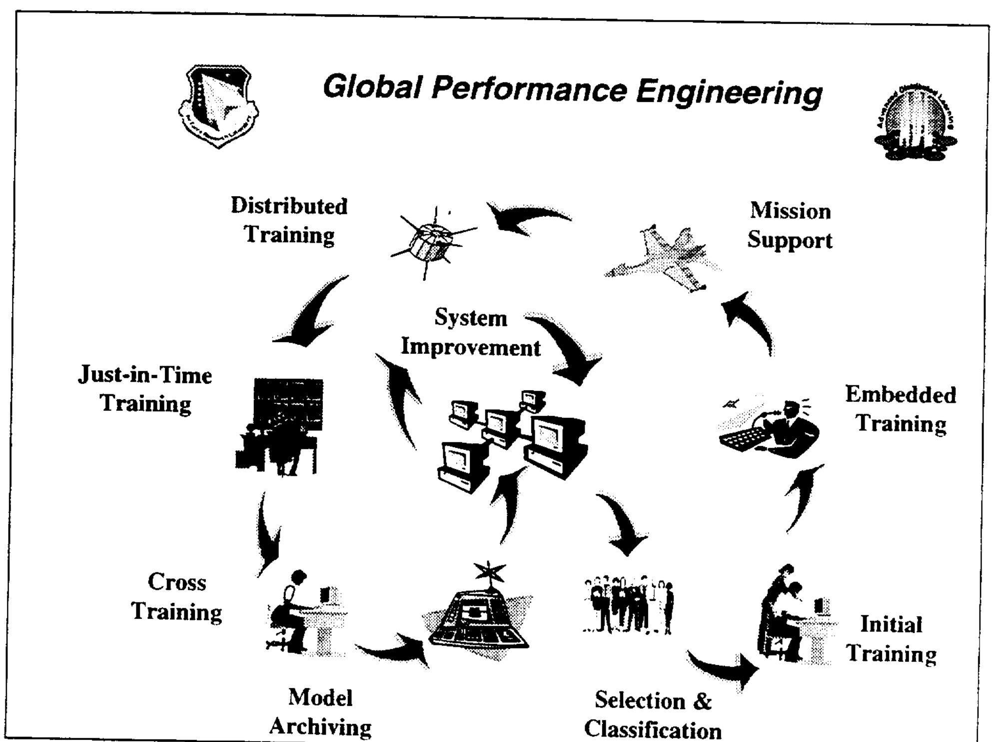
Our current approaches to training and managing personnel, and our approaches to designing complex human-machine systems, are often not optimized for high levels of performance."

- Human Systems Biotechnology Panel Concept Paper
AF Scientific Advisory Board
New World Vistas

Global Performance Engineering

Contemporary cognitive science has developed knowledge representation (KR) technologies that support the acquisition, storage, maintenance, retrieval, and application of digitally-coded human knowledge and skill. These convergent technologies support efficient methods for knowledge engineering (extraction and coding of human expert knowledge) and knowledge re-coding (extraction and re-coding of existing, knowledge-bearing digital data); authoring tools for knowledge-based (individual and team, local and distal) courseware that capitalizes on these KR technologies; tools for planning, deploying, and lifecycle management of courseware over local, wide-area, and global networks; tools for human performance assessment, remediation, and support over local, wide-area, and global networks; tools for rapid deployment of just-in-time and theater-specific training; tools for automated feedback of human performance data to centralized human resource tracking facilities in order to support ongoing empirical improvement of personnel selection, training, job-aiding, and workspace design; and tools for searching, browsing, and otherwise utilizing archived KRs to support all forms of training, education, performance enhancement, and media.

In the long run, policy definition and inter-community coordination activities must help create a commercial and federal environment that encourages the creation of libraries of KRs and compatible commercial authoring, decision support and delivery systems. We are currently working with the Office of the Secretary of Defense, as well as leading companies from the commercial hardware and software industries to accomplish the required coordination.



HS.04.06 Knowledge Representation Technologies for Human Performance Enhancement

A key element of this work will be the scaling up of existing knowledge representation technologies to support very large scale knowledge management over a massively distributed, variable bandwidth global network. In some cases our current generation KR technologies are tedious and labor intensive. We must overcome this hurdle as practical knowledge representation and management technologies are core requirements for global performance engineering. Practical KR technologies are the focus of Defense Technology Objective HS.04.06.

Preliminary results suggest that knowledge re-coding techniques, which enable the efficient recovery of knowledge and skill information from existing sources (e.g., occupational surveys, systems technical databases, digitized courseware, html documents), can reduce the development time for knowledge models by up to 90%.

HS.04.06 Knowledge Representation Technologies for Human Performance Enhancement

OBJECTIVES

- Standardized, general-purpose knowledge representation technology
- Single digital format for acquisition, storage, maintenance, retrieval, and application of knowledge bases
- Authoring tools for knowledge-based (individual and team, local and distal) courseware that capitalizes on these knowledge technologies
- Tools for planning, deploying, and lifecycle management of courseware over networks
- Tools for human performance assessment, remediation, and support over local, wide-area, and global networks

TECHNICAL CHALLENGES

- Efficient methods for extraction and coding of human expert knowledge
- Automated methods for extraction and recoding of existing, knowledge-bearing digital data
- Automated feedback of human performance data to centralized human resource tracking facilities in order to support ongoing empirical improvement of personnel selection, training, job-aiding, and workspace design
- Automated indexing and searching of archived knowledge to support all forms of training/education/performance enhancement

Service/Agency POC	USD(A&T) POC	Customer POCs	
Dr. W. Regian	Dr. A. Johnson-Winegar	Don Johnson (OSD/P&R)	Mariellen Cogan (HQ JSF/PO)
AL/HEJT	Dir. E&LS (DDR&E)	Office of the Sec. Def.	Joint Strike Fighter SPO
DSN 240-2034	DSN 227- 8714	Dennis Richburgh, SES	TC George Selix (AETC)
		Air Intelligence Agency	Air Ed. & Trng. Cmnd.

Distributed Human Performance Engineering
Progress

Our past efforts have focused on concepts, technologies, and methods to support individual human performance engineering. That work has matured sufficiently to support very general application. Building on these capabilities, we are currently scaling up our KR technologies to support massively distributed performance engineering for individuals and teams. These KR technologies are a critical requirement to enable the DoD vector toward Global Performance Engineering.

Distributed Human Performance Engineering

Progress

- **Phase I (Done)**
  - **Instructional Engineering Achieved**
    - **Knowledge Based Training (ICAI) highly effective**
    - **R&D spins off Knowledge Representation Technologies**
    - **KR Technologies enable Cognitive Engineering**
- **Phase II (You are Here)**
  - **Practical Knowledge Representation Technologies**
    - **Standardized & Streamlined Cognitive (KR) Modeling**
    - **Practical knowledge acquisition, maintenance, retrieval, application**
    - **Rapid Development tools for distributed KR-based agents**
- **Phase III (Stay Tuned)**
  - **Global Performance Engineering**

HS.04.06 Knowledge Representation Technologies for Human Performance Enhancement

Practical Knowledge Representation Technologies

Global Performance Engineering will capitalize on knowledge archiving, distributed simulations, deployed smart agents, and hybrid performance modeling to support distributed performance engineering for individuals and teams. Knowledge archiving will provide central online repositories of performance models that support model-based training and performance aiding. Distributed simulations will provide an effective deployable training and assessment platform that has low bandwidth requirements and is cost-effective, maintainable and reusable. Deployed smart agents will provide intelligent deployed performance aiding, real-time reasoning and time-critical performance support. Hybrid performance modeling will provide a modeling language that parsimoniously supports both individual and team level performance engineering.

HS.04.06 Knowledge Representation Technologies for Human Performance Enhancement

Practical Knowledge Representation Technologies

HS.04.06 Knowledge Representation Technologies for Human Performance Enhancement

The TRAIN Lab at AFRL Brooks has recently completed the largest data collection effort ever conducted under true experimental conditions - over 300,000 hours of tightly controlled studies comparing rival theory-based approaches to teaching all categories of cognitive tasks. Consider this general description of an ICAI evaluation paradigm.

The instructional effectiveness of ICAI varies as a function of four classes of independent variables, including which ICAI features are implemented, which pedagogies are implemented, the nature of the targeted knowledge or skill, and sometimes, the characteristics of the student. Although this problem is combinatorially explosive, it was approachable and we obtained very useful results.

This empirical refinement of design guidelines into instructional engineering principles was tedious, but well worth the effort. The results are applicable to instruction in general, not just to ICAI. We derived a completely parameterized, fully specified model of human knowledge and skill acquisition that formally selects instructional approaches for any task that might be taught with computers. We have developed over forty ICAI systems based on the model, with an average effect size of 1 Sigma.

Although it is beyond the scope of this paper to fully describe the model, I will lay out the basic structure as it is the basis for derivation of the streamlined model to follow.

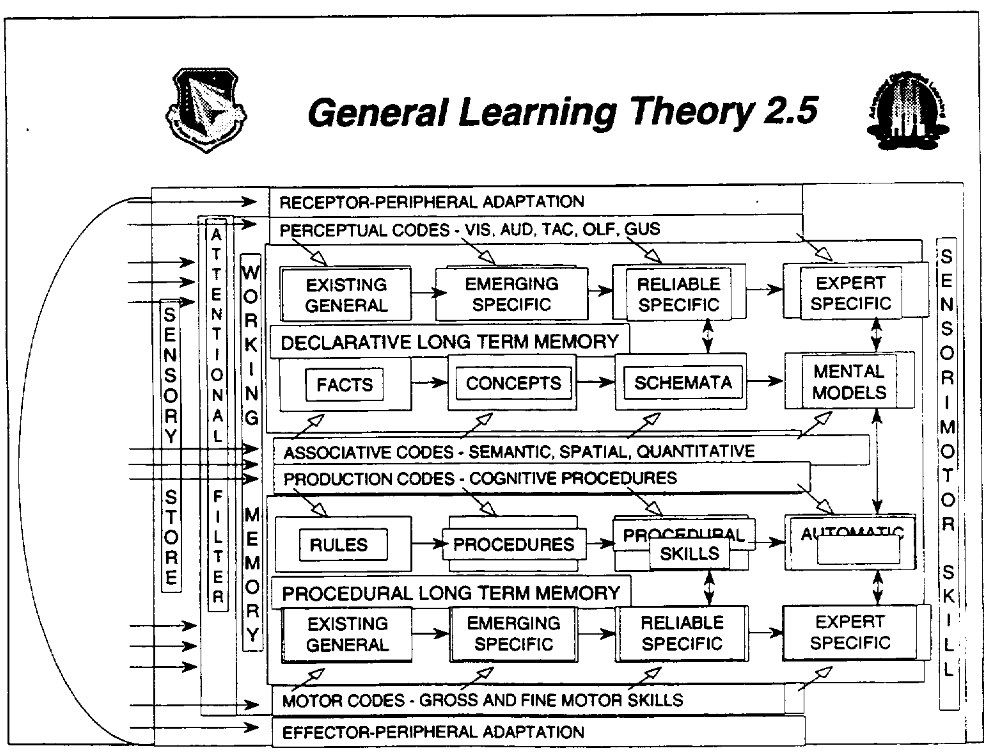
HS.04.06 Knowledge Representation Technologies for Human Performance Enhancement

- **Practical Knowledge Management Technologies**
  - **Standardized & Streamlined Cognitive (KR) Modeling**
  - **Efficient knowledge acquisition, maintenance, retrieval**
  - **Rapid Development tools for distributed KR-based agents**

The General Learning Theory 2.5

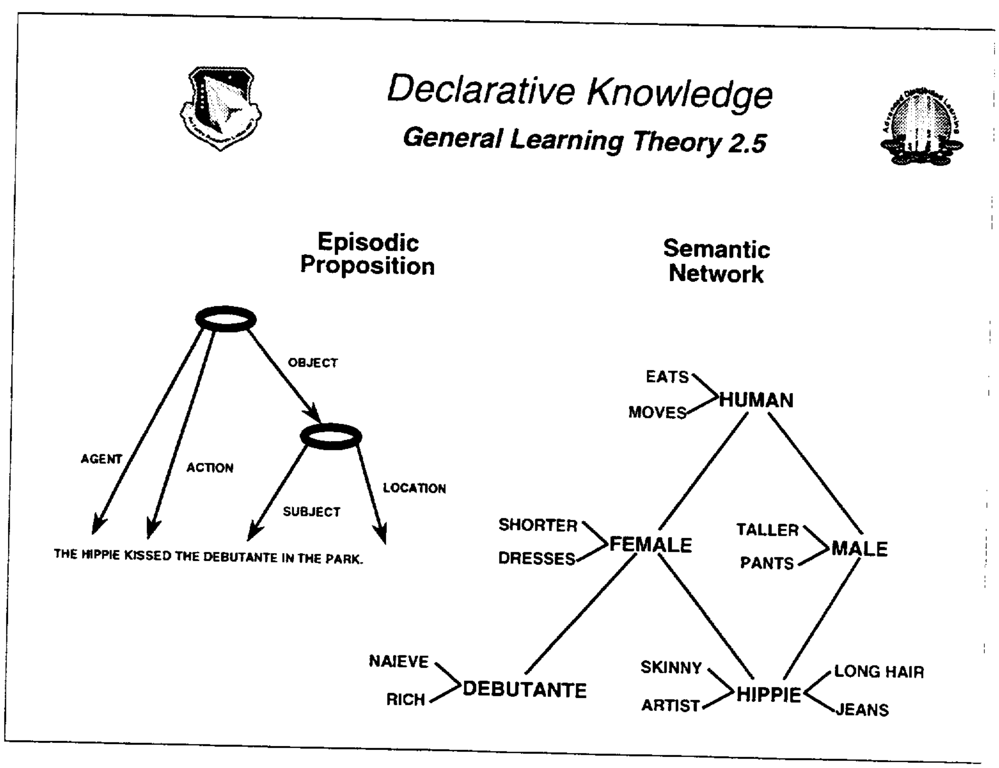
The General Learning Theory is intended to characterize acquisition processes associated with human knowledge and skill types. We sought sufficient precision and specificity to generate instructional manipulations for tasks and to predict the transferability of instructional manipulations across tasks. The degree to which any task is effected by an instructional manipulation depends on the degree to which the manipulation supports or inhibits task acquisition as specified by theory. Two similar tasks should be similarly effected by a given instructional manipulation, only to the degree that the two tasks require the same knowledge and skill types, in the same proportions.

According to General Learning Theory, learning occurs when incoming perceptions are encoded, synthesized and organized into increasingly complex long-term memory representations that result in enhanced performance capabilities. Selective attention to perceptual information encodes some of the information in working memory, and constitutes a learning opportunity. Learning occurs if some of the information is subsequently encoded into long-term memory. Early learning is primarily a process of accretion, where new knowledge/skill is added to existing knowledge/skill, and later learning is primarily a process of assimilation, where new connections are established among existing knowledge/skill, and abstractions are generated based on the common characteristics of existing knowledge/skill. Information in long-term memory can be organized along declarative or procedural lines, supporting task knowledge and performance.



Declarative Knowledge
General Learning Theory 2.5

Declarative LTM is associative in nature, and the basic unit of representation is an associative fact (e.g., X is Y). Declarative LTM associations are among basic propositional codes that support semantic, spatial, quantitative or perceptual information. Declarative knowledge is factual information that includes general world knowledge (semantic) and autobiographical knowledge (episodic). A formal distinction is often made between declarative knowledge that is episodic (e.g., I was bitten by a big, mean Doberman Pinscher named Charles), and declarative knowledge that is semantic (e.g., dogs sometimes bite). Episodic knowledge is thought to precede and underlie semantic knowledge. Declarative knowledge can be functionally represented as a hierarchical network of nodes and links, often called a semantic network (Collins and Quillian, 1969). Semantic networks have been shown to be cognitively plausible by studies showing that the hypothesized organization of the network structure is predictive of how long it takes people to answer questions. For intelligent tutoring in declarative domains, semantic networks have been used as student models by instantiating the network with the knowledge to be taught, and then tagging nodes as to whether the student has learned it or not (Carbonell, 1970). These networks are an economical way to represent large amounts of interrelated information, are easily inspectable, and support mixed-initiative dialogs between user and tutoring system. They are considerably less effective for representing procedural information (i.e., knowledge or skill related to doing things).

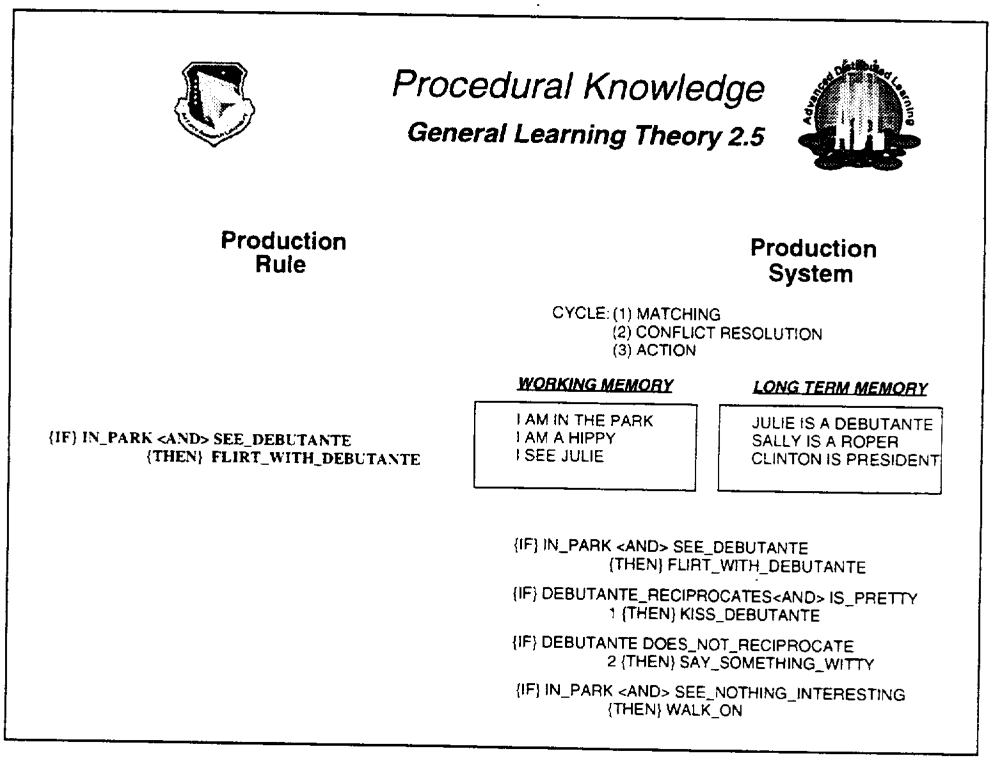


Procedural Knowledge
General Learning Theory 2.5

Procedural LTM is productive in nature, and the basic unit of representation is a production rule (e.g., if X then Y). Procedural LTM rules support cognitive information processing and related motor task performance. Procedural knowledge is knowing how to do something(s), and procedural skill is the demonstrable capability of doing so. For example, one may know how to cook pancakes, but not do it very well. Or one may know how to cook chili, and also do it quite well. In the former case (pancakes), one may be said to have procedural knowledge but not procedural skill. In the latter case (chili), one would have both procedural knowledge and skill. Procedural knowledge/skill can be functionally represented using a rule-based formalism, often called a production system (Anderson, 1983). Production systems have been shown to be cognitively plausible by studies showing that the hypothesized structure of the rule-base is predictive of what kinds of errors people make in solving problems. For intelligent tutoring in procedural domains, production systems have been used as student models in several ways. One way is to instantiate an expert (production) system with the knowledge/skill to be taught, and then teach the knowledge/skill to the student, keeping track of what is, and isn't, learned by tagging productions appropriately (Anderson, 1987; Goldstein, 1979). In another approach, expertise is modeled through negation by matching student errors to previously identified, common patterns of errors that are associated with incorrect or missing productions, or procedural "bugs" (Brown and Burton, 1978; Van Lehn, 1990).

Procedural Knowledge
General Learning Theory 2.5 (Cont'd.)

Production systems are a finely-grained way to represent procedural knowledge or skill, are easily implemented in most programming languages, and support a variety of straightforward ways to automate instruction because they directly represent the performance steps to be taught. They are, however, sub-optimal for representing declarative information, and the level of feedback that is most easily obtained may be too elemental for efficient instruction. The "bug library" approach to teaching procedural knowledge/skill is limited in that it is not possible to anticipate all possible procedural errors that students might manifest, and procedural bugs tend to be transient before disappearing altogether.

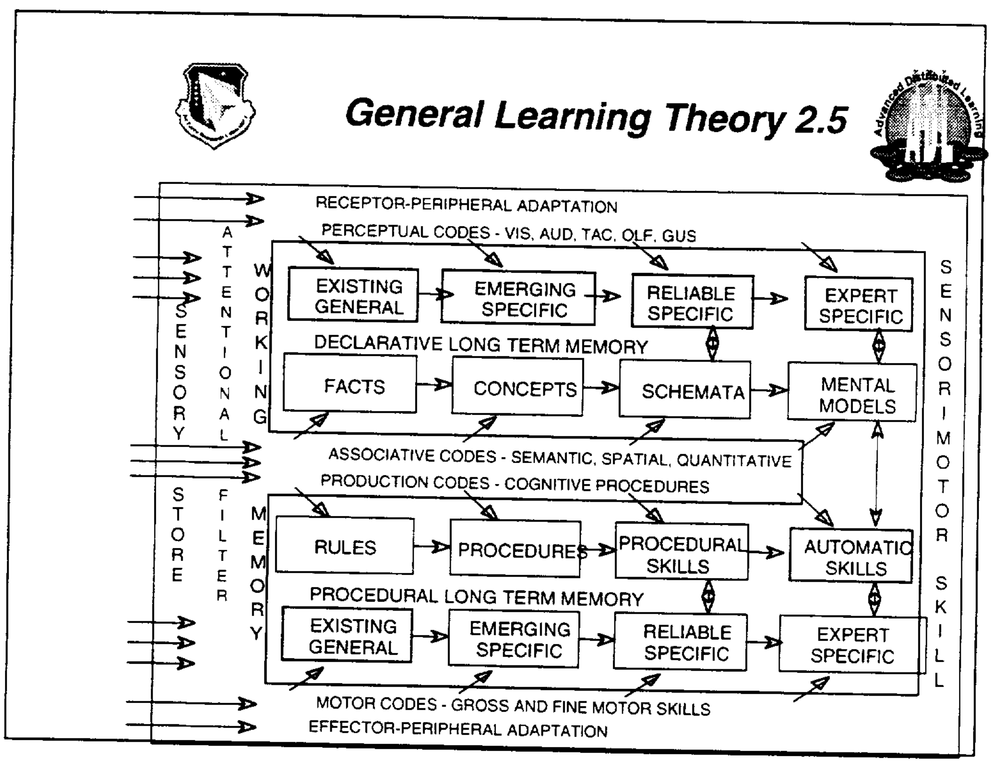


General Learning Theory 2.5

With appropriate instruction or experience, and effort, closely related facts are organized into concepts, and rules are organized into procedures. With further application, concepts are organized into more elaborate schemata and procedures are organized into production systems. With continued frequent application, schemata become refined into autonomous mental models, and procedural skills are compiled into autonomous skills. This highest level of organization, characteristic of task expertise, also produces tightly interrelated declarative and procedural representations, reducing conscious awareness of task steps. Mental models support qualitative reasoning and constitute a specialized, hybrid category of knowledge not well handled by either semantic networks or production systems. For example, reasoning about principles of electricity, complex weather systems, or human behavior seems to involve internalized mental models that contain both declarative information (e.g., knowledge about electrical components) and procedural information (e.g., knowledge about how electrical systems behave). Mental models allow humans to reason about how a system will behave under changing input conditions, either accurately or inaccurately.

General Learning Theory 2.5 (Cont'd.)

Regarding inaccurate mental models, students who think that electricity flows through wires (analogous to water flowing through pipes) will make predictable errors in reasoning about electricity. For purposes of intelligent tutoring, certain kinds of qualitative reasoning can be modeled by matching the student's beliefs and predictions to the beliefs and predictions associated with mental models that have been previously identified as characteristic of various levels of understanding or expertise. It is possible to infer what mental model the student currently holds, and contrive a way to show the student situations in which the model is wrong, thus pushing the student toward a more accurate mental model. This "progression of mental models" approach (White and Frederiksen, 1987) to teaching qualitative reasoning is ideal for remediating misconceptions, but cannot easily address other kinds of declarative knowledge or procedural knowledge/skill.



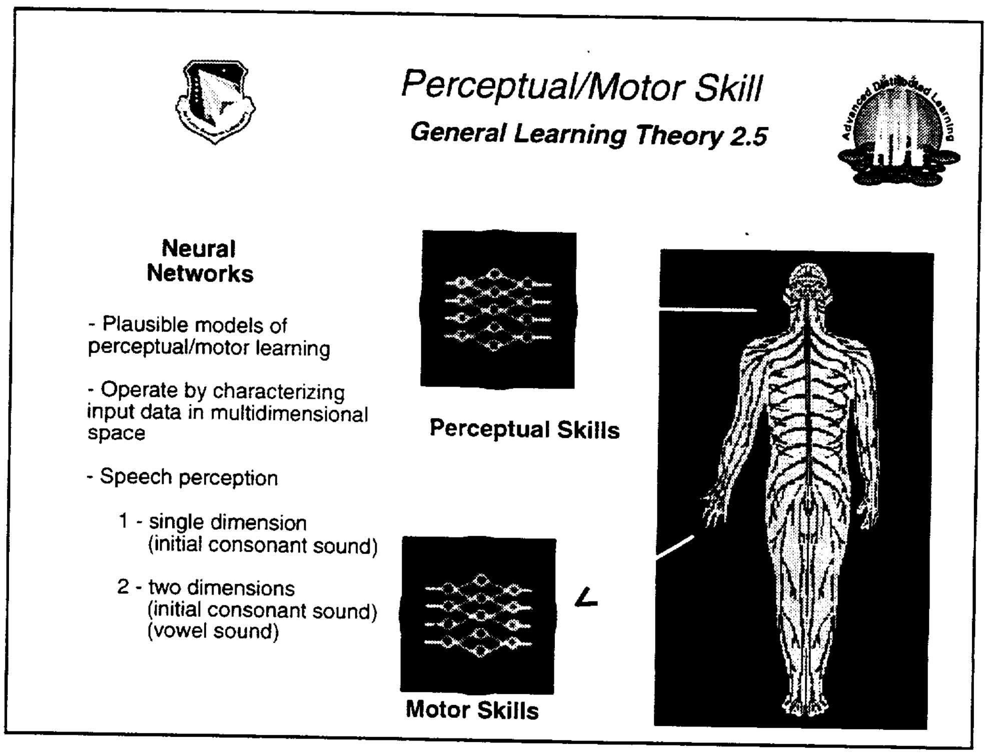
Perceptual/Motor Skill

General Learning Theory 2.5

Note that the most basic mechanisms of learning do not look at all like semantic networks or production systems. The two basic mechanisms involved in learning are 1) changes in cell membrane permeability, and 2) dendritic tree enrichment. These mechanisms may be driven by external stimuli (exogeneous), by internal stimuli (endogeneous), and/or by cognitive intention (thinking). However, for reasoning about or attempting to influence higher (more central) cognitive processes, it has decisively proven most useful to model learning using higher order abstractions - even though these are undoubtedly supported ultimately by dendritic reorganization. The basic mechanisms do warrant discussion with regard to more peripheral learning. The most peripheral learning phenomenon (adaptation) is not conscious to the learner, and is fully explained as dendritic reorganization with no CNS control. That is, as self organizing neural networks. A general statement would be that more peripheral learning is less influenced by learner awareness, strategies, etc., and more central learning is more dependent on conscious control by the learner. The more peripheral learning phenomena are best modeled as neural networks, which is exactly what they are. The effective manipulations for influencing learning are thus things like repetition, spacing, stimulus ordering, and feedback proximity, rather than semantic organization, mnemonics, or conceptual set.

Perceptual/Motor Skill (Cont'd.)
General Learning Theory 2.5

Skilled performance (perceptual, cognitive and motor) is primarily due to sufficient and appropriately organized/abstracted central nervous system (cognitive) knowledge representations. Skilled performance (perceptual and motor, respectively) is secondarily due to dendritic enrichment in receptor-peripheral and effector-peripheral neural ganglia. Skilled performance (sensorimotor) is primarily due to dendritic reorganization in receptor-peripheral and effector-peripheral neural ganglia. In both cases, neural networks are the appropriate model to apply. A neural network is best defined as a set of simple, highly interconnected processing elements that are capable of associating stimulus conditions to feedback states.



HS.04.06
FY98 Achievements - 1

As stated, it is beyond the scope of this paper to fully describe the General Learning Theory. For purposes of practical knowledge modeling on a very large scale, I adopted the goal of reducing the GLT by eliminating any information or distinction that would not lead one to do something differently in task training or support. I sought to develop the simplest possible description of human skill acquisition that was yet sufficient to support practical application. Further, I attempted to organize (both graphically and serially) the information in such a way as to enhance recall and utilization. Essentially I wanted to create a mental model of the human cognitive system that I could reliably teach to knowledge workers and that they could then reliably apply.

HS.04.06
FY98 Achievements - 1

- **Knowledge Representation Technologies for Human Performance Enhancement**
  - **HS.04.06: FY98 - FY01 KR-Tech**
  - **Practical Knowledge Representation Technologies**
    - **Standardized & Streamlined Cognitive (KR) Modeling**
    - **Efficient knowledge acquisition, maintenance, retrieval**
    - **Rapid Development tools for distributed KR-based agents**

ENGRAMS

Efficient Normative Grammar for Reasoning About Memory and Skill

We have developed a hierarchical set of knowledge representation schemes that cover the full range of human performance. ENGRAMS (Empirical Normative Grammar for Representing Acquired Memory and Skill) is a general purpose performance modeling scheme for all aspects of all tasks performed by individuals. ENGRAMS provides a taxonomy consisting of three dimensions, knowledge/skill type, performance level,and instructional approach. Knowledge/skill types include perceptual skill, declarative knowledge, procedural knowledge, and motor skill. Performance levels include novice, beginner, skilled, and expert. ENGRAMS is applied by first decomposing a target task into the knowledge/skill types and performance levels represented by these first two dimension. One then extracts instructional approaches from the third dimension of ENGRAMS.

Mental Models that support qualitative reasoning constitute a specialized category of knowledge not well handled by either semantic networks or production systems. For example, reasoning about principles of electricity, complex weather systems, or human behavior seems to involve internalized mental models that contain both declarative information (e.g., knowledge about electrical components) and procedural information (e.g., knowledge about how electrical systems behave). Mental models allow humans to reason about how a system will behave under changing input conditions, either accurately or inaccurately. Regarding inaccurate mental models, students who think that electricity flows through wires analogous to water flowing through pipes will make predictable errors in reasoning about electricity. For purposes of intelligent tutoring, certain kinds of qualitative reasoning can be modeled by matching the student's beliefs and predictions to the beliefs and predictions associated with mental models that have been previously identified as characteristic of various levels of understanding or expertise.

ENGRAMS
Efficient Normative Grammar for Reasoning About Memory and Skill (Cont'd.)

It is possible to infer what mental model the student currently holds, and contrive a way to show the student situations in which the model is wrong, thus pushing the student toward a more accurate mental model. This "progression of mental models" approach (White and Frederiksen, 1987) to teaching qualitative reasoning is ideal for remediating misconceptions, but cannot easily address other kinds of declarative knowledge or procedural knowledge/skill.

ENGRAMS
Efficient Normative Grammar for Reasoning About Memory and Skill

- **What Is It?**
  - **general purpose cognitive modeling scheme, notational system and planning tool for performance engineering on tasks performed by individuals**
  - **A normative lexicon and a set of related models setting forth a standard of usage for the practical engineering of human performance.**
  - **A simple organizational framework that simultaneously embodies knowledge and skill categories, the relationships among these categories, and their acquisition sequence.**
- **Strengths?**
  - **Provides a mental model of cognition for Instructional Engineers**
  - **During task analysis, the framework categories serve as prompts for decomposing the task into standardized performance-supporting representations.**
  - **After task analysis, the model then supports the planning of instruction by virtue of associated instructional principles that derive from the general learning theory.**

PCA Task Loop

Tasks that are performed by individuals include perceptual, cognitive, and motor demands. All such tasks can be broken down into one or more sequences of perception, cognition, and action. That is, the person must perceive the situation, decide what to do, and do it, with this sequence possibly repeating until the task is complete.

While virtually all tasks involve perceptual skills that must be available to achieve expert performance, for training purposes we focus on special perceptual skills that are not acquired outside of training. At each stage of learning, we build on available perceptual skills to develop next-stage perceptual skills.

Cognition refers to the operations on internal representations of task-relevant information. These representations may be first-order derivatives that maintain critical information about percepts, second-order representations that combine selected information from spatially or temporally distal percepts, or abstract representations with no perceptual basis. We focus on the type of representation that is developed in working memory and maintained over time, and the extent to which the operation draws on perceptually available information or information in long term memory. At each stage of learning, we build on available knowledge and skill to develop next-stage knowledge and skill.

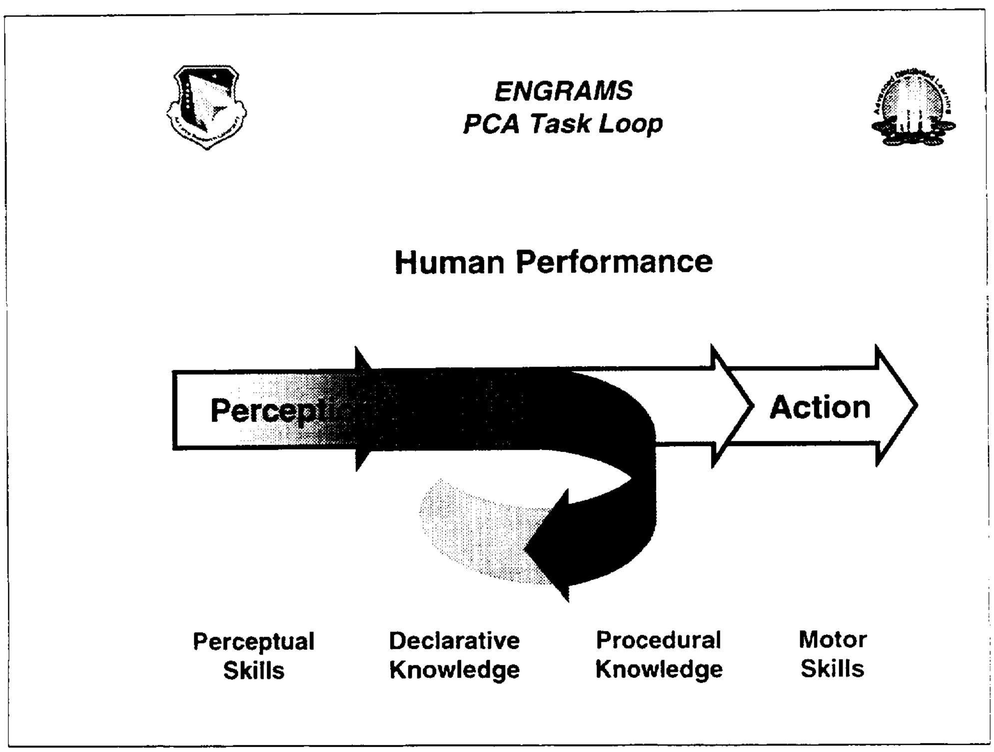
Every task requires some response, implying motor activity. We focus on those motor activities that are required to do the task but are not already available to the student. At each stage of learning, we build on available motor skills to develop next-stage motor skills.

Notice the interrelated nature of learning and performance. Task acquisition and task performance actually occur in sets of related iterative series of Perception-Cognition-Action loops. During learning, instruction and experience provide perceptual input to drive construction of appropriate working memory representations and their transfer to long-term memory, which serves to create, accrete, or modify knowledge representations in long-term memory. Task performance, then, capitalizes on input from the current task-relevant perceptual environment, including feedback from any previous action, to drive retrieval of appropriate long-term memory representations and their application in working memory, in order to select or generate and then execute a goal-driven action. Learning and performance are not independent constructs, as learning involves performances that are more or less related to the desired outcome performance. Similarly, ongoing task performance with feedback produces ongoing learning.

ENGRAMS
PCA Task Loop (Cont'd.)

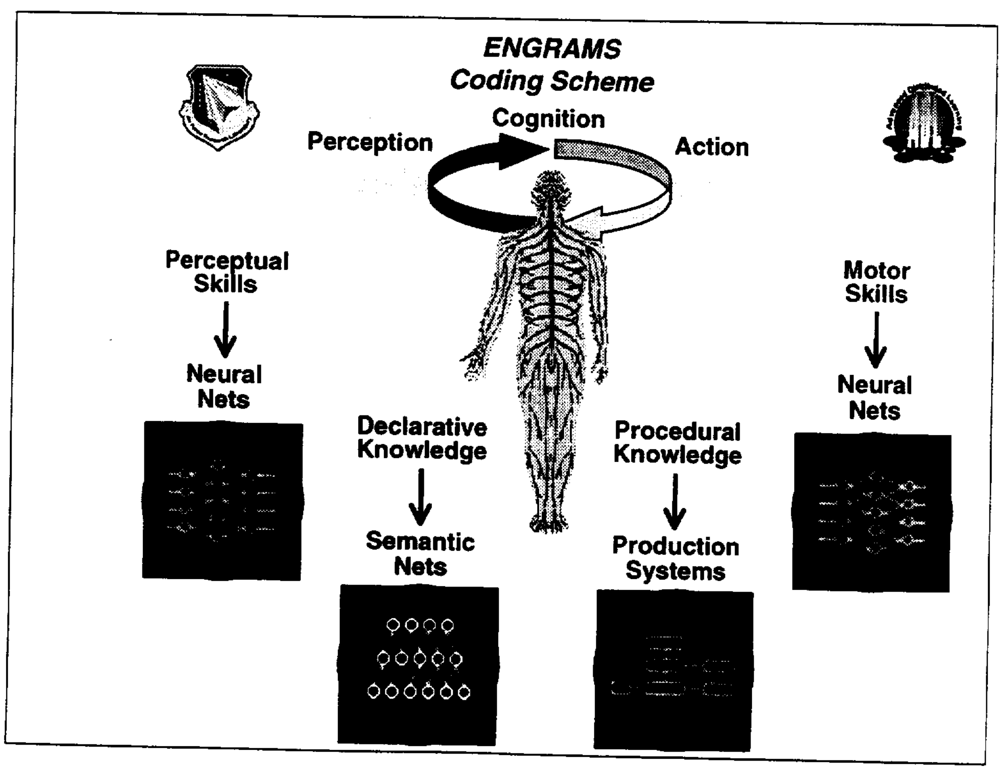
The design of any training program should reflect the underlying cognitive operations that support performance in the targeted task. Tasks may depend on greater or lesser contributions from declarative knowledge, procedural knowledge and skill, or performance skill determinants. These categories of cognitive operations may be said to lie along a continuum which runs from more knowledge-based to more performance-based (see, for example, Kyllonen and Shute, 1989).

While most complex tasks are supported to some degree by all of these categories of cognitive operations, many tasks are heavily weighted toward one end of the continuum. Some tasks, for example, are very knowledge-based, such as electronic troubleshooting or medical diagnosis. Others tend to be much more performance-based, such as cutting a diamond or piloting a high-performance aircraft. While both types of tasks require a certain amount of knowledge in order to perform them properly, knowledge-rich tasks tend to require more depth-of-knowledge and conscious effort on the part of the individual performing the task. The more performance-based tasks, on the other hand, require an assimilation of the required knowledge to the point where conscious effort is no longer required --- or may even be detrimental --- for superior task performance. Such an assimilation, or "automatization," of the task has the benefits mentioned above, namely: automated task performance allows the individual to perform other functions at the same time, and renders task performance highly reliable under stress (Schneider and Shiffrin, 1977).



ENGRAMS
Coding Scheme

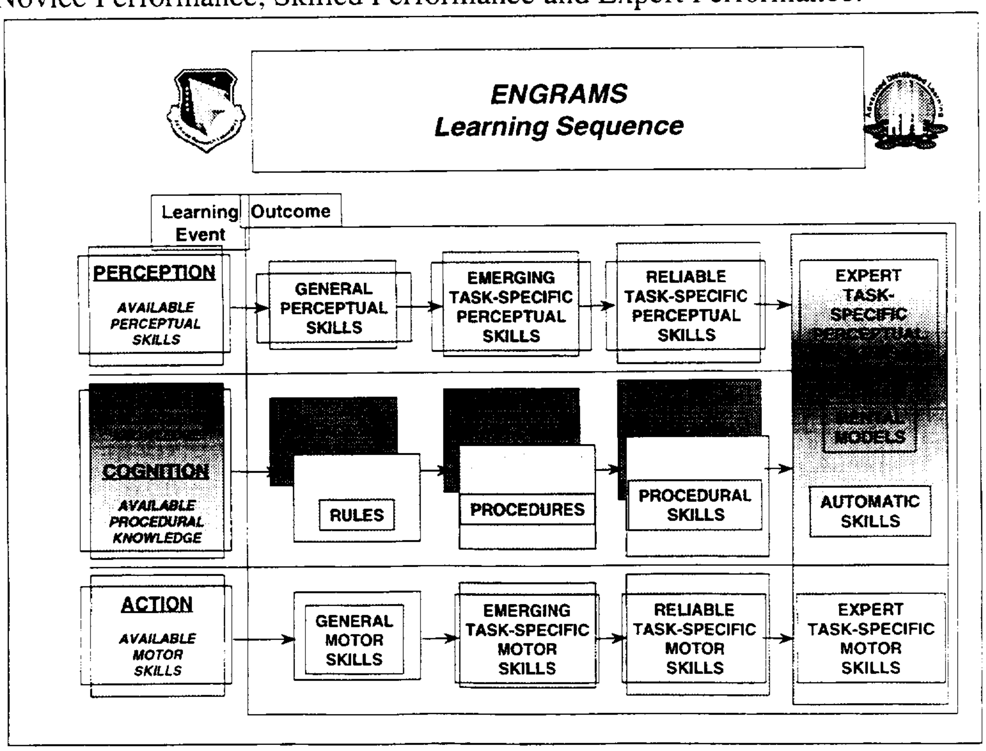
The design of any training program should reflect the underlying cognitive operations that support performance in the targeted task. Tasks may depend on greater or lesser contributions from perceptual skill, declarative knowledge, procedural knowledge, or performance skill determinants. These categories of cognitive operations may be said to lie along a continuum which runs from more knowledge-based to more performance-based . While most complex tasks are supported to some degree by all of these categories of cognitive operations, many tasks are heavily weighted toward one end of the continuum. Some tasks, for example, are very knowledge-based, such as electronic troubleshooting or medical diagnosis. Others tend to be much more performance-based, such as cutting a diamond or piloting a high-performance aircraft. While both types of tasks require a certain amount of knowledge in order to perform them properly, knowledge-rich tasks tend to require more depth-of-knowledge and conscious effort on the part of the individual performing the task. The more performance-based tasks, on the other hand, require an assimilation of the required knowledge to the point where conscious effort is no longer required, or may even be detrimental, for superior task performance. Such an assimilation, or "automatization," of the task has the benefits mentioned above, namely: automated task performance allows the individual to perform other functions at the same time, and renders task performance highly reliable under stress.



ENGRAMS
Learning Sequence

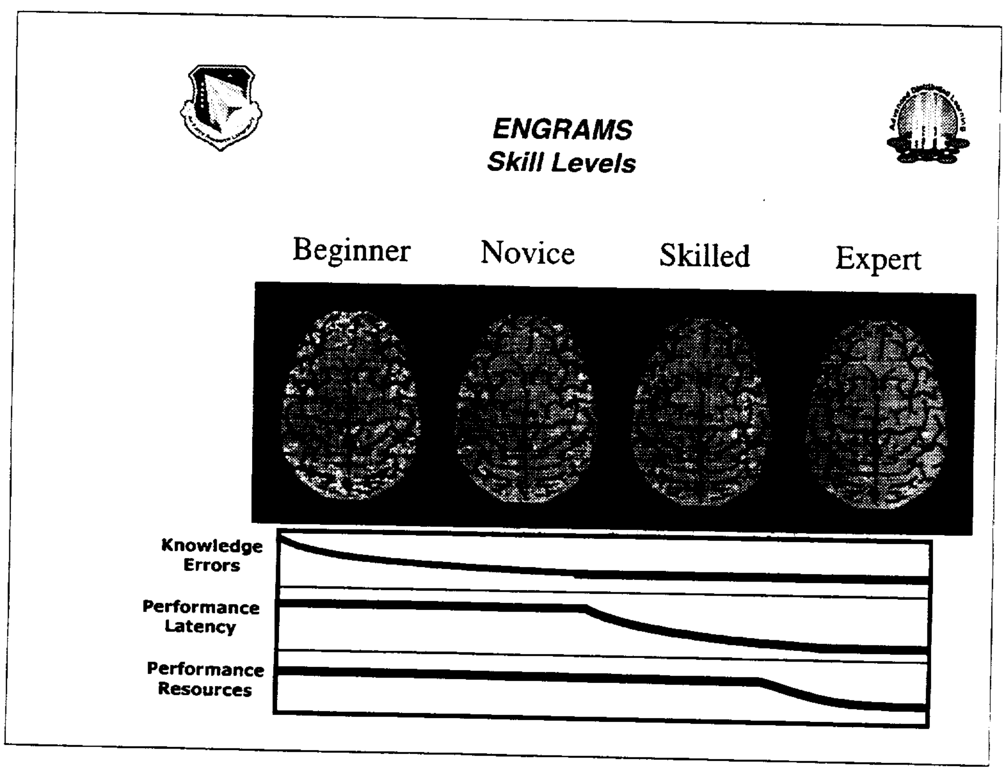
The ENGRAMS Learning Sequence simultaneously embodies a simplified cognitive taxonomy and learning model. It supports task analysis by providing a simple organizational framework that simultaneously embodies knowledge and skill categories, the relationships among these categories, and their acquisition sequence. During task analysis, the framework categories serve as prompts for decomposing the task into standardized performance-supporting representations. After task analysis, the model then supports the planning of instruction by virtue of associated instructional principles that derive from the general learning theory.

Performance levels on any task are characterized in terms of the representations supporting performance and of gross performance measures reflecting acquisition. The unit of input is a learning event. Learning events consist of PCA (Perception-Cognition-Action) sequences that produce enduring mental representations, which subsequently support task performance. The level of expertise that is reflected in task performance stems from the availability, quality, reliability, and applicability of representations supporting task performance. The goal of the Instructional Engineer is to cause students to efficiently acquire these representations. This is accomplished through an optimized sequence of learning events which cause the student to build up the appropriate representations. The sequence is optimized by a planned program of learning events where a) each event capitalizes on available knowledge and skill, and b) the sequence of events causes the student to encode representations in order of prerequisition. We define 4 levels of task expertise that provide manageable sub-goals in the planning of instruction. These are Basic Knowledge, Novice Performance, Skilled Performance and Expert Performance.



ENGRAMS
Skill Levels

Performance levels on any task are characterized in terms of the representations supporting performance and of gross performance measures reflecting acquisition. The unit of input is a learning event. Learning events consist of Perception-Cognition-Action sequences that produce enduring mental representations, which subsequently support task performance. The level of expertise that is reflected in task performance stems from the availability, quality, reliability, and applicability of representations supporting task performance. The goal of the instructional engineer is to cause students to efficiently acquire these representations. This is accomplished through an optimized sequence of learning events that cause the student to build up the appropriate representations. The sequence is optimized by a planned program of learning events where each event capitalizes on available knowledge and skill, and the sequence of events causes the student to encode representations in order of prerequisition. We define four levels of task expertise that provide manageable sub-goals in the planning of instruction. These levels are Basic Knowledge, Novice Performance, Skilled Performance, and Expert Performance. At any stage in learning to perform the task, the focus of training is on those trainable factors that limit performance improvements. Early training focuses on acquisition of accurate task knowledge and then basic-task performance. Next, focus shifts to efficient basic-task performance and then comprehensive full-task performance. Finally, focus shifts to effortless full-task performance.



ENGRAMS
Efficient Normative Grammar for Reasoning About Memory and Skill

Basic knowledge is defined as gross knowledge of the nature of the task, the context in which it applies, it's goals and methods, and the (seven or fewer) top-level steps involved in performance. Learning events to produce Basic Knowledge should require only general perceptual and motor skills and should capitalize heavily on existing knowledge and cognitive skill. Initial acquisition of any task should begin by assuring the encoding of task-requisite basic knowledge, which may be diagnosed during acquisition by accuracy of recognition and then recall. Supervisors of task performers should have at least basic knowledge of the task.

Novice Performance assumes the presence of Basic Knowledge but further requires detailed knowledge of all facts, concepts, rules, and procedures required for task performance. Novice performers manifest errorful performance, with errors decreasing as requisite knowledge, cognitive skill, perceptual skill, and/or motor skill develops. Learning events to produce Novice Performance should be ordered (bottom-up) according to hierarchical information subservience and capitalize on accuracy-based diagnosis of this knowledge and skill.

Skilled Performance assumes error-free and increasingly efficient performance. With frequent application, concepts are organized into more elaborate schemata and procedures are organized into production systems. Improvements in Skilled Performance are reflected in continued low error rates and decreasing latency of task performance.

ENGRAMS

Efficient Normative Grammar for Reasoning About Memory and Skill (Cont'd.)

Expert Performance assumes fast, low-error, and increasingly low-effort performance. "Low-effort" refers to task performance with minimized cognitive resource requirements. With continued frequent application, schemata become refined into autonomous mental models and procedural skills are compiled into efficient and autonomous skills, which may be indirectly measured by monitoring reductions in resource requirements.

ENGRAMS

Efficient Normative Grammar for Reasoning About Memory and Skill

	Perception	Cognition		Action
	Neural Networks	*Semantic Networks*	*Production Systems*	*Neural Networks*
	Perceptual Skill	**Declarative Knowledge**	**Procedural Knowledge/Skill**	**Motor Skill**
Expert Performance (effortless)	Expert Task-Specific Perceptual Skills	Mental Models Facts/Concepts/ Schemata organized as Autonomous Models	Automatic Skills Production Systems fully refined for Autonomous Execution	Expert Task-Specific Motor Skills
Skilled Performance (fast)	Reliable Task-Specific Perceptual Skills	Schemata Facts/Concepts organized as Abstract Explanatory Systems	Skills Production Systems sufficiently refined for Reliable Application	Reliable Task-Specific Motor Skills
Novice Performance (accurate)	Emerging Task-Specific Perceptual Skills	Concepts Facts organized as Simple Abstractions	Procedures Productions organized as Goal-Seeking Systems	Emerging Task-Specific Motor Skills
Basic Knowledge	Existing General Perceptual Skills	Facts Propositions organized as Associations	Rules Propositions organized as Productions	Existing General Motor Skills

HS.04.06
FY98 Achievements - 2

ENGRAMS is a general purpose cognitive modeling scheme for tasks performed by individuals. ENGRAMS represents any cognitive domain as a hierarchy of nodes including declarative knowledge (DK), procedural knowledge (PK), and perceptual/motor skill (S). In order to speed the process of developing cognitive tutors, we have developed DNA (Decompose, Network, Assess), an automated system for knowledge engineering that yields an ENGRAMS student model. The DNA program automates knowledge engineering by questioning subject matter experts in order to decompose the relevant domain expertise into its DK, PK, and S components. Then, DNA arranges these elements into hierarchies of task elements, creating an ENGRAMS knowledge model. Finally, DNA generates assessment probes to be used in automated tutoring. The assessment probes are embedded within problem-solving activities as part of the tutor. During tutoring, performance measures are used to update ENGRAMS' estimates of student mastery of the task. As a consequence of this diagnosis and assessment, ENGRAMS can prescribe different types of instruction and remediation for each type of curricular element until a student demonstrates through performance that he/she has attained mastery of the material.

HS.04.06
FY98 Achievements - 2

- **Practical tools for knowledge acquisition, storage, maintenance, retrieval, & application**
  - **ROSETTA (Rational Overlay System for Exploiting Traditional Task Analyses)**
    - **converts existing behavioral task analysis output to ENGRAMS model format**
    - **the first knowledge recoding Codec**
  - **DNA (Decompose, Network, Assess)**
    - **an automated system for knowledge engineering that yields an ENGRAMS model**

ROSETTA

Rational Overlay System for Exploiting Traditional Task Analyses

Rosetta is an algorithm for converting existing behavioral task analysis output to ENGRAMS model format. This is significant because most of the tasks that are performed in the DoD have already undergone behavioral task analysis. Rosetta provides a semi-automated method for converting these analyses to knowledge models - enabling cognitive engineering

ROSETTA

Rational Overlay System for Exploiting Traditional Task Analyses

- **What Is It?**
  - **Translates between two task analysis schemes schemes that cover the full range of human performance.**
  - **Traditional task analysis focuses on observable behaviors and has been in use for 25 years.**
  - **Cognitive task analysis focuses on the knowledge representations that are postulated to support the behaviors which were the focus of traditional task analysis.**
- **Strengths?**
  - **Provides an efficient method (80% solution) for translating existing behavioral task analysis output to ENGRAMS model format.**
  - **This is significant because most of the tasks that are performed in the DoD have already undergone behavioral task analysis.**
  - **Rosetta provides a semi-automated method for converting these analyses to cognitive task analyses**

ROSETTA

A Rosetta Stone for converting existing behavioral task analysis output to ENGRAMS model format. This is significant because most of the tasks that are performed in the DoD have already undergone behavioral task analysis. Rosetta provides a semi-automated method for converting these analyses to cognitive task analyses - which is required for automated tutoring.

ROSETTA

Behavioral Taxonomy
AETC Proficiency Codes

Cognitive Taxonomy
ENGRAMS Framework

Subject Knowledge	Task Knowledge	Task Performance		Declarative Knowledge	Procedural Knowledge	Perceptual/Motor Skill
D Evaluation Can evaluate conditions and make proper decisions about the subject.	d Advanced Theory Can predict, isolate, and resolve problems about the task.	4 Highly Proficient Can do the complete task quickly and accurately. Can show or tell others how to do the task.	Expert	Mental Models Facts/Concepts/ Schemata organized as Autonomous Models	Automatic Skills Production Systems fully refined for Autonomous Execution	Expert Task-Specific Perceptual Skills and/or Motor Skills
C Analysis Can analyze facts and principles and draw conclusions about the subject.	c Operating Principles Can identify why and when the task must be done and why each step is important.	3 Competent Can do all parts of the task. Needs only a spot check of completed work.	Skilled	Schemata Facts/Concepts organized as Abstract Explanatory Systems	Skills Production Systems sufficiently refined for Reliable Application	Reliable Task-Specific Perceptual Skills and/or Motor Skills
B Principles Can identify relationship of basic facts and state principles about the subject.	b Procedures Can determine step-by-step procedures for doing the task.	2 Partially Proficient Can do most parts of the task. Needs help only on hardest parts.	Novice	Concepts Facts organized as Simple Abstractions	Procedures Productions organized as Goal-Seeking Systems	Emerging Task-Specific Perceptual Skills and/or Motor Skills
A Facts Can identify basic facts and terms about the subject.	a Nomenclature Can name parts, tools, and simple facts about the task.	1 Extremely Limited Can do simple parts of the task. Needs to be told or shown how to do most of the task.	Basic	Facts Propositions organized as Associations	Rules Propositions organized as Productions	Existing General Perceptual Skills and/or Motor Skills

DNA
Decompose, Network, Assess

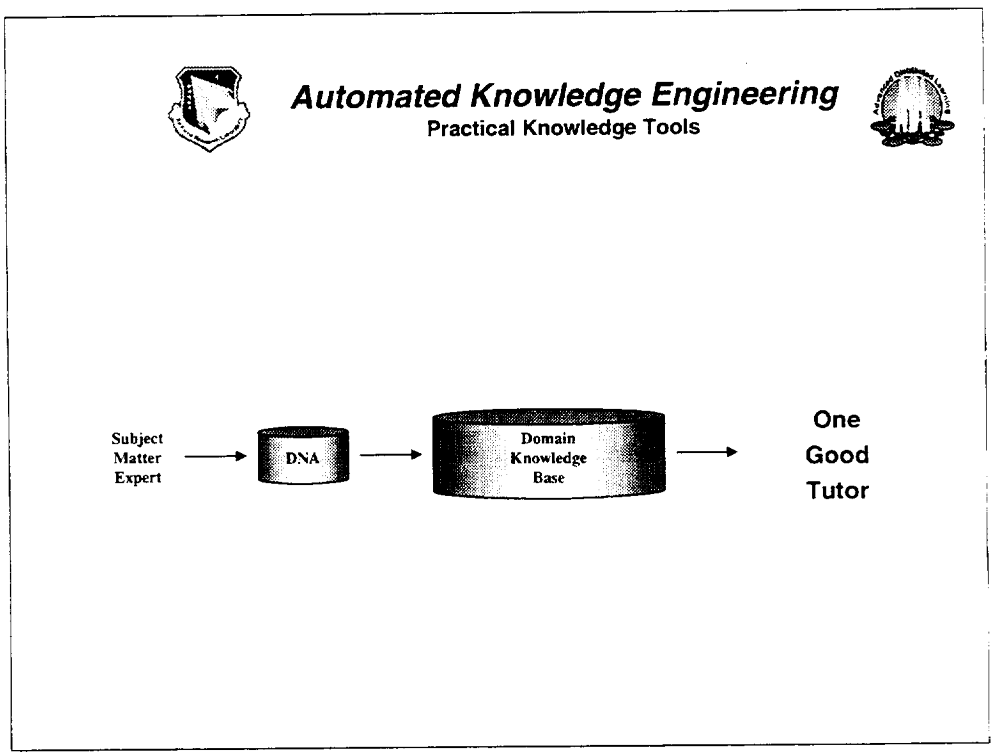
DNA (Decompose, Network, Assess) is an automated system for knowledge engineering that yields an ENGRAMS task model. The DNA program automates knowledge engineering by questioning subject matter experts in order to decompose the relevant domain expertise into knowledge components. Then, DNA arranges these elements into hierarchies, creating an ENGRAMS task model. Finally, DNA generates assessment probes to be used in automated tutoring. The assessment probes are embedded within problem-solving activities as part of the tutor.

DNA
Decompose, Network, Assess

- **What Is It?**
  - **DNA is an automated program for cognitive task analysis that decomposes and classifies experts' knowledge into organized knowledge structures suitable for training/instruction.**
  - **Networks extracted knowledge components into integrated hierarchical structures of curriculum elements, Knowledge Models**
  - **Decomposes knowledge into ENGRAMS knowledge types, then assesses the validity & reliability of the knowledge structures extracted.**
- **Strengths?**
  - **Automated: Offers the first practical automated CTA technique.**
  - **Theory-based: Produces broadly applicable methodology by combining proven CTA techniques (e.g., conceptual graph analysis, probed interview techniques).**
  - **User Friendly: Contains a lot of scaffolding (e.g., diverse examples, help files, orientation modules), so is easy to use for both instructional designer and SME.**

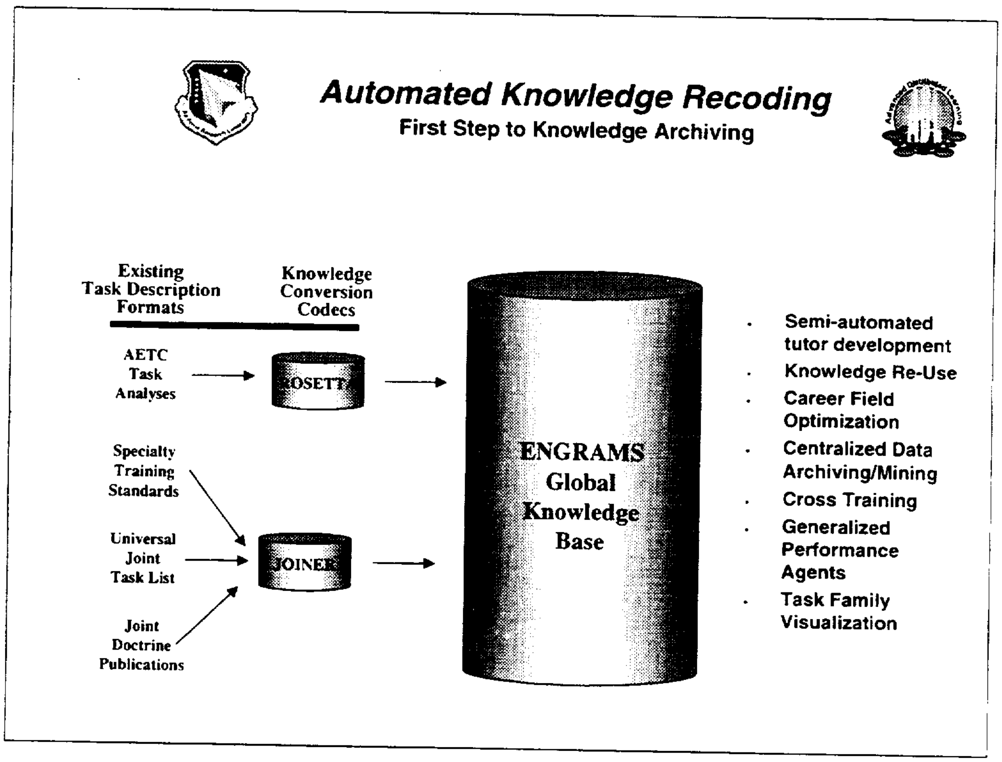
Automated Knowledge Engineering
Practical Knowledge Tools

DNA provides an efficient method for knowledge engineering (extraction and coding of human expert knowledge). Practical and scalable Knowledge Representation (KR) Technologies are the key to the future of human performance enhancement. A practical, reliable, standardized KR technology (efficient knowledge engineering plus standard knowledge representation) is crucial to achieving the goals of Vision 2010 as well as realizing the performance potential of expeditionary Air Forces. Given the distributed information era that we are entering - standardized KR technologies, will also lead quickly to practical KR-based job aiding, knowledge warehousing, cradle-to-grave (enlistee to retiree) student management, and career field management.



Automated Knowledge Recoding
First Step to Knowledge Archiving

ENGRAMS standardized knowledge representation (KR) method simplifies the coordinated acquisition, storage, maintenance, retrieval, and application of digitally coded human knowledge and skill. ENGRAMS opens the door for knowledge re-coding techniques (extraction and re-coding of existing, knowledge-bearing digital data) to bring economies of scale to knowledge management. Rosetta is the first knowledge re-coding Codec, but a second Codec is under development for re-coding of Joint Doctrine documents.



HS.04.06
FY98 Achievements - 3

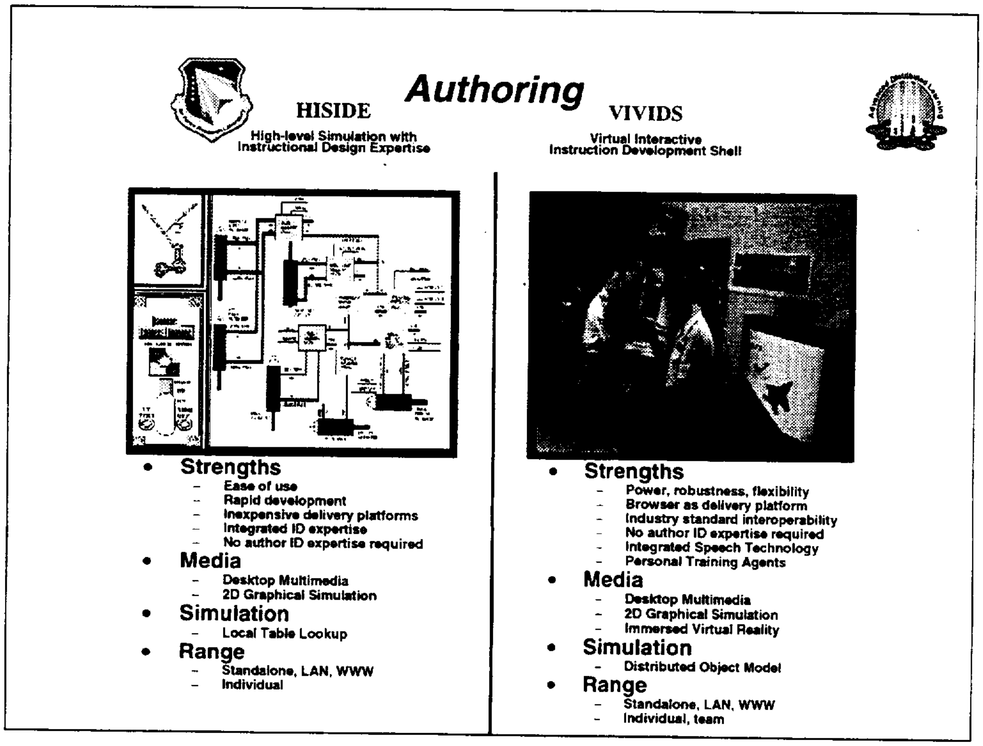
VIVIDS (Virtual Interactive Intelligent Tutoring System Development Shell) is an authoring shell that allows AETC schools to develop, deliver, and maintain virtual environment and simulation-based adaptive tutors based on the ENGRAMS. In 1998, we delivered the Advanced Virtual Adaptive Tutor for Air Traffic Control Readiness (AVATAR) to the 334th TRS, Keesler AFB. AVATAR saves one week of training time and costs only $75K for per station as compared to $750K for the commercial-off-the-shelf simulator it replaces. In addition, TRS personnel can use VIVIDS to develop new lessons. VIVIDS is also being used to develop the F15E Armament Maintenance Tutor for 3-level training at the 363rd TRS, Sheppard AFB. This tutor will enable the 363rd to produce mission ready technicians for this and future aircraft at reduced cost, and without requiring the TRS to have a representative airframe for students to practice on.

HS.04.06
FY98 Achievements - 3

- **Authoring tools for distributed simulation-based & knowledge-based courseware**
  - **HISIDE** (High-level Simulation with Instructional Design Expertise)
    - easy to use authoring shell with integrated instructional design expertise, for simulation-based tutors delivered on low-end platforms
  - **VIVIDS** (Virtual Interactive Intelligent Tutoring System Development Shell)
    - authoring shell that allows users to develop, deliver, and maintain virtual environment and simulation-based adaptive tutors
  - **AVATAR** (Advanced Virtual Agent Technologies for Air-Traffic-Control Readiness)
    - 334th TRS, Keesler AFB
    - saves one week of training time
    - costs 90% less than the commercial-off-the-shelf simulator it replaces
  - **F15E** (F15E Armament Maintenance Tutor)
    - 363rd TRS, Sheppard AFB
    - mission ready technicians at reduced cost
    - eliminates requirement for the TRS to have a representative airframe

Authoring

VIVIDS (Virtual Interactive Intelligent Tutoring System Development Shell) is an authoring shell that allows AETC schools to develop, deliver and maintain virtual environment and simulation-based adaptive tutors based on ENGRAMS



Global Performance Engineering

Stay Tuned

Global P.E.

By having a complete understanding as to the nature of the complex skill necessary to operate a proposed system, and of the stages of skill acquisition involved, it is possible to (1) design selection procedures which accurately predict later performance of the complex skill, (2) maximize the efficiency and effectiveness of training regimens, and (3) estimate training times that will be required to obtain and maintain specific desired skill levels. Policy makers should determine, therefore, those systems for which a high level of expertise is most critical, and then they should provide the opportunity and strongly reinforce trainees for achievement of expert performance. Given desired mission capabilities, skill acquisition specialists should be called upon to identify the level of proficiency required for successful system operation, to develop training technologies and protocols, and to estimate training times and practice schedules required to obtain/maintain that level of performance (see Newell and Rosenbloom, 1981, for a fascinating discussion of the relationship between extended practice and performance). In cases where there is a long delay between skill acquisition and skill application, such as is the case with reserve troops, skill acquisition specialists should be called upon to determine which task components will decay quickly and which will remain available over time, and to design training protocols for skill maintenance.

Global P.E.

- **Global Performance Engineering**
  - **Distributed knowledge management**
  - **Distributed Knowledge Agents**
  - **Distributed Cognitive Engineering**
  - **Distributed TEAM Performance Engineering**

Distributed Knowledge Management
Challenges

The greatest challenge in this area derives from converting the DoD training establishment from a behavioral psychology of human performance to a cognitive psychology of human performance. While the data supporting the power of this new approach is overwhelming, the behavioral approach to human performance is very entrenched in the internal DoD training hierarchies as well as among DoD training contractors. Policy definition/coordination activities must help create a commercial and federal environment that encourages this technology upgrade by providing carefully targeted examples. We are currently working with the Office of the Secretary of Defense, as well as leading companies from the commercial hardware and software industries to provide the required leadership.

Distributed Knowledge Management

- **Distributed Knowledge Management**
- **Centralized Data Archiving and Mining**
- **Knowledge Re-Use**
- **Real-time Knowledge-Based Decision Making**
- **Autonomous Situation Monitoring**

Distributed Knowledge Agents

Distributed Knowledge Agents

- Deployed personal agents for training and remediation
  - Highly Effective Deployed Training with Extremely Low Bandwidth Requirements
  - Intelligent Deployed Performance Aiding
  - Real-time Data Acquisition and Reasoning for Time-Critical Performance Support
- Distributed Simulations for Training & Assessment
  - Effective Deployable Training & Monitoring
  - Low Bandwidth Requirements
  - Supports Refresher & Just-in-Time Training
  - Cost-effective, Maintainable, and Reusable

Distributed Cognitive Engineering
Challenges

The greatest challenge in this area derives from converting the DoD training establishment from a behavioral psychology of human performance to a cognitive psychology of human performance. While the data supporting the power of this new approach is overwhelming, the behavioral approach to human performance is very entrenched in the internal DoD training hierarchies as well as among DoD training contractors. Policy definition/coordination activities must help create a commercial and federal environment that encourages this technology upgrade by providing carefully targeted examples. We are currently working with the Office of the Secretary of Defense, as well as leading companies from the commercial hardware and software industries to provide the required leadership.

Distributed Cognitive Engineering

- **Distributed Cognitive Engineering**
  - Tools for distributed human performance assessment, remediation, and support
  - Tools for distributed monitoring and lifecycle management of human performance
- **Career Performance Engineering for Individuals**
  - Training (initial, deployed, refresher, just-in-time, and embedded)
  - Aiding (tailored to the recipient and to the tactical environment)
  - Formal Methods for Training Planning (initial, deployed, and embedded)
- **Focus on Career-long Maintenance of Peak Performance Readiness**

Distributed TEAM Performance Engineering

Distributed TEAM Performance Engineering

- **Individual Performance Models in Team Context**
  - Model ideal team members to optimize learning for others
  - Model less-than-ideal team members for further learning
  - Train individuals when other team members are unavailable
  - Model various person-type configurations in teams
- **Team-Level Performance Models**
  - Team performance engineering
  - Model-based team selection, composition, and training
  - Knowledge-based team support systems
  - Focus on team decision errors under extreme stress
- **Deployed performance agents**
  - Intelligent Deployed Performance Aiding
  - Real-time Data Acquisition and Reasoning
  - Time-Critical Performance Support

ATP and the Adaptive Learning Systems Program

Harris L. Liebergot
Program Manager
Information Technology and Applications
Advanced Technology Program
National Institute of Standards and Technology
U.S. Department of Commerce
Gaithersburg, MD

ATP and the Adaptive Learning Systems Program

Dr. Harris L. Liebergot, Program Manager
Information Technology and Applications
Advanced Technology Program
National Institute of Standards and Technology
U.S. Department of Commerce
Gaithersburg, MD

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
atp™ ADVANCED TECHNOLOGY PROGRAM

ATP and the Adaptive Learning Systems Program

Dr. Harris L. Liebergot, Program Manager
Information Technology and Applications
Advanced Technology Program
National Institute of Standards and Technology
U.S. Department of Commerce

Tel: 301-975-5196 Fax: 301-926-9524
harris.liebergot@nist.gov http://www.atp.nist.gov/

Workshop on Advanced Training Technologies
and Learning Environments
NASA Langley Research Center
March 8, 1999

Advanced Technology Program

There are four main organizations at the National Institute of Standards and Technology (see below). The Measurements and Standards Laboratories are the technical leaders for the development and maintenance of U.S. standards and measurements. MEP operates a nationwide network of regionally based extension centers that help smaller manufacturers adopt advanced technologies and business practices. The Malcolm Baldridge National Quality Program has become both the U.S. standard of performance excellence in business and a comprehensive guide to quality improvement.

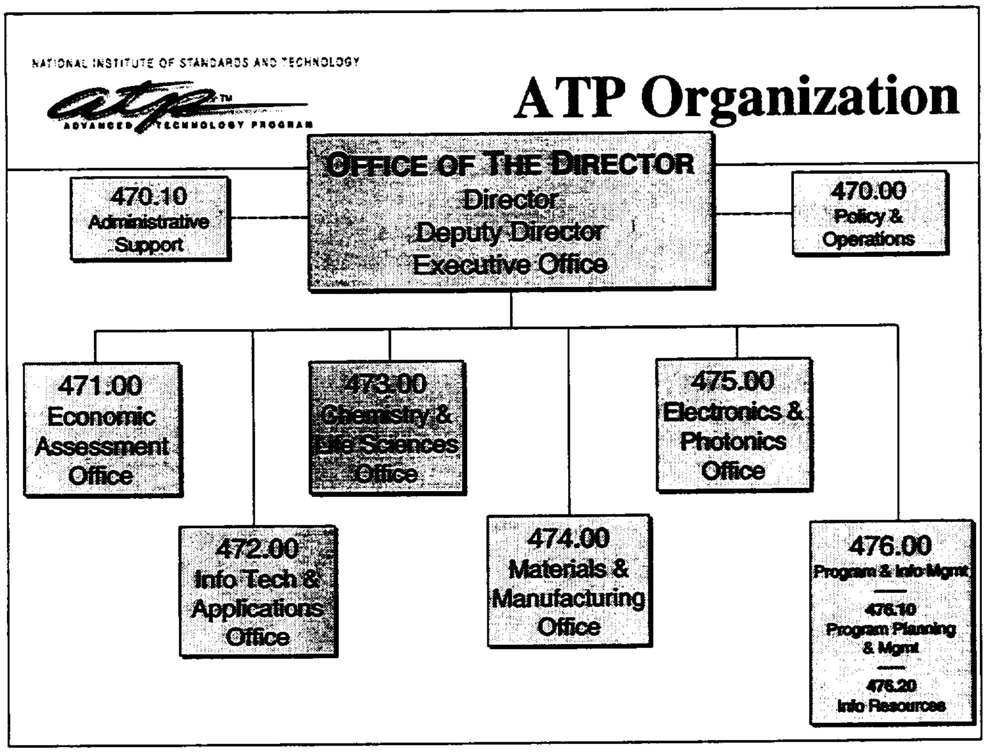
ADVANCED TECHNOLOGY PROGRAM

Who We Are ... One of 4 parts of the National Institute of Standards & Technology (NIST)

- Advanced Technology Program - $200M
- Measurements & Standards Laboratories - $300M
- Manufacturing Extension Partnership - $100M
- National Quality Program - $5M

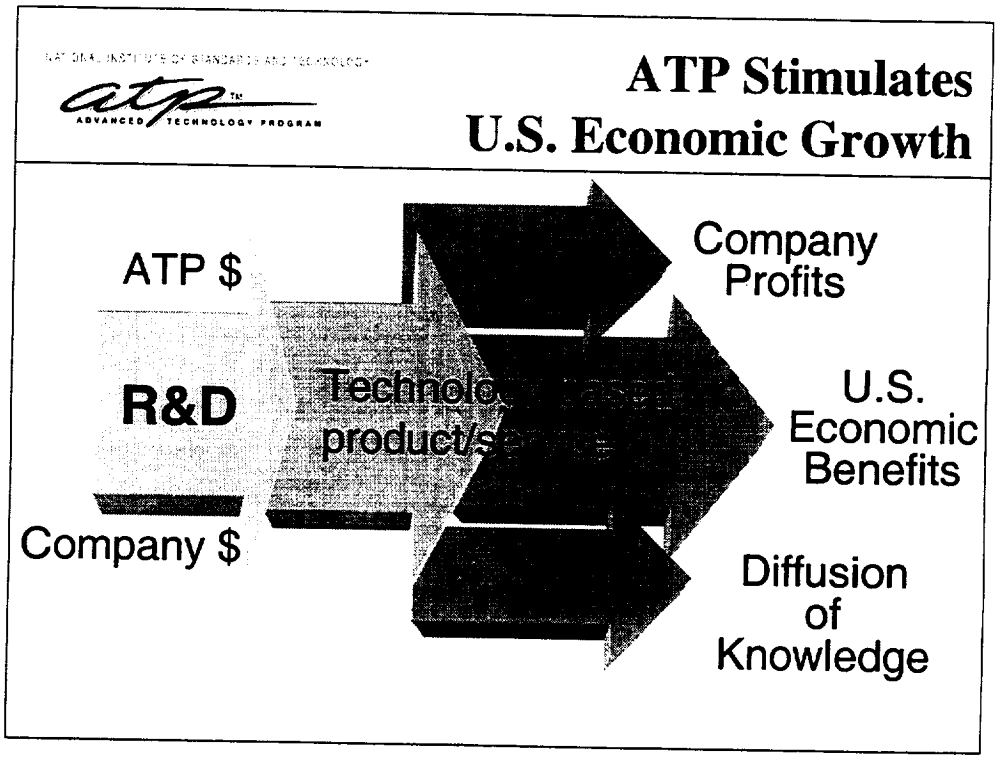
ATP Organization

ATP has approximately 80 personnel, shown organizationally below. About half work in the four offices on technology proposal evaluation and project management, others work on assessing the economic benefits of the technologies developed by companies working with ATP, and there are support people to help handle the volume of proposals and evaluation processes.



ATP Aims

ATP funds, leveraged with company funds, are used to develop technologies leading to eventual commercialization of products and services that yield widespread economic benefits, diffuse technological knowledge, and return profits to the companies that invest their dollars and personnel in working with ATP.

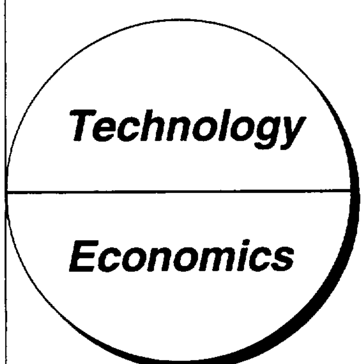


ATP Criteria

Projects which receive ATP funds must show that innovative technology will be developed and that widespread economic benefits will be realized. Proposals are judged equally on this basis. To meet technology criteria, the proposed project should be high risk but feasible, have a credible research plan, and contribute to the U.S. technology knowledge base. On the economic side, the proposal must demonstrate the technology's potential for broad-based economic benefits and explain why ATP funding is necessary and what difference ATP involvement is expected to make.

ATP Criteria

What We Look For in a Proposal



- Scientific and Technological Merit (50%)
  - ✓ Innovations in the Technology
  - ✓ High Technical Risk & Feasibility
  - ✓ Quality of R&D Plan
- Broad-Based Economic Benefits (50%)
  - ✓ Economic Benefits
  - ✓ Need for ATP Funding
  - ✓ Pathway to Economic Benefits

ATP Mission

ATP managers monitor the projects after awards are made to insure that the projects meet their maximum potential. ATP also helps companies to work together on projects where mutual benefits accrue to the participants and eventually to the U.S. as a whole.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

ADVANCED TECHNOLOGY PROGRAM

ATP Mission

The ATP does...

- Fund *applied R&D* where underlying science is known and there are clear objectives
- Insist that project proposals present a clear, coherent *research plan* illustrating how technical uncertainties will be tackled
- Manage awards with *cooperative agreements* that allow the ATP to be a partner
- Encourage companies to *work together* as collaborative R&D partners

ATP Mission

ATP's mission has some fundamental restrictions. ATP funding is not intended to replace private capital, nor is basic research on one end or commercialization on the other end, funded through an ATP award.

ADVANCED TECHNOLOGY PROGRAM

ATP Mission

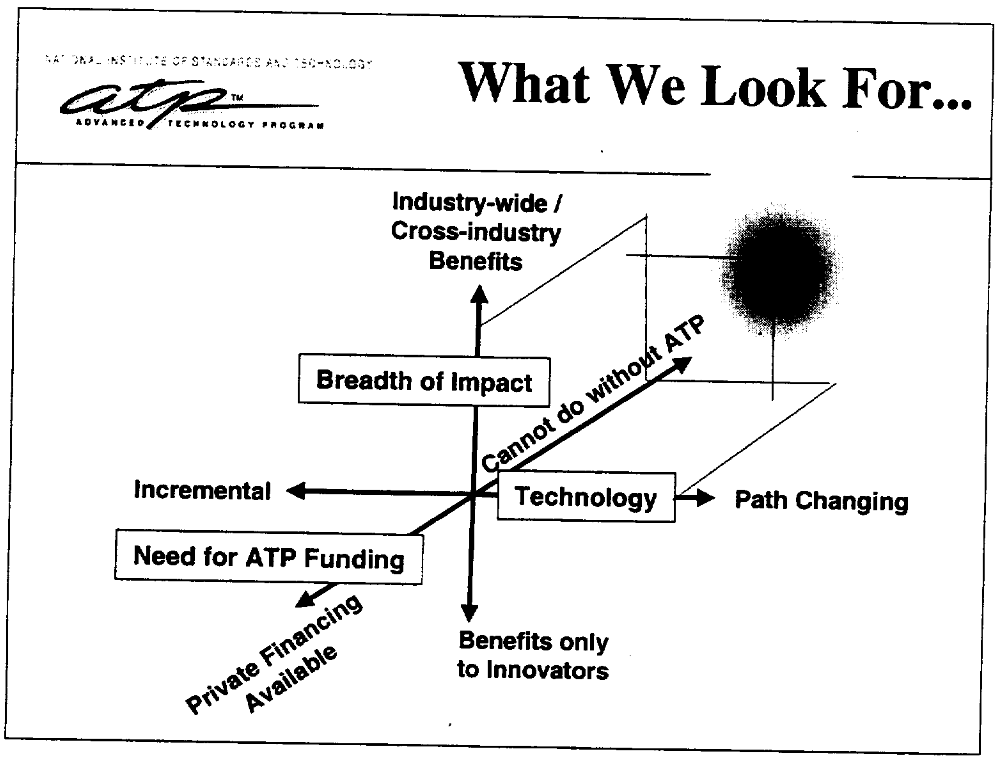
The ATP does NOT...

- Supplant private capital
- Fund basic research
- Fund product development

ATP Sphere of Activity

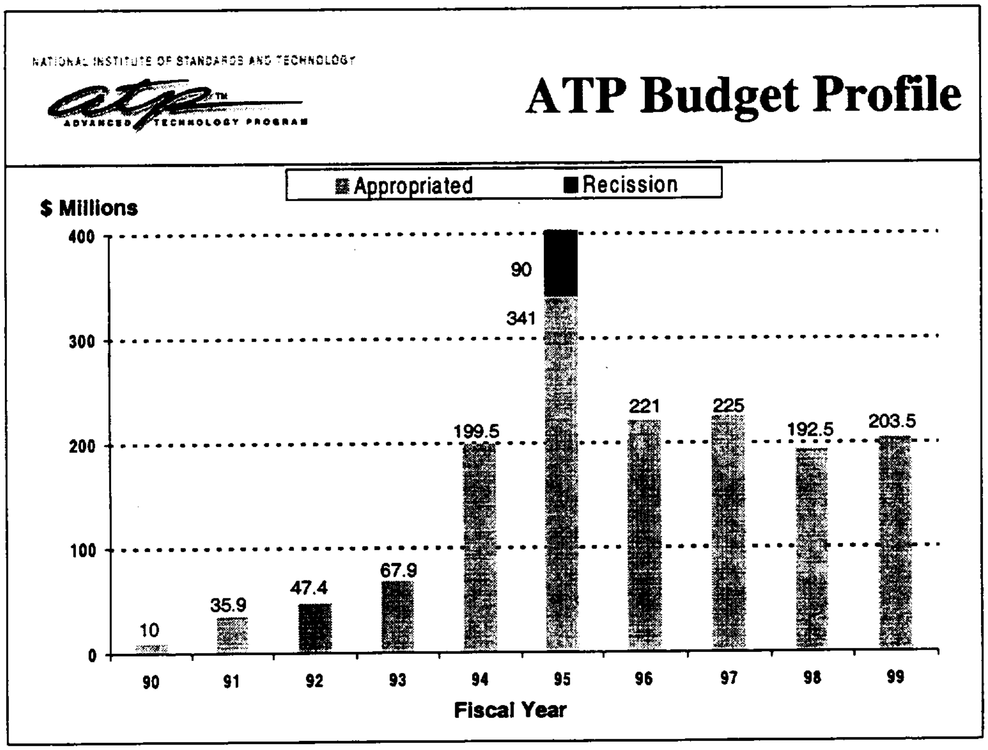
ATP invests in projects that:

- Develop path-changing technologies - revolutionary rather than incremental and that open up new possibilities;
- Offer industry-wide (broad-based) benefits, resulting in both infra-structural and multi-use technologies; and
- Would not be done without ATP funding - because of their tendency to be high-risk, have a long-range ROI, or are outside the normal bounds of the enterprise (i.e., needing collaboration among competitors, supply chain, etc.).



ATP Budget History

ATP's yearly budget history is shown below. In any given year, funds are used for both older multi-year projects, as well as new projects resulting from current year awards.



ATP Status

Shown below are the general results from proposal competitions which have been held to date.

Status of ATP

Our Status to Date

- 3,585 proposals submitted by industry
- 431 projects co-funded with 1,010 participants
- $2.783 billion of advanced technology development funded
  - ✓ ATP Share = $1.386 billion
  - ✓ Industry Share = $1.397 billion

ATP Rules

Shown here are basic eligibility criteria for single applicants.

ATP Eligibility

Rules for Single Applicants

- For-profit company
- ATP funding for up to 3 years
- ATP funding up to $2 million
- Company pays indirect project costs
  - ✓ Large company must pay 60% of total costs
- Intellectual property is owned by for-profit companies

ATP Rules

Shown here are basic eligibility criteria for joint venture applicants.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

ADVANCED TECHNOLOGY PROGRAM

ATP Eligibility

Rules for Joint Ventures

- At least 2 for-profit companies participating
- ATP funding for up to 5 years
- No limit on award amount
- Industry must pay at least 50% of the total project costs
- Intellectual property is owned by for-profit companies
- ATP encourages teaming arrangements

Other Statistics

Small businesses play a major role in ATP awards, and although universities and federal labs may not apply directly, they may participate as joint venture partners or subcontractors.

Our Partners

Our Partners

- Small businesses are thriving
  - > 50% of projects led by small businesses
  - Joint ventures have many small business participants
- Universities play a significant role
  - > 125 different universities involved
  - > 400 instances of participation
- Federal laboratories participate

Focused Programs

From 1994-1999, ATP sponsored both general competitions, in which any technology could be proposed, and focused competitions, aimed at technology areas which industry has indicated offer especially important opportunities for economic growth. For 1999, there will be only one competition, but ATP still maintains strong interest in the focused program technology areas shown below. Subsequent slides relate to the Adaptive Learning Systems program.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

ADVANCED TECHNOLOGY PROGRAM

ATP Focused Programs

- Tools for DNA Diagnostics
- Information Infrastructure for Healthcare
- Manufacturing Composite Structures
- Component-Based Software
- Technologies for the Integration of Manufacturing Applications (TIMA)
- Tissue Engineering
- Photonics Manufacturing
- Premium Power
- Selective-Membrane Platforms
- Catalysis and Biocatalysis Technologies
- Motor Vehicle Manufacturing Technology
- Digital Data Storage
- Digital Video in Information Networks
- Vapor Compression Refrigeration Technology
- Materials Processing for Heavy Manufacturing
- Microelectronics Manufacturing Infrastructure
- **Adaptive Learning Systems**

1998 Competition Results

Shown here are the numbers of proposals received and awards made as a result of the 1998 competition.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
ADVANCED TECHNOLOGY PROGRAM

1998 Competitions

Competition	*# of Proposals Received*	*# of Awards Made*
98-01 General Competition	167	23
98-02 Photonics Manufacturing	60	10
98-03 Premium Power	66	13
98-04 Digital Video in Information Networks	25	5
98-05 Catalysis & Biocatalysis Technologies	31	6
98-06 Microelectronics Manufacturing Infrastructure	55	9
98-07 Selective-Membrane Platforms	18	5
98-08 Tools for DNA Diagnostics	29	7
98-09 Adaptive Learning Systems	51	3
	502	81

ALS Program Overview

Both learners and trainers are dealing with systems that are often rather costly and difficult to modify. The learning system arena seems to offer an opportunity for economic growth for some time to come.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
atp ADVANCED TECHNOLOGY PROGRAM

Adaptive Learning Systems Program Drivers

End-user Demands

> learners and workers

> educators and trainers

Current Situation

> ed-tech too costly, isolated, static, difficult to use

> re-evaluation and restructuring in training and education

Opportunity

> potential spillovers, i.e.,gaming, office products

> growing export markets, i.e., software and hardware

> emergence of a National Learning Infrastructure

ALS Goals

The ATP ALS program seeks to stimulate high quality training or teaching curriculum development and delivery, while holding costs to a minimum. The resultant systems should be adaptable to a variety of learners and educators.

Program Goals

TECHNICAL

Revolutionary improvements in:

- ✓ production and delivery
- ✓ search and retrieval
- ✓ quality of service (QoS) in distributed instructional systems

BUSINESS

Transform ed-tech markets, by delivering content that is:

- ✓ affordable
- ✓ accessible
- ✓ adapted to learner and educator requirements

ALS Areas

Working with industry, the scope of the ALS program was defined to include four areas.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

ADVANCED TECHNOLOGY PROGRAM

Scope

CONTENT & AUTHORING

✓ intelligent authoring tools

✓ integrated learning resources

MODES OF DELIVERY

✓ simulated environments

✓ collaborative interaction

SEARCH & RETRIEVAL

✓ automated knowledge acquisition

✓ instructional component libraries

QUALITY OF SERVICE

✓ n-tiered architecture middleware

✓ performance support & services

Inclusions

ATP is definitely interested in proposals which suggest products or services that could be used in many different situations, are adaptable to a variety of educators and learners, and show marked improvements over existing applications.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

ADVANCED TECHNOLOGY PROGRAM

Inclusions

- Designed for open, Web-based, net-centric environment
- Reusable, integrated, and scalable
- Support to synchronous and asynchronous interaction
- Adapted to user requirements and diverse contexts
- Creative uses of client, server, and network
- Appropriate workload on learner and educator
- Measurable improvements in affordability, accessibility, adaptability

General Approaches

While high risk areas are targets, the proposals should still be grounded in good science and focus on the workplace. New ideas in handling resultant products and services are encouraged.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
ATP ADVANCED TECHNOLOGY PROGRAM

Encouraged

- Use of proven theory and practice
- Focus on U.S. workforce
- Innovative business models

Not Funded

Proposals that do not demonstrate improvements in widespread adaptability, cost, usability, etc., are not considered for funding.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
ADVANCED TECHNOLOGY PROGRAM

Exclusions

- Single-point, stand-alone, unscalable solutions
- Technologies incompatible with collaboration and distributed access
- Inability to demonstrate measurable gains in
  - ✓ affordability
  - ✓ usability or
  - ✓ accessibility
  - ✓ scalability
  - ✓ time-to-market

1998 ATP ALS Award

Real Education, Inc., was one of the winners of the 1998 competition in the Adaptive Learning Systems competition.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
ADVANCED TECHNOLOGY PROGRAM

Adaptive Intelligent Learning System

Real Education, Inc.

Aim: <u>A Web-Based Intelligent Tutoring System</u>

Innovation:

Combining Latent Semantic Analysis (meaning-based text retrieval) and Radial Basis Function (a neural network) to improve search accuracy through machine learning.

Benefits:

Improved retrieval rates for <u>relevant</u> documents; automatic meta-tagging by accepted classification schema; better hyper-links across disparate knowledge sources.

1998 ATP ALS Award

Teknowledge Corporation was one of the winners of the 1998 competition in the Adaptive Learning Systems competition.

Courseware Conversion Factory

Teknowledge Corporation

Aim: Rapid, High-Volume Conversion of Heterogeneous Content

Innovation:

KB systems for capture, conversion, interoperation, reuse;
system of hierarchies and meta-tags tailored to instruction;
interoperable software for wrapping and scripting.

Benefits:

Robust, usable tools for authoring and conversion;
diverse education and training testbed;
participation by ed-tech industry leaders.

1998 ATP ALS Award

ADAM Software was one of the winners of the 1998 competition in the Adaptive Learning Systems competition.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
ADVANCED TECHNOLOGY PROGRAM

Net-Based Authoring System
ADAM Software

Aim: Three-Tiered, Web-Based Authoring System for Anatomical Data

Innovation:
Support for interactive, online authoring with diverse inputs, e.g., annotations and 2-/3-D images; high-risk development at client tier, middle tier, and database tier.

Benefits:
Improvements in distribution, reuse and quality of content; wide application in education and training for K-12 and medicine; high-bandwidth solutions will transfer to other sectors.

1999 ATP Competition

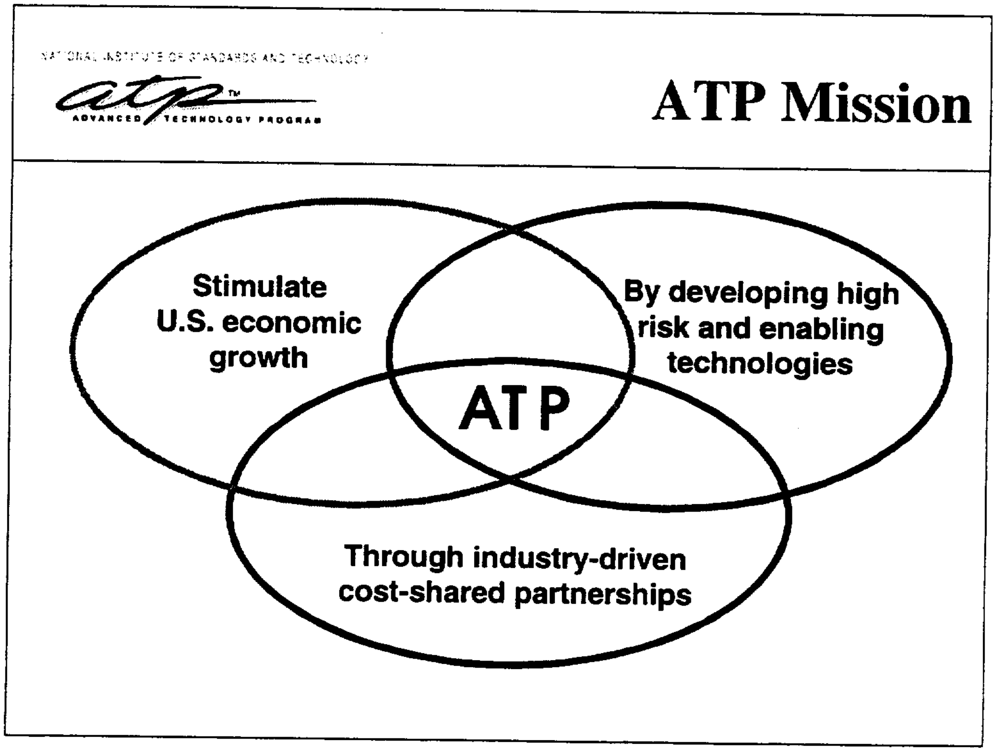
Logistics for the current year competition.

1999 Competition

- Single Competition
- Competition 99-01
- Proposals due April 14, 1999 at 3 p.m. Eastern Time
- $66 million in first-year funding for new awards

ATP Mission

ATP focuses on helping U.S. economic growth via cost shared projects with industry that result in advanced technology based products and services reaching marketplace potential as rapidly as possible.



Engineering a Successful Corporate Technology Supported Learning Program: A Department of Energy Progress Report

Tanya Luckett
Office of Training and Human Resource Development
U.S. Department of Energy
Washington, D.C.

Presented by

Paul R. Aaron
U.S. Department of Energy
Washington, D.C.

and

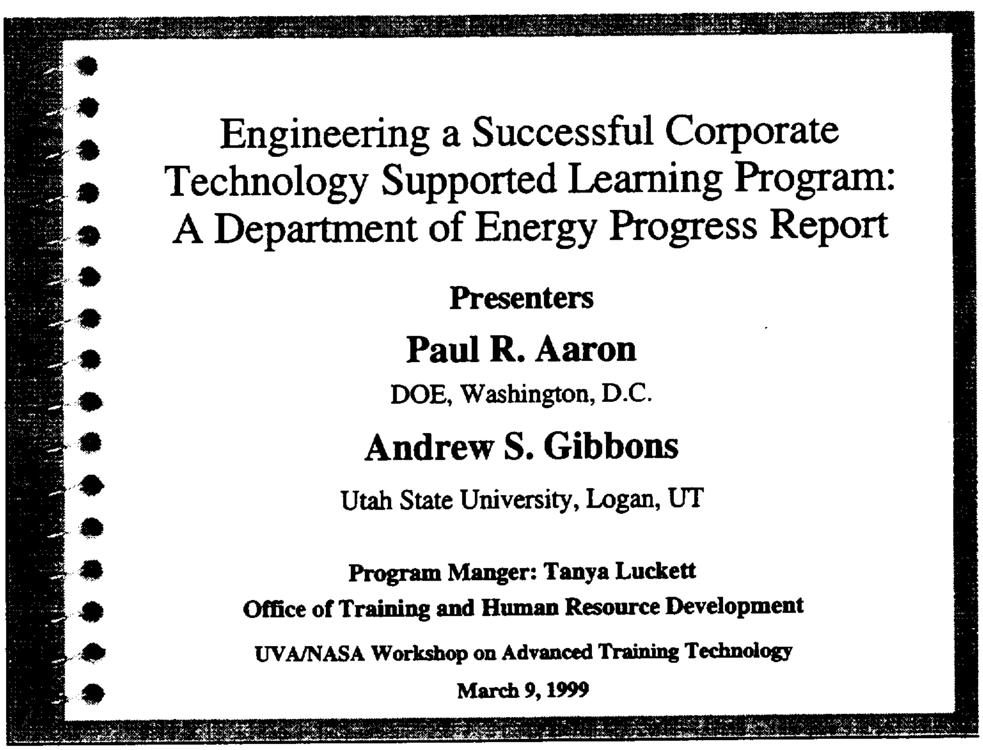
Andrew S. Gibbons
Utah State University
Logan, UT

Engineering a Successful Corporate Technology Supported Learning Program: A Department of Energy Progress Report

Presented by
Paul R. Aaron
DOE, Washington, D.C.
and
Andrew S. Gibbons
Utah State University, Logan, UT

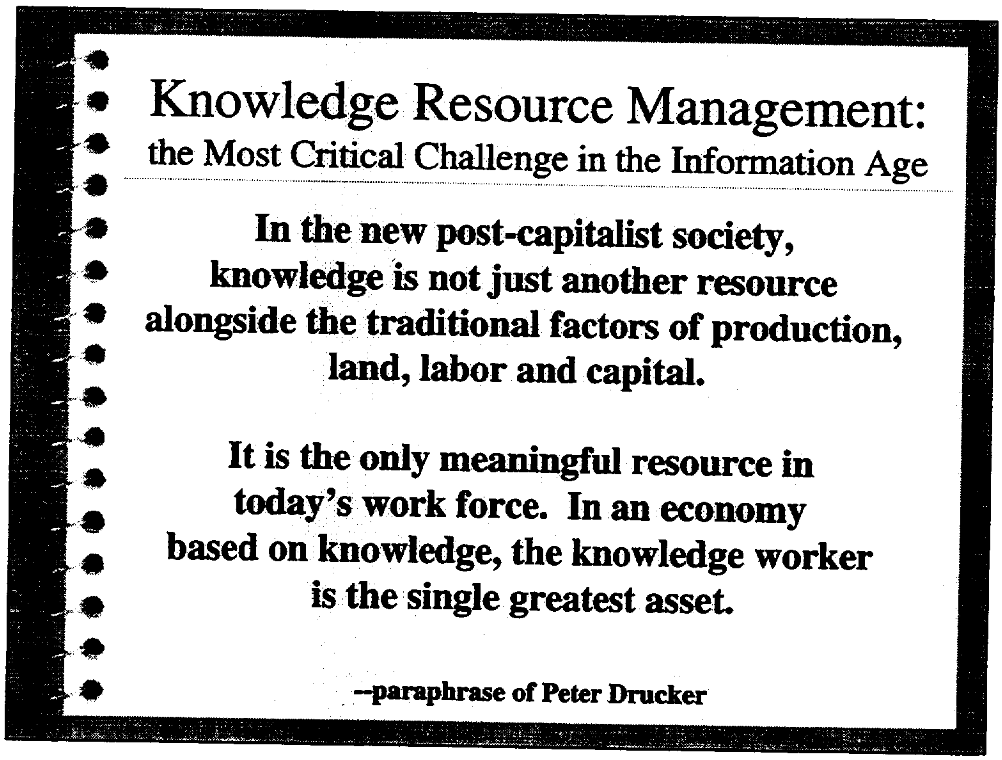
Program Manager: Tanya Luckett
Office of Training and Human Resource Development

Presentation given at the NASA Workshop on Advanced Training Technology, March 9, 1999. Tanya Luckett of the Department of Energy (DOE), Office of Training and Human Resource Development was to be the presenter. Other obligations necessitated asking Paul Aaron, an Information Management Specialist within DOE and Dr. Andy Gibbons, a consultant to DOE (Professor of Instructional Technology at Utah State University) to make the presentation.



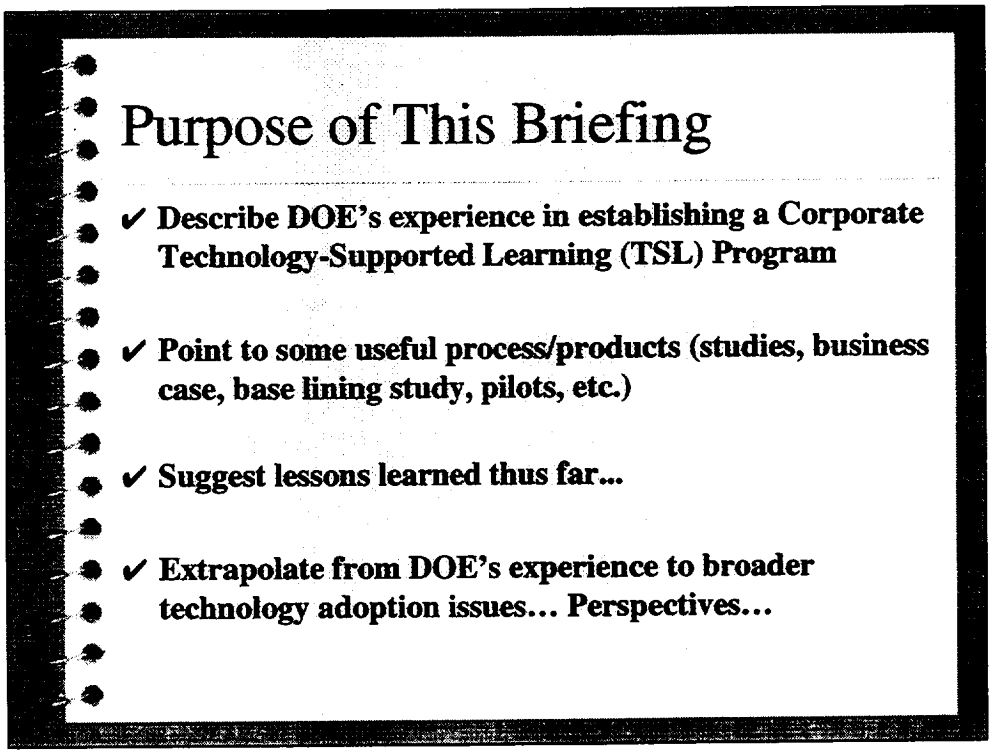
Knowledge Resource Management: The Most Critical Challenge in the Information Age

At DOE we are vitally concerned with maintaining and enhancing our single greatest asset, our knowledge worker. Given the complex nature of work at DOE, most of the federal and contractor employees are knowledge workers, many exclusively so.



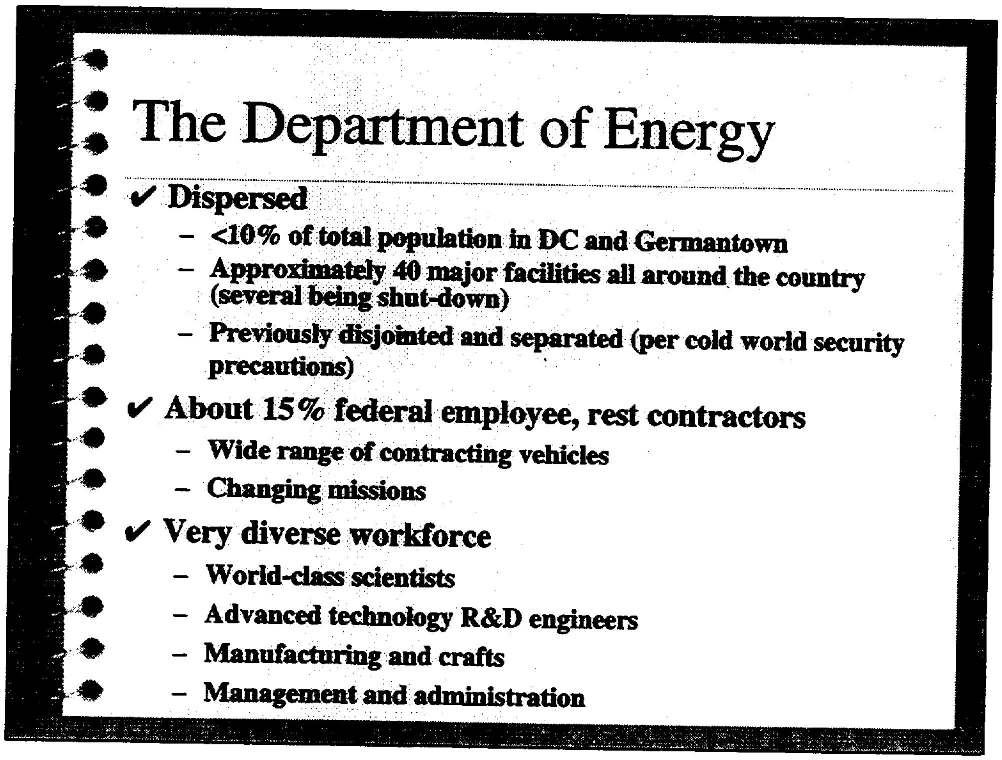
Purpose of This Briefing

We would like to share our experiences thus far in implementing a Corporate Technology-Supported Learning program.



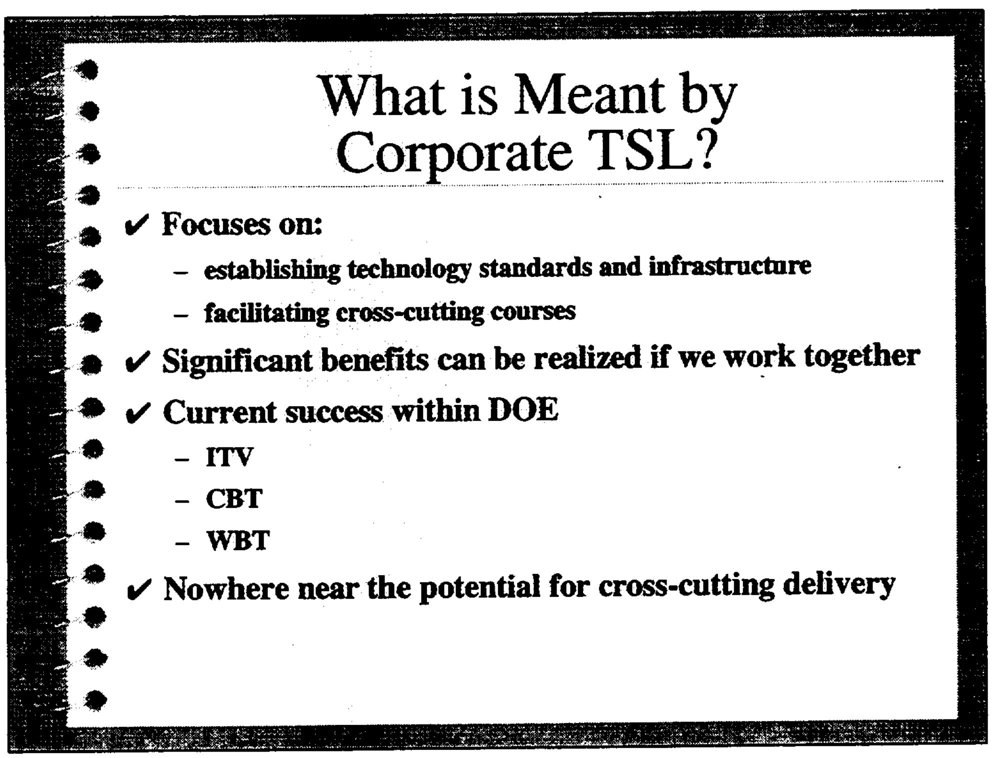
The Department of Energy

Creating a DOE-wide program is a tremendous challenge because of its dispersion, mixture of federal and contractor roles, and tremendous diversity in mission, roles and expertise.



What is Meant by Corporate TSL?

At DOE, we have long had significant success with TSL at local sites including several of the National Labs and within our general employee training, radiation worker, emergency preparedness, environmental and safeguards and security areas. We are now in the process of creating a "Corporate" TSL system and infrastructure focused on setting up an infrastructure and standards for distributing cross-cutting content. While DOE is realizing millions of dollars in savings locally through the use of TSL, we believe that even greater savings are possible through corporate programs.



Advanced Training Technology Formats Currently Being Used Within DOE

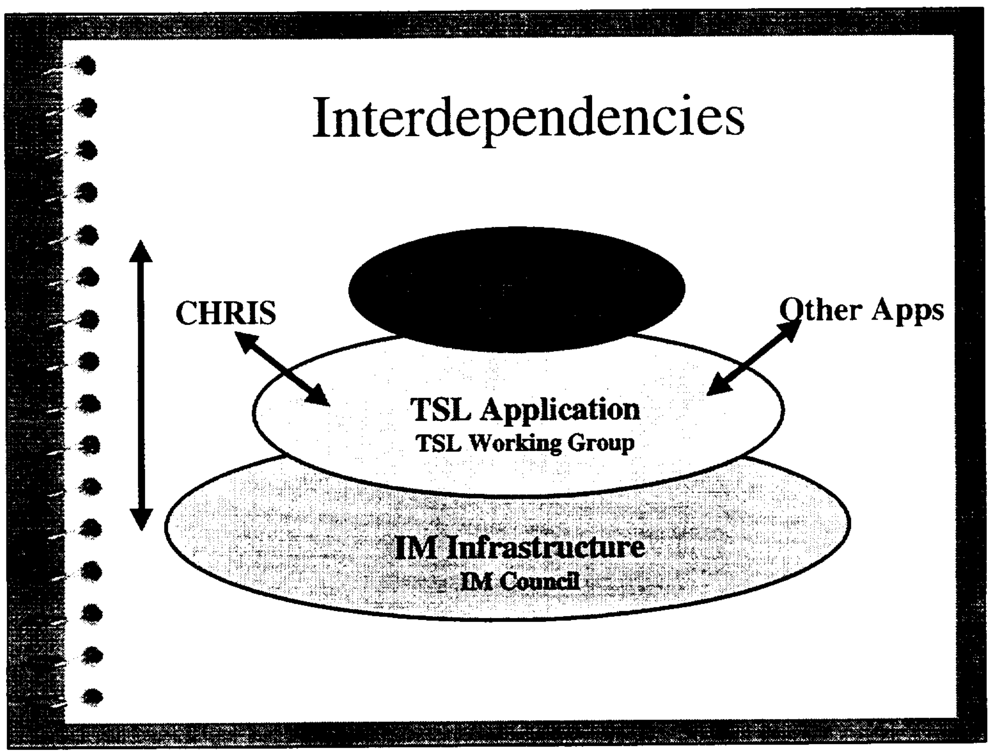
DOE has an excellent Interactive TeleVision system, primarily used for safeguards and security courses. We also have numerous learning centers and more and more desktop multimedia systems capable of LAN and web-based training.

Advanced Training Technology Formats Currently Being Used Within DOE

- ✔ **Interactive TeleVision (ITV)**
  - Satellite 1-way
  - Satellite 1-way video/ 2-way audio (NNSI)
  - Terrestrial 2-way audio and video (mostly for meetings)
- ✔ **Learning Centers with Multimedia**
  - Several established using video discs in early 90s
  - At more advanced sites, centers are actually disappearing
- ✔ **Desktop (and LAN-Based) Multimedia**
  - Simple page turners to advanced simulations
- ✔ **Web-Based**
  - Asynchronous courses (varied sophistication) and tests
  - Experimenting with synchronous classes

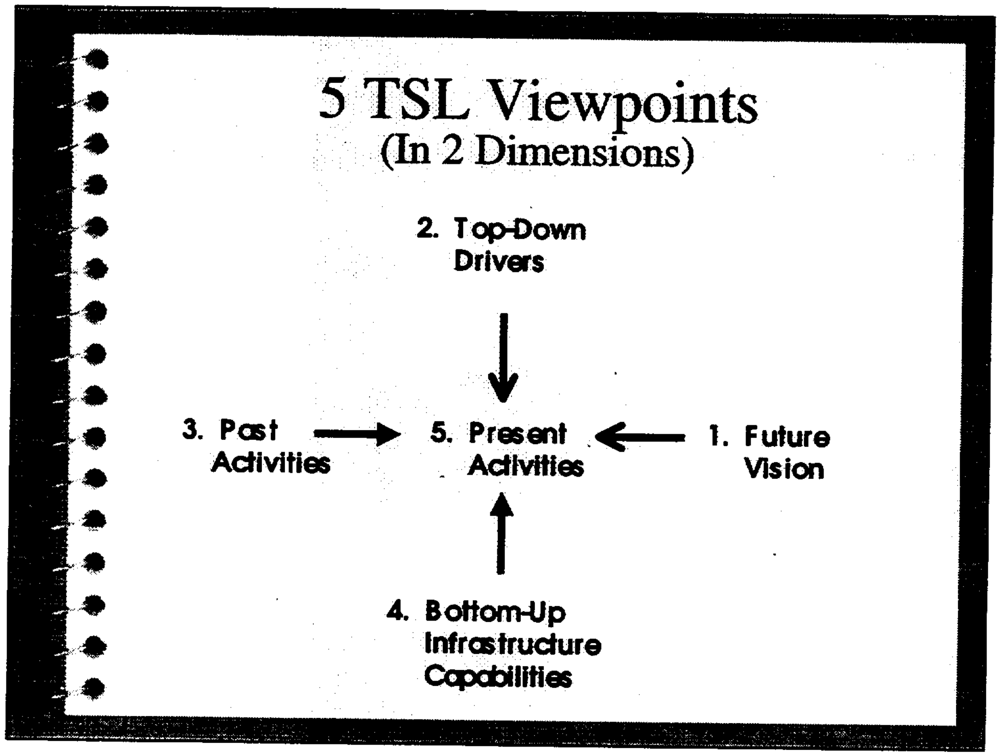
Interdependencies

The TSL Program sees itself as a bridge or conduit between content providers and the hardware, software and standards infrastructure. The TSL Program helps translate the needs of the centers of excellence and other corporate content owners and those responsible for designing, building and maintaining the information management infrastructure. Going the opposite direction, the TSL Program helps the content owners understand the current and future capabilities of the infrastructure.



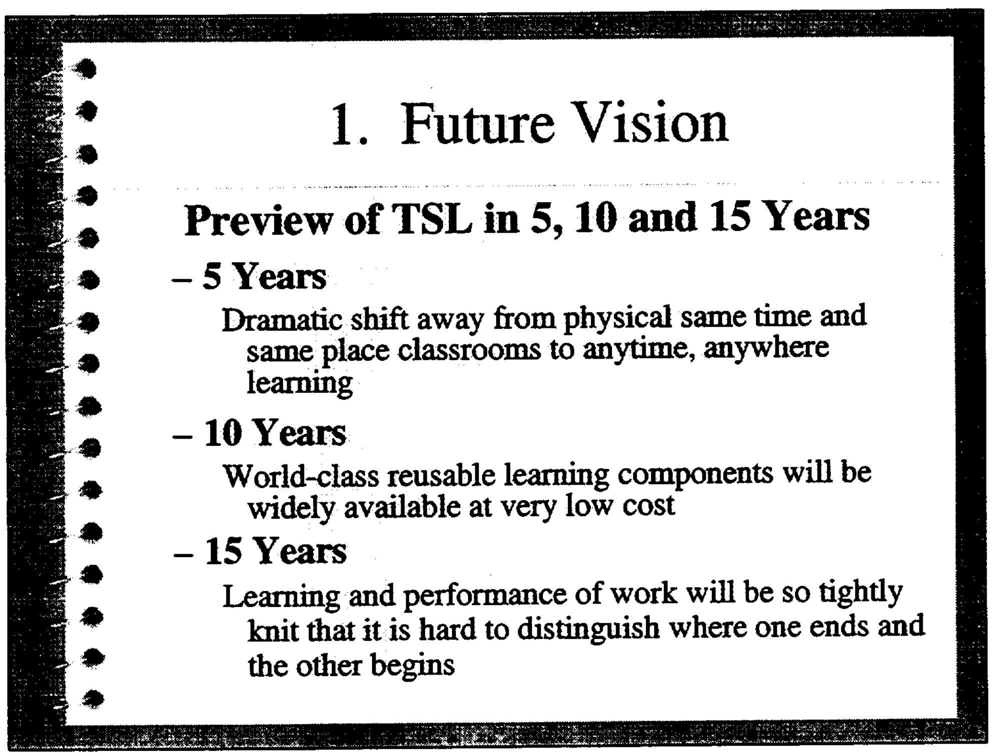
Five TSL Viewpoints
(In Two Dimensions)

Our past and present activities should make the most sense when viewed in the context of our future vision, the top-down drivers and the "as is" realities we are identifying.



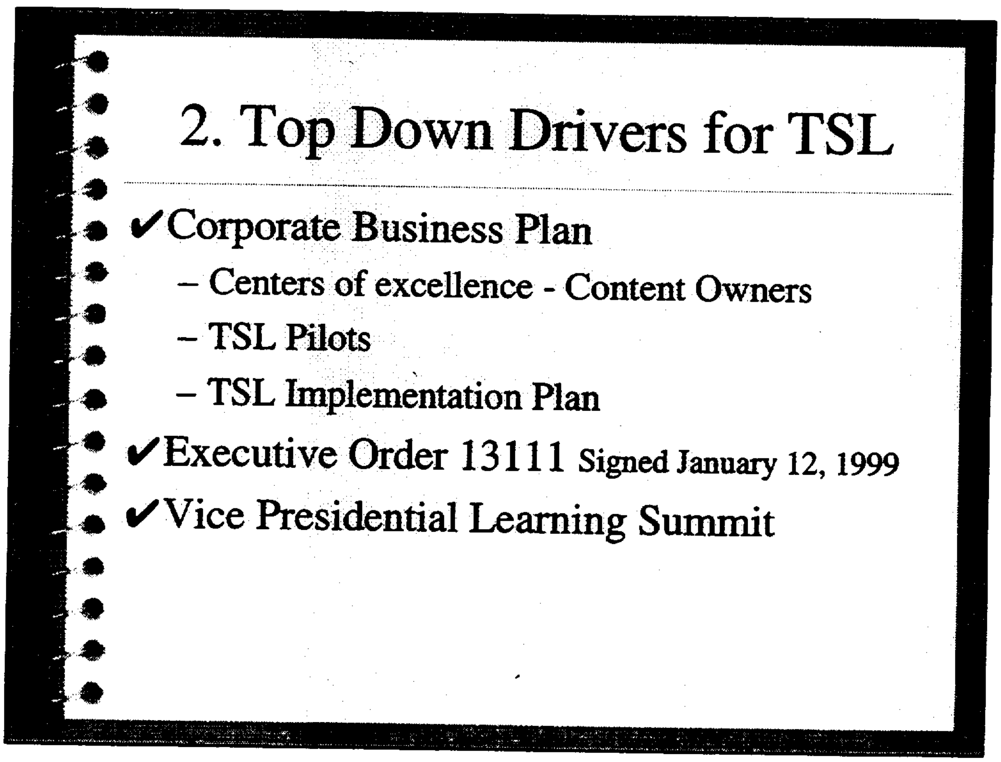
1. Future Vision

We are anxious to help make the future happen rather than at some point down the road be wondering what happened. We have already seen dramatic shifts within DOE in the acceptance of technology for getting work accomplished. We think that acceptance and utilization of TSL will "explode" within the next ten years and become tightly woven into our work--so much so that within 15 years we will notice it no more than we do our telephones and copy machines today.



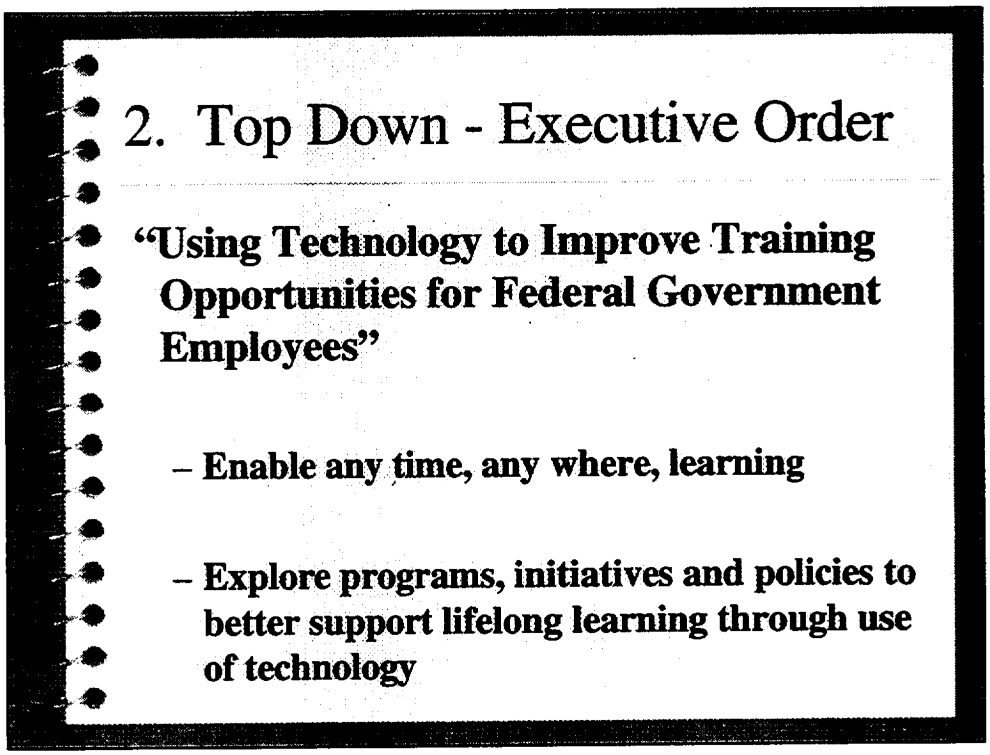
2. Top Down Drivers for TSL

The importance of building a viable corporate TSL program has come from within DOE for a number of years and most recently from the White House. Whereas corporate seemed sufficiently global for us in the recent past, we are adjusting to federal government-wide and nation-wide as a new meaning for global. We are grateful for our early start but still breathless in trying to accommodate the increased scope and excitement generated by Executive Order 13111.



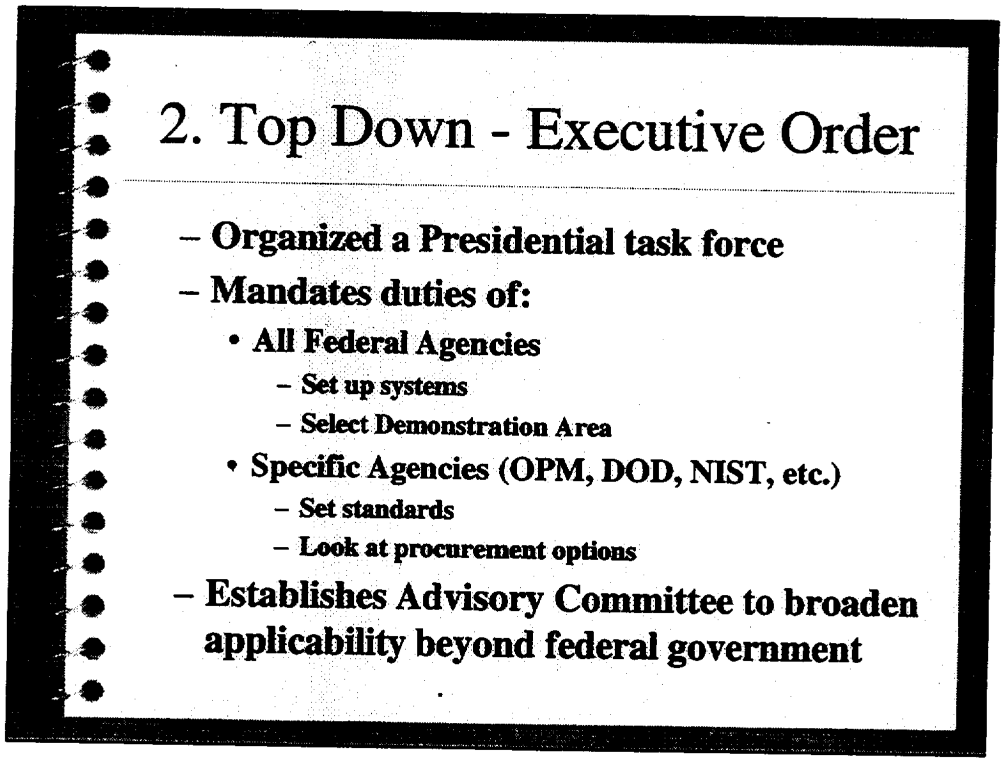
2. Top Down - Executive Order

These goals are 100% consistent with the vision and mission DOE espoused three years ago except that, being primarily contractor-staffed, DOE is looking beyond the federal government employees and seeking to make TSL inclusive of both federal and contractor employees.



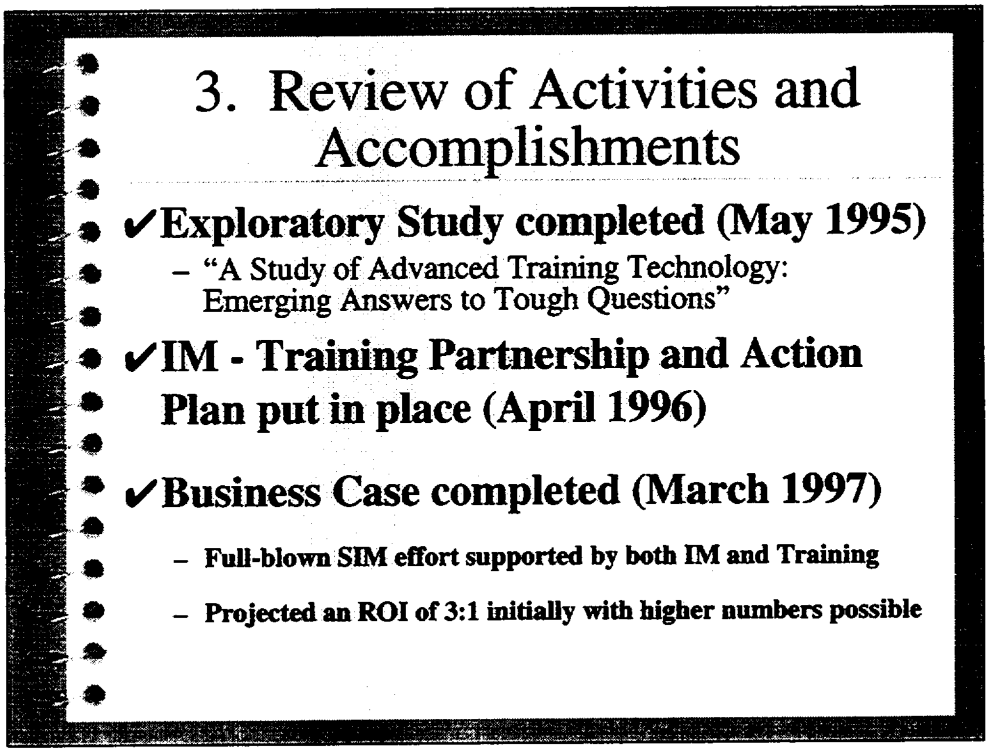
2. Top Down - Executive Order

DOE is a charter member of the task force and intends to be fully involved.



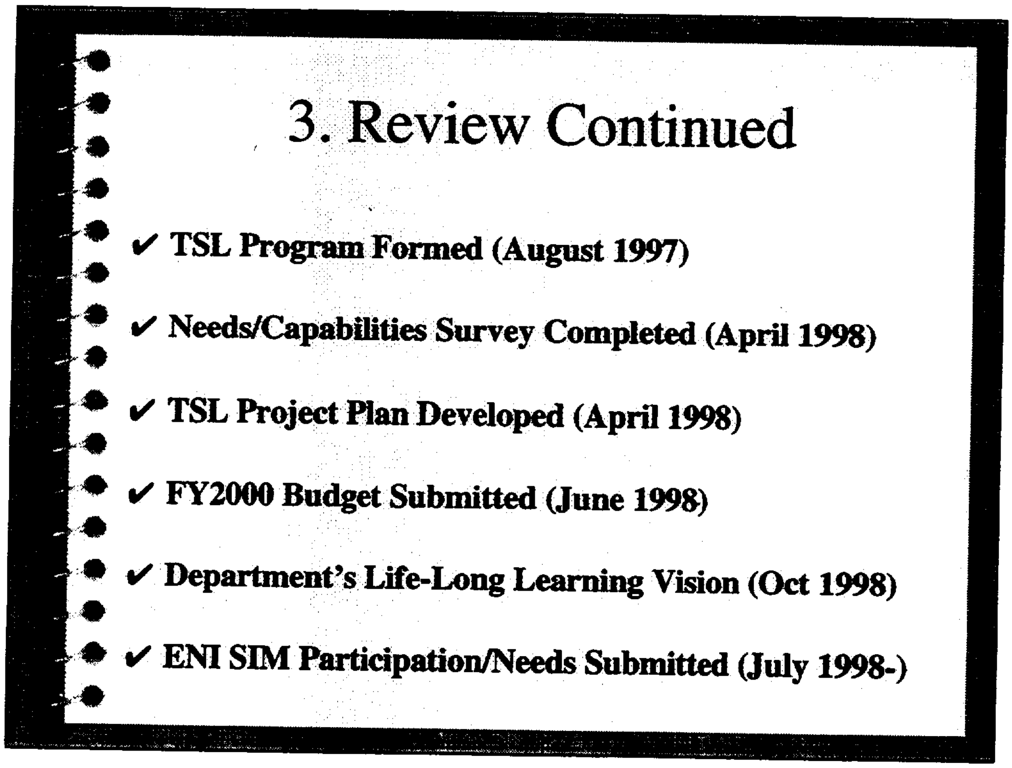
3. Review of Activities and Accomplishments

All of our products are posted on the Clearinghouse for Training, Education and Development (CTED). See cted.inel.gov/cted. We started cautiously with studies and video conferences to explore the potential and develop a firm foundation. A full-blown business case cemented the partnership between training and information management and uncovered an indisputable need and promising potential for a corporate TSL Program.



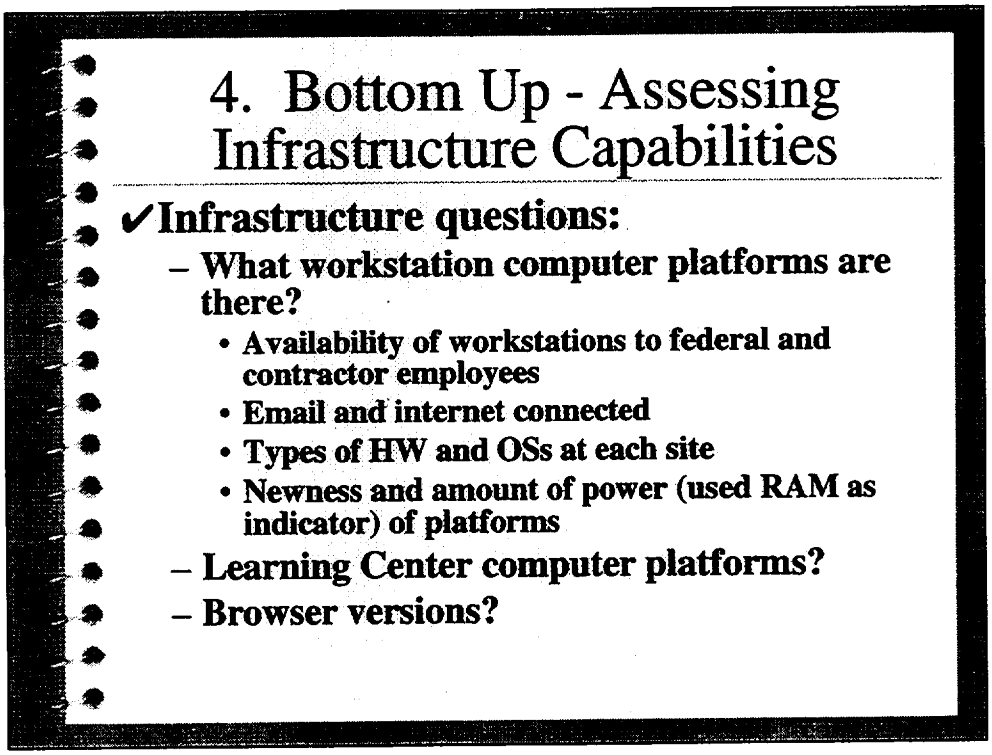
3. Review Continued

Progress has been slow but steady as the work of the Program has been almost entirely done by volunteers working at a distance and occasionally meeting together. Close coordination has been maintained with Information Management. FY2000 funding has been secured and the two existing centers of excellence have matured so that progress should accelerate.



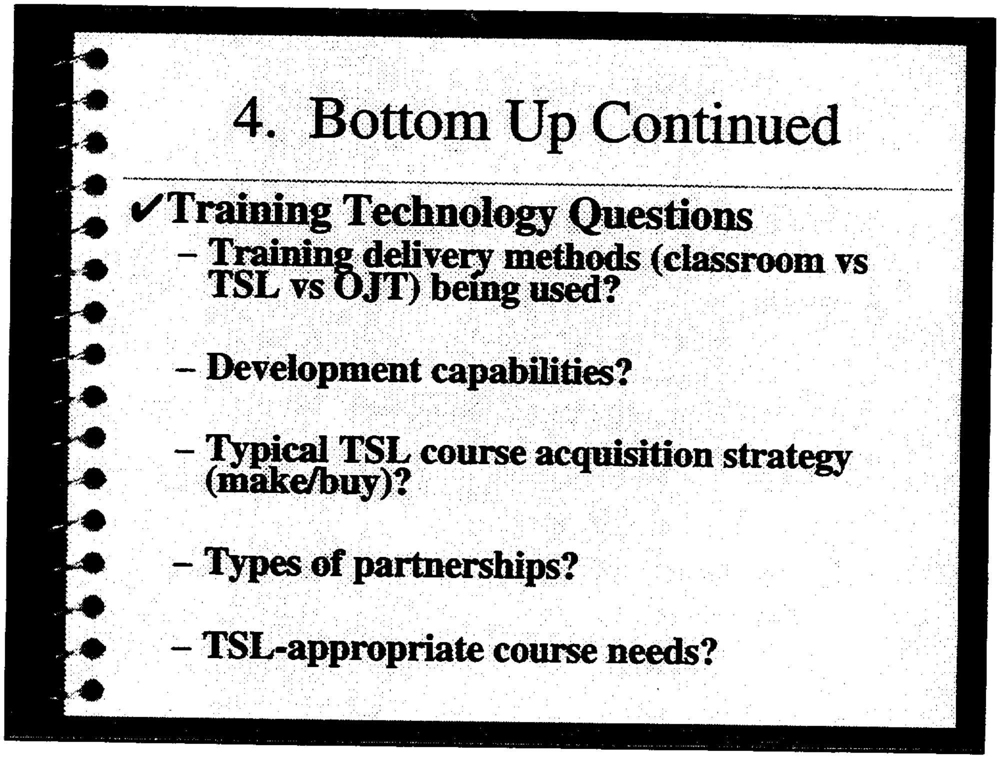
4. Bottom Up - Assessing Infrastructure Capabilities

A detailed online capabilities and needs assessment was completed to enable reality-based planning of next steps.



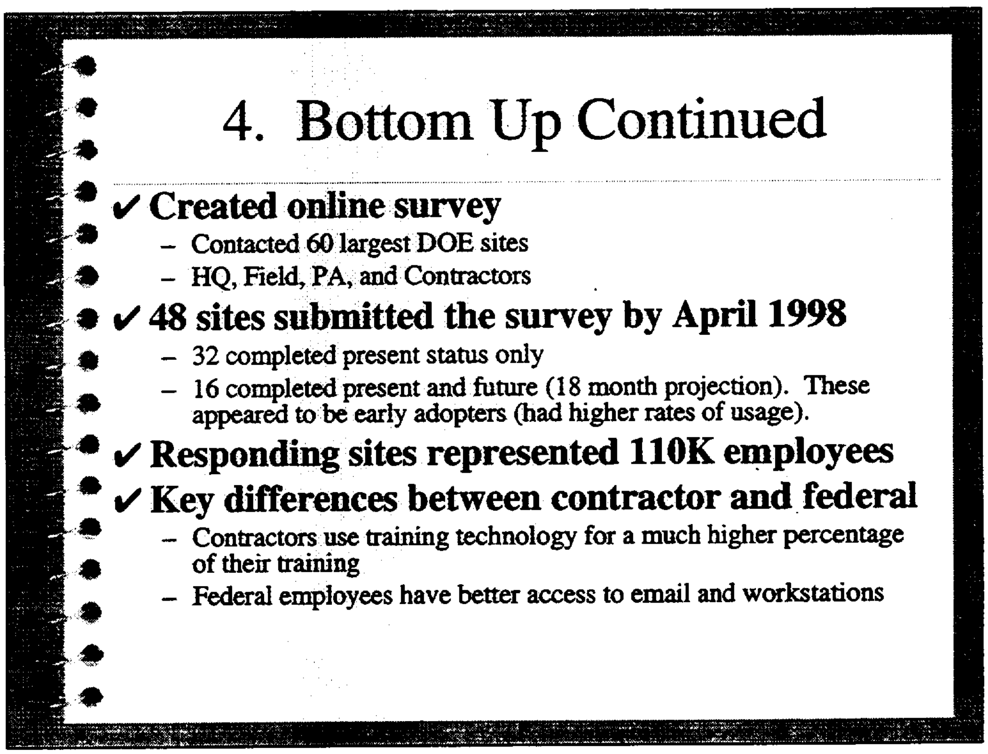
4. Bottom Up Continued

The results of this assessment established a baseline against which progress can be measured.



4. Bottom Up Continued

Patience and arm twisting bore fruit as 75% of the targeted sites submitted current data and 25% also completed 18 month projections. The quantitative data is available in summarized form on CTED (see the Technology-Supported Learning page). The qualitative data has been summarized and can be shared (with appropriate blank-outs of names) if you contact Tanya Luckett directly.



4. Bottom Up - Platform Information

The DOE federal employees are subdivided into Field, Headquarters (HQ), and Power Administration. Our data indicated that the contractor community lags the federal employees in both access to desktop computers and email connectivity. Overall, however, the percentages of those with access is very high.

4. Bottom Up - Platform Info

Data Collected April 1998

	Contractor	Field	HQ	Power Ad.	All
% employees w/ desktop computer	82	97	100	95	84
% employees w/ email	79	97	100	95	82
% computers w/ T1connection	72	70	85	26	69
% computers w/ >=32 MB RAM	42	51	35	26	42
% computers w/ multimedia cap.	56	37	39	14	51

4. Bottom Up - Delivery Methods

In the use of Computer-Based Training (CBT), Web-Based Training (WBT) and Interactive TeleVision (ITV), the contractor community is ahead of the federal training community.

4. Bottom Up - Delivery Methods

Data Collected April 1998 (All Respondents)

	Contractor	Field	HQ	Power Ad.	All
% Classroom	60	70	78	59	61
% CBT	13	7	2	1	12
% WBT	4	2	4	0	4
% ITV (all kinds)	6	5	1	2	6
% On the Job Training/Other	17	16	15	38	17

4. The Early Adopters

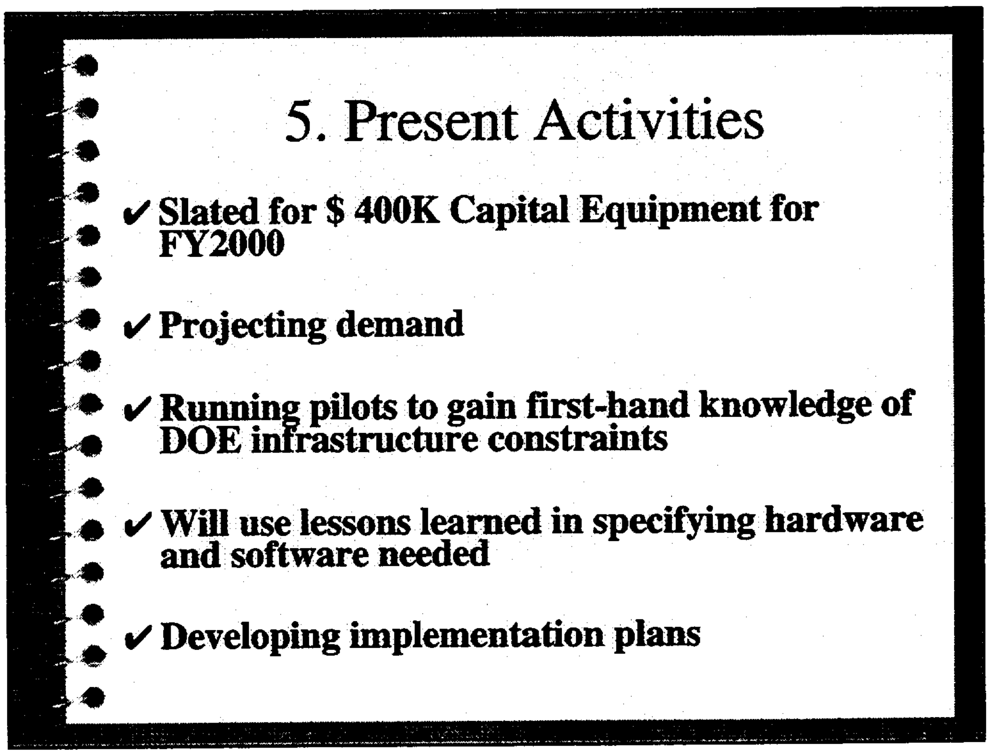
Looking at the 25% who also projected their future situation, we see that there is a group of early adopters within DOE. This group expects that the percentage of training to be delivered via the classroom will drop from 60% to 51% in the near future, with the largest increase being in the web-based training area (from 1% to 10%). As time goes by, these predictions seem to be very accurate.

4. The Early Adopters

Type of Delivery	Contractor	Field	HQ	PA	Comb
% Classroom	60 => 51	84 => 68			61 => 53
% CBT	20 => 23	3 => 9			19 => 22
% WBT	1 => 10	0 => 2			1 => 9
% ITV (all kinds)	2 => 3	5 => 12			3 => 4
% On the Job Training/Other	17 => 13	8 => 9			16 => 12

5. Present Activities

DOE is gearing up to effectively utilize promised FY2000 infrastructure funds. Various planning efforts are underway, with a major source of input to those plans being the results of a series of pilots being run. The first pilot was a synchronous web-based course using COTS software. Other courses are being identified for piloting and roll-out during the next 15 months. Sufficient information management support has been critical to successes to date.



5. Projected Corporate Needs

Here are some of the figures we provided to our internal network infrastructure planners to help them determine what bandwidth they need to make available through their planning efforts. Acquiring the content (free, modified, self-developed, licensed or otherwise) will be at least as great a challenge as the hardware and software issues.

5. Projected Corporate Needs

- ✔ CBT - Very rich media and interactive
- ✔ WBT
  - Asynchronous… varied media richness
  - Synchronous… may make extensive demands
- ✔ ITV - Unclear future. Certainly NNSI will continue.

				Courses per year			
Method	Learners	Hrs/Cours	FY 2000	FY 2001	FY 2002	FY 2003	FY 2004
CBT	3000	2	10	23	27	52	67
WBT	10000	1.5	14	29	45	64	83
	Tot Hrs	CBT	60000	138000	162000	312000	402000
		WBT	210000	435000	675000	960000	1245000

5. Online Learning Pilot

As noted earlier, the Online Learning Pilot has been very successful in meeting our objectives of introducing technology to DOE and in identifying the capabilities and weaknesses of our existing systems.

5. Online Learning Pilot

Using COTS Tool (PlaceWare) to:

Run Synchronous Course over the Web

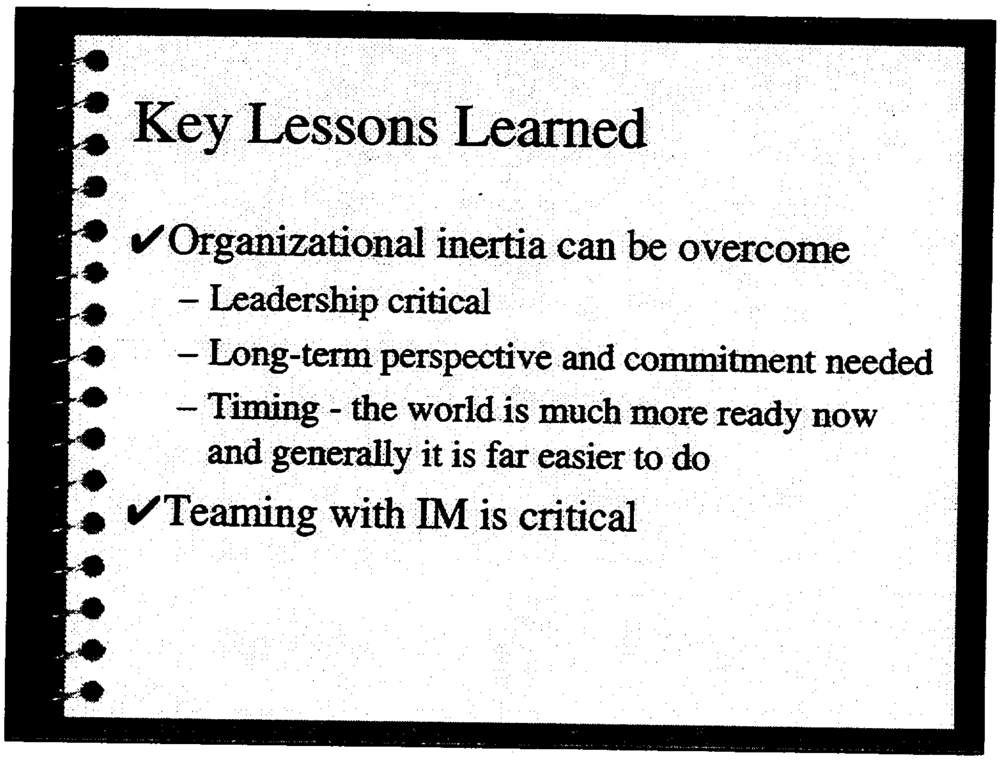
- **Live audio presentations**
- **Shared screens (e.g. Powerpoint created slides)**
- **Shared whiteboard, WWW tours**

"Using Interactivity to Boost Online Learning Effectiveness"

- World Class WBT Developer (David Metcalf)
- Discuss principles
- Show online examples (awarding winning courses)

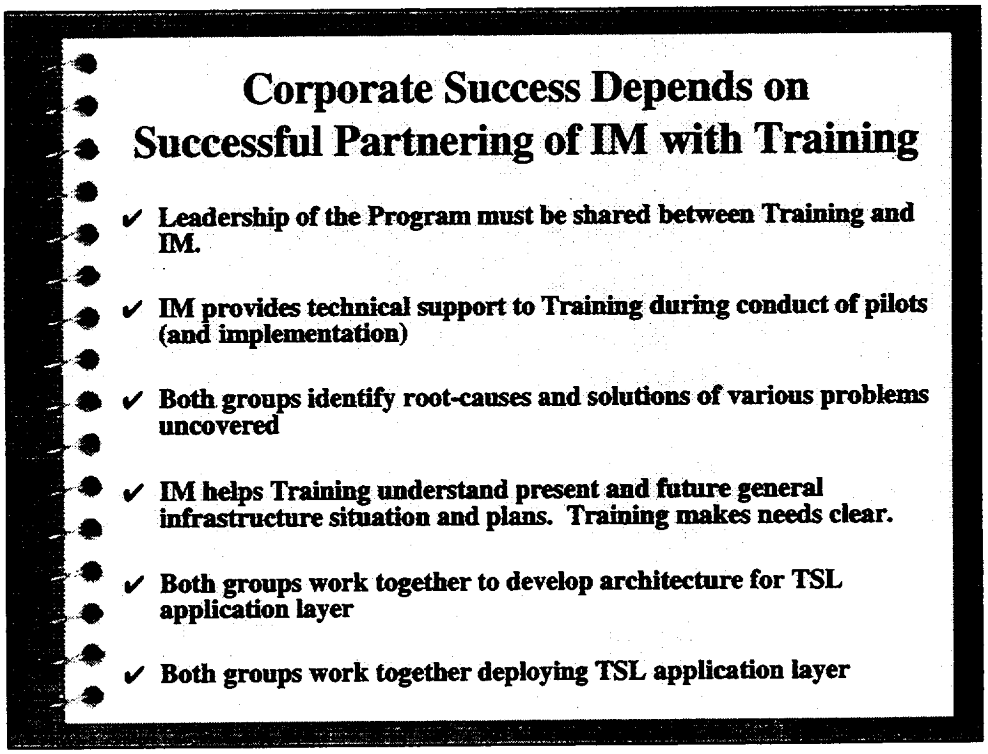
Key Lessons Learned

Our TSL Program is maturing rapidly. As with any and all innovation efforts, we are learning how vital championship and leadership are. Overall, we are finding far more readiness and acceptance that existed three short years ago.



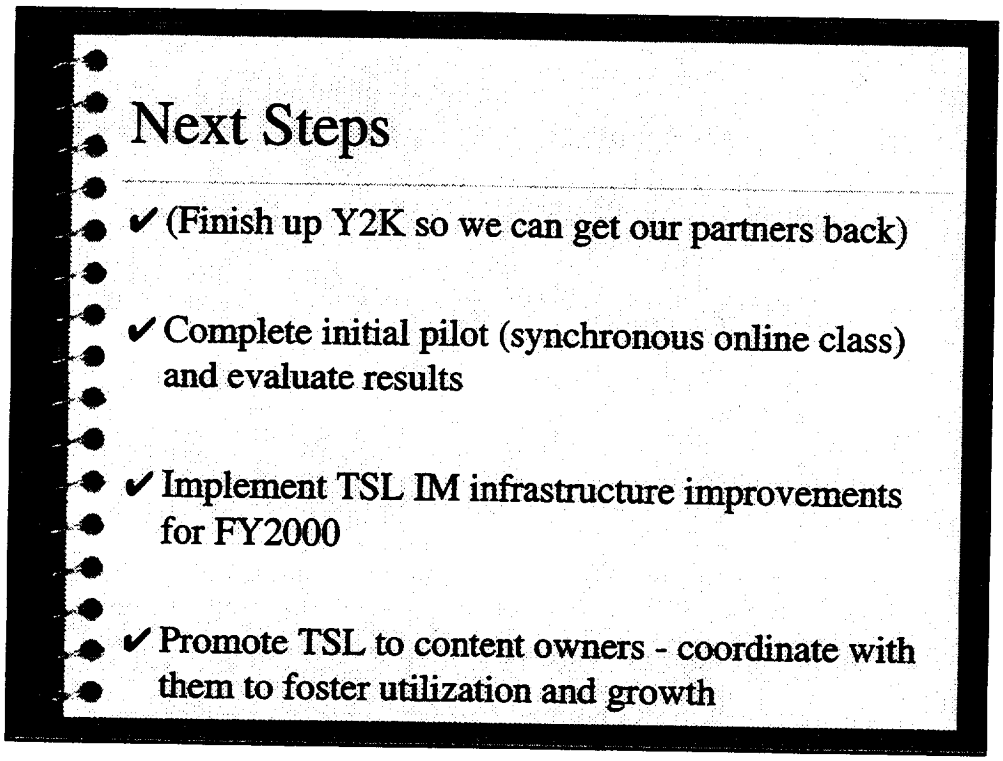
Corporate Success Depends on Successful Partnering of IM with Training

Our internal partnering efforts with Information Management are starting to pay off!



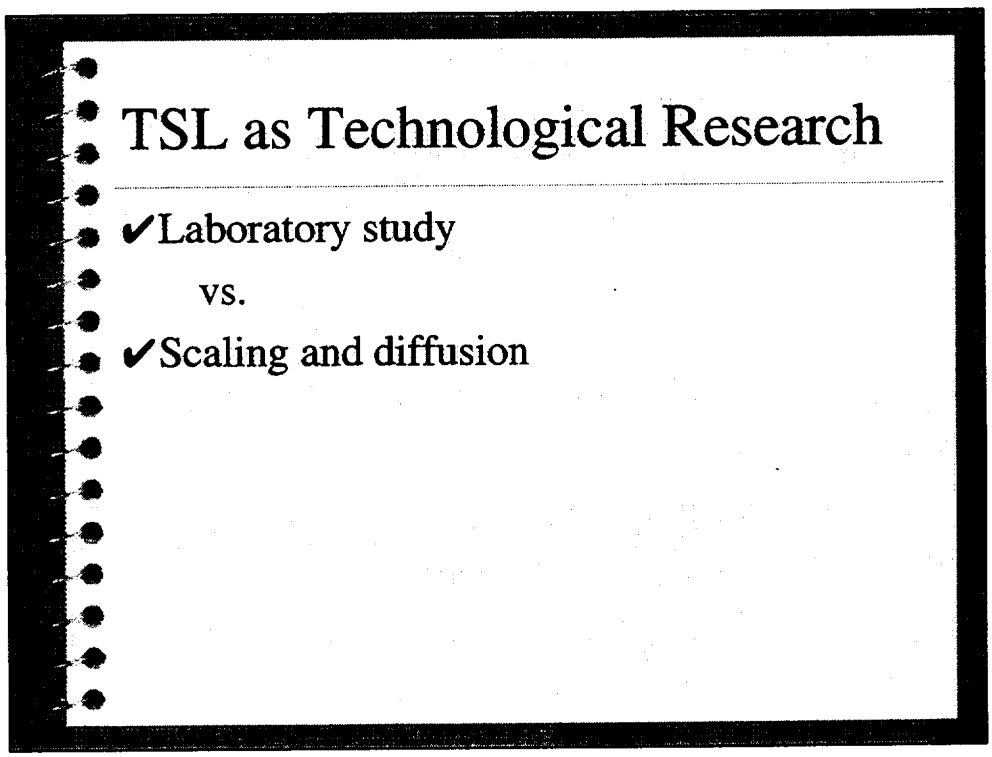
Next Steps

We feel much like orchestra conductors. Our job is not so much to build the instruments or auditorium (infrastructure) or compose the music (learning content), but to help the music happen (melodically) - get everyone to play together from the same sheet of music.



TSL as Technological Research

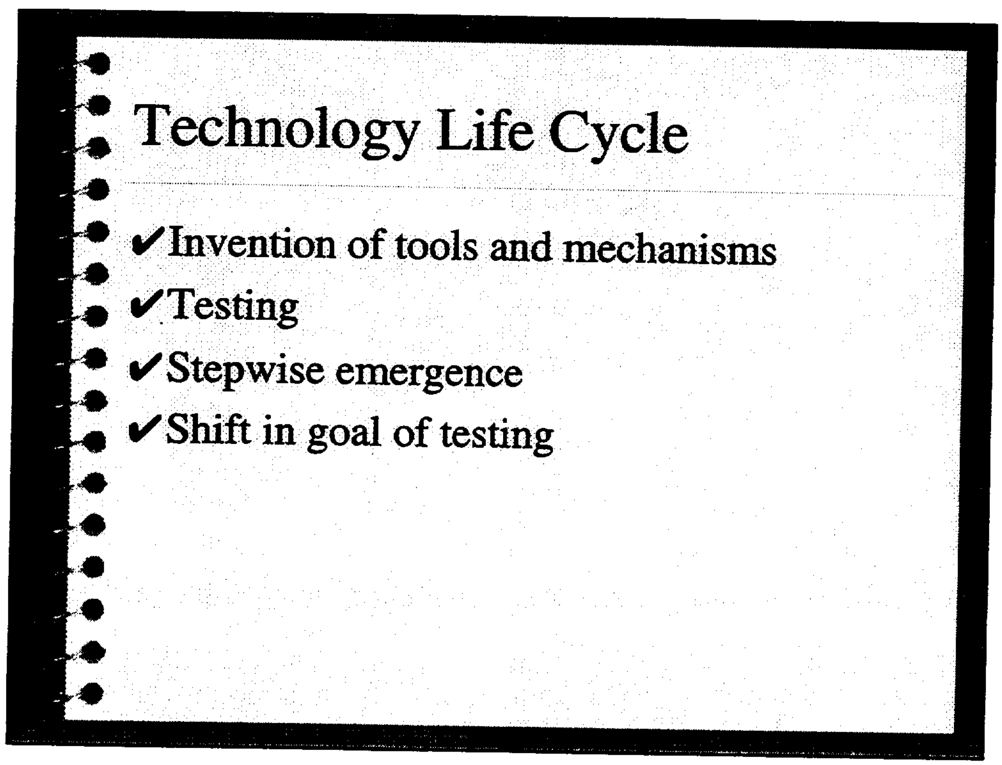
We hope to see the TSL project in its larger perspective because it is not just a project to install new technical devices into organizational offices and classrooms. This workshop is centered on technological research, and that research ranges from laboratory studies to research on the scaling and diffusion of technology in a sustainable way. The work we are presenting is of this latter type.



Technology Life Cycle

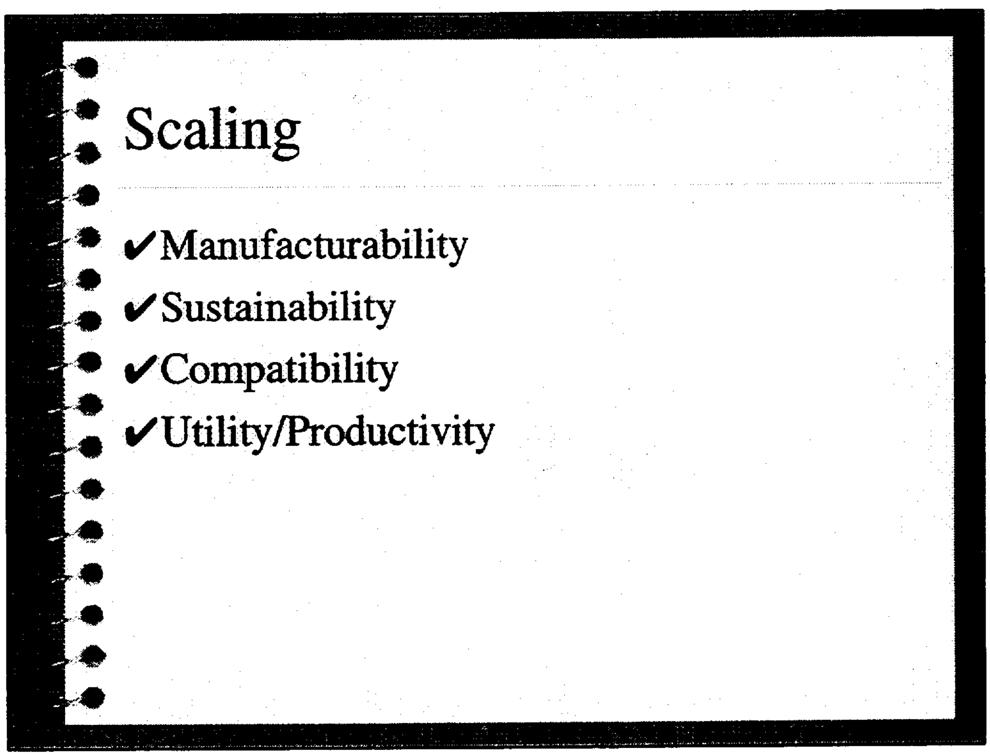
Technologies have a predictable life cycle. In early stages of that cycle, invention of tools and mechanisms is the rule, hopefully followed by rigorous testing. This work takes place in the relatively structured laboratory environment. Thus, in one or more steps, successfully tested technology products gradually emerge into the less-protected settings of everyday usage.

At some point, however, there is a shift in the testing of the technology from proving the technology's effectiveness to proving its viability in the real world. An idea can work in a laboratory but be useless as a mass product because of many factors.



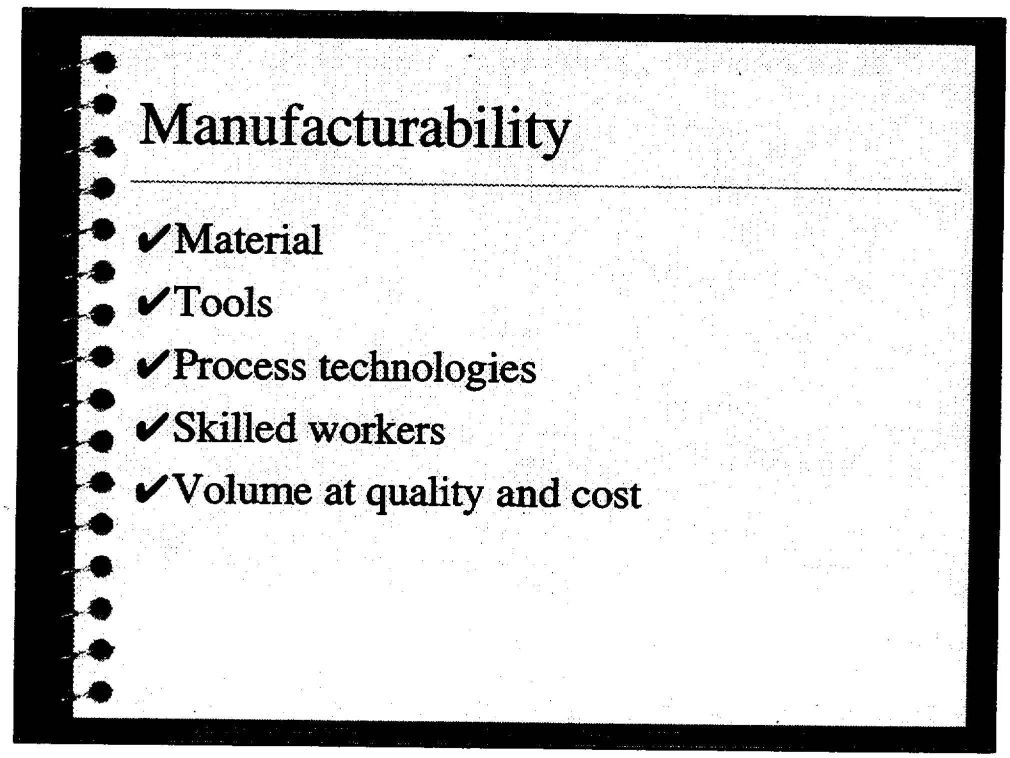
Scaling

The problem of proving a technology's viability involves the problem of scaling. Scaling involves establishing a technology as a reliable everyday tool. The tests of scaling involve proving its manufacturability, its sustainability, its compatibility, and its productivity. Though special financial and logistical scaffolds may be used in early stages of this proving, they must eventually be removed, and the technology must prove it can be sustained in regular use within reasonable constraints. One of the main goals of the TSL effort is to establish that technologies that have proven themselves in controlled settings can be brought into sustained use in standard work settings; that is, that the technologies can be successfully scaled at the organizational level.



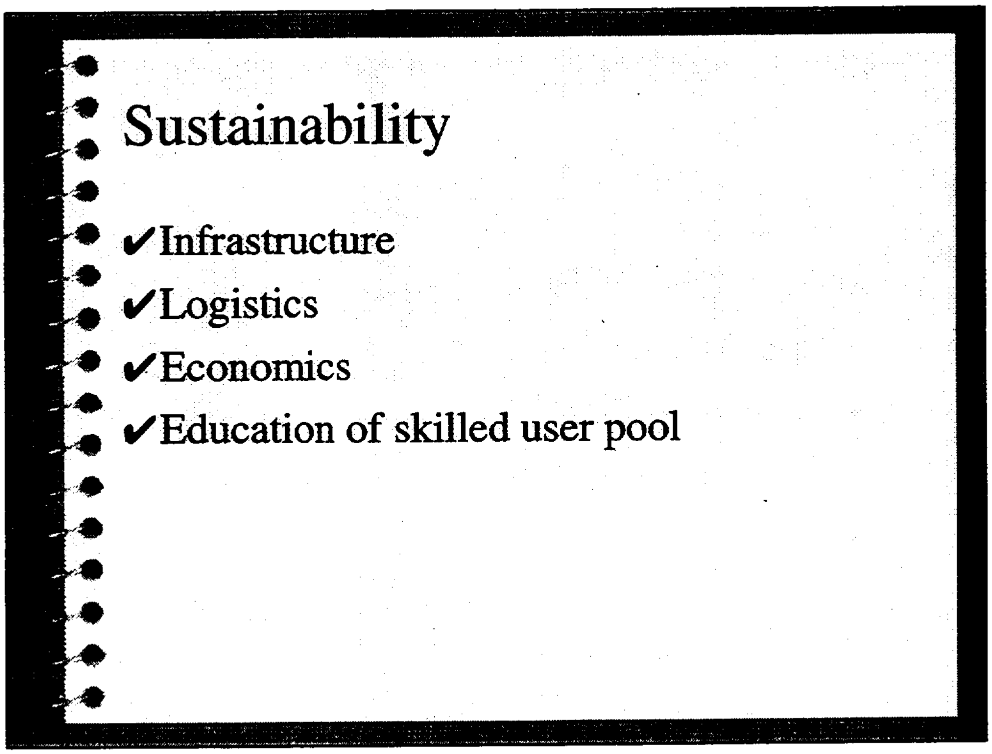
Manufacturability

To prove the manufacturability of a technology, it must be shown that the materials used for manufacture can be obtained and that tools exist that allow manfacturers following specific process technologies to apply an adequate pool of skilled workers to the production of adequate quantities at required levels of quality and cost. A technology that fails any of these standards is not a viable technology. In the technology of instruction, the materials involved in manufacture are not tangible, but the requirement for tools, process technologies, and skilled workers is as critical as if it were a hard product technology.



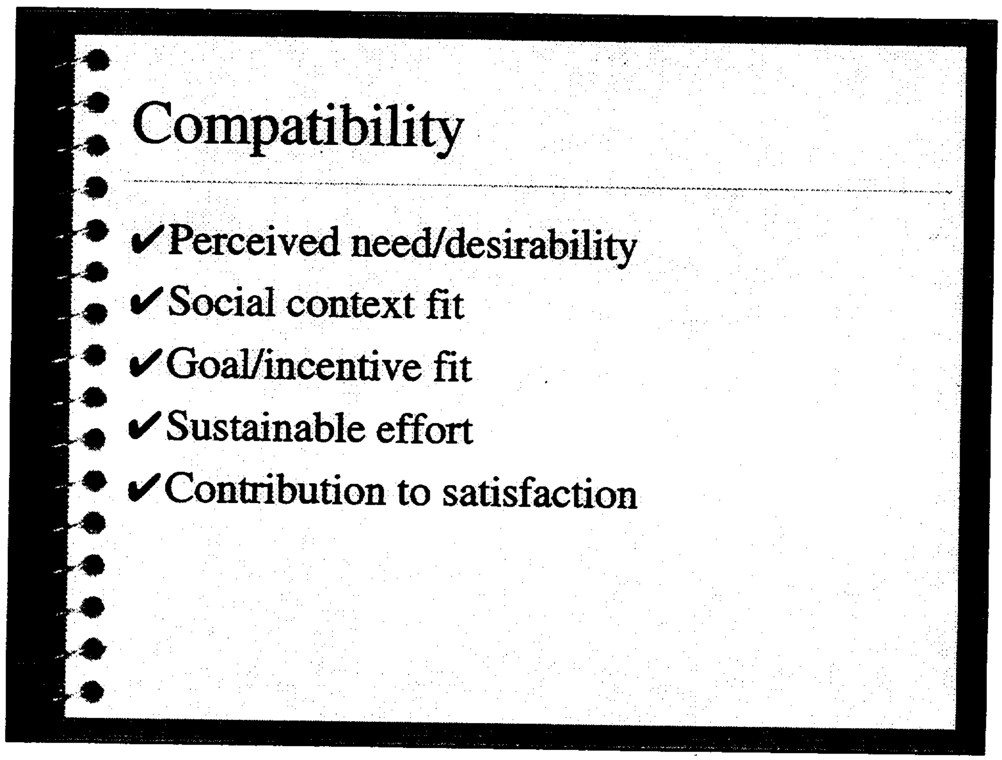
Sustainability

To prove the sustainability of a technology, it must be shown that there is possible an infrastructure that allows the technology to be used and that it is possible to supply and maintain the technology. The economics of using the technology on a larger scale must be shown to be tolerable, and there must be provision for the education of new skilled personnel to use the technology.



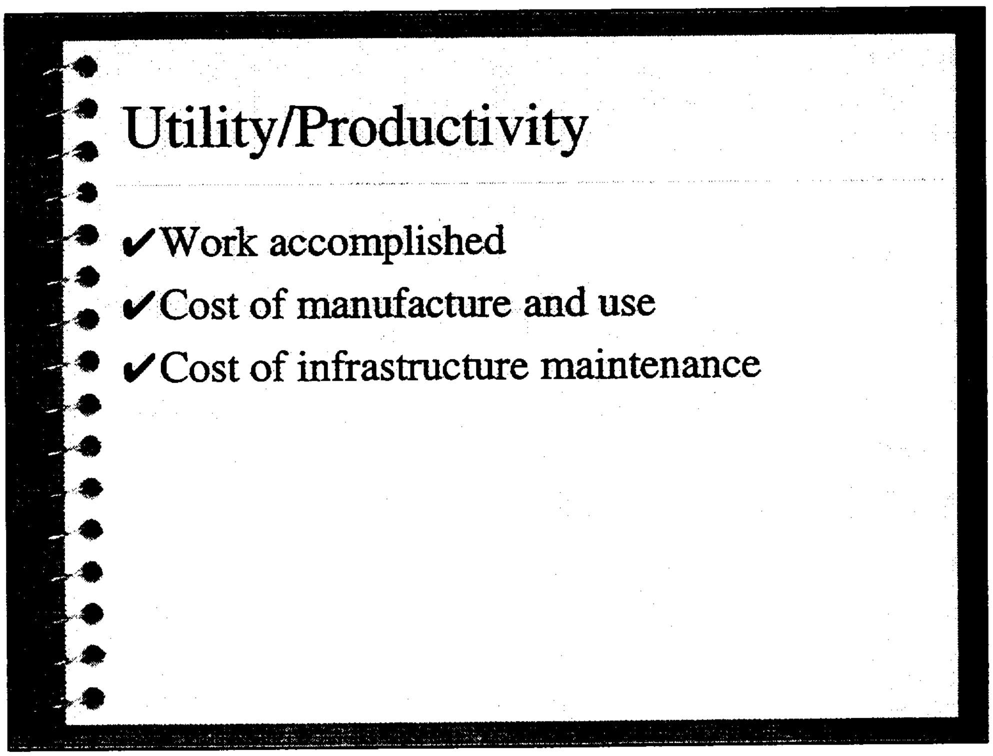
Compatibility

To prove the compatibility of a technology, it must be shown that people can and will use the technology. This involves issues of the perceived need and desirability of the technology in the eyes of the user, the degree to which it fits with social factors of use and the goals and incentives of the user. The technology must show that the effort of using the technology is tolerable day to day and that the technology contributes sufficiently to the satisfactions of the user.



Utility/Productivity

Finally, to prove the utility and productivity of a technology in everyday use, it must be shown that the work accomplished is comparable or better than the replaced technology and that the costs associated with continued use are supportable within an achievable or existing economic context.



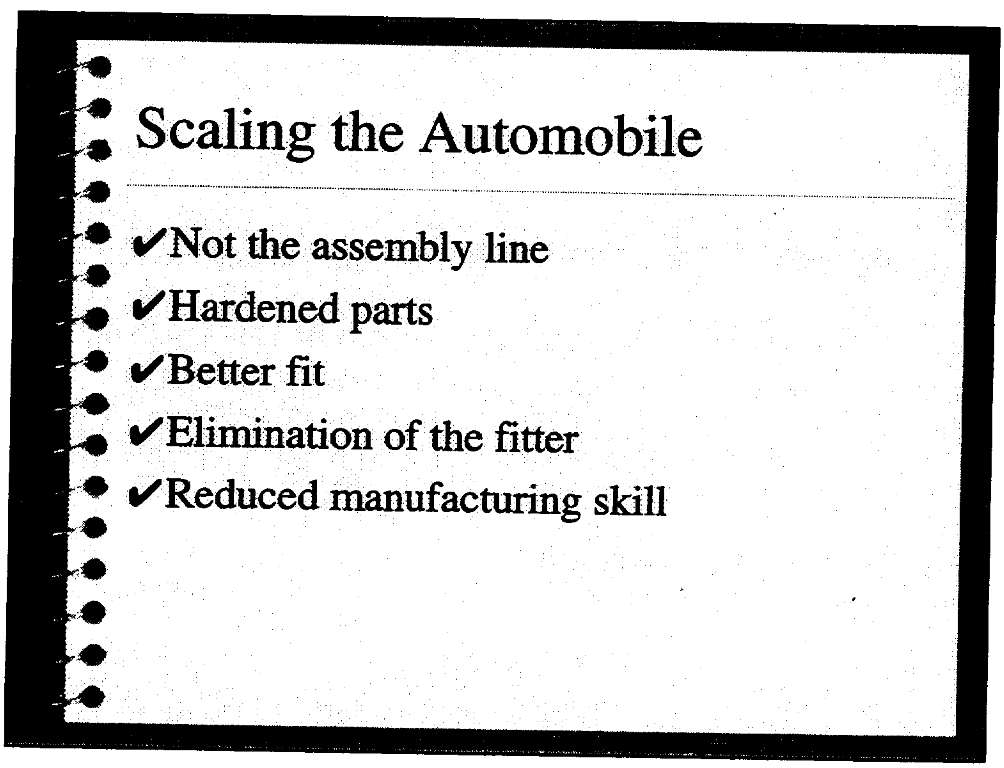
Scaling the Automobile

Henry Ford solved the problem of scaling for the automobile but not in the way most people understand. It was not the assembly line that Ford invented, because the assembly line existed in many industries well before Ford's success. Ford solved the problem of manufacturing precisely-dimensioned parts.

Due to the variability in parts dimensions, Ford and other auto manufacturers found they had to hire large numbers of highly-experienced "fitters". These skilled and expensive workers would use metal files to "fit" individual parts to a specific car, making each car unique and parts not truly interchangeable.

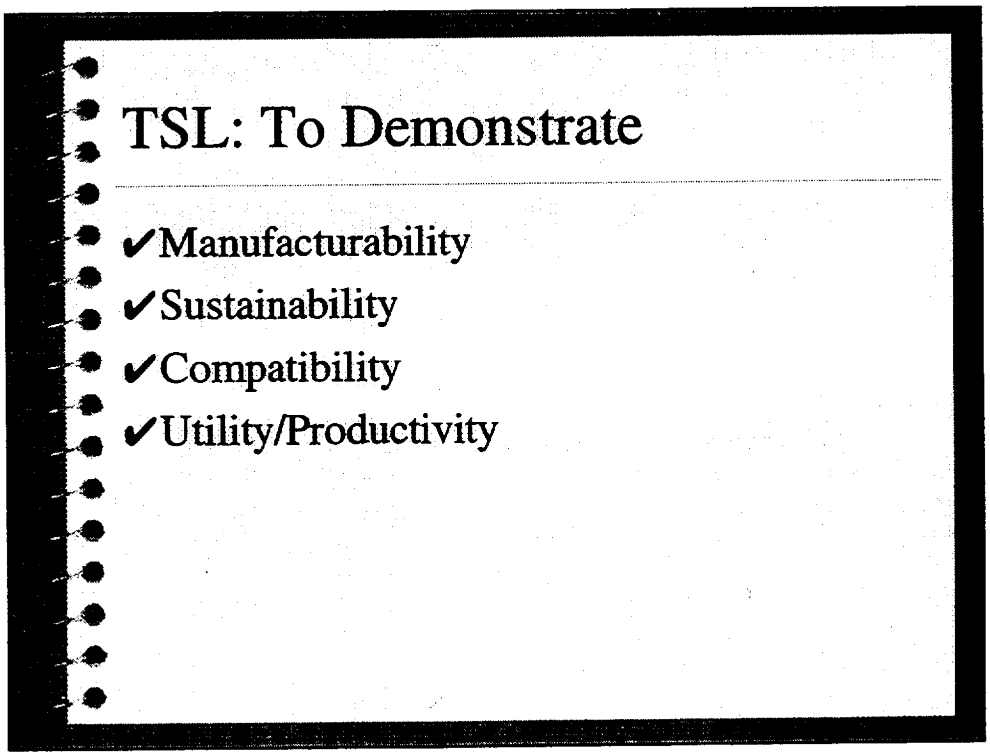
Ford's accomplishment was a method of parts manufacturing (using pre-hardened steel) that allowed parts to be truly interchangeable, eliminating the need for the fitter. This development allowed less-skilled workers to build autos while improving the quality and durability of autos.

The elimination of the fitter eventually took place in other industries, from steel to steel cans.



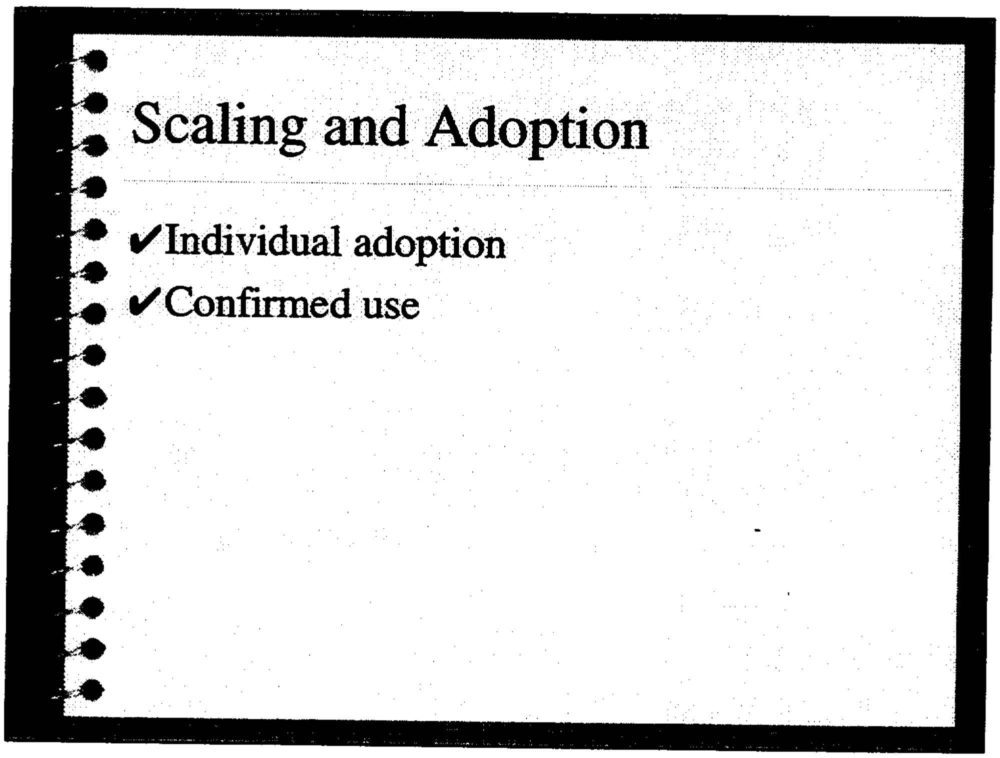
TSL: To Demonstrate

The TSL initiative of DOE is a test of the ability of a large organization to implement new instructional technologies according to a general plan that allows the technology to prove itself in terms of manufacturability, sustainability, compatibility, and utility/productivity. It is a test of scaling technology up to operate reliably within a large organization as a normal part of daily routines rather than as a special, fitted activity.



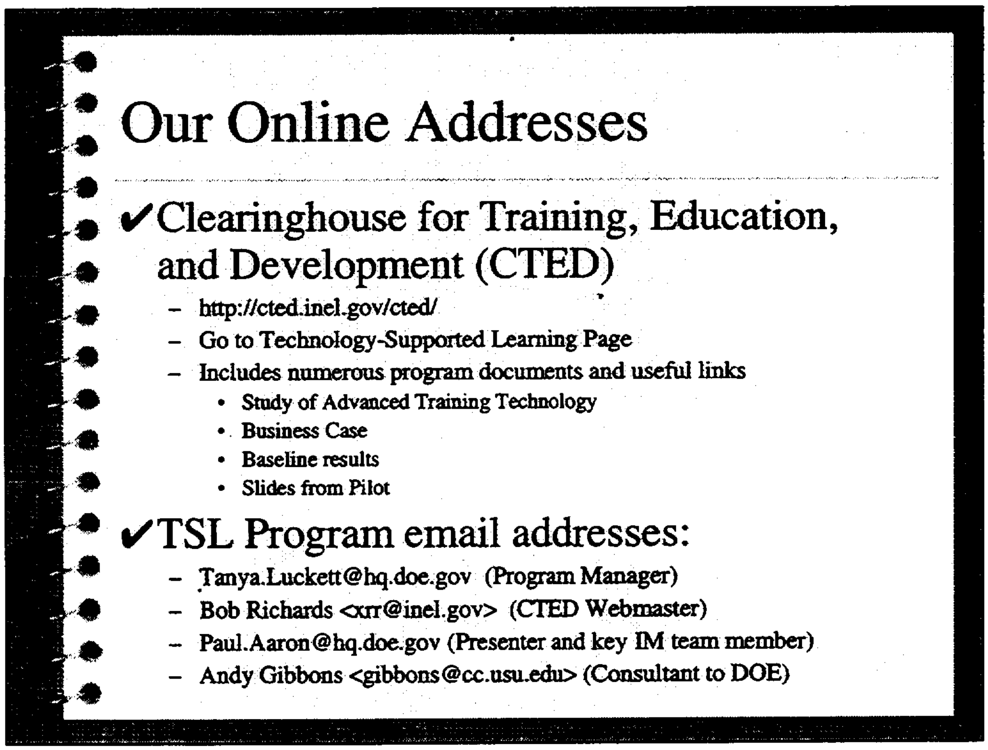
Scaling and Adoption

The major part of the solution sought by the TSL research probably exists on the opposite side of the scaling coin. When seen through the eyes of the individual, scaling of training technology is in large part a matter of individual adoption and confirmed use, according to processes described by Everett Rogers in his book, The Diffusion of Innovations. Every case of technology success is a case of large-scale adoption that is the result of individual adoption and confirmation decisions. Every case of technology failure is simply a case of large-scale adoption of what already exists or of some other alternative or of many uncoordinated alternatives. If a group of users were determined to use a new technology it would be difficult to prevent them. If a group of users were confirmed in the decision to reject, it would result in chronic non-compliance.



Our Online Addesses

We have a long way to go but are excited by the prospects. Send us information about what you are learning that would benefit us. You are welcome to use any/all the content on our page as a starting point for your efforts.



Experience Working with Five Architectures for Learning by Doing

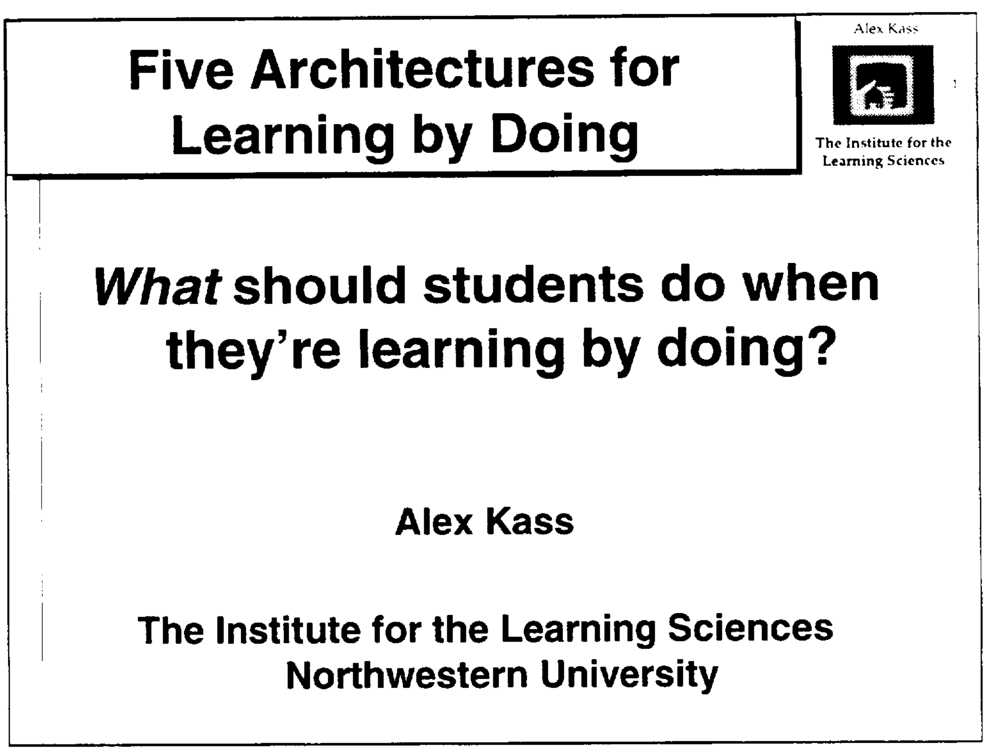
Alex Kass
The Institute for the Learning Sciences
Evanston, IL

Experience Working with Five Architectures for Learning By Doing

Alex Kass
The Institute for the Learning Sciences
Northwestern University
Evanston, IL 60201

There is widespread agreement among educational researchers that people learn best by doing. However, the choice of learning by doing over passive absorption of information is not the end of the educational design discussion - it is really just the beginning. The next logical question is, *what* should learners be doing?

At the Institute for the Learning Sciences we have been looking at that question for several years now, and developing technology infrastructure to support several different kinds of learn-by-doing task types. We call these independent systems *educational architectures*. In this talk, I will discuss five of those architectures very briefly and show examples of how some of them look in action.



Who We Are

Before we get into the heart of the talk, I would like to set some context by describing a unique organization created at Northwestern University to pursue this type of research. The Institute for the Learning Sciences (ILS) is an interdisciplinary center at NU that brings faculty and graduate assistants from several fields together with multimedia production staff to advance the state-of-the-art of learning environments.

About The Institute for the Learning Sciences

Alex Kass

2

The Institute for the Learning Sciences

- An interdisciplinary R+D Center at Northwestern U.
- Dedicated to applying principles of AI, cognitive science, and learning theory to the improvement of education and training
  - Faculty and grad students in Computer Science, Psychology, and Education
  - Programming staff, content staff, graphic artists, video production crew
- Sponsors have included:
  - Andersen Consulting, Ameritech, IBM, US DoD, US EPA, Markle Foundation
- Some clients of spin-off company (Cognitive Arts):
  - IBM, Prudential, HP, Target, US DVA, First Union, G.E.

Underlying Model of Education

Possibly the gravest misconception about education is that it is essentially a matter of transmitting information or providing learners with correct answers. This is the vessel-filling model of education - filling the student's head with knowledge. In fact, education is much more profitably thought of as lighting a fire rather than filling a vessel. *Lectures that answer questions that the student isn't asking generally result in very little learning regardless of how well the material is presented.* The challenge in teaching is creating that burning desire in the student to answer a certain question. Once the question is genuinely raised in the student's mind, the battle has essentially been won. The reason that learning-by-doing works is that goals raise questions when they are difficult to achieve. When you can't immediately do something, you want to know why and you want to understand this aspect of how the world works.

Alex Kass

Education and Training Are About Lighting a Fire, Not Filling a Vessel

The Institute for the Learning Sciences

- **Education and training are NOT equal to "Transmission of Information"**
- **People learn most effectively while in the pursuit of specific, authentic, intrinsic goals.**
- **Presenting answers is not the key to teaching, raising questions is the key to teaching.**
  - Clever presentation of an "answer" is useless if the student/trainee isn't asking the question.
- **The *right kind* of learning-by-doing environment motivates inquiry. Doing raises questions.**

What It Takes

The biggest problem with most traditional learning environments is that they are too passive, and they don't give the learner a chance to learn from his/her mistakes. In addition, they generally try to teach abstract principles rather than focusing first on concrete cases from which the generalizations can grow.

In contrast, effective learning environments revolve around situations in which the learner can pursue meaningful goals and learn from his/her own mistakes with the support of first person stories told by experts at just the time when they are relevant to a goal that the learner is trying to achieve. We use the term GBS to describe such learning environments.

Key Elements of Effective Learning Environments

Alex Kass

The Institute for the Learning Sciences

- **Learning by doing**
  - No lectures, no drill and practice, no listening until *after* trying
- **Learning from failure**
  - A safe environment in which to learn from mistakes
- **Learning from first-person stories told by experts**
  - People remember cases, not general principles
  - People learn from other's mistakes, if provided with the context needed to understand them

A Goal-Based Scenario (GBS) is a course structured according to these principles

What It Looks Like On A Computer

The GBS concept described on the previous page is a general approach, not specific to computer-based learning environments, but which is very compatible with a computer-based approach. When done on a computer, GBSs require three main software components: a practice environment in which the student can operate to pursue a mission; a set of agents that help the student pursue the mission and learn from mistakes; and reference materials which are available to help answer questions once the practice environment has raised them.

The requirements that different learning objectives place on the helpers and the reference materials are fairly constant, so the structure of these components remain the same across various GBSs. However, the structure of a practice environment itself changes because different task structures are called for to teach different things. The structure of a practice environment for flight simulation is very different from one for diagnosing a disease.

Alex Kass

The Institute for the Learning Sciences

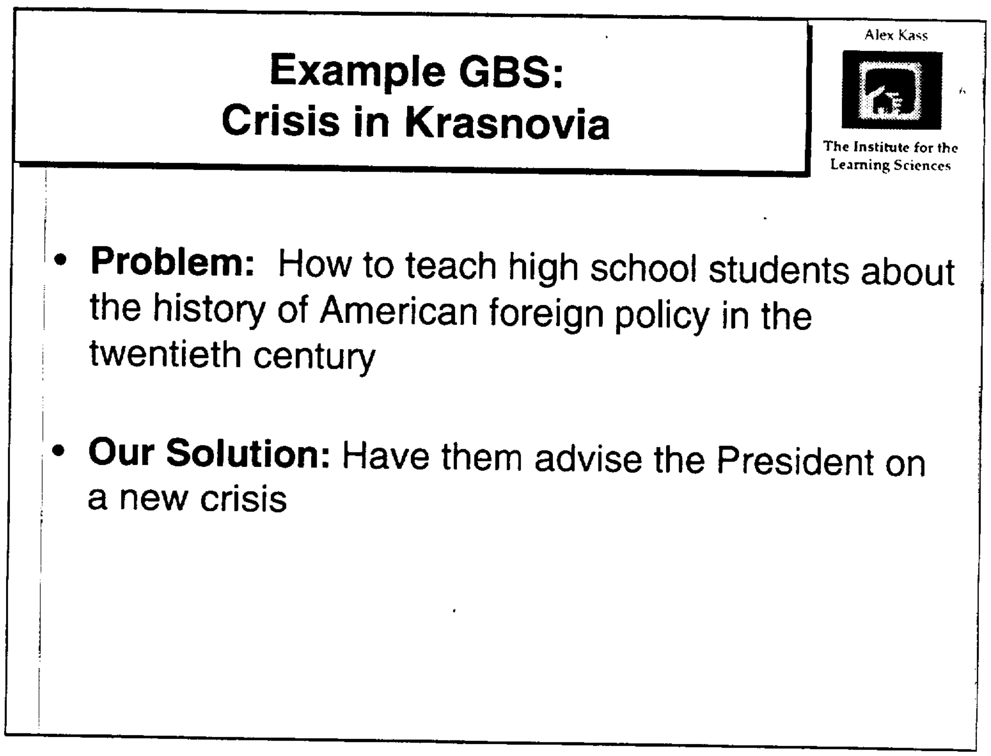
Three Main Layers of Software-Based GBSs

- **Practice environment**
  - Constructions sets, editors, simulations
- **Helpers**
  - Explanation, coaching, case presentation, Socratic tutoring
- **Hypermedia reference materials, organized to support task**
  - Organized according to cognitive principles of conversation and memory
  - Nodes in network designed to answer questions raised by task, *and* to raise new questions, which are answered by other nodes

Learning About History by Applying the Lessons to a New Situation

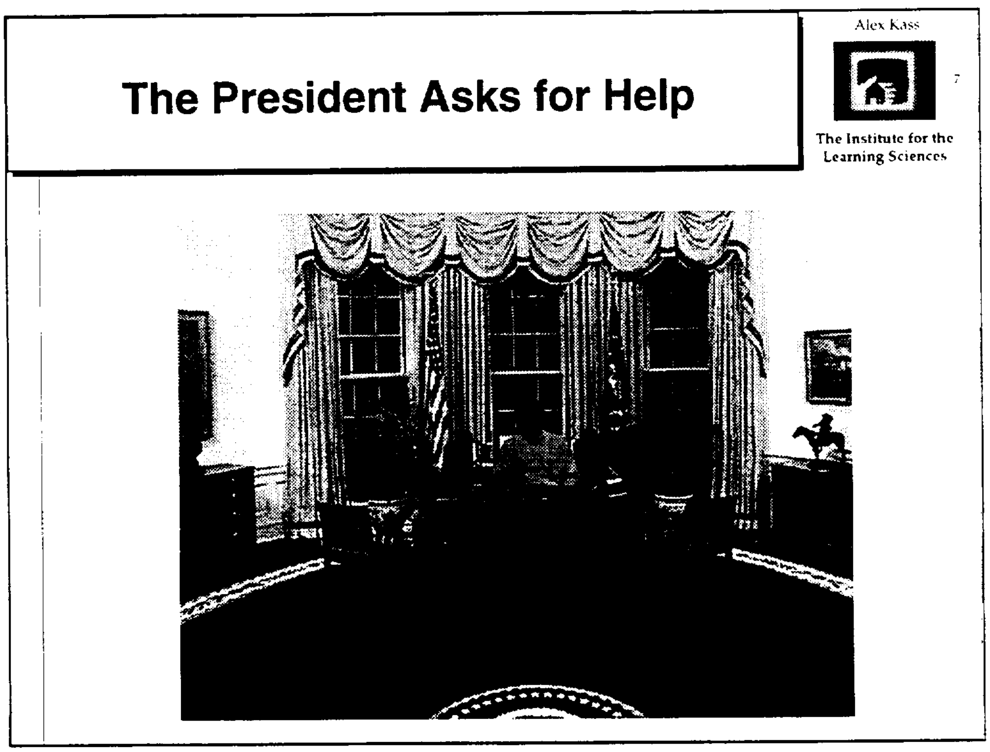
History is typically taught chronologically. For example, a high school course on American Foreign Policy in the 20th Century might start in 1900, and march through the various conflicts and alliances that the U.S. has gone through. This does not work. Students retain little and they wind up thinking of history as boring and pointless.

When we were asked to create a program to teach about this subject we built something that starts in the present day - with the President calling you to get your advice on how to handle a (fictional) crisis. To back your advice you must use evidence drawn from the history of what has and hasn't worked in the past. This is a learn-by-doing approach to learning history.



The President Gives You A Mission

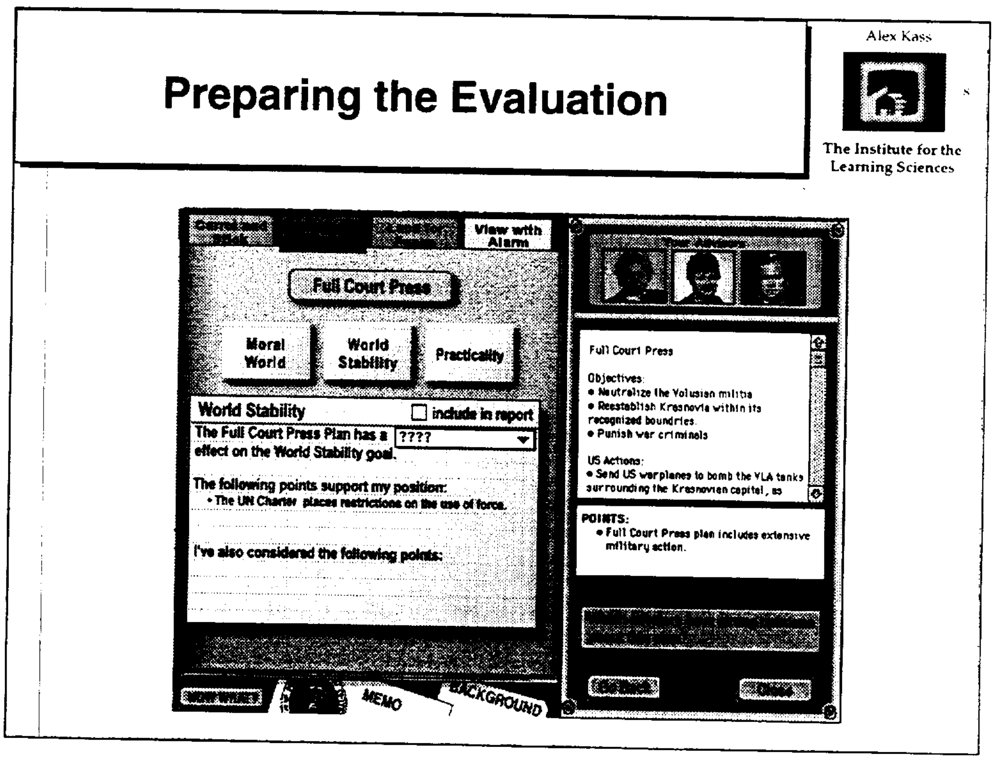
All GBSs start with the learner adopting a mission which is designed to be as engaging as possible. In the case of Crisis in Krasnovia, this takes the form of the (simulated) President of the U.S. bringing you into the (simulated) oval office, and asking for your advice. He tells you that there is an ethnic conflict going on in Krasnovia, and that he needs your advice about what to do. He also tells you what goals he wants to achieve: protect American interests, minimize our own casualties, reduce civilian suffering, and prevent future conflicts.



Resources at Your Disposal

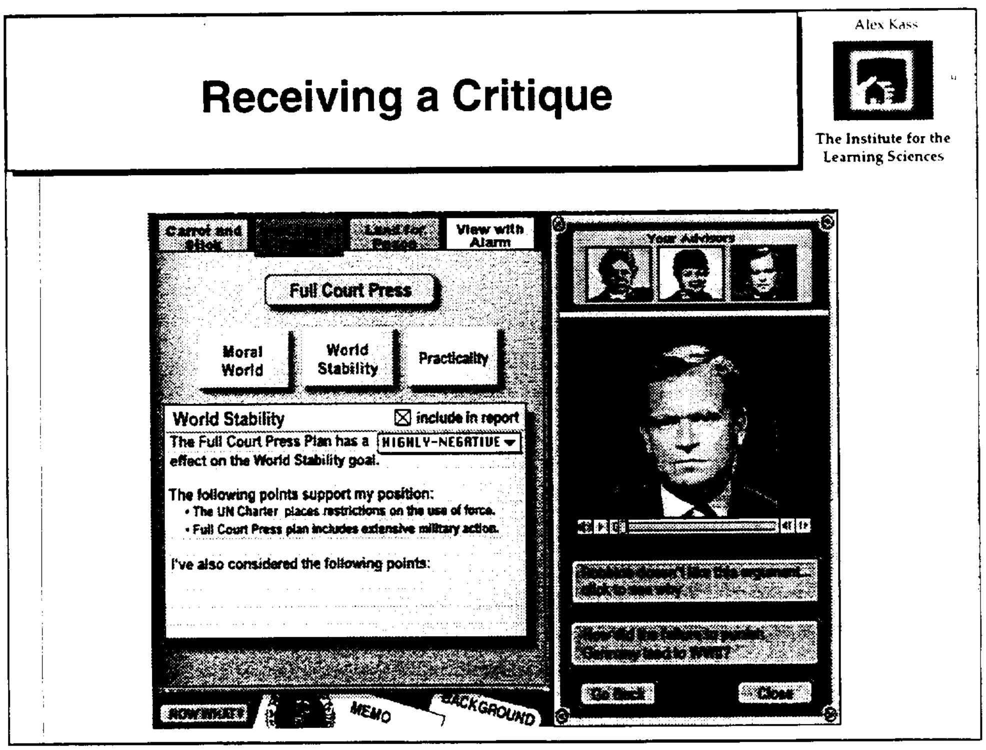
To put together your recommendation, you must evaluate how well each proposed plan will do on each of the President's goals. Simulated advisors (pictured in the upper-left corner) interject, criticizing your reasoning or your chosen course of action. You can turn to real-world experts, captured on video, to collect evidence for your recommendation, and to counter the arguments made by your critics.

When you listen to an expert describe an historical case, for example, the outline of the description appears in itemized text below the video of the expert story. When you find points from that outline that serve your argument, you can drag them into your recommendation.



Advisors Link Between Simulation and the Real World

When you make an argument in Crisis in Krasnovia, you are reasoning in the context of a fictional scenario. The simulated advisors critique you in terms that point toward real scenarios. For example, if you argue that we should stop the "terrorism" in Krasnovia by sending in troops, an advisor would interrupt you with an objection, and suggest that you listen to some experts talk about Vietnam. You would then have the opportunity to hear a (real) history professor (on video) talk about Vietnam, and the limits of our ability to accomplish goals with force. On the other hand, if you argue that we should do nothing, another advisor will interrupt with the opposite objection, and suggest that you listen to a real-world expert describe what happened when we failed to stop Hitler before WWII. Either way, you get critiqued. The point is to learn to defend your position using real-world arguments based on history.



Feedback on Sins of Omission and Sins of Commission

The advisors provide critiques on the basis of several different types of issues: when you make a claim that is not well supported; when you fail to address some issue that an advisor considers important; and when you make claims that are mutually contradictory.

When you indicate that you are done with your recommendation, it is submitted to the President for approval. The President then critiques it on the basis of overall completeness, reacting to broad sins of omission; for example, if you ignore one of the proposed plans altogether rather than arguing against it, or if you give too little attention to one of his important goals.

Each of these claim types is authored by the application designer in advance, using a specially-structured tool which, for example, attaches critiques to plan/goal behaviors.

Type of Critiques

Alex Kass

10

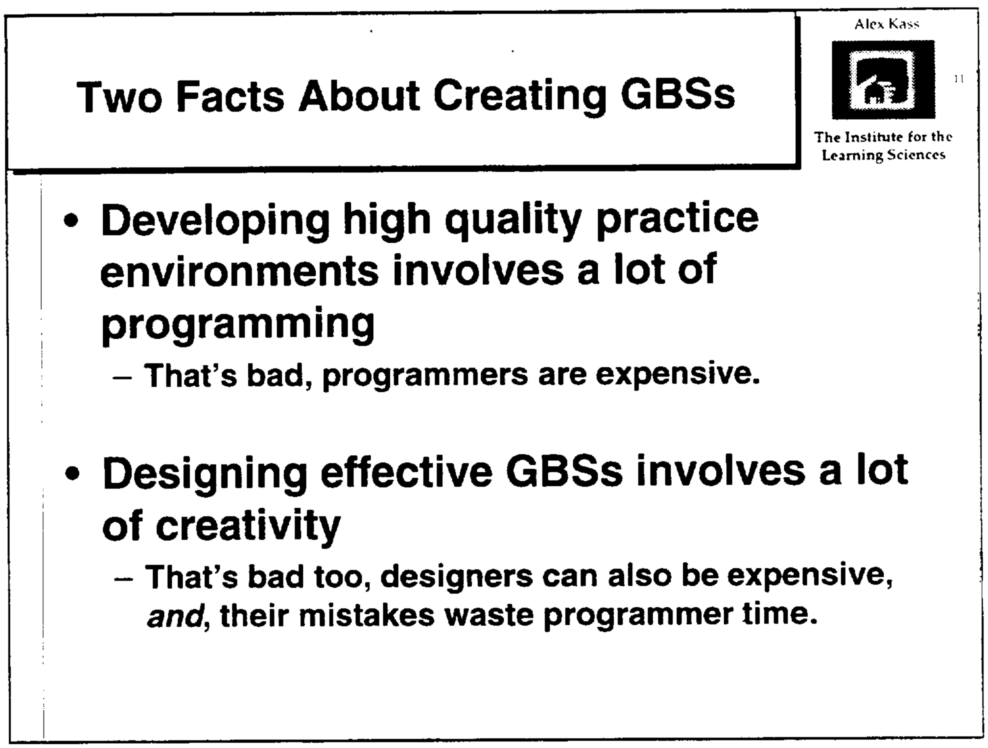
The Institute for the Learning Sciences

- **Critiquing the claimed plan/goal relationship**
  - When student argues for a different relationship than an advisor supports, advisor announces counter claim supported with evidence.
  - Critiquing evidence.
  - Advisors challenge pieces of evidence they don't believe in.
- **Critiquing unstated assumptions**
  - Advisors make an unstated assumption explicit, and then challenge it.
- **Critiquing completeness**
  - President rejects report that doesn't evaluate all plans adequately.
- **Critiquing consistency**
  - President rejects report that is internally inconsistent

Bringing the Cost Down

Crisis in Krasnovia is a successful application, off in many senses, but also highlights the costs associated with doing this kind of work. The designs involve a great deal of creativity, and the implementation requires a great deal of custom production work in terms of video, art and programming. The video, art and programming costs are compounded by the design complexity, since it results in numerous false starts and redone work as the designers struggle to come up with a structure that works. In short, designing an application like Crisis in Krasnovia from scratch is a very costly proposition. To make such applications practical, the cost issue had to be addressed.

In the next few slides I will discuss our approach to addressing this issue.

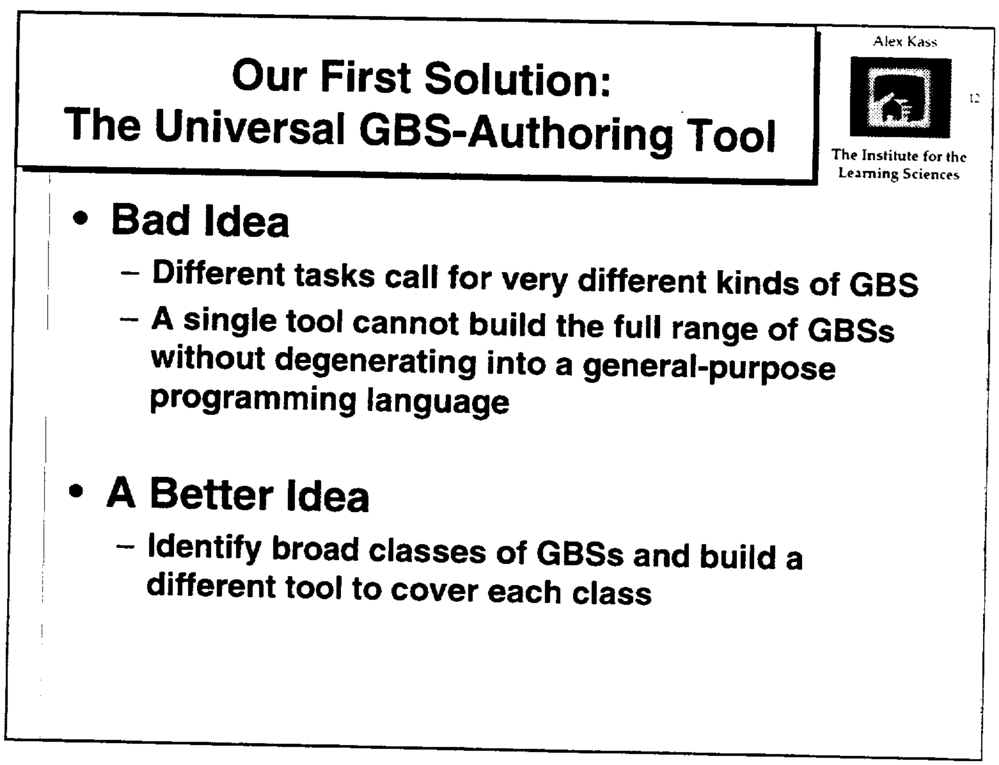


Approaches to Tools

Our first approach to solving the expense problem was to develop the universal GBS construction tool. It seemed like a good idea at the time to develop a single powerful tool that could eliminate the programming cost from all GBSs.

In practice, this tool did not live up to expectation. The essential flaw was that there are just too many different kinds of GBS structures; that is, too many different kinds of doing, to be easily handled by a universal tool. Such a tool becomes something close to a programming language in itself, thus eliminating the benefits that were initially sought by reintroducing the programming effort and by allowing so much design freedom that the design process became as open-ended as it ever was.

An alternate idea was more successful: Identify the various types of GBSs and create a separate tool set to handle each kind. Because there are not a large number of different types, this approach has worked out fairly well.



What Do Other GBSs Look Like?

Before we discuss the general architectures, it will help motivate the idea of architectures to examine one more GBS, drawn from another architecture, to illustrate what the architectures have in common, and what separates them.

Our second example was developed under the sponsorship of the Environmental Protection Agency to teach environmental protection agency professionals how to build consensus in a community.

Alex Kass

An Example of Another Type of GBS Structure

The Institute for the Learning Sciences

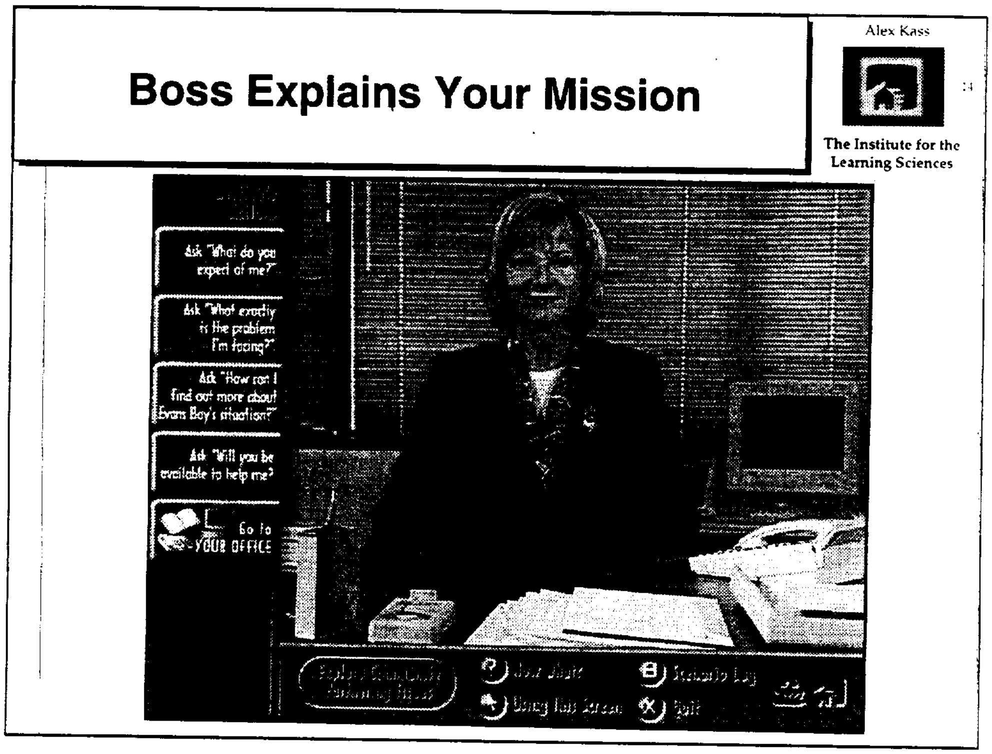
- **Problem: Teaching EPA staff to change the way they conceptualize environmental protection**
  - Replace the "command and control" model with a community-based model in which the agency attempts to develop the long-term capacity of the community to solve its environmental problems

- **Our solution:**
  - Put the staff in the role of a community-relations specialist charged with building consensus for an important environmental problem

Meeting Your Boss

The scenario begins with the "boss" giving the learner a mission, just like in Crisis in Krasnovia. However, the learner's mission and role are rather different. The learner is put in the role of an area coordinator, in charge of a region that has a PCB cleanup problem and is having trouble (with all parties involved) agreeing on how it should be handled.

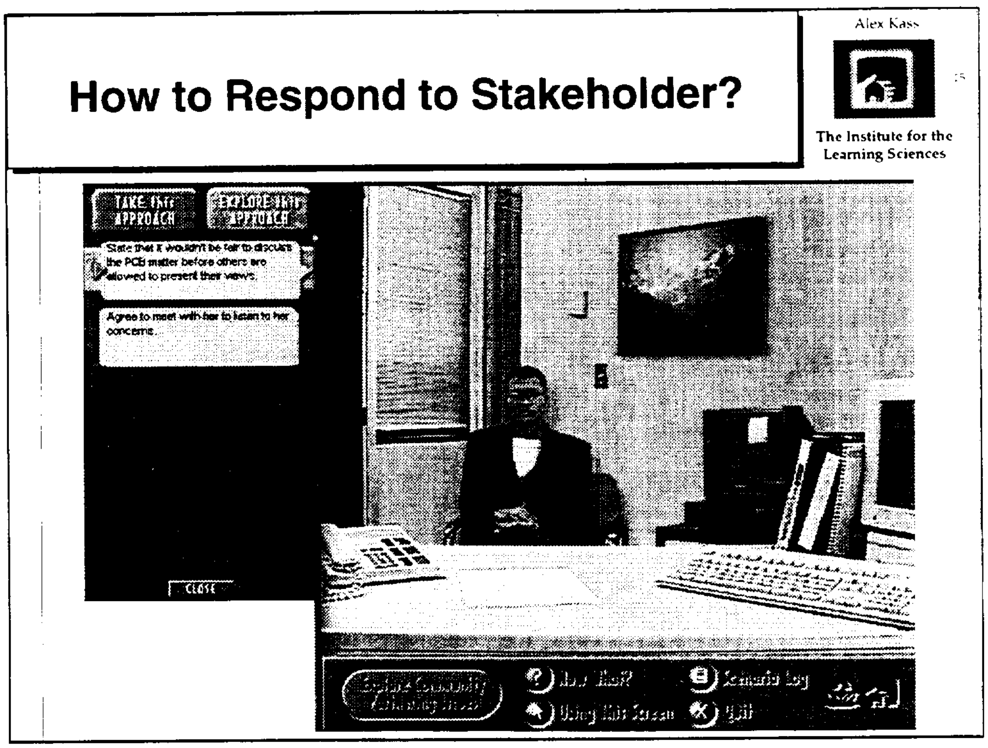
The learner must attend public meetings, meet with stakeholders, and craft a compromise solution that will address the problem with agreement from as much of the community as possible.



Interacting with Simulated Characters

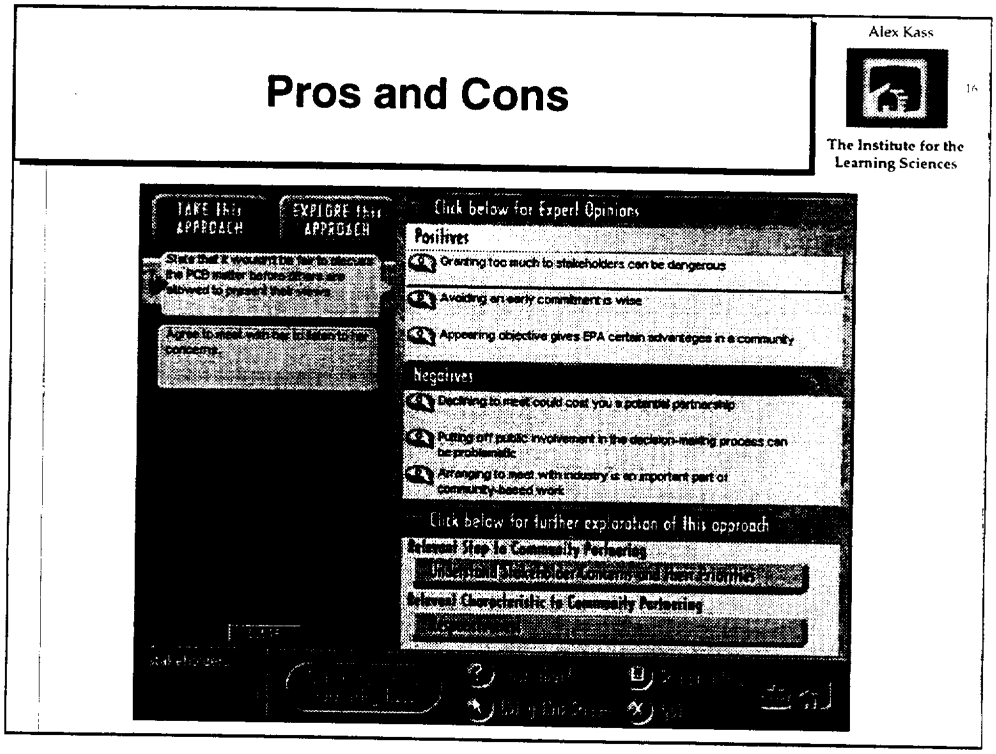
The screen below depicts the scene as the learner sees it from behind his/her desk. In the current scene a representative of one of the polluters has come to request a private meeting.

The learner is confronted with a problem for which there is really no easy or correct answer: grant the meeting and risk showing bias toward the polluter. or deny the request and alienate an important party to the solution. Before making the choice, the learner may choose to explore the ramifications of either option.



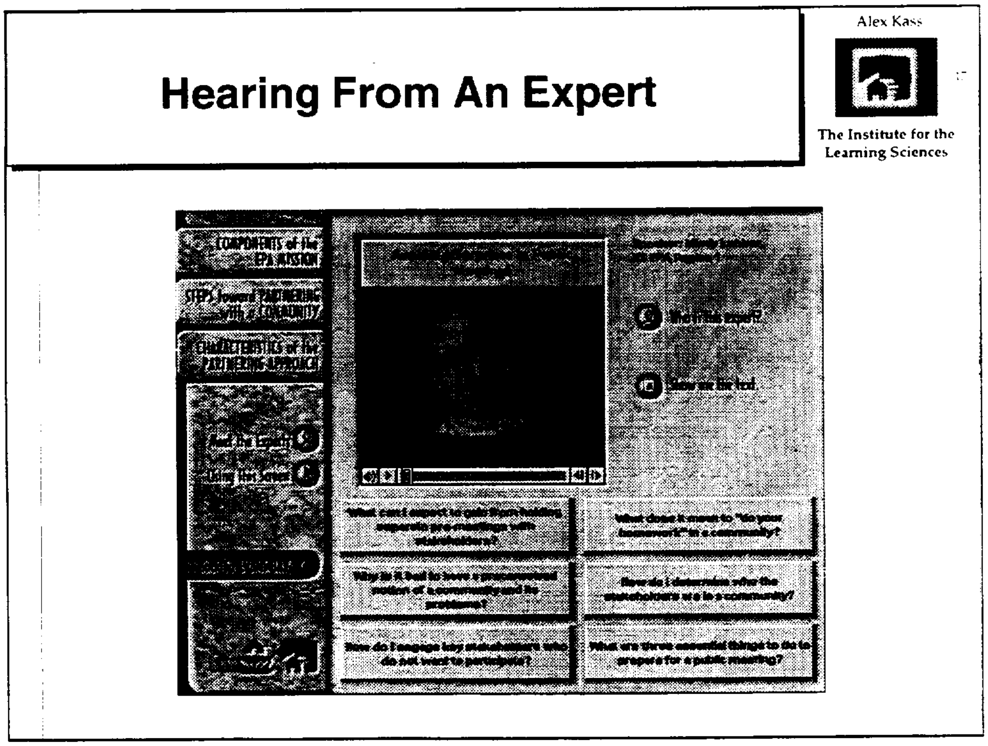
Exploring the Options

If the learner chooses to explore an option, he/she sees a list of *potential* strengths and weaknesses of the approach in question. All of these pros and cons are clickable. Clicking on one brings up an experienced expert who can talk about real-world experience with the option.



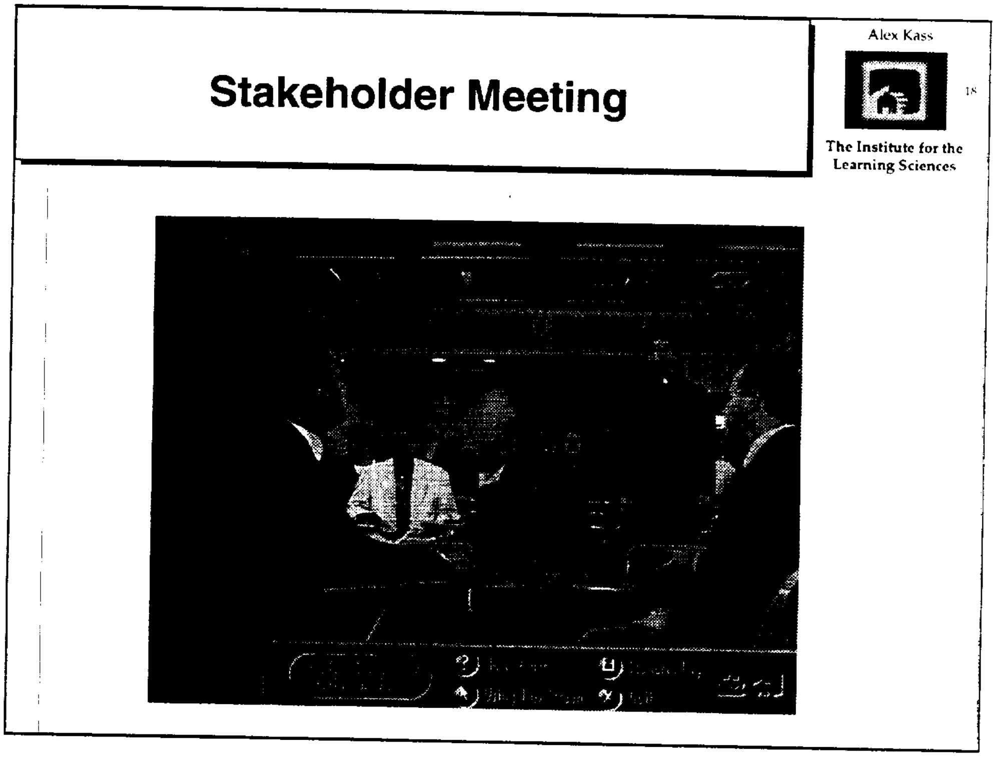
Specific Cases

If the learner clicks on one of the pros or cons related to a potential course of action, he/she hears from someone about the ways they have seen the potential effect occur in real life. Each clip is accompanied by follow-up questions that the learner may choose to ask after hearing the initial clip.



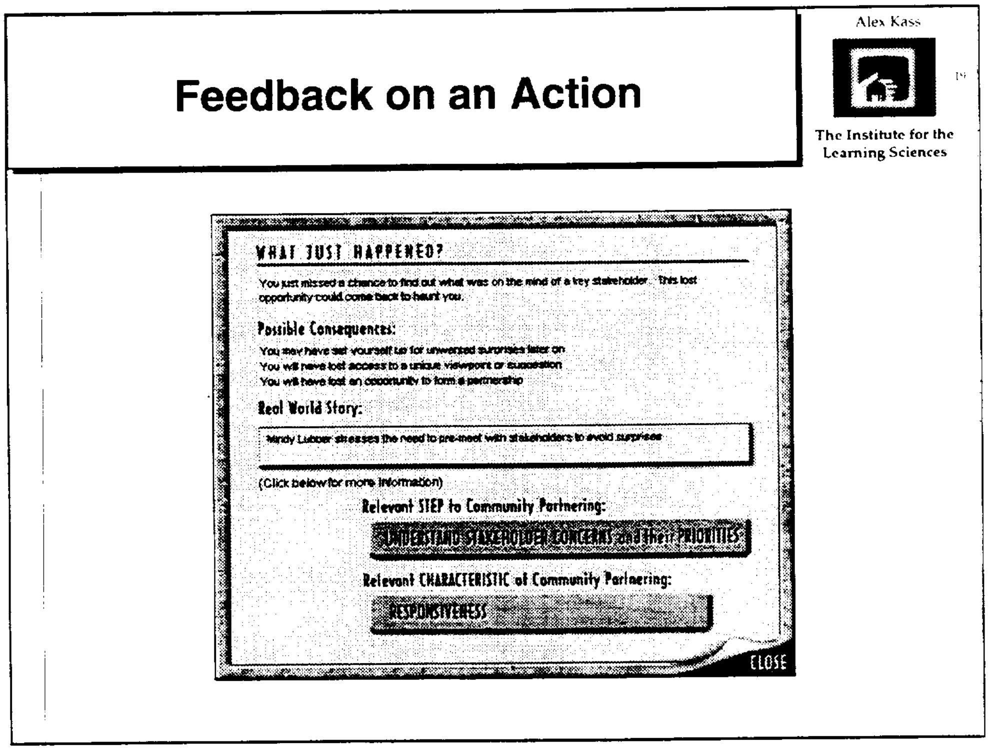
Dealing with Groups

When the learner is ready to attend the public meeting he/she sees a scene like the one below in which many different parties pursue various agendas, some helpful, and some not so helpful. The learner must decide how to run the meeting and how to respond to both expected and unexpected questions and arguments.



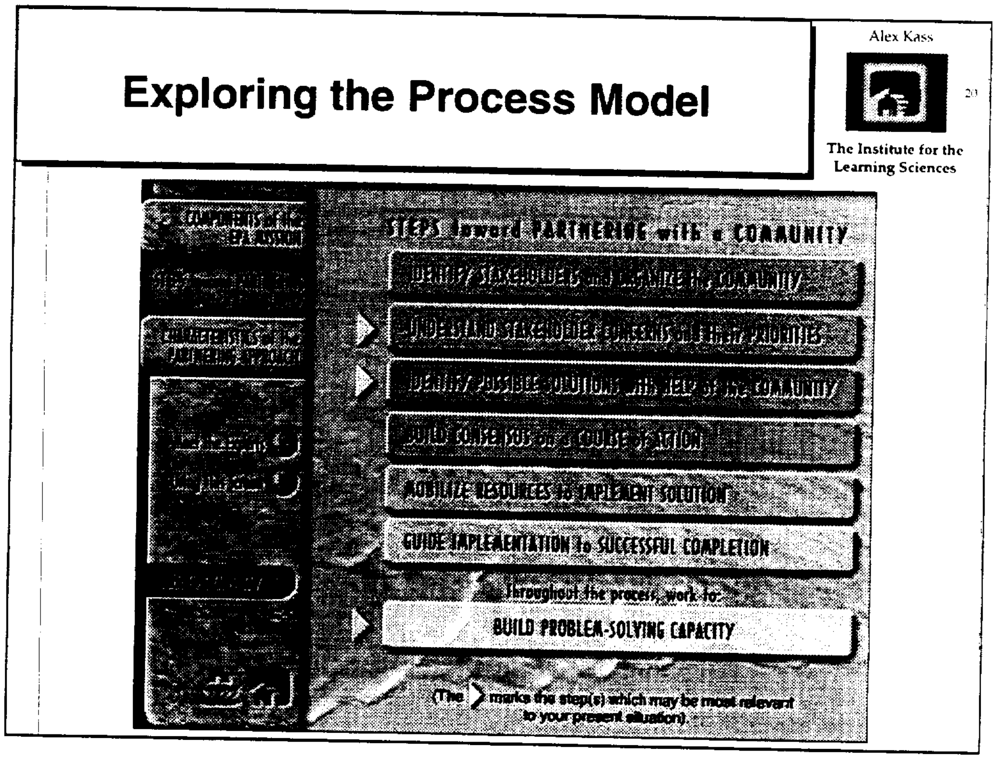
When Bad Things Happen

When certain especially painful things happen, such as a stakeholder becoming enraged, the system interrupts the simulation with some hints about what might have caused the problem, and a just-in-time opportunity to investigate the related issues.



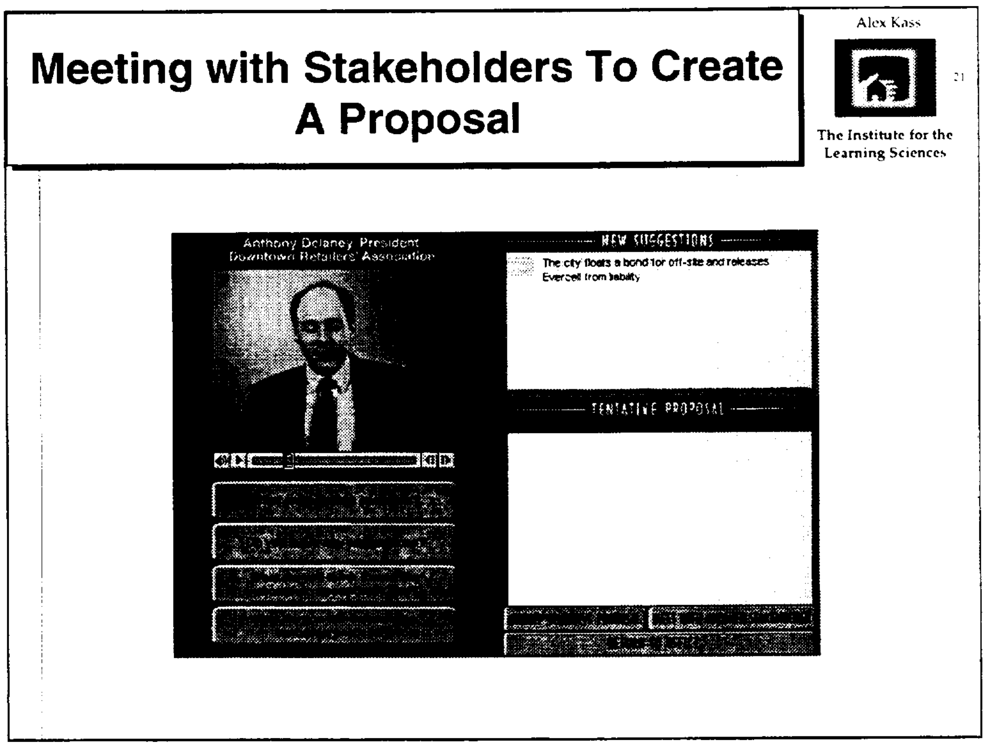
Digging Deeper

When the learner's interest is piqued, but not before, he/she has the opportunity to explore a hypermedia representation of the standard steps in community-based environmental protection. This model is only presented after it is given relevance by events in the simulation.



After the Group Meeting

After the group meeting is completed, the learner meets with individual stakeholders where he/she can hear the stakeholders' concerns and suggestions and can combine some of the suggestions into a proposal. The learner can also hear what each stakeholder thinks of his/her proposal as it is developed. Eventually, a conscientious student who is understanding the stakeholders and the ramifications of different options, can construct a proposal that will be acceptable to most of the community. In fact, there are several different approaches that will work – but even more that won't!



OK, So Let's Talk About Architectures

As clearly shown from the discussion of the two systems presented in this talk, all GBSs have many key features in common, but they also can differ in terms of the structure of the core activity.

Fortunately, our experience has shown that the number of different activity structures needed to cover a very broad range of learning objectives is not infinite. In fact, the five architectures below have proven to cover a surprising range of subjects. We have built the first four and are in the process of prototyping the fifth.

Let's talk a little about what they are.

Alex Kass

The Institute for the Learning Sciences

Learn by Doing *What?* (Equipment for a GBS Factory)

Architecture Name
Advise
Investigate
Run-a-system
Persuade
Design-an-artifact

Advise

Crisis in Krasnovia is an example of the architecture that we call Advise. In Advise applications, the main task is to recommend a course of action. In addition to Crisis in Krasnovia, in which the learner advises the President of the United States, we have deployed a second Advise system to teach business school students about emerging economies. In that system, the learner advises the president of a company about what course of action to take in expanding into the fictional country of Chernova (a former Soviet Republic). Various advisors, including a marketing vice president, a head of finance, etc., critique the learner's recommendation. The learner can support his/her argument with evidence taken from experts' stories about real-world emerging economy expansions that either have or have not worked in the past. The point is that GBS architectures are structure specific, but subject-matter independent.

Alex Kass

Learn by Doing *What?*
(Equipment for a GBS Factory)

The Institute for the Learning Sciences

Architecture Name	Activity	Sample Applications
Advise	Choose a course of action, develop a well-reasoned recommendation.	• Crisis in Kransovia • Emerging Markets
Investigate		
Run-a-system		
Persuade		
Design-an-artifact		

The Main Elements of Advise

This slide outlines the main elements of the Advise GBS architecture. Like all architectures, it is defined by the role and the mission of the learner, and by the types of coaching and critiquing that the learner gets.

Alex Kass

Advise: Architecture Overview

The Institute for the Learning Sciences

24

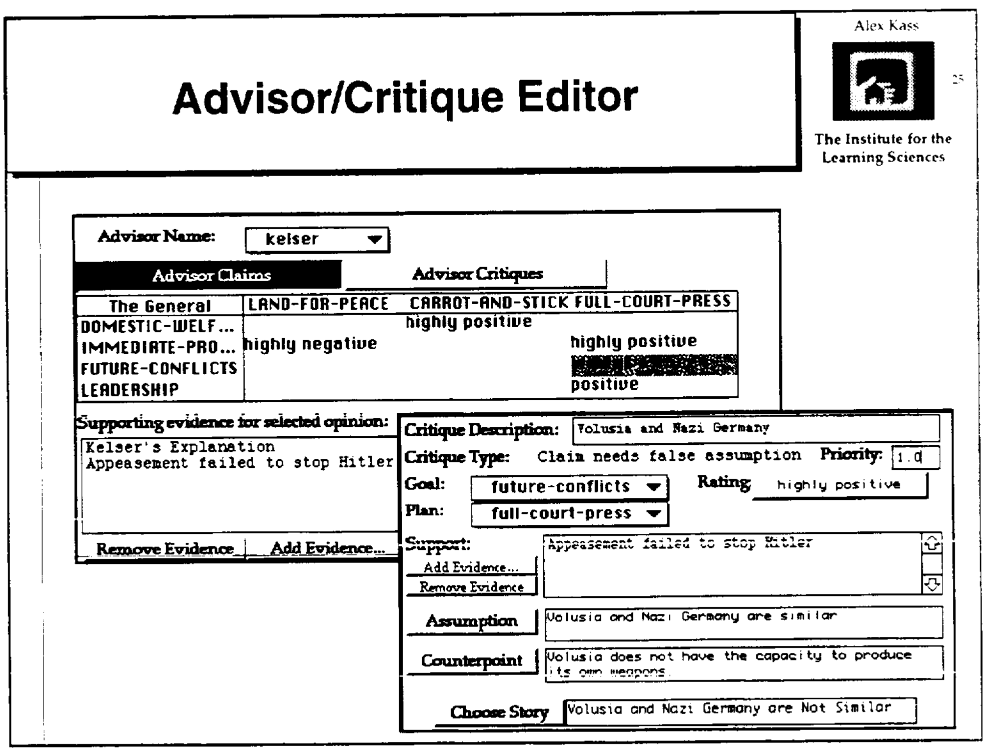
- **Student's role: Recommend a course of action.**
  - "Customer" provides a set of goals and a set of proposed plans.
  - Student must
    - analyze the relationships between the plans and the goals
    - choose a plan to recommend
    - give reasons and support with evidence
- **Coaching, critiquing and support materials:**
  - Opinionated advisors challenge student's decisions.
  - Evidence comes from real world experts and primary sources, such as scientific studies, historical documents, instrumentation readouts.
  - Decision-maker rejects unconvincing recommendations.

Tools to Support the Architecture

The architectures, which are conceptual entities, are all supported by software tools. The tool sets consist of authoring tools and run-time simulation engines. The image below depicts the interface of one component of the Advise tool.

The tool allows the author to set up critiques by defining a three-dimensional matrix relating advisors to plan-goal relationships, and indexing critiques to any plan-goal relationship.

The structure of the architecture is directly represented by the structure of the tool interface, making it much easier for the author to think in terms of the educational architecture rather than in terms of computer screens, buttons or video clips.



Investigate

Investigate is an entirely different architecture. Whereas Advise is about choosing a course of action and then defending it, Investigate is about running tests and making a diagnosis.

Investigate GBSs are commonly used to teach about natural sciences, in which empirical data is used to deduce unseen causes of detectable symptoms. For example, an Investigate system has been used to teach medical students about nutrition by having them diagnose a series of patients with a variety of nutritional deficiencies.

Investigate, however, is not limited to science. A recent application was built for an Art History course. The student was put in the role of art investigator, asked to attribute a painting, which might or might not have been done by the hand of Rembrandt.

Learn by Doing *What?*
(Equipment for a GBS Factory)

Alex Kass

The Institute for the Learning Sciences

Architecture Name	Activity	Sample Applications
Advise	Choose a course of action, develop a well-reasoned recommendation.	• Crisis in Kransovia • Emerging Markets
Investigate	Collect and interpret evidence to develop and test hypotheses	• Nutrition Clinician • Is it a Rembrandt?
Run-a-system		
Persuade		
Design-an-artifact		

Run-a-System

In the Run-a-System (or Run) application, the learner makes a series of decisions aimed at keeping/managing an ongoing crisis, or at keeping a dynamic system (such as an economy, an ecosystem or a small company) in equilibrium.

Learn by Doing *What*? (Equipment for a GBS Factory)

Alex Kass

The Institute for the Learning Sciences

Architecture Name	Activity	Sample Applications
Advise	Choose a course of action, develop a well-reasoned recommendation.	• Crisis in Kransovia • Emerging Markets
Investigate	Collect and interpret evidence to develop and test hypotheses	• Nutrition Clinician • Is it a Rembrandt?
Run-a-system	Keep a dynamic system in equilibrium over time, or get one out of crisis	• Run the powerplant • Fire commander
Persuade		
Design-an-artifact		

Persuade

Persuade is the architecture used to build the EPA system described earlier in the talk. In Persuade applications, the learner interacts with a community of simulated characters to try to build consensus for a solution to some problem. The key skill exercise is to understand the stakeholders' concerns well enough to craft a solution or to convince the stakeholders to adopt a solution.

A second application was built to teach college students about the French Revolution by sending them back in time (within the simulation) to Versailles just before the Revolution, to try to head off the violence by building support for the peaceful reform of France. The learner must meet with various parties, listen to their concerns and convince them.

In a third application, built to teach middle school students about global warming, the learner is sent to a global warming conference to build consensus for a treaty that can stop global warming from creating havoc.

Learn by Doing *What?*
(Equipment for a GBS Factory)

Alex Kass

28

The Institute for the Learning Sciences

Architecture Name	Activity	Sample Applications
Advise	Choose a course of action, develop a well-reasoned recommendation.	• Crisis in Kransovia • Emerging Markets
Investigate	Collect and interpret evidence to develop and test hypotheses	• Nutrition Clinician • Is it a Rembrandt?
Run-a-system	Keep a dynamic system in equilibrium over time, or get one out of crisis	• Run the powerplant • Fire commander
Persuade	Understand a range of constituencies with conflicting interests, build consensus around a plan	• CERES (EPA) • French Revolution • Global Warming
Design-an-artifact		

Design-an-Artifact

The Design architecture, currently being developed to support a project, teaches physics through engineering problems. The learner is asked to configure a set of features to produce an artifact with certain properties. For example, in our physics program, the learner will have to configure an elevator to meet a set of constraints. It is, of course, only possible to do that by mastering the underlying physical relationships between energy, acceleration, velocity and mass.

Learn by Doing *What*?
(Equipment for a GBS Factory)

Alex Kass

The Institute for the Learning Sciences

Architecture Name	Activity	Sample Applications
Advise	Choose a course of action, develop a well-reasoned recommendation.	• Crisis in Kransovia • Emerging Markets
Investigate	Collect and interpret evidence to develop and test hypotheses	• Nutrition Clinician • Is it a Rembrandt?
Run-a-system	Keep a dynamic system in equilibrium over time, or get one out of crisis	• Run the power plant • Fire commander
Persuade	Understand a range of constituencies with conflicting interests, build consensus around a plan	• CERES (EPA) • French Revolution • Global warming
Design-an-artifact	Configure device or artifact to meet a set of constraints	• Physics-through-engineering

Where We Are Going

The idea of using GBS architectures has a lot of power because it allows us to reuse both design ideas and code, across subject areas, thus reducing the design and implementation costs, sometimes dramatically. In addition, by codifying design ideas that work into specific architectures, it is possible to produce high quality designs with more consistency.

Our current work attempts to strengthen our collection of tools and architectures by increasing the number of architectures supported, increasing the amount of intelligence performance support given to authors, and integrating the various architectures into coherent higher-level curriculum-building solutions suitable for building large, multi-faceted courses.

Alex Kass

The Next Generation of Tools

The Institute for the Learning Sciences

- **More Architectures**
- **Smarter Authoring Tools**
  - Coaching for authors
  - Automatic Indexing
- **Friendlier Authoring Tools**
  - Pre-defined interface templates
  - Tools take author all the way from initial pedagogical design through interface without any programming
- **Integrated tool sets for producing multi-faceted courses that combine several types of activities into a comprehensive learning experience**

An Open Architecture for Simulation-Centered Tutors

Allen Munro
Director, Behavioral Technology Laboratories
University of Southern California
Redondo Beach, CA

An Open Architecture for Simulation Centered Tutors

Allen Munro
Behavioral Technology Laboratories
University of Southern California
250 North Harbor Drive, Suite 309
Redondo Beach CA 90277

Over a period of more than fifteen years, my colleagues and I have conducted a body of research on simulation centered tutoring and on authoring tools for simulation centered authoring. We have developed a number of experimental monolithic authoring applications. This presentation reports on that progress and outlines a new approach based on lighter-weight collaborating components. The presentation has three themes:

- Robust interactive simulations provide a context in which instructional intelligence can be consistently exploited.
- The authoring of both simulations and tutors is more cost effective than programming.
- Component-based systems offer advantages over monolithic systems, including lightweight delivery over intranets.

I gratefully acknowledge the support of the Air Force Research Laboratory under Contract F33615-90-C-0001and of the Office of Naval Research under Grant N00014-98-1-0510. The work reported here is the result of collaborations with my colleagues at Behavioral Technology Laboratories, especially Mark Johnson, Quentin Pizzini, David Surmon, Doug Towne and Josh Walker. Intellectual contributions have also been made by Carol Horwitz, Jeff Rickel, Lewis Johnson, Randy Stiles and many others.

An Open Architecture for Simulation-Centered Tutors

Allen Munro
University of Southern California
Behavioral Technology Laboratories
250 North Harbor Drive, Suite 309
Redondo Beach CA 90277
munro@usc.edu

CBT and ITS

In ITS (Intelligent Tutoring Systems), knowledge is explicitly represented. CBT (Computer Based Training) developers often make use of deep and sophisticated knowledge about the subject matter, but this knowledge is only implicitly represented in the structure of the lesson code that they write. In an ITS, there is an explicit representation of the knowledge that can be processed by the ITS to make run-time instructional decisions.

CBT and ITS

CBT: Computer-Based Training
ITS: Intelligent Tutoring System

- CBT—Page-based instruction
- ITS—Knowledge-based instruction

The Evolution of Development Methodologies in CBT and ITS

Both conventional computer-based training (CBT) and intelligent tutoring systems (ITS) exhibit a progression of development methodologies that have improved development productivity and the reliability and maintainability of the training and tutoring products produced. Since intelligent tutoring system technologies are newer, it is no wonder that the process of specialized authoring system development for ITS is still in a very active and formative state.

Parallel Evolution?

Increasing Productivity ↓

CBT Development	ITS Development
• Programming Languages (*e.g.*, Assembly, Fortran, 1960s)	• Programming Languages (*e.g.*, Lisp, 1970s-1980s)
• Special-purpose Languages (*e.g.*, Tutor, 1970s)	• Special-purpose Languages (*e.g.*, Soar, 1980s-1990s)
• Authoring Systems (*e.g.*, Authorware, 1980s-1990s)	• Authoring Systems (*e.g.*, RIDES/VIVIDS, DIAG 1990s....)

Leveraging the Information in Simulations for Tutoring

Simulations offer the opportunity to undergo informative interactive experiences, but, of themselves, do not constitute training or instruction. Tutoring can enhance learning in a simulation context by directing the student's attention to relevant aspects of the simulation by guiding the student through an educational series of actions and observations, and by helping the student overcome task impasses. In our laboratory, we have sought to enhance the process of tutoring system development by creating tools that encourage the development of simulations that can be automatically exploited for instruction. The considerable intellectual effort that is required to produce a simulation can thus serve the second purpose of supporting the automatic development and/or delivery of tutorial instruction.

Pedagogical Leverage

What can ***computer-based*** learning environments offer that simpler pedagogical technologies (books, videos, etc.) can't?

- Simulations
  - Vast potential for educational experiences
  - Hypothesis testing, discovery learning, etc.
- Intelligent Tutoring
  - Exploits a deep knowledge of the subject matter
    - Beyond presentation knowledge
  - May exploit a sophisticated representation of the student
  - Makes pedagogical decisions based on deep knowledge of the topic (and of the student)

Simulations + Intelligence

The knowledge that is required to make a simulation work correctly can be, if properly structured, used to support tutorials, both during instruction development and during instruction delivery. This concept underlies much of the work done in our laboratory to develop experimental simulation-centered tutor authoring systems.

It is important to be aware that simulation alone does not equal tutoring. Many military simulation training systems not only have no built-in intelligence, they also have no built-in instruction of any kind. The deep integration of tutorial features in a simulation system offers potential benefits for improving the speed of learning in the context of the simulation, for increasing the consistency of pedagogical results, and for reducing routine simulation supervision and coaching demands on expert human tutors.

Simulations + Intelligence

A simulation represents *knowledge about how things work.* How can we exploit the knowledge embodied in a simulation?

- During Instruction **Development**?
- During Instruction **Delivery**?

Note: Simulation alone is not instruction.

Background

We have been involved in the development of a series of authoring tools for the development of tutors for many years (Towne, Munro, Pizzini, Surmon, Coller and Wogulis 1990; Towne and Munro 1991, 1992; Munro and Towne 1994; Munro, Johnson, Surmon and Wogulis 1993; Munro 1997). A primary motivation for building such authoring systems for developing tutors is to increase the cost effectiveness of tutor development. Using a well-designed authoring system, an author should be able to much more rapidly develop, test and modify a tutor than if a number of lower-level development tools, such as programming languages and expert system shells, are used. A cost effective approach will make feasible the development of a much larger number and wider variety of tutors than would be possible using more expensive techniques.

Munro, A., *RIDES QuickStart*, Behavioral Technology Laboratories, University of Southern California, Los Angeles, 1997.

Munro, A., Johnson, M. C., Pizzini, Q. A., Surmon, D. S. and Wogulis, J. L., "A Tool for Building Simulation-Based Learning Environments, in *Simulation-Based Learning Technology Workshop Proc., ITS'96*, Montreal, Québec, Canada, June 1996.

Towne, D. M. and Munro, A., "Simulation-Based Instruction of Technical Skills," *Human Factors*,Vol. 33, 1991, pp. 325-341.

Towne, D. M. and Munro, A., "Supporting Diverse Instructional Strategies in a Simulation-Oriented Training Environment," in *Cognitive Approaches to Automated Instruction*, J. W. Regian and V. J. Shute (eds.), Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1992.

Towne, D. M., Munro, A., Pizzini, Q. A., Surmon, D. S., Coller, L. D. and Wogulis, J. L., "Model-Building Tools for Simulation-Based Training," *Interactive Learning Environments*, Vol. 1, 1990, pp. 33-50.

A Sequence of Simulation-Centered Tutoring Projects

- GMTS
  - Generalized Maintenance Trainer Simulator •1979-82
- IMTS
  - Intelligent Maintenance Trainer Simulator •1984-87
- RAPIDS
  - Rapid Prototyping of Instructional Delivery Systems •1988-89
- RIDES
  - Rapid ITS Development System •1990-95
- VIVIDS
  - Virtual Interactive Visual ITS Development Shell •1995-2000

GMTS

The Generalized Maintenance Trainer Simulator was a system that allowed authors to develop an interactive simulation based on photographs (accessed by a computer-controlled microfiche viewer) or video images (accessed from videodisk). The authors built tables of conditions that described what pictures should be shown based on the actions that students had taken and on the initial state of the simulated system. Systems whose behaviors could be adequately simulated with a few thousand pictures were an appropriate domain for the use of this technology. Many systems, however, have sufficiently complex interactive behaviors that such a 'static image set' approach is too constrained for practical application. Authoring and maintaining a simulation was difficult because it required editing large textual tables that prescribed the conditions under which various simulation scenes would be shown.

Towne, D. M., Munro, A., Johnson, M. C. and Lahey, G. F., *Generalized Maintenance Trainer Simulator: Test and Evaluation in the Laboratory Environment*, Navy Personnel Research and Development Center, San Diego, NPRDC TR 83-28, Aug. 1983.

GMTS

Generalized Maintenance Trainer Simulator •1979-82

- Simulation based on graphical scene states
  - Videodisk images
  - Thousands of possible view states
- Authored conditions for depicting simulation states
  - Tables of condition-scene correspondences
- Lesson Learned
  - Complex systems require unbounded simulation states

IMTS: The Intelligent Maintenance Training System

The IMTS authoring system described by Towne and Munro (1988), supported the development of interactive graphical simulations based on schematics (electrical, hydraulic or mechanical). Authors could create behaving simulations by connecting behaving components from a parts library. Additional information about system level symptoms and causes could be entered in table form. IMTS could assist a maintenance trainee in selecting tests and in evaluating those tests.

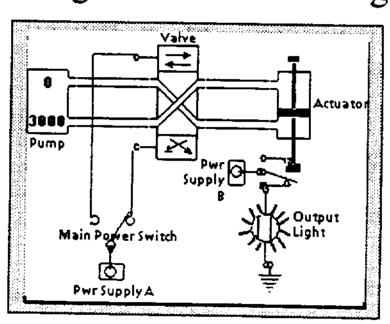
In this figure, there is a switch under the control of the student. The student can observe changes in a valve, an actuator and an output light. These are the *indicators* in the simulation. A good deal of useful instructional interaction with students can be phrased in terms of the states of *controls* (such as the switch) and such indicators. The simulation authoring system let developers draw objects in all their possible states and enter rules for transitioning among the states. A separate simulation scene editor was used to compose these authored objects into connected systems.

IMTS included an intelligent monitor of student performance called *Profile*. Profile watched students manipulating controls and noticed changes in indicators. It was designed for equipment troubleshooting (fault isolation) training. It made judgments about what malfunctions students should suspect based on their actions and the indications they had observed. It could offer advice on the potential usefulness of tests and observations being considered by the student, and advise about what should and should not be suspected based on what had been done in the session.

IMTS

Intelligent Maintenance Trainer Simulator •1984-87

- Simulation based on graphical object states
- Behaviors partially authored, partially universal
- Intelligent aiding for troubleshooting diagnosis



Lessons Learned From IMTS

IMTS could be used to build equipment simulations of a certain class. With additional authoring about malfunction effects, it could also be used to deliver equipment troubleshooting instruction. Unfortunately, the naturalness and ease of use of the simulation authoring environment encouraged its application in domains for which it was not designed.

Simulation authoring systems should not be too special purpose. The ease of use of the simulation composition system generated demand from a wider authoring community than was originally envisioned. The special-purpose maintenance training focus was a negative for this larger group of authors. The range of systems that could be simulated was also found to be too restrictive. Simulations were based on object states and did not support continuous effects. Some authors needed deeper control over simulation objects and simulated systems. One way of describing the simulation authoring limitations of IMTS would be to say that it tried to be too intelligent. The authoring system itself had an understanding of hydraulic effects, electrical effects and mechanical effects. As soon as authors tried to apply the system to simulating economic systems, chemical reactions, satellite orbits, or any other domain that did not fit into the 'equipment' orientation of IMTS, the authoring system kept authors from accomplishing what they wanted.

It should be possible to author instruction, as well as to generate it at run-time. IMTS also offered too little authoring control over instructional intelligence. Tutor development was fast and effective only if a particular type of maintenance tutor was wanted; no other type of tutor could be developed using the system.

IMTS—Lessons Learned

- *Simulation authoring systems should be general-purpose.*
- *It should be possible to author instruction, as well as to generate it at run-time.*
    - To modify what is generated by default
    - To create new types of instruction not envisioned by the tool developers

RAPIDS: Rapid Prototyping of Instructional Delivery Systems

The lessons learned from IMTS were first applied to our RAPIDS project (Towne and Munro 1991). The RAPIDS authoring system increased the openness and power of the simulation authoring system by letting authors build simulations with continuous value changes, as well as with state-oriented objects. It required authors to more fully specify how values were propagated in a system of connected objects instead of trying to automatically apply physical laws to every object connection. This made it possible for authors to build systems that were outside the narrower domain of equipment simulations permitted by IMTS. This slide displays a simulation scene for a RAPIDS tutor about a helicopter blade-folding system.

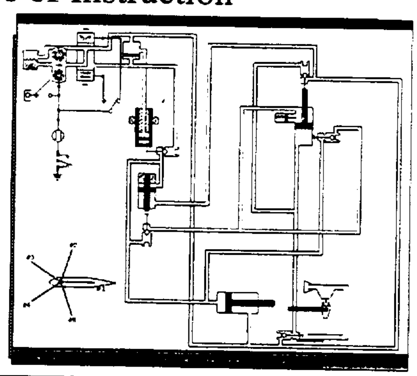
RAPIDS also provided a more open instructional system. It supported a variety of simple instruction templates that authors could use to create simulation-centered tutorials. Authors would build complete graphical simulations using one set of authoring tools. Then, using a different set of instruction authoring editors, they would build lessons for delivery in the context of those simulations.

Towne, D. M. and Munro, A., "Simulation-Based Instruction of Technical Skills," *Human Factors*, Vol. 33, 1991, pp. 325-341.

RAPIDS

Rapid Prototyping of Instructional Delivery Systems •1988-89

- More Powerful Simulation
  - Graphics controlled by continuous values, not merely states
  - All behaviors authored, not built in
- More Types of Instruction



Lessons Learned From RAPIDS

A lesson of the RAPIDS project was that still more control should be provided to authors. There was also demand for an easy-to-use simulation-centered authoring system that could be used on widely available workstations rather than only on specialized Lisp workstations.

Simulation authors sometimes require very detailed control over object behavior. Although the simulation approach was much more general purpose in RAPIDS than in IMTS, some authors needed finer control over graphical features than it provided. Authors could control universal characteristics of objects like location, rotation, scale and visibility, but they could not control the detailed, unique characteristics of most graphic primitives, such as fill pattern or color. RAPIDS still encouraged a state-oriented 'bitmap-style' representation of objects.

Ease of instruction authoring is good, but detailed control is necessary. The instructional authoring system was easy to use, but it provided only a few types of structured lessons. Nonetheless, there was considerable interest in using RAPIDS. Authors were sometimes frustrated by their inability to make simple changes in the wording or structure of generated lessons.

Multiple development and delivery platforms should be supported. Potential users were put off by the fact the RAPIDS, like IMTS, was available only on specialized Lisp machines.

RAPIDS—Lessons Learned

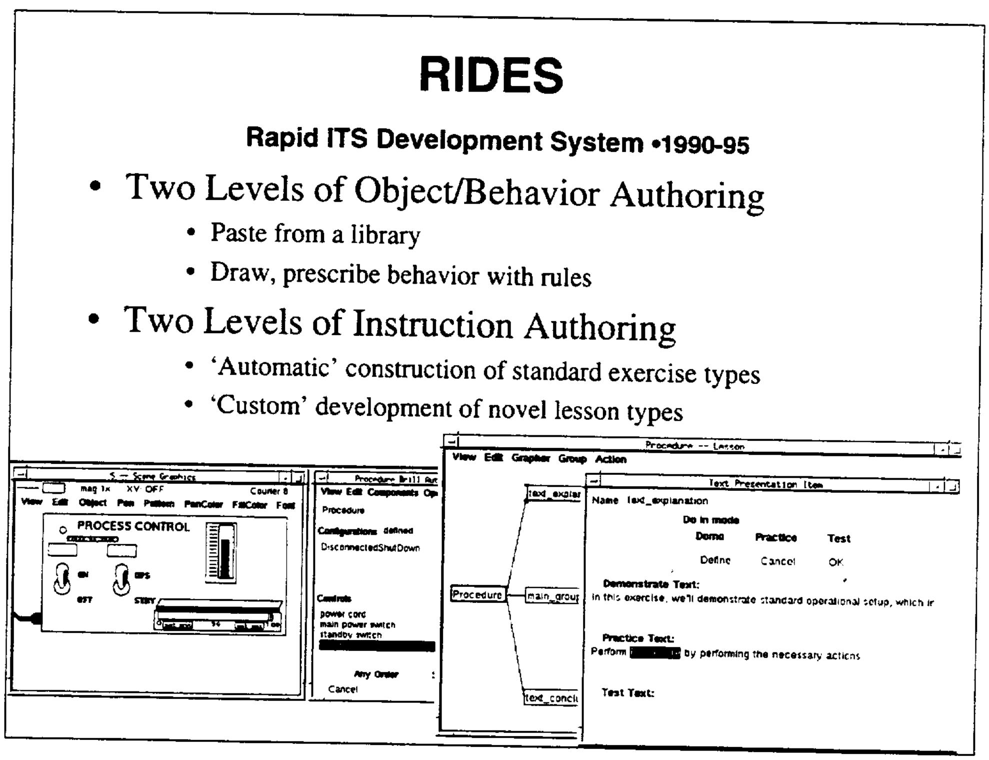
- *Simulation authors sometimes require very detailed control over object behavior.*
- *Ease of instruction development is good, but detailed authoring is sometimes necessary.*
- *Multiple development and delivery platforms should be supported.*

RIDES

The RIDES authoring system provides a rich simulation authoring environment, one that encourages the development of continuous behaviors and that offers very fine control over the behavior of graphical objects. Simulations can control the specialized attributes of certain types of graphic primitives, including the colors and patterns of objects, the text of text primitives, and so on.

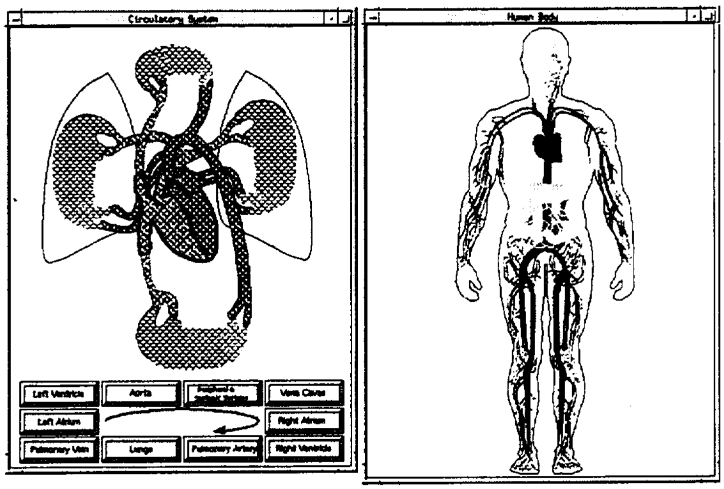
RIDES offers two levels of authoring for creating behaving objects in a simulation. The 'libraries' authoring level lets inexperienced authors select behaving objects (or groups of related objects) from libraries. More expert simulation authors can develop novel new objects by drawing them (or by importing their graphics) and then writing rules, using the RIDES event and constraint editors that control the behavior of those objects in RIDES simulations.

RIDES also provides two levels of instruction authoring. A high-level authoring system permits subject matter experts who are not experienced instructional developers to quickly and interactively produce several different kinds of simulation-centered lessons. This approach to authoring is called *patterned exercise authoring*. A lower-level system lets authors who are more expert develop lessons with varied internal structures and with fine control over what is said to the students. This approach is called *custom instruction authoring*.



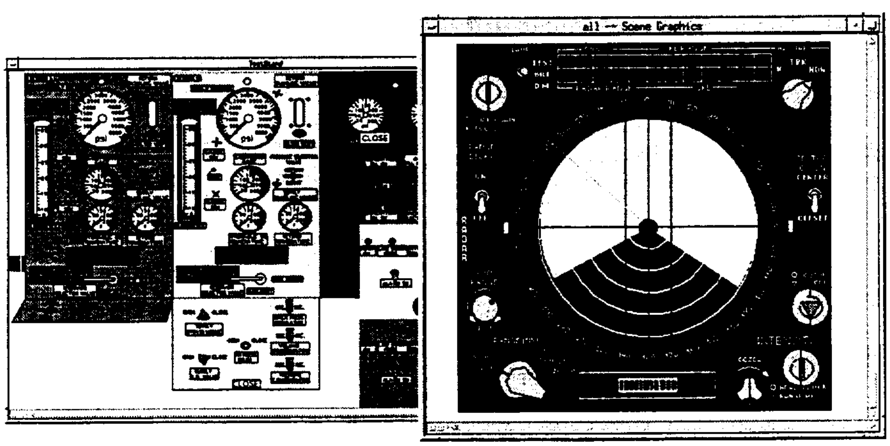
Rich Simulation Variety

Scores of simulations and related tutors have now been developed, many (such as the two shown in this slide) by defense contractors. Carol Horwitz at the Air Force Research Laboratory, has developed a simulation of the human circulatory system.



We believe that any claims for the tool status of a research product must be validated by the successful use of the product by parties other than the developers of the product.

Wide Variety of RIDES Simulations



- Developed at other institutions
- Many topics: orbital mechanics, human circulatory system, complex devices, ...

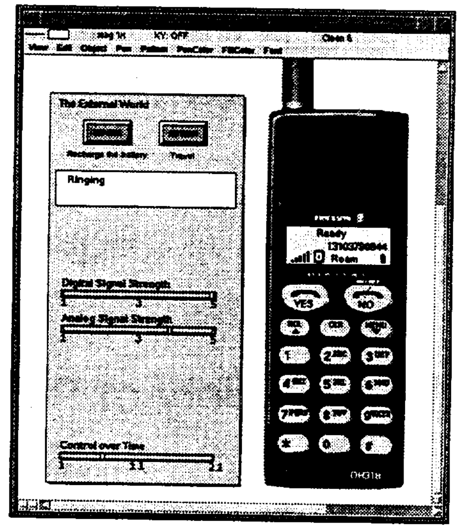
Authoring Productivity

In preparation for another workshop last year, I decided to quickly develop a new simple simulation and tutorial. In about eight hours I built and tested a rich simulation of my own cellular telephone. The simulation included such features as number recall, sound effects, randomly occurring 'system busy' states, cell signal strength, battery decay, and so on. In another forty-five minutes I built brief tutorials on how to use the phone to dial a new number or to access a previously dialed number.

Authoring Productivity

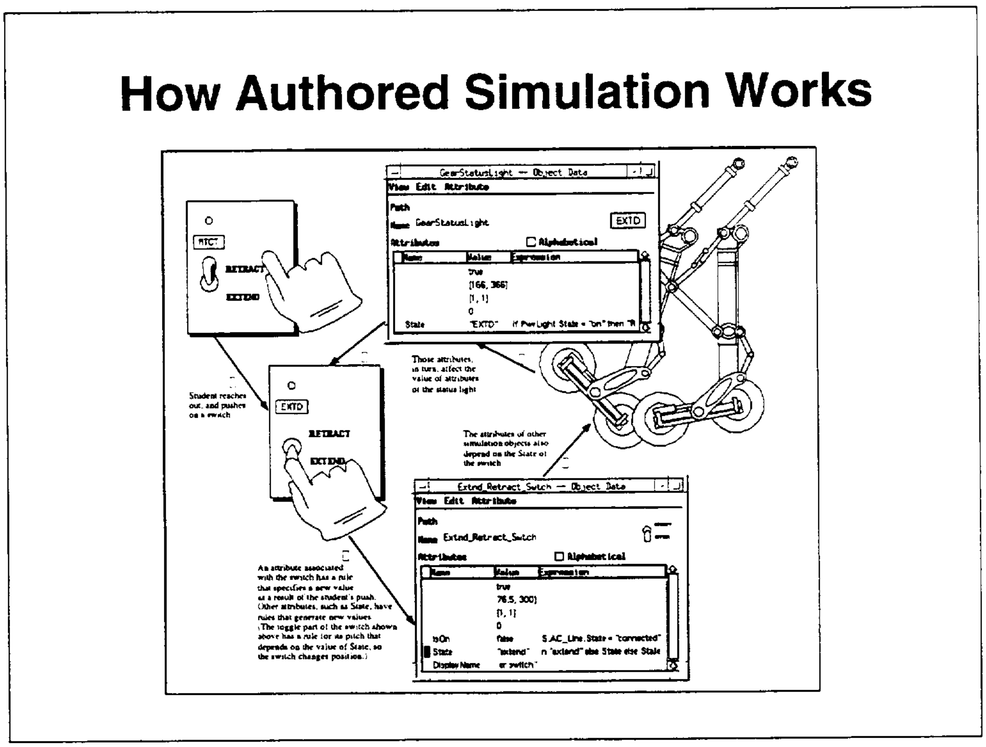
Simulations:
Easily modified
maintained,
debugged.

Instruction:
Quickly authored
tested,
revised.



The RIDES Simulation Engine

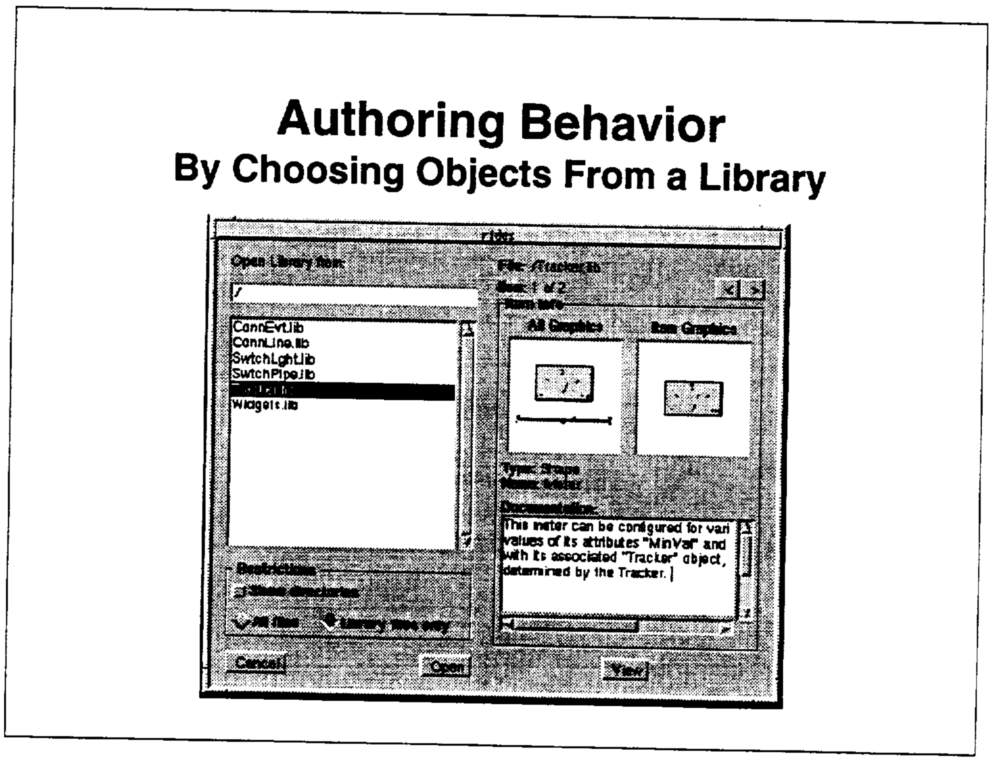
The RIDES simulation engine is responsible for the handling of simulation events and for the maintenance of simulated relational constraints. In addition to dealing with student actions and with clock events, the simulator coordinates with the instruction manager. At the request of the instruction manager, the simulator passes information back about certain student actions, starts and stops the simulation process, installs requested simulation states at the start of exercises, and provides a number of other services.



Library Objects

When an author copies a library object and pastes it into a simulation, the new object has whatever behavior was authored for the library object. If several objects are pasted together, their behavioral interactions are preserved. If pasted objects are to be influenced by other objects in the simulation, the author must edit the rules of the new objects so that they refer to the objects that are the source for their behavior in this context.

New library objects can be created simply by selecting authored objects on a RIDES simulation scene and saving them in a library.



Authoring Novel Behaviors

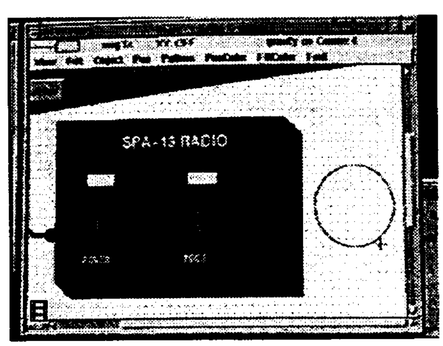
Any graphical object can be given a name and *attributes*. Some object attributes are *intrinsic* while others are created by authors. For example, an author can add a new number attribute to an object, give the attribute a name, specify its current value and, if desired, write a constraint that prescribes the attribute's value in terms of the values of other attributes. Attributes that are explicitly created by the commands of authors are called *authored attributes*.

Most intrinsic attributes control aspects of the appearance of objects. Every graphical object has the attributes Visibility, Location, Scale and Rotation. Each of these attributes controls some aspect of the appearance of an object. If Visibility is false, the object is not shown on its scene. Location is a point type value that prescribes the Cartesian coordinates of the object on the 'page' that is shown in a scene window. Changing the value of an object's Location will cause it to move to a different spot on its scene. The Scale attribute is also of type point; it specifies scaling factors that should be applied to the X and Y dimensions of the object. Rotation is a number attribute that controls the orientation of the object. In addition to these four universal attributes, objects of particular graphic types have other intrinsic attributes. Some objects also have attributes that control their color, text values, and so on.

Simulation authors can view an object's attributes by selecting the object and then opening its object data view. Relation rules that determine the attribute's value can be composed in this view or in a more detailed *attribute data view*.

Authoring Novel Behavior

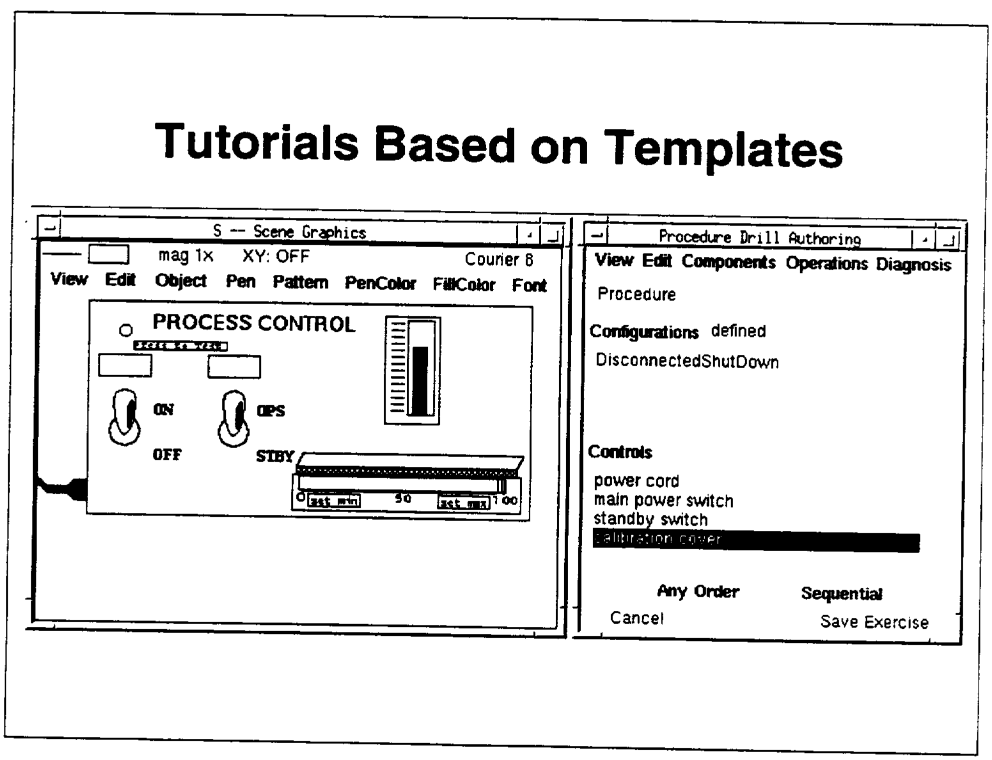
- Drawing new types of objects
- Creating object attributes
- Writing relational rules
- Prescribing event behaviors



Tutorial Lessons Based on Templates

RIDES provides a very simple interface for creating instructional vignettes that have one of a number of constrained structures, or patterns. Instructional units created using this interface are called patterned exercises. The figure on this slide shows a patterned exercise being created. The window at the left displays a simulation scene, while the window at the right shows the patterned exercise editor after the author has specified that a new exercise about operational setup (called "Procedure") should be created. The author has carried out several operations by clicking on the mouse-sensitive graphical objects, such as the power cord and switches. As each such control object is clicked, its name appears in the Controls list in the patterned exercise editor. At the same time that the actions are being recorded in the patterned exercise editor, they are generating graphical simulation effects in the simulation scene—switches change position, lights change colors or begin flashing, and so on. These graphical effects are brought about by the execution of previously authored simulation rules that determine the values of attributes of the graphical objects.

By carrying out the sequence of actions that a student will be required to carry out, an author creates a *specification* of the exercise. When the author clicks the "Save Exercise" button, a detailed instructional vignette is actually created and stored for later presentation to students.



Editing Authored Tutorials

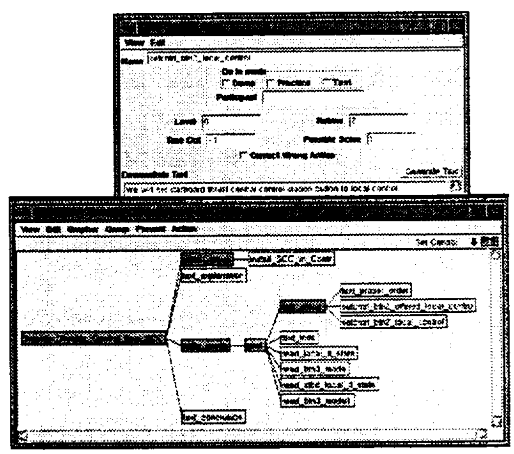
In the custom lesson editor, the elements of the lesson are displayed in a tree structure. Several types of nodes are found in an instructional tree: group nodes, presentation nodes and student requirement nodes, for example. The terminal (non-group) nodes represent elementary instructional presentations of different types. For example, some nodes present generated text presentations, which can be modified using this editor. Double-clicking on a node opens a dialog in which a presentation can be customized.

Many instructional items specify more than textual presentations. For example, the nodes that specify the control actions that a student is to take in carrying out a procedure are called Control instructional items. In the dialog for this type of lesson step, an author can specify not only the textual prompts that should be given to students, but also: what attribute value is the goal of the control item, and what object is the control that should be manipulated.

Edit Authored Tutorials

The VIVIDS Custom Lesson Editor

- Reorder or restructure a lesson by dragging lesson steps or groups of steps
- Add new instruction steps
- Open any step to edit it—changing the generated prompts and remediations, etc.



Lessons Learned From RIDES

RIDES offers a wider range of development approaches with differing levels of required expertise, offering different levels of control than did our earlier authoring and delivery systems. In fact, the range of options for controlling simulations seems to be about right. However, ...

Instruction authors need as much power and flexibility as do simulation authors. Even the *custom* instruction authoring in RIDES is still somewhat too constrained. It needs to be more flexible in order to give advanced developers more control over the structure and flow of tutorials. There should be a deep level of representation of instruction that provides the power and flexibility of a programming language. Many casual authors would never use this deep level of instruction, but it should be available for those who need it.

More platforms should be supported. In addition to Unix workstations, many developers have requested support for Windows NT and Windows 95, and ideally, future platforms as well.

An open architecture of collaborative components. Not every feature of the RIDES simulation-centered tutorial development and delivery system is required for every project to which it has been applied. In particular, some of our research colleagues at other institutions have wanted to use portions of the RIDES system, such as its simulation engine or its course administration system, while substituting for other portions of the system their own applications. An architecture of collaborative components, some providing simulation, others instructional or course management, defined with open communication standards, would facilitate the integration of our work with that of our colleagues.

RIDES—Lessons Learned

- *There should be more power and flexibility for **instruction** authors.*
- *More platforms should be supported.*
- *A more open architecture of collaborative components is required.*
    - Support the use of only some components
    - Be able to incorporate 'foreign' components

VIVIDS

VIVIDS—an extension of RIDES—is an authoring environment for building interactive graphical simulations and tutorials based on those simulations. The graphical simulations can be two-dimensional simulations, which are viewed in graphical windows that are part of the VIVIDS program, and they can also be three-dimensional simulations which are viewed in a collaborating *three-dimensional browser*. In addition, VIVIDS can communicate with an *autonomous agent* that can observe and critique student actions and play a role in simulated team tasks.

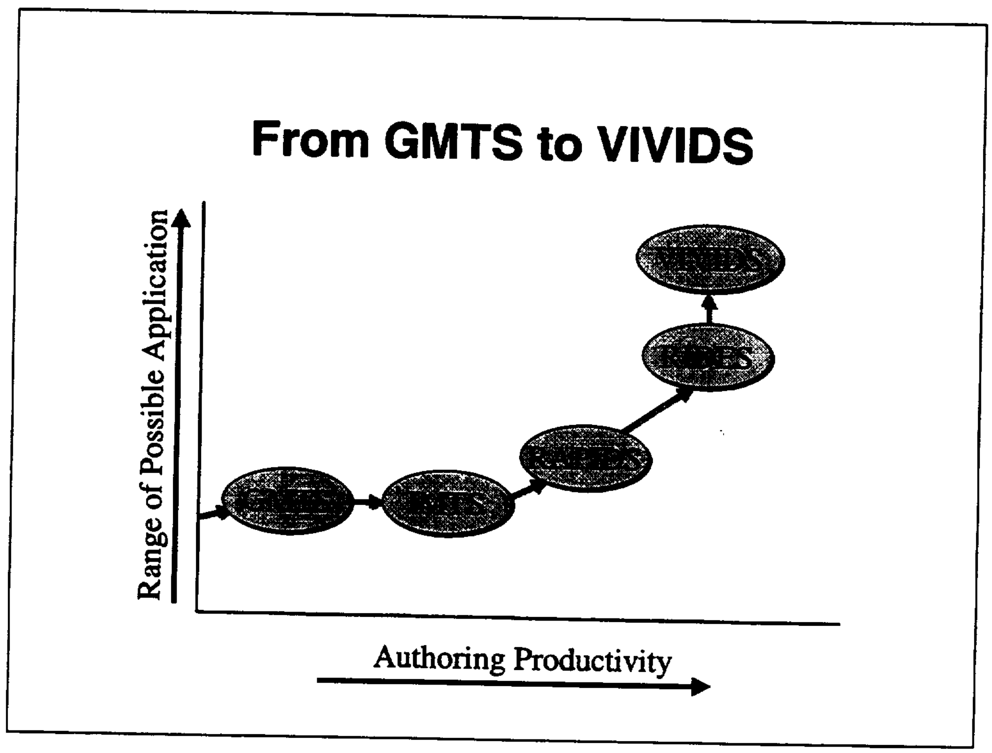
VIVIDS

Virtual Interactive Visual ITS Development Shell •1995-now

- Collaboration with Virtual Environment Views
- Collaboration with Pedagogical Agents
- Enhanced Instructional Control

Progress
From the Generalized Maintenance Training System to VIVIDS

The progression of simulation-centered tutorial authoring-and-delivery systems that we have developed exhibits progress on at least three dimensions, two of which are depicted here. GMTS and IMTS were restricted to maintenance training requirements. Subsequent systems have offered successively wider ranges of application, including operator training and complex concept learning. Authoring productivity has also increased. Not only do the more recent authoring systems support faster development, they also support effective maintenance in a number of ways.



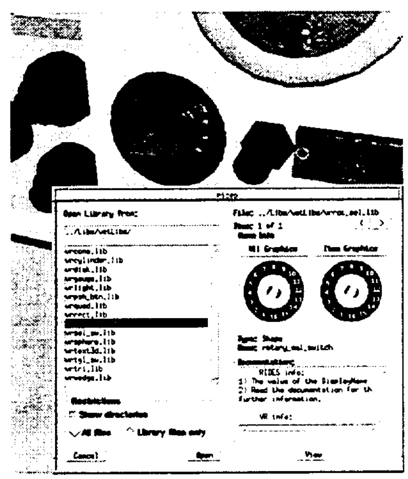
VIVIDS
RIDES Extended for Virtual Environments

In some cases, authors can reuse objects that were developed for other VE simulations. These can be directly copied from one simulation and pasted into the other. Alternatively, a number of useful simulation objects are available in the VIVIDS library files. In this picture, the VIVIDS "Open Library" dialog has been used to import a rotary control object into the simulation. To link the behavior of the simulation object to the three-dimensional graphical objects of Vista, the author must enter data into the attribute value fields of the simulation object in its *object data view*. See Munro and Pizzini (1998).

Munro, A. and Pizzini, Q. A., *VIVIDS Reference Manual*, Behavioral Technology Laboratories, University of Southern California, Los Angeles, 1998.

VIVIDS—RIDES Extended for Virtual Environments

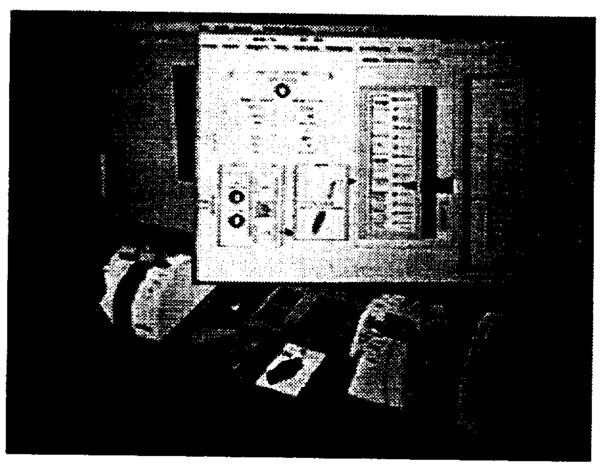
- 2D graphical simulation objects control the appearance of 3D model elements, in collaboration with a viewer.
- Libraries of such objects foster reuse.



Locus of Behavior

Authored VIVIDS simulations are the source of propagating simulation effects. Actions taken in the two-dimensional simulation control the behavior of corresponding objects in the three-dimensional view. Most actions taken in the three-dimensional view are referred to VIVIDS to determine any simulation effects. A three-dimensional browser can support *immediate behaviors* in order to ensure a high level of responsiveness to user actions, avoiding inter-application communications. In such cases, the browser-supported behavior is reported to VIVIDS so that any 'downstream' effects can be appropriately simulated.

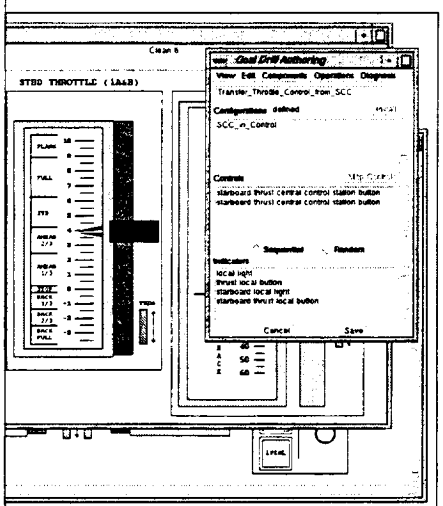
VE Behaviors Under the Control of a 2D Authored Simulation



Authoring Procedural Tutorials

The patterned exercise editors of VIVIDS support instructional authoring only in conjunction with the two-dimensional views of VIVIDS. Instruction can be delivered in either the two- or three-dimensional (virtual environment) contexts, however.

Authoring Procedural Tutorials

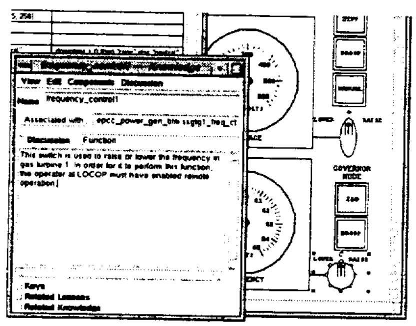


- Authored largely by carrying out the procedure in the 2D view.
- Students can be prompted to carry out the same steps (in either the 2D or the 3D environment) with automatic remediation of errors.

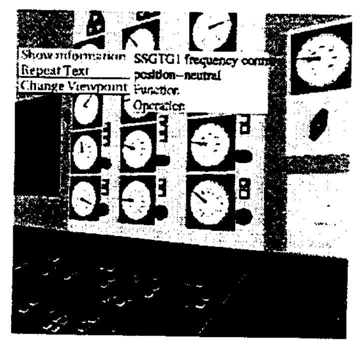
Knowledge Units

Authors can create *knowledge units*, which bundle textual descriptions, links to authored lessons, and links to other knowledge units. These knowledge units can be associated with simulation objects. At the author's discretion, students can be given access to the information in these knowledge units.

Authoring and Browsing Related Information



- Authoring information in VIVIDS



- Accessing the information in VE

Collaboration with Tutorial Agents

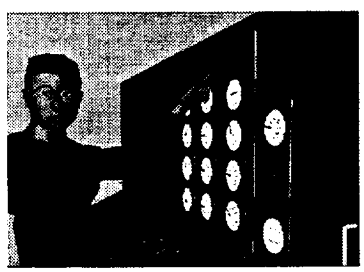
The pedagogical agent, *Steve*, developed by Lewis Johnson and Jeff Rickel at Information Sciences Institute, can access VIVIDS services in a simulation learning environment. Steve can be informed about user actions and relevant changes in the simulation. VIVIDS also supports action taken by Steve to manipulate the world. Such actions have effects that are indistinguishable to a student observer from the actions taken by a real person in the simulation.

In addition, Johnson and Rickel have provided features in Steve that make it possible for structured VIVIDS lessons to control Steve as an aid to learning. Steve can be used to point out objects and to demonstrate actions in the simulated world.

A sequence of structured VIVIDS lessons and Steve lessons can be controlled by the authored course management system of VIVIDS.

Johnson, W. L., Rickle, J., Stiles, R. and Munro, A., "Integrating Pedagogical Agents Into Virtual Environments," *Presence*, 1998.

VIVIDS
Collaboration with Tutorial Agents



- Inform the agent of the state of the world.
- Respond to the agent's actions as a participant.
- Use the agent to indicate or to demonstrate.
- Manage instructional sequencing.

Motivations for a New Architecture

While the RIDES and VIVIDS systems achieved practical success and met a number of research objectives, we found that ambitious developers were pushing our authoring systems in ways we had never anticipated. These developers found ways to couple RIDES and VIVIDS to new technologies, such as speech input systems and virtual environment viewers. The marriage of our monolithic authoring and delivery system to unanticipated technologies was sometimes awkward, and failed to provide completely satisfactory results. Often only a subset of the features of VIVIDS and of the newly coupled technologies could be practically applied in tutorials, and the design, development and maintenance of complex tutorial systems was more difficult than it seemed it should be. This was painfully evident by contrast with the natural interactive approach used for the development and delivery of simulations and simulation-centered tutorials when only the native capabilities of the monolithic VIVIDS system were employed.

It is also desirable that students have on-line access to course materials over the internet or via an intranet. This approach has many obvious advantages for course maintenance and version control. The use of Java enables distribution of the student 'player' directly over the net. In addition, certain new net-based technologies, such as Marimba Castanet, support the automatic redistribution to registered students of only that portion of tutorial material at the server site that has been revised since the student last used the tutor.

Why a New Architecture?

- Avoid the problems of a monolithic application
  - Cumbersome authoring requirements for VE behavior and tutorials
  - Large size of delivery application—inappropriate for Internet delivery
- Openness to collaboration
  - Open to 'foreign' simulations
    - Including real devices or systems
  - Open to other instructional approaches
    - Pedagogical agents
- Platform independence

Architectural Themes

1. *Avoid unnecessary baggage.*
2. *Achieve software cost savings through component reuse.*
3. *Be better positioned to meet unanticipated needs.*
4. *Be better able to exploit new technologies as they emerge.*
5. *Be able to support tutorial applications with widely differing requirements.*

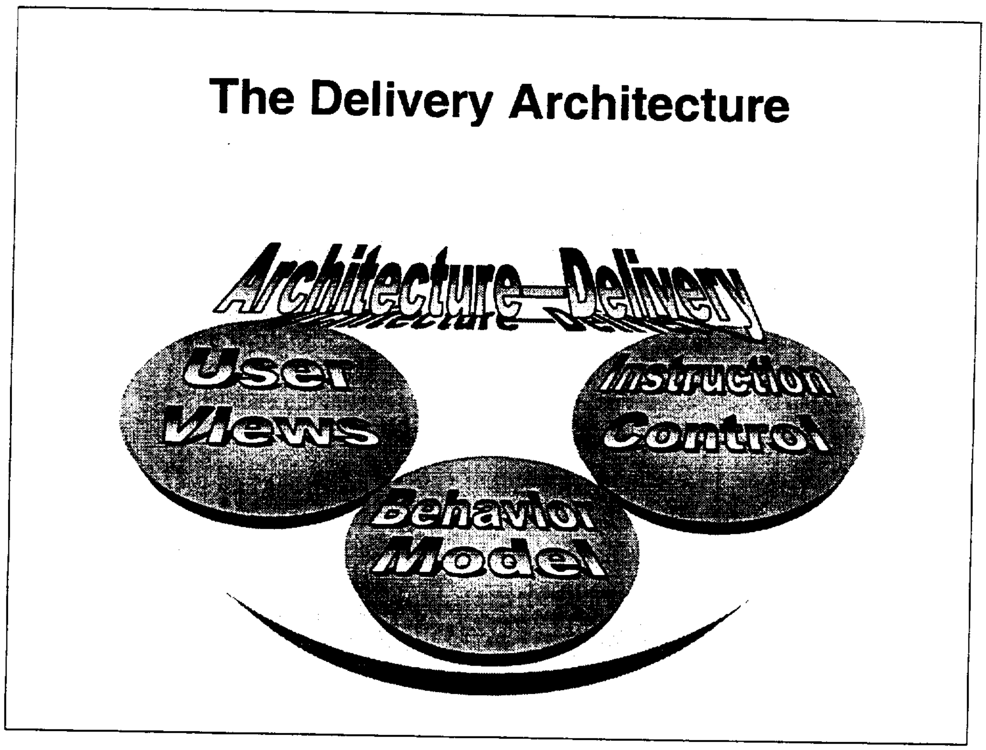
Major Architectural Themes

- Separate authoring from delivery
  - But delivery applications and authoring applications will share components
- Define components in terms of services
  - Support component replacement
- Enable a wide range of simulation-centered tutoring systems
  - Lightweight, Internet-delivered 2D simulation tutorials
  - Networked team training
  - Virtual environment tutorials
  - Either authored or hard-coded simulations

The Delivery Architecture

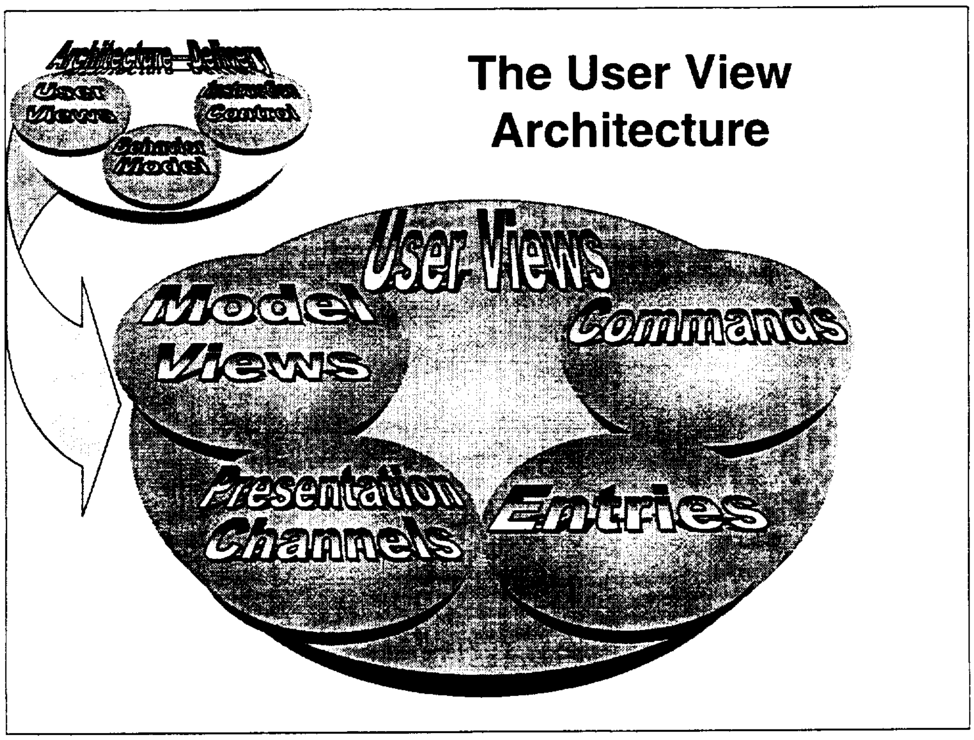
A lightweight delivery system should contain no elements that are used only for authoring, but not for instructional delivery. Three major components are found in the new delivery architecture:

- User Views—a component responsible for all the student user interface elements.
- Behavior Model—a component that specifies the behavior of the simulation in a simulation-centered tutor.
- Instruction Control—a component that is responsible for the structure of the entire learning environment, as well as for moment-to-moment tutorial interactions.



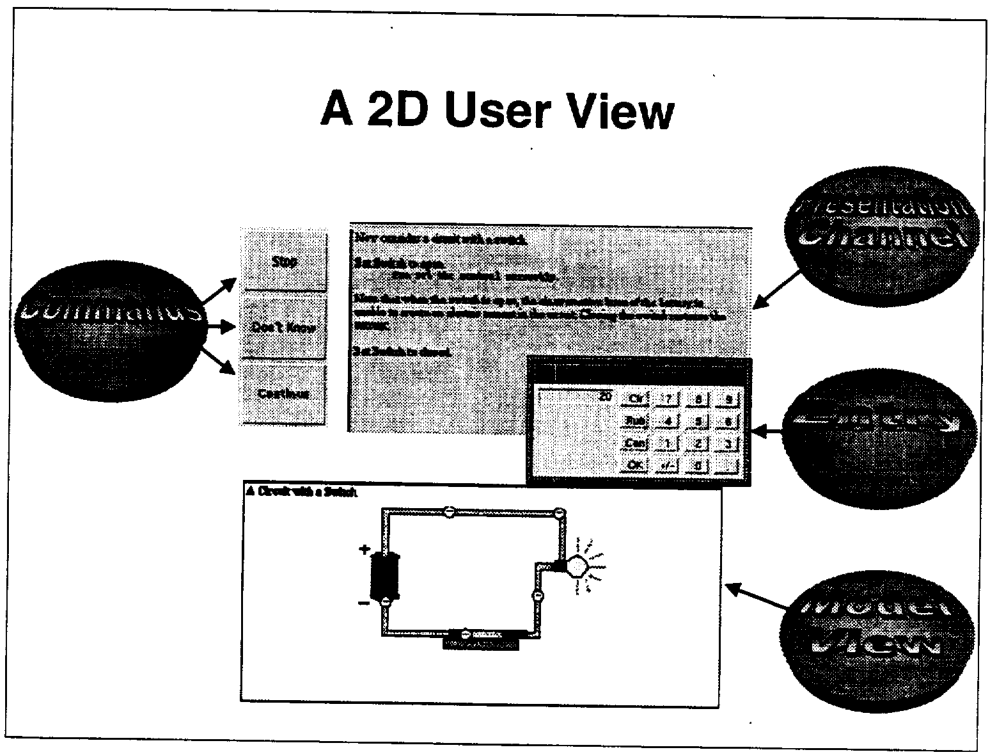
The User View Architecture

A user view is responsible for everything that can be seen (or heard, or touched, etc.) by a student during simulation learning. Its model view component is responsible for rendering the simulated world prescribed by the behavior model, and for detecting user events and passing them on to the behavior model for semantic processing. Presentation channels may include text, speech, audio, HTML and video presenters, to name a few. Entries are user interface components that students can use to answer questions that cannot be answered by actions in the simulated model view. Commands are interfaces for higher-level student interactions with the tutorial system, such as pacing instruction, asking for help, and so on.



A Two-Dimensional User View

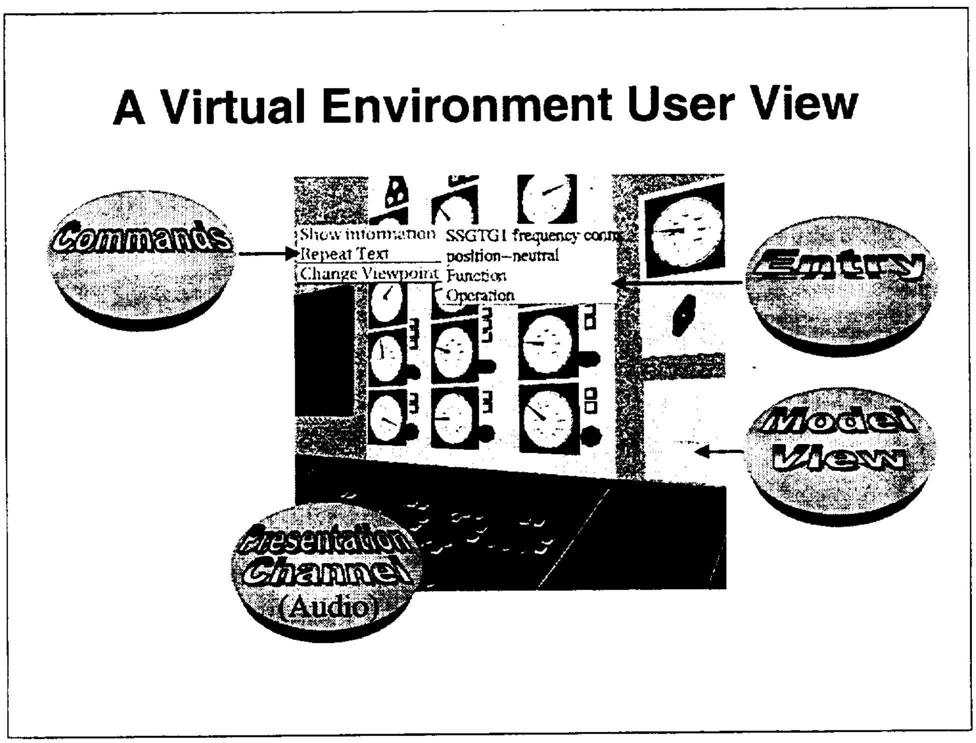
An example of the user view architecture is shown here. Commands provide a user interface and control mechanism by which a student can engage in higher-level interactions with the tutorials system. In this example, the student can issue commands to stop the training session, to request help via the "Don't Know" button, and to direct the tutor to continue with a lesson, by pressing the "Continue" button. A simple text presentation channel is shown in this example, along with a numeric keypad for entering answers to questions posed by the tutor. At the bottom of the figure, an interactive simulation model view shows a simple circuit consisting of a battery, a switch and a light.



A Virtual Environment User View

In this example, the same generic model view architecture is embodied in a three-dimensional context. In this case, there is a menu of high level commands available to the student. The presentation channel is an audio channel that uses text-to-speech to produce explanations, guidance and directives. One example of an entry interface is also a menu of choices. The three-dimensional visual representation of the simulated explorable world is the model view.

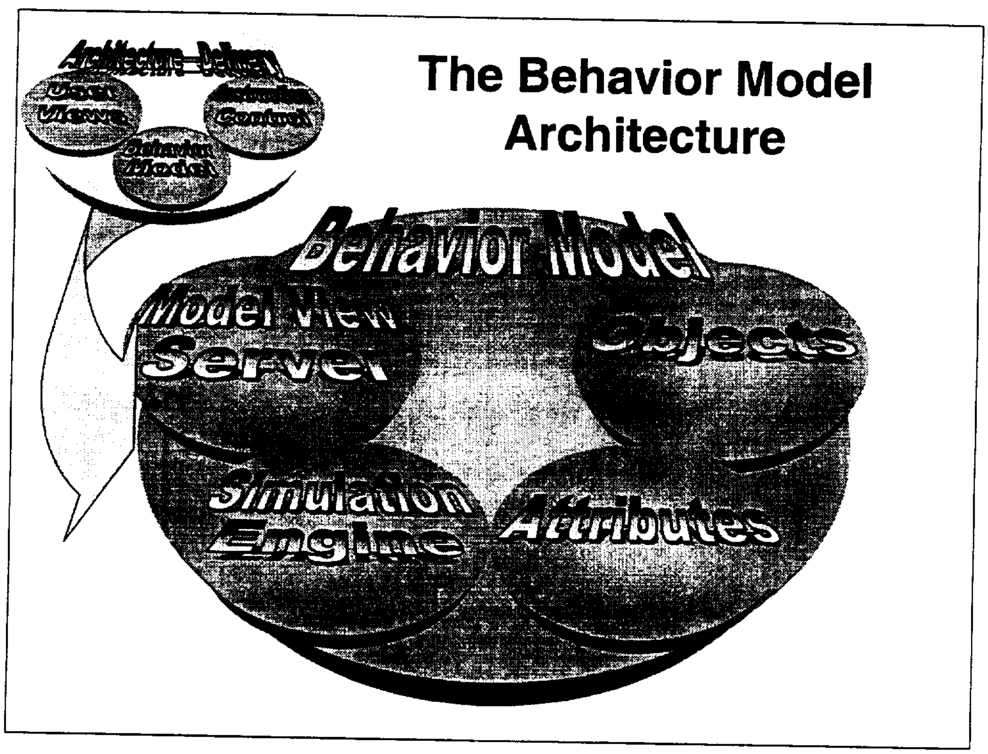
Stiles, R., McCarthy, L., Munro, A., Pizzini, Q., Johnson, L. and Rickel, J., *Virtual Environments for Shipboard Training*, Intelligent Ship Symposium, American Society of Naval Engineers, Pittsburgh PA, 1996.



The Behavior Model Architecture

A behavior model is responsible for dealing with user actions at the semantic level, by propagating simulation effects. It informs a model view of attribute value changes that would require a change in the view. It also is prepared to render services to the tutorial component--services such as starting or stopping a simulation, changing an attribute value, reporting on a change in a value that is of interest to a tutor, and so on.

Many different behavior model architectures can accommodate the requirements of the newVivids architecture, including some that do not include all of the internal components shown here. This behavior model architecture is one that we have developed to accommodate a universal simulation engine and attribute model, which can be used with either two- or three-dimensional model views to deliver interactive graphical simulations. Only the model view server component of the behavior model needs to be replaced to create a behavior model that delivers authored data-driven simulations for the chosen type of model view.



Why Are Model Views Separated From Behavior Models?

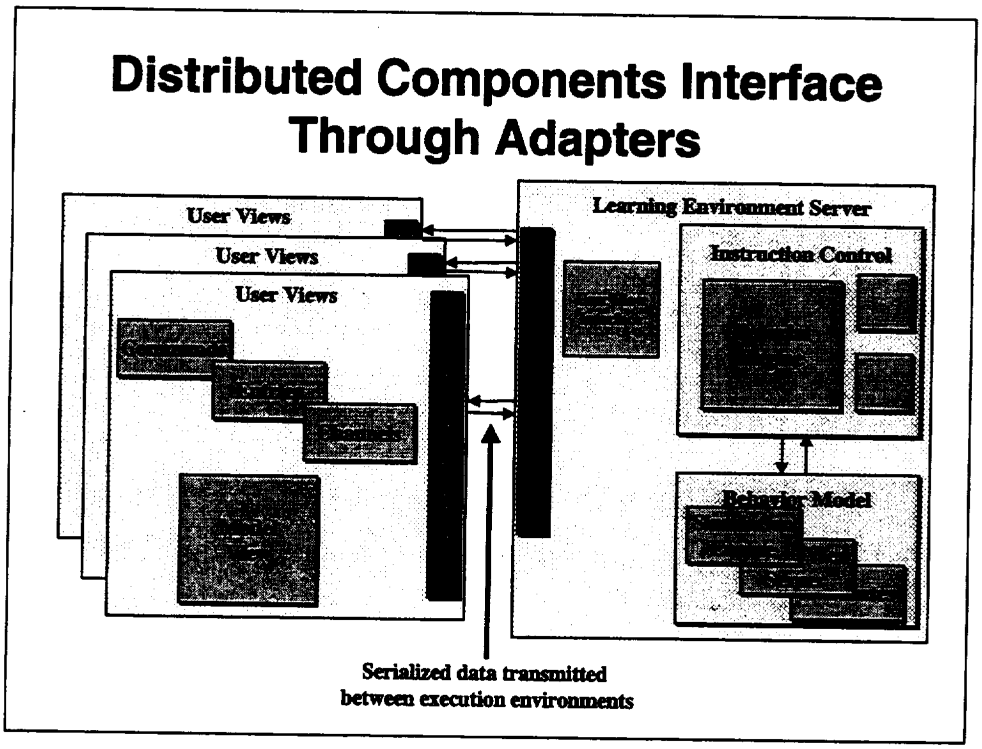
The reason that model view and behavior model components are shown separately from each other is that the architecture supports team training applications. Each student can have his own computer with its own view of the simulation. These model views need not be the same, and for many applications they will not be. Team members often work with different parts of a complex interconnected system, and are able to see only one part of it at a time. The single behavior model determines the values of simulation attributes, and some of these value changes cause changes in what is displayed in the model views.

Model Views Disentangled From Behavior Models

- Distributed simulation views for team training
  - Different team members may see the same or different views
  - But behavior in all views is determined by the central model
  - Future support for 'external' behaviors
- Core behavior model components can be universal
  - Same simulation engine for both 2D and VE simulations

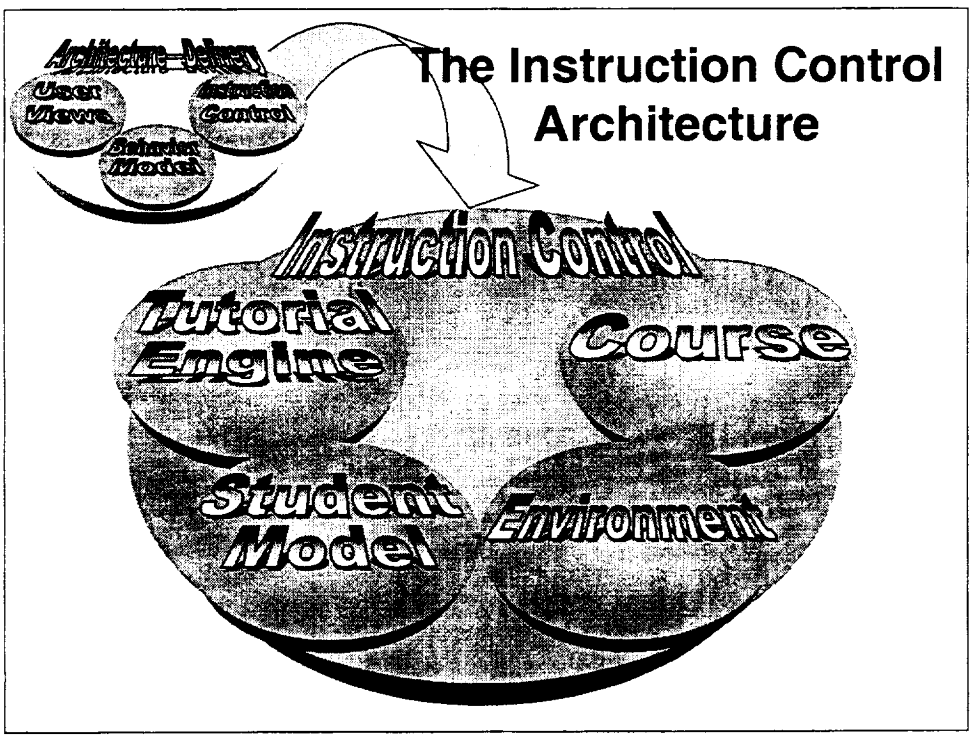
Distributed Components Interface Through Software Adapters

In the team training context, each team member has his or her own user view interface, including a model view of the underlying simulation. In the particular tutorial system represented schematically here, the behavior model and instructional control systems are on the same server, but these elements, too, could be distributed. Distributed components are coupled with adapters that play the role of the components with which they must collaborate. These adapters are responsible for the serialization of data flows and service requests. We have developed experimental sample adapters based on several communications standards, including CORBA.



The Instruction Control Architecture

The instruction control components are responsible for the configuration of the complete learning environment, including the availability and positioning of the possible user interface elements. The tutorial engine component is a service requester for many of the other major components: it can ask a presentation channel to fetch and present a video. It can ask the behavior model to put the simulation in a particular state. It can ask the model view to stop responding to user actions for a time so that the tutor can demonstrate a sequence of actions without student interference.



Advantages of the New Architecture

A new, open architecture approach to simulation-centered tutors offers advantages such as lightweight tutors, cost savings due to component reuse and improved maintainability, more rapid adaptation to new presentation technologies, and support for a wider range of tutorial applications than would be possible with a monolithic system.

The open architecture's components can be abstractly defined in terms of the sets of services that the instances of these components provide. This abstract approach encourages the extension of the simulation-centered tutoring component model to tutoring in the context of real devices or systems, in addition to tutoring in the context of simulations.

Advantages of the New Architecture

- Lightweight delivery
  - Application
  - Applet
  - Marimba Tuner
- Open to new components
  - New types of presentations
  - New tutorial components
  - New simulators
- Both individual and team training supported

First Implementations

Our first implementations based on the new architecture are of delivery systems for simulation-centered tutoring, not development or authoring systems. Under the sponsorship of an Air Force research contract monitored by the Air Force Research Laboratory, we have developed the architecture outlined here, together with a tutorial engine component, other elements of the tutorial control system, and a number of presentation channels, entries and commands.

Under the sponsorship of an Office of Naval Research grant, we are developing a run-time delivery system for the delivery of authored graphical simulations for tutoring over intranets. This project is creating a two-dimensional model view and collaborating behavior model for authored simulation delivery on a variety of platforms.

First Implementations—Delivery

- The *VIVIDS New Architecture* Project—AFRL
  - Overall Architecture
  - Tutorial Engine Component
  - Environment Control
  - Presentation Channels, Entries, Commands
- The *Intranet Delivery of Authored Simulations* Project—ONR
  - Behavior Model—Simulation Engine, Model View Server, Attributes, Objects
  - 2D Model View

The *VIVIDS New Architecture* Project

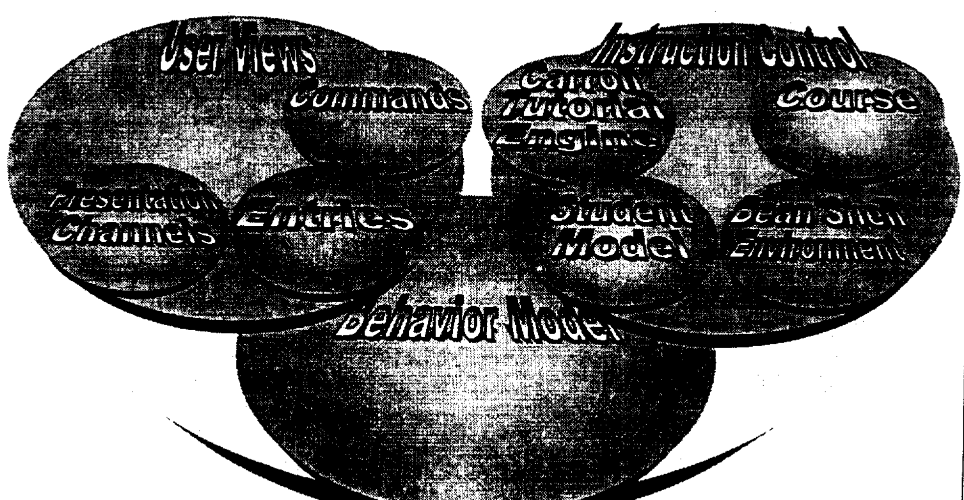
The *VIVIDS New Architecture* Project supports the development of the newVivids

- Overall Architecture
- Tutorial Engine Component—Carroll
  - Carroll delivers instruction defined in lesson meta-language (LML), an XML compliant data format.
- Environment Control
- Presentation Channels, Entries, Commands

Air Force Contract No. F33615-90-C-0001; Contract Technical Monitor, Jim Fleming, AFHRL:fleming@alhrt.brooks.af.mil

First Implementations

***VIVIDS New Architecture* Project**

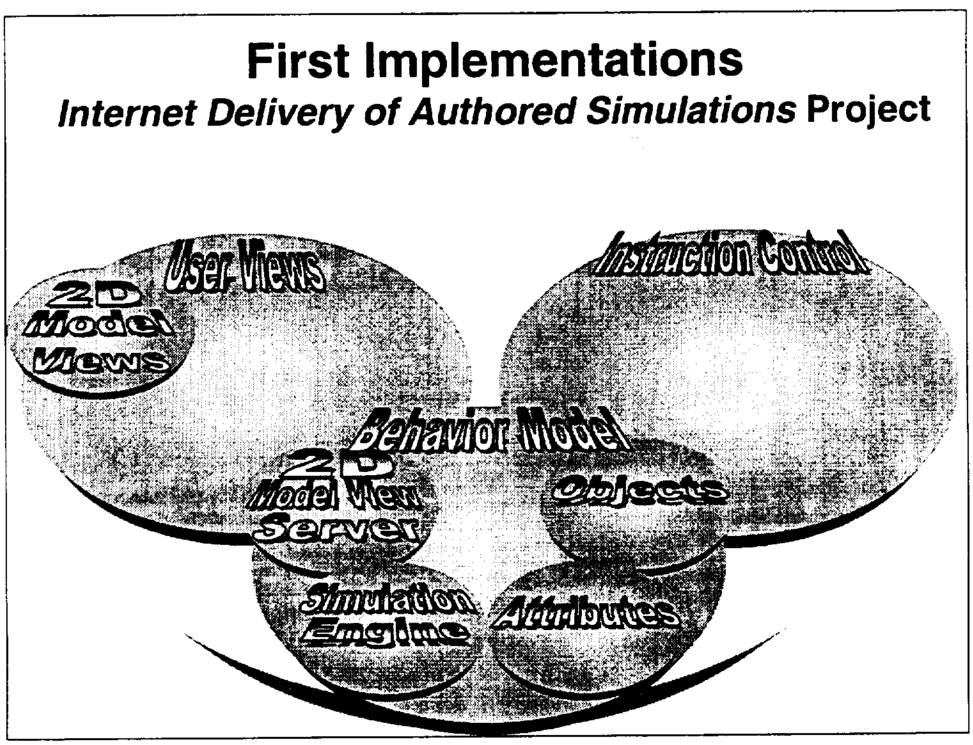


The *Intranet Delivery of Authored Simulations* Project

The *Intranet Delivery of Authored Simulations* Project supports the development of the newVivids

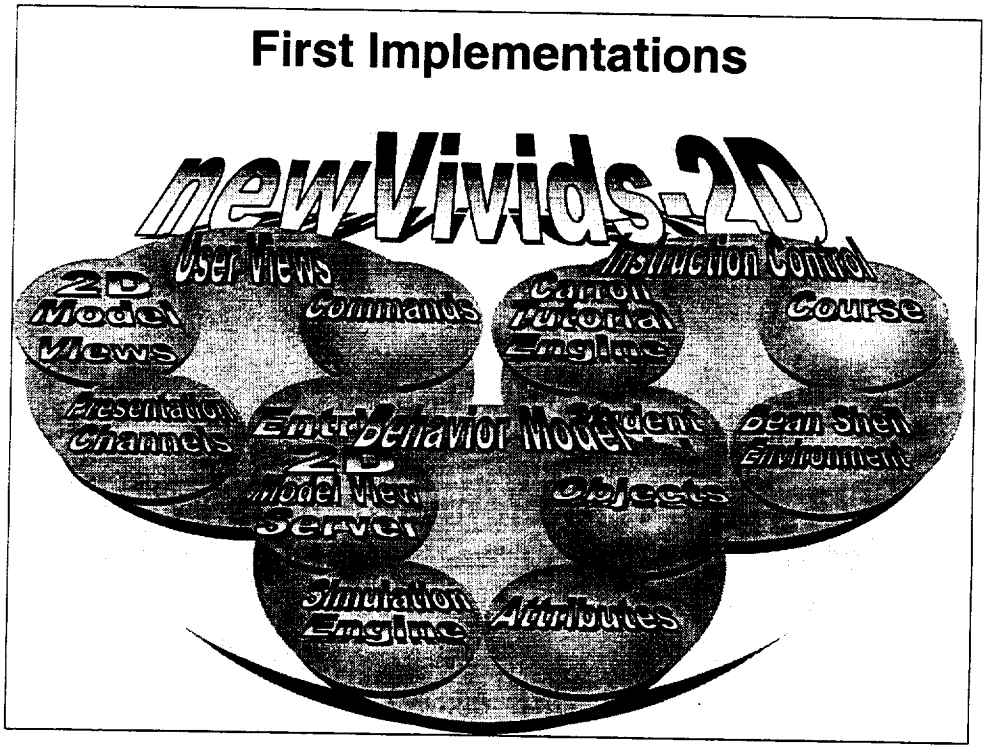
Behavior Model—Simulation Engine, Model View Server,
Attributes, Objects
2D Model View

Office of Naval Research Grant No. N00014-98-1-0510. Technical monitor, Dr. Susan Chipman: chipmas@onr.navy.mil



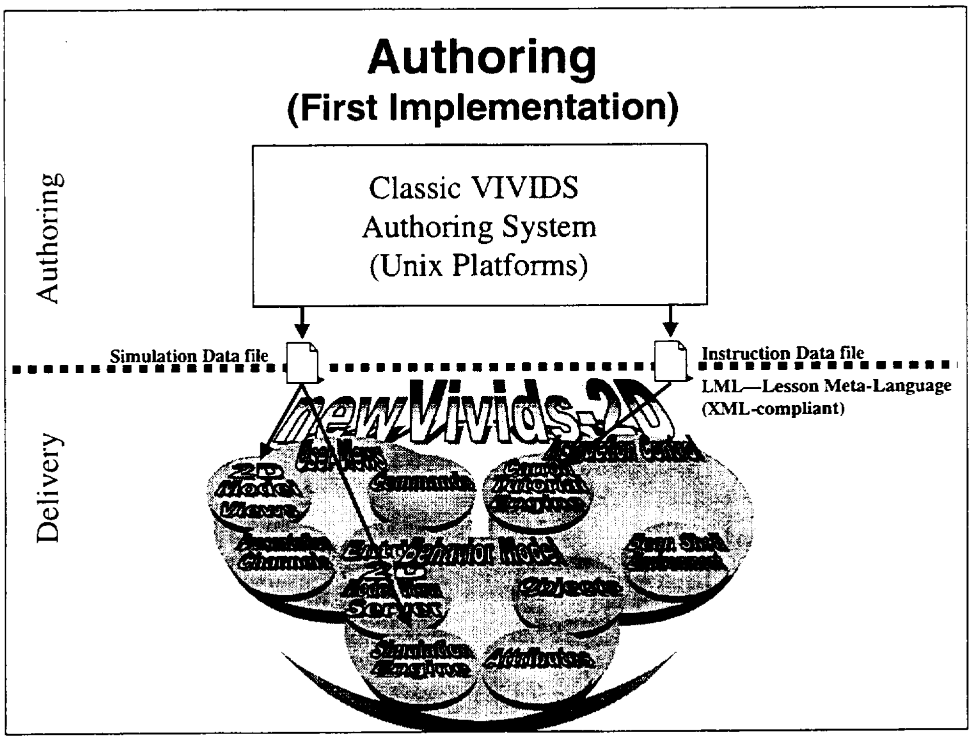
newVivids-2D

The combination of the components being developed for these two projects results in a complete system for the delivery of authored tutorials in the context of authored two-dimensional graphical simulations.



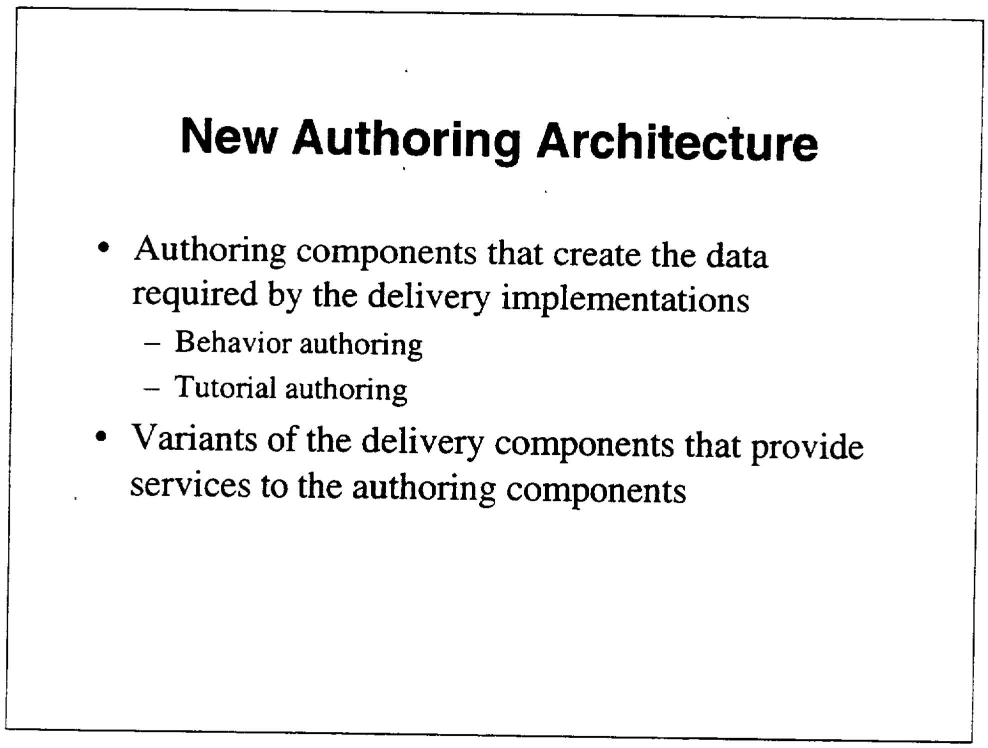
Authoring for newVivids-2D

The approach that is being used to author the simulations and the tutorials that are delivered by the newVivids-2D delivery system is to use a modified version of the existing classic VIVIDS authoring application, which runs on Unix platforms. Authors can save simulation data and tutorial specification data in new file formats that can be read and utilized by the newVivids-2D components.



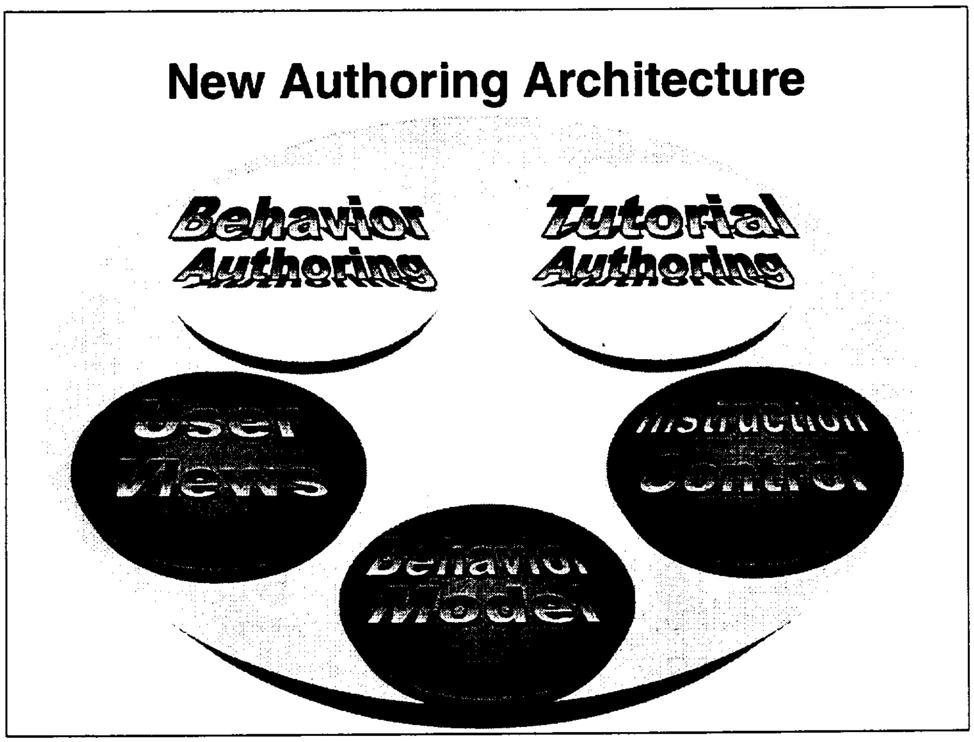
Next: A New Authoring Architecture

We believe that the use of modified classic VIVIDS for authoring *newVivids* simulations and tutorials is only an interim solution to the problem of authoring for tutors based on the new architecture. Authors would like to author on the same platforms on which tutorials will be delivered. Furthermore, both the simulation and the tutorial components of the newVivids architecture offer powerful features that are not available in classic VIVIDS. To exploit these features fully, a new authoring architecture is called for.



A New Authoring Architecture

Implementations of the new authoring architecture will consist of two new authoring components that work with versions of the delivery components already discussed in this presentation. These new authoring modules will work with any architecture-compliant user views, behavior model, and instructional control system components.



Advantages of Authoring Components

The approach outlined here offers a number of advantages over the monolithic approach utilized in our earlier projects. First, it encourages the use of right-sized components that permit lightweight tutorial delivery applications. Second, it encourages the development of authoring tools that can be used in a variety of disparate simulation and training contexts. These include authoring simulation behavior for both two- and three-dimensional simulations, and the authoring of tutorials that can collaborate with hard-coded, non-authored simulations as well as with authored ones. In addition, new types of tutorial components and tutorial authoring tools can be developed that can make use of the defined services of the other major components of the architecture.

For further information on RIDES, VIVIDS, and the new architecture described here, please visit our web site: http://btl.usc.edu/.

Advantages

of the Authoring Components Approach

- Non-monolithic tutor development systems
- Reuse of authoring components in different environments
  - Same interfaces for authoring in both 2D and VE simulations
  - New tutor authoring tools can be developed (e.g., pedagogical-agent-authoring) that will work in a variety of simulation environments.
- Authoring on more platforms

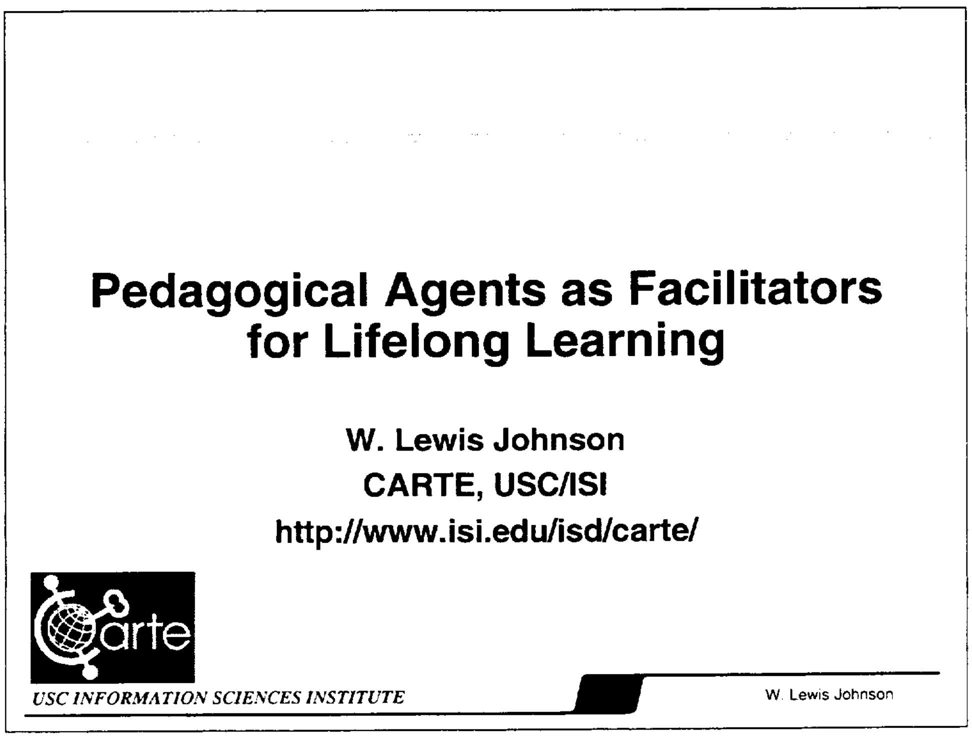
Pedagogical Agents as Facilitators for Lifelong Learning

W. Lewis Johnson
Director, Center for Advanced Research in Technology for Education
University of Southern California
Marina del Ray, CA

Pedagogical Agents as Facilitators for Lifelong Learning

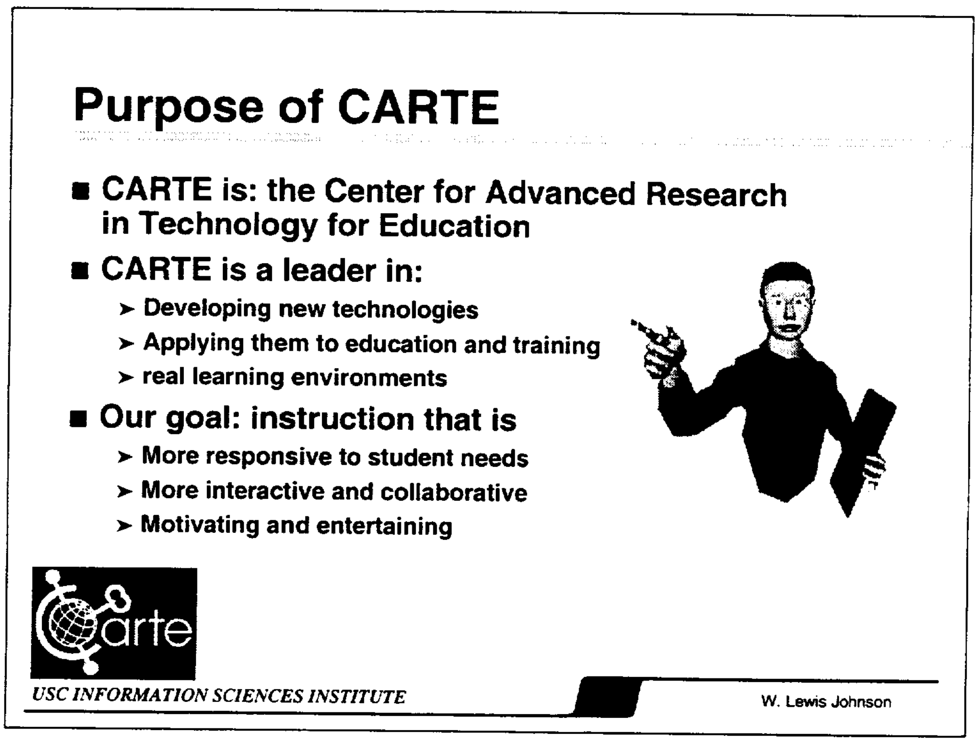
W. Lewis Johnson
Center for Advanced Research in Technology for Education
University of Southern California
Marina del Rey, CA

This talk was presented at the Workshop on Advanced Training Technologies and Learning Environments, NASA Langley Research Center, Hampton, VA, March 9-10, 1999. It presents some work on pedagogical agents as facilitators for lifelong learning at USC's Center for Advanced Research in Technology Education (CARTE), and explains how such agents can help make learning environments more effective at fostering lifelong learning.



Purpose of CARTE

CARTE is engaged in the development of new learning technologies and their application in the field. CARTE is involved in projects aimed at exploring new technologies such as autonomous agents and virtual environments, and developing methods for applying them to education and training. It also collaborates with educators to integrate technology into curricula. CARTE is an interdisciplinary team including computer scientists, educational researchers, and multimedia specialists.



Wanted: Lifelong Learners

In our modern knowledge-oriented economy there is an increasing need for workers who can continually improve their knowledge and skills. Such workers need to be able to learn what they need when they need it. At the same time, learning must make efficient use of time and money. Workers should not have to spend long periods away from work in order to attend classes, and they should not require constant supervision from instructors. As noted in the introductory briefing, we are witnessing a confluence of training and education concerns, and this applies to lifelong learning as well. In particular, the new skills that workers acquire must be tied to new knowledge, e.g., learners need to understand the rationales underlying recommended procedures. This helps to ensure that workers will be able to apply their skills adaptively to changing situations. Also, it is important that workers understand how to apply new skills to real work contexts. For example, workers frequently must work in teams, but in typical computer-based training, learners work individually. Finally, lifelong learners must have the necessary metacognitive skills to learn effectively, e.g., they need to learn to ask questions and reflect on the skills that they have learned.

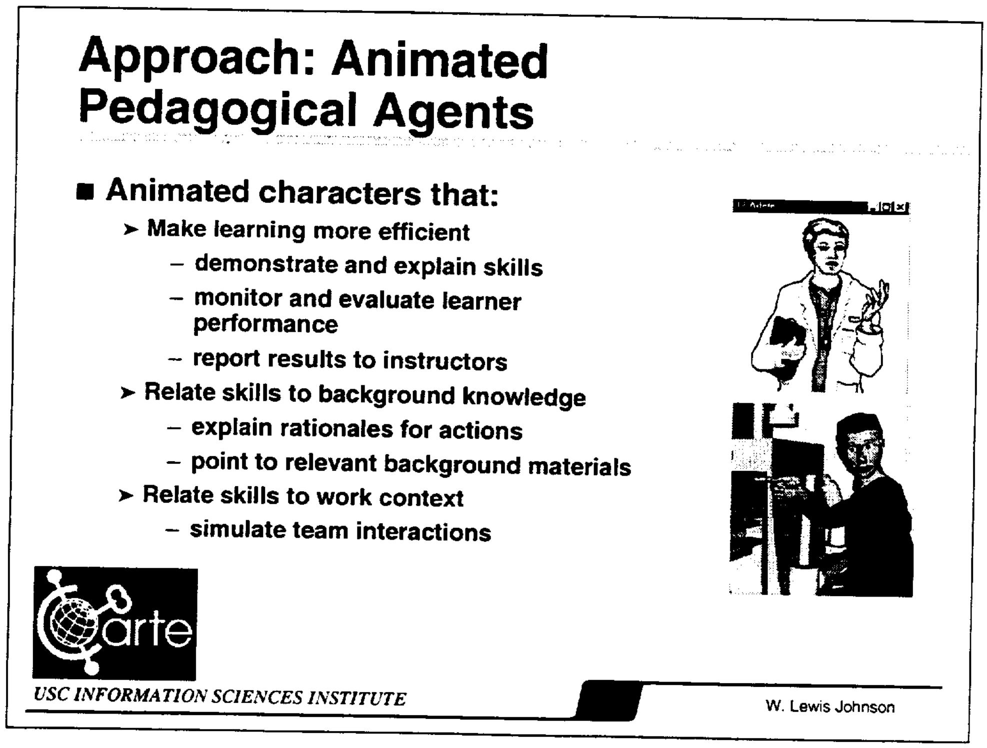
Wanted: Lifelong Learners

- **Workers must continually improve their knowledge and skills**
  - They must learn what they need when they need it
  - Learning must be efficient
    - without lengthy absences from work
    - without constant support from instructors
  - New knowledge must be tied to new skills
    - e.g., rationales underlying recommended procedures
  - Skills must be tied to work context
    - e.g., applying skills in teams
- **Lifelong learners must know how to learn**

USC INFORMATION SCIENCES INSTITUTE W. Lewis Johnson

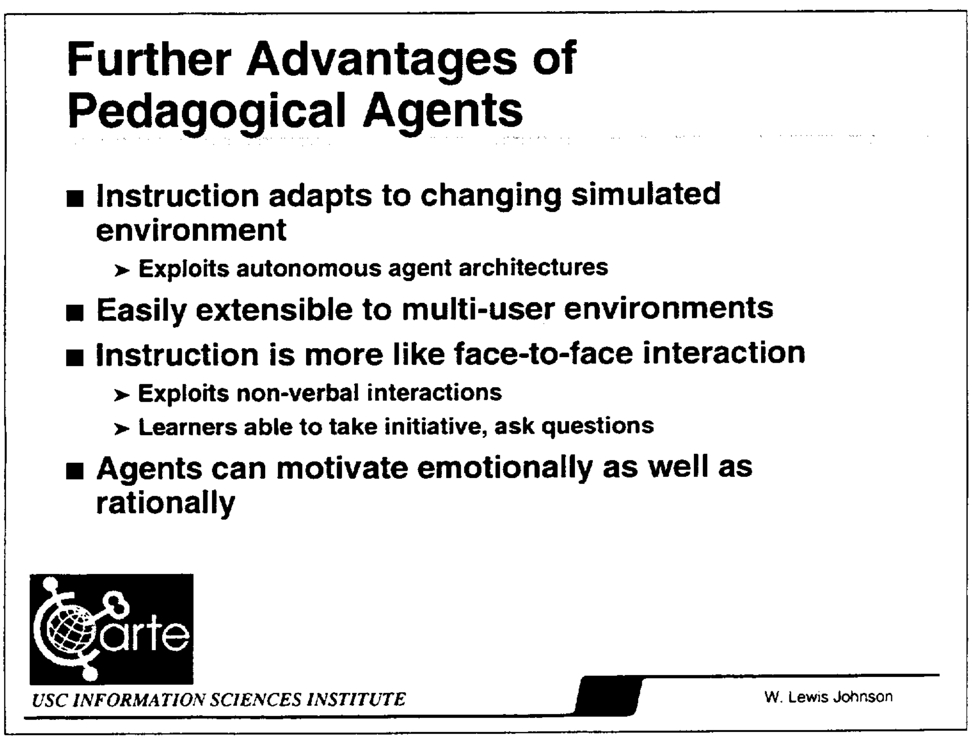
Approach: Animated Pedagogical Agents

Animated pedagogical agents can help support effective lifelong learning. Animated pedagogical agents are animated characters that are integrated into computer-based learning environments in order to facilitate learning. They can help make learning more efficient by interacting with the learners. Agents have been developed that can demonstrate and explain skills on request, and monitor and evaluate learner performance. They can also report on learner performance to instructors who are not physically present. They, thus, can act both as a learner's assistant and as an instructor's assistant. Agents help relate skills to be background knowledge by explaining the rationales for recommended actions and by pointing the learner to relevant background materials. They can simulate team interactions, allowing learners to practice skills in situations that more authentically model real work practice.



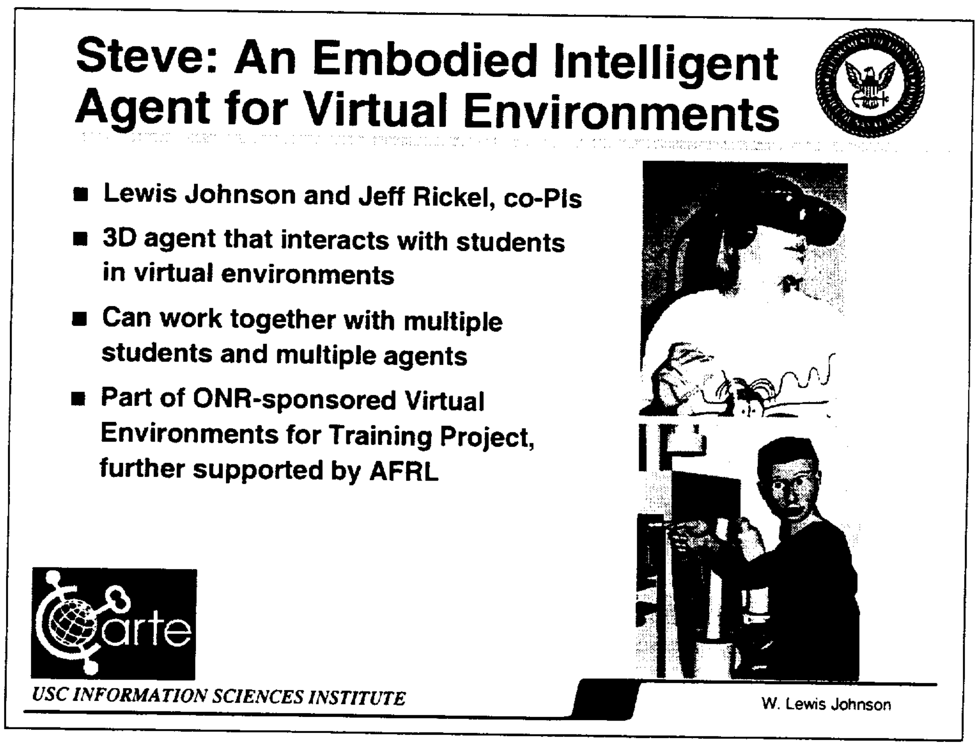
Further Advantages of Pedagogical Agents

Animated pedagogical agents offer a number of additional advantages. They can adapt instruction to the changing simulation environment. New advances in autonomous agent architectures help to make this possible. It is relatively straightforward to extend agents to operate in multi-user and multi-agent settings. Typical agent-assisted learning environments are designed as multi-agent systems with a minimum of two agents - one human learner and one animated agent. It is easy to extend such environments to support more than two agents, although one must then make sure that the additional agents can coordinate their activities effectively. Interaction with animated agents is more like face-to-face interaction. Agents can exploit non-verbal cues, such as direction of gaze, that are not available in menu-based or text-based interfaces. Learners are free to take the initiative in the face-to-face dialog. Finally, agents can motivate learners emotionally as well as rationally. Emotion plays an important role in learning - any football coach or drill sergeant knows this and takes advantage of it. In order for an animated agent to appear lifelike it must be capable of expressing emotion as well via facial expressions and body posture. We are just beginning to understand how to exploit such emotional expression to foster learning.



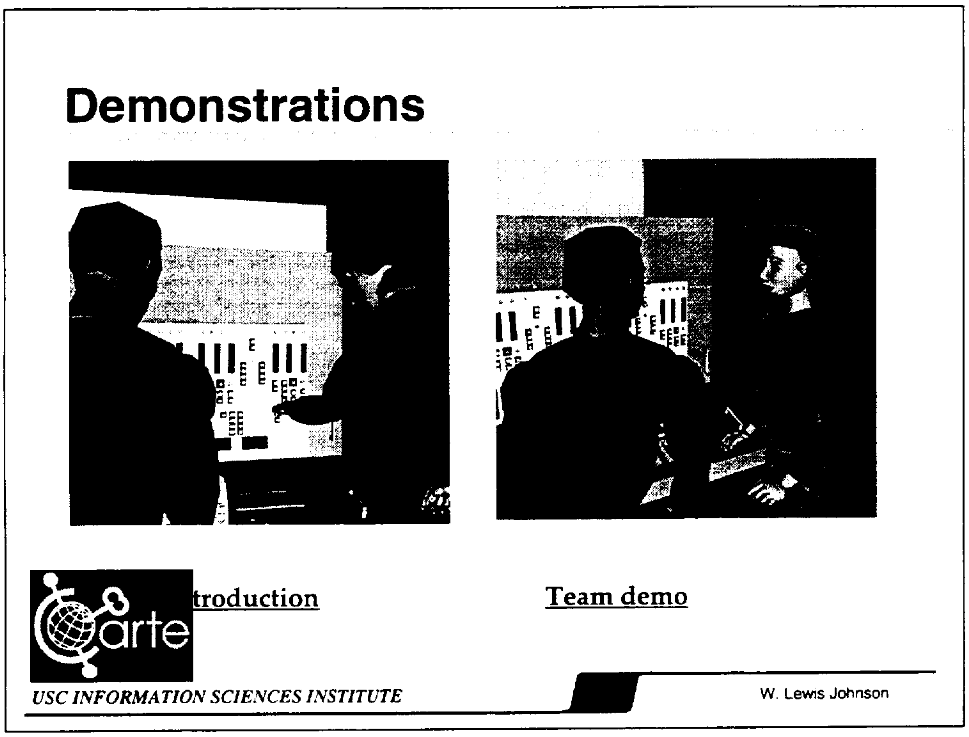
Steve: An Embodied Intelligent Agent for Virtual Environments

The first agent that I will talk about is Steve (Soar Training Expert for Virtual Environments), developed by a team headed by Jeff Rickel and myself. Steve is a three-dimensional animated agent that interacts with students in immersive virtual environments. Steve can work with individual students or with combinations of multiple students and multiple agents. Steve was developed as part of the ONR-sponsored Virtual Environments for Training project, a collaboration between Lockheed Martin (Randy Stiles, PI), USC Behavioral Technology Laboratories (Allen Munro, PI), and USC CARTE. Further development is funded by ONR and the Air Force Research Laboratory.



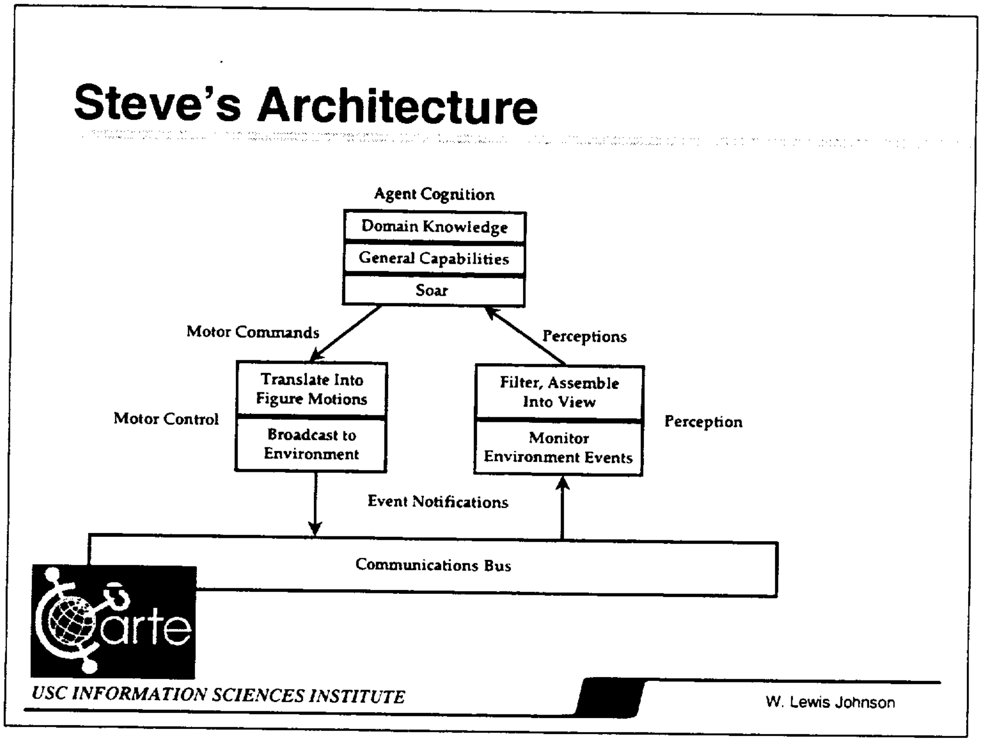
Demonstrations

The presentation includes two video demonstrations, one of Steve, agent training an individual learner, and one showing a learner working in a team with multiple Steve agents. Click on the underlined text to play the videos or access them on the world wide web at http://www.isi.edu/isd/VET/steve-demo.html



Steve's Architecture

Steve, the Soar Training Expert for Virtual Environments, is built on top of the Soar cognitive architecture developed at USC/Information Sciences Institute, Carnegie Mellon University and the University of Michigan. Steve consists of three main components, responsible for perception, cognition, and motor control. The perception module monitors events that occur in the simulation environment, and creates a representation of the current state of the environment. The cognition component continually monitors the state of the environment, and decides what actions to perform. These actions are passed as motor commands to the motor control component, which translates the motor commands into actions performed by Steve's virtual body. A number of general-purpose instructionally-relevant capabilities were built in Soar for Steve to use as needed. In order to apply Steve to a particular skill, one provides Steve with the domain knowledge necessary to perform the skill, and the instructional capabilities apply that knowledge as needed in the instructional interactions.



General Capabilities

The following general capabilities were provided for Steve. An adaptive plan execution capability enables Steve to adapt plans to changing circumstances. Steve uses a path planning function to plan routes between points in the virtual environment as he moves around and performs tasks. Support for collaborative, mixed initiative dialog has been developed. Steve has a student monitoring capability that enables him to interpret student actions and relate them to known procedures. A question answering capability enables Steve to answer a learner's questions about the task being trained. An episodic memory component makes it possible for Steve to answer questions about actions that he has previously demonstrated. Finally, a set of human figure behaviors were created together with control mechanisms for activating and interleaving them.

General Capabilities

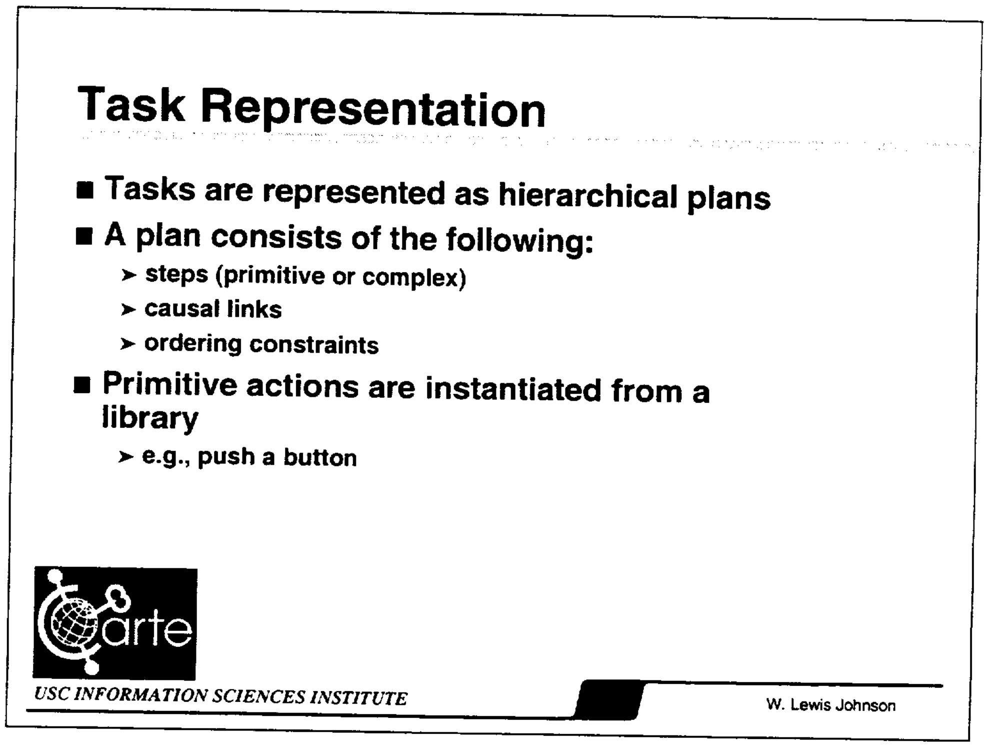
- Planning, replanning, and plan execution
- Path planning
- Collaborative, mixed initiative dialogue
- Student monitoring
- Question answering
- Episodic memory
- Control of a human figure

carte

USC INFORMATION SCIENCES INSTITUTE W. Lewis Johnson

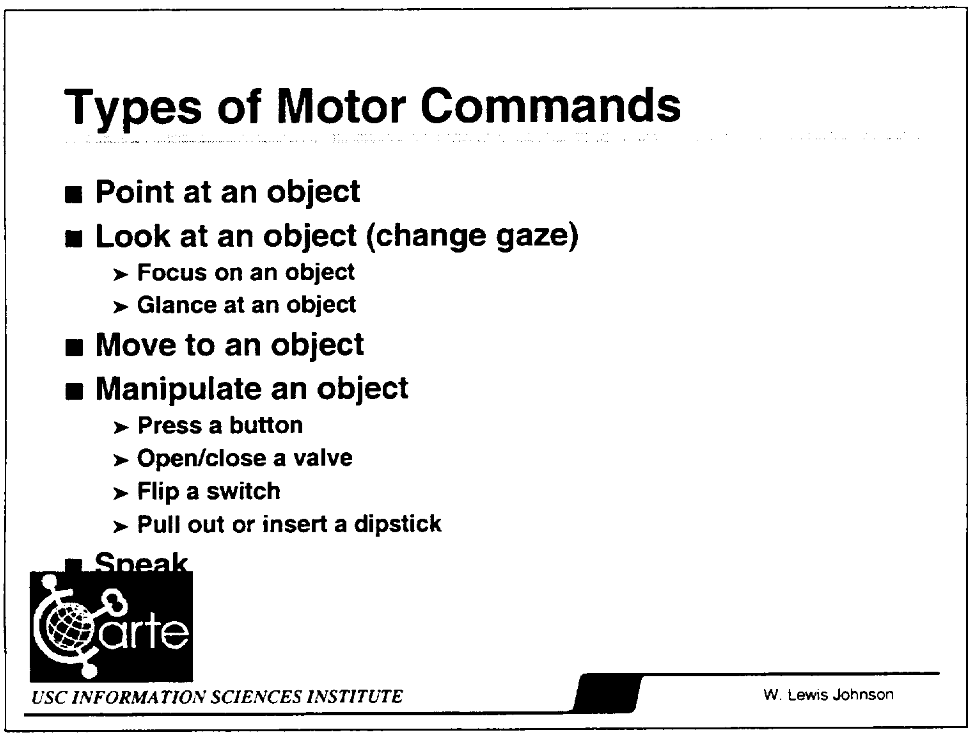
Task Representation

Task knowledge is represented in Steve in the form of hierarchical plans. A plan consists of a set of steps, each of which may be either a primitive action or a complex step with substeps. Causal links and ordering constraints may be specified between steps in the plan. Causal links express the relationships among steps and the goals that they accomplish, e.g., action A achieves condition C that enables action B to be performed. These links and ordering constraints are used by Steve to determine which steps are relevant in the current situation and to explain the rationales for actions. Primitive actions are instantiated from a library of action types.



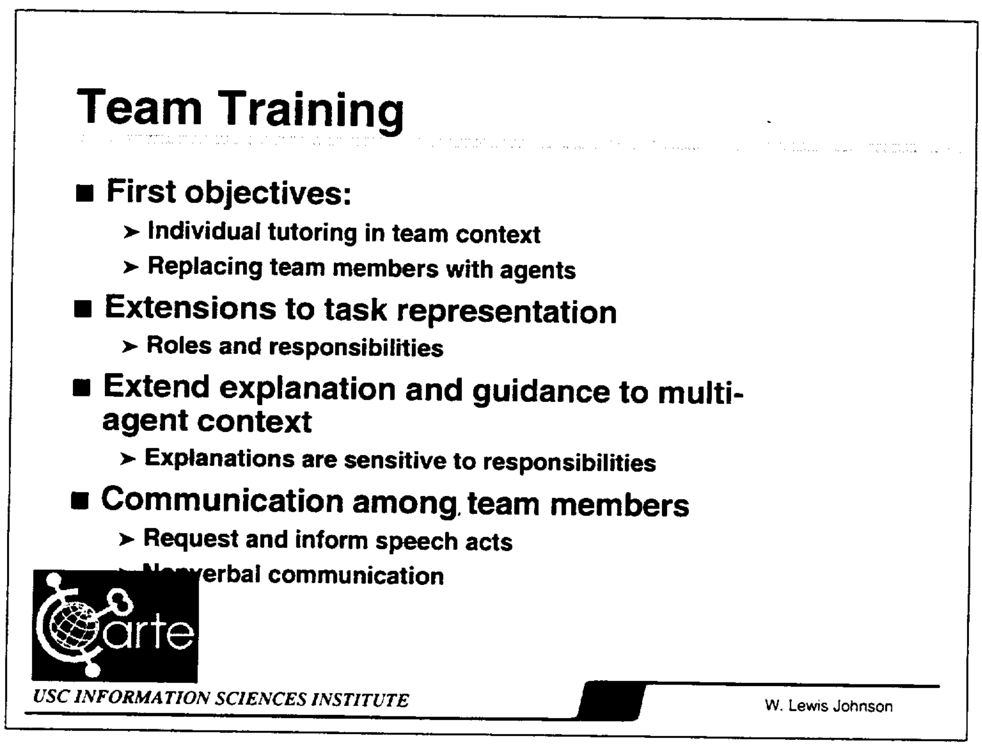
Types of Motor Commands

The following are the types of primitive actions that Steve can perform, each of which is initiated by a different motor command. Steve can point to an object, moving his extended finger toward the object while positioning his upper body so that the student can see where he is pointing. Steve can glance at an object, briefly turning his eyes and head toward the object, or focus on the object, which involves orienting the body as well. Steve can move to an object, which may involve planning a path to arrive at the object. Steve can manipulate objects in different ways depending upon the object, e.g., he can press buttons, open and close valves, flip switches, and pull out or insert a dipstick. Speaking either to a learner or to another agent is also a first-class primitive action.



Team Training

The task representation described above was then extended in order to support team training. Our initial objectives in the team training area was to emulate the kind of training scenarios that we observed at the U.S. Navy's Great Lakes Training Center, where teams of trainees are supervised by teams of instructors (one instructor for each trainee). Each Steve tutor serves as the personal instructor for an individual trainee within a team scenario. In addition, Steve was extended to perform the roles of missing team members. To support team skills, Steve's task representation was extended to define roles and responsibilities. When a given team scenario is conducted, each trainee or agent is assigned the responsibility to assume some role within the team task. Explanation, demonstration, and monitoring capabilities were then extended to take the roles and responsibilities into account. Each Steve agent coordinates his actions with the other team members, and is able to explain how his actions affect the activities of others. Finally, both verbal and nonverbal communication needed to be extended. Speech utterances are represented explicitly as Request or Inform speech acts, to which other team members can respond. Agents also use nonverbal communication, such as gaze and head nods, to make their coordination explicit.



Authoring by Demonstration

Steve requires a new kind of instructional authoring, focusing on skills to be taught instead of information to be conveyed. One of our students, Richard Angros, has developed an extension to Steve called Diligent that enables nonprogrammers to instruct Steve through the use of demonstration examples. The author enters the virtual environment and demonstrates how the task is performed. Diligent records the sequence of actions performed by the author, and notes the changes to the virtual environment that result. At the end of the demonstration Diligent presents to the author a list of the simulation attributes that were affected by the demonstration, and asks the author to select the subset of state changes where the intended result of the procedure. Diligent then conducts a series of experiments, modifying the demonstration procedure in various ways, to see whether or not they have an impact on the result of the procedure. Through this process Diligent is able to discover causal links between steps, and is able to generalize from the example demonstration. The resulting generalization is presented back to the author for validation. Empirical studies have shown that this approach enables authors quickly to construct accurate task models for use by Steve in instruction.

Authoring by Demonstration (Angros)

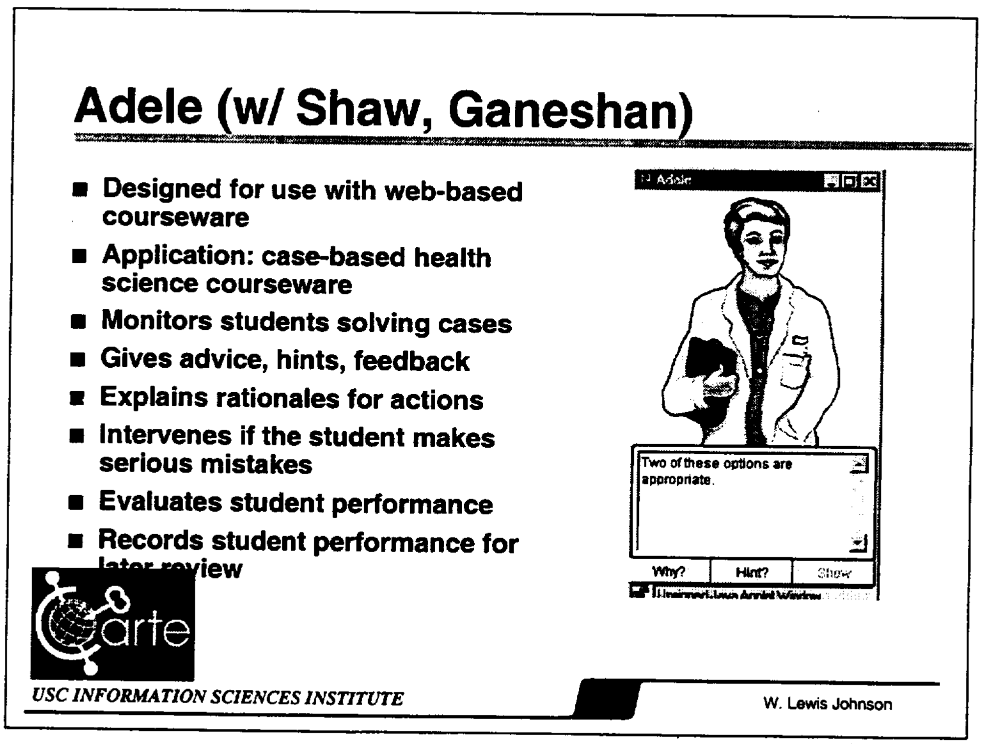
- **Objective: enable nonprogrammers to define new tasks for Steve to train**
- **Author demonstrates task**
  - Perform actions
  - Describe actions
- **Agent experiments with task**
  - Try variants to uncover dependencies
- **Agent discusses task with author**
  - Verify inductive hypotheses

carte

USC INFORMATION SCIENCES INSTITUTE W. Lewis Johnson

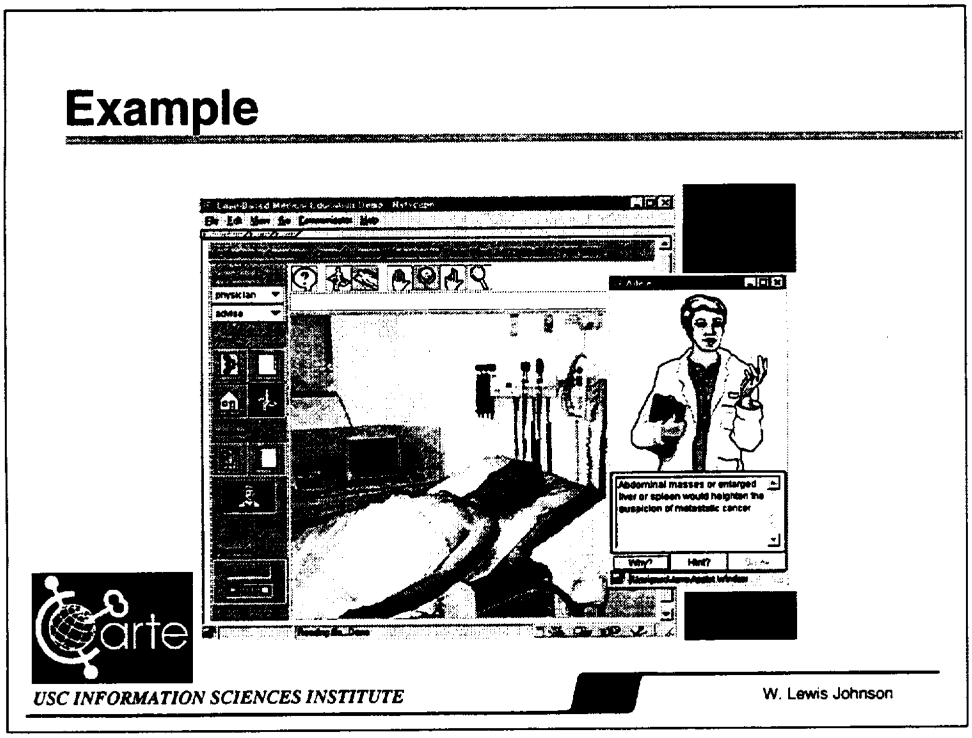
Adele (with Shaw, Ganeshan)

Although interaction between people and animated pedagogical agents occurs most naturally in immersive virtual environments, it is not yet practical to use virtual environment technology to train all subjects. Furthermore, there is an ever increasing amount of learning material that is available outside of virtual environments, e.g., in hypertext on the World Wide Web. We, therefore, developed Adele (Agent for Distance Education - Light Edition) to support agent-assisted learning over the Web. The Adele architecture is designed for use in any application area where students can learn through the process of trying to solve problems. Our initial area of focus has been the health sciences - clinical decision making, dentistry, and emergency trauma care. Adele monitors students as they solve cases, and gives advice, hints, and feedback. She can explain the rationales for her recommendation. She intervenes if the student makes serious mistakes, and performs a continual evaluation of student performance. The record and evaluation of student performance is transmitted automatically to human instructors for subsequent review.



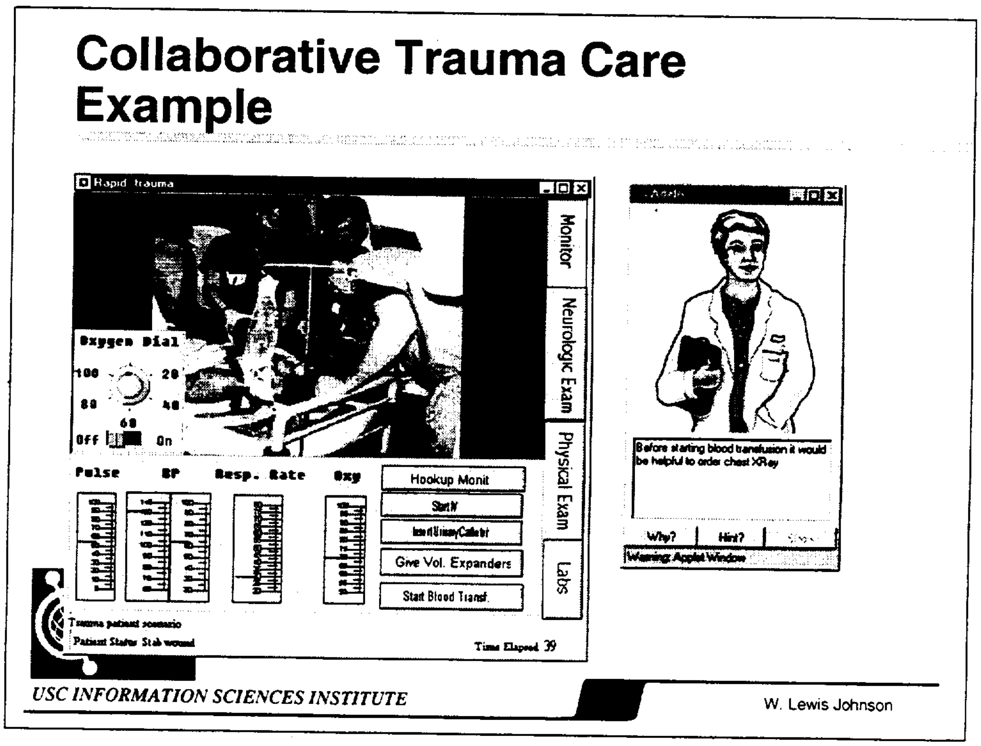
Adele Demonstration

At this point in the presentation I showed a live demonstration of Adele assisting in a clinical decision making scenario.



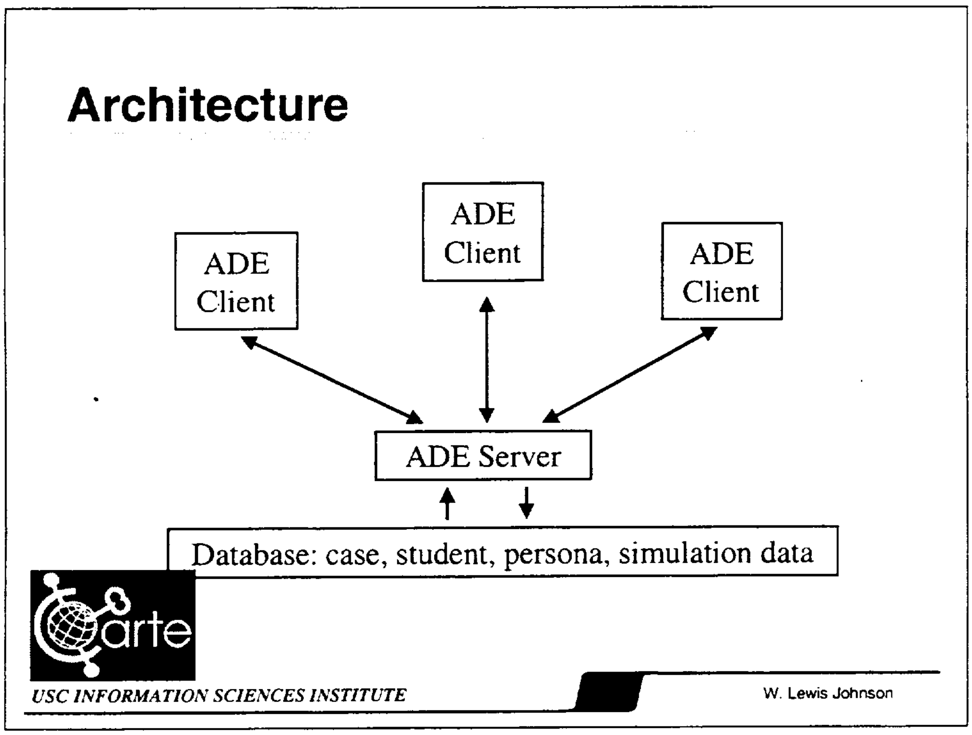
Collaborative Trauma Care Example

These shots were taken from another Adele application, in the area of trauma care. Students play the role of trauma care specialists in an emergency room. There are two important points to note in this example. First, the trauma simulation was authored using an off-the-shelf simulation authoring tool, called Rapid. This demonstrates the ability of Adele to interface to externally authored simulations. Second, this is a networked application in which multiple students working at different workstations can collaborate. Each student has his own version of Adele commenting on his activities. The agent-based architecture made it easy to extend the application from single-user to multi-user delivery.



Architecture

Adele is part of a distributed learning architecture called ADE (Advanced Distance Education) developed at CARTE. Each user has his own ADE client that runs on the local computer in conjunction with a web browser. The Adele persona runs as part of the ADE client in order to enable rapid response to the student's actions. The central ADE server downloads client software if needed as well as data on each individual case. It uploads student performance data and stores it in a central database.



Case-Based Approach

Adele instruction is authored using a case-based approach. Subject matter knowledge and background information is provided specifically for each case. The advantage of this approach is that the information needed for a given case is concrete and relatively easy for instruction authors to understand and create. Furthermore, the amount of data required for each case is limited. Instead of providing Adele with a complete knowledge base of medical information, for example, we provide just the information needed to help a student work through a given case. This makes it possible to download cases to client computers on an as-needed basis.

Case-Based Approach

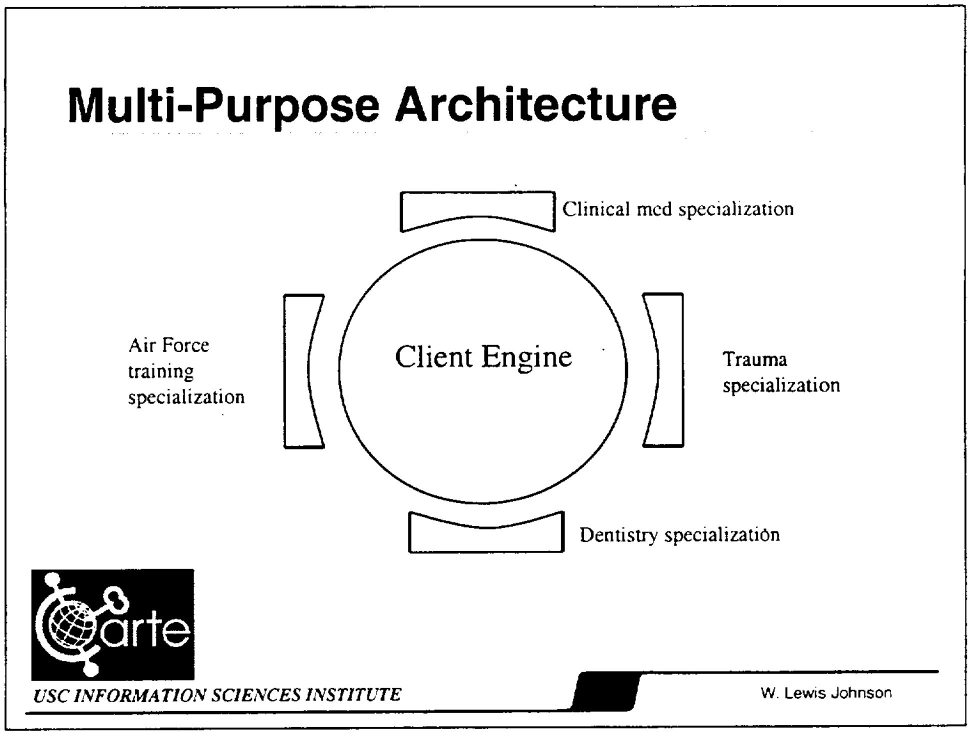
- **Background knowledge is provided specifically for each case**
- **Advantages:**
  - **Information is concrete, easy for instructors to understand**
  - **Amount of data required for each case is limited**
  - **Agent software can run on client computer**

carte

USC INFORMATION SCIENCES INSTITUTE W. Lewis Johnson

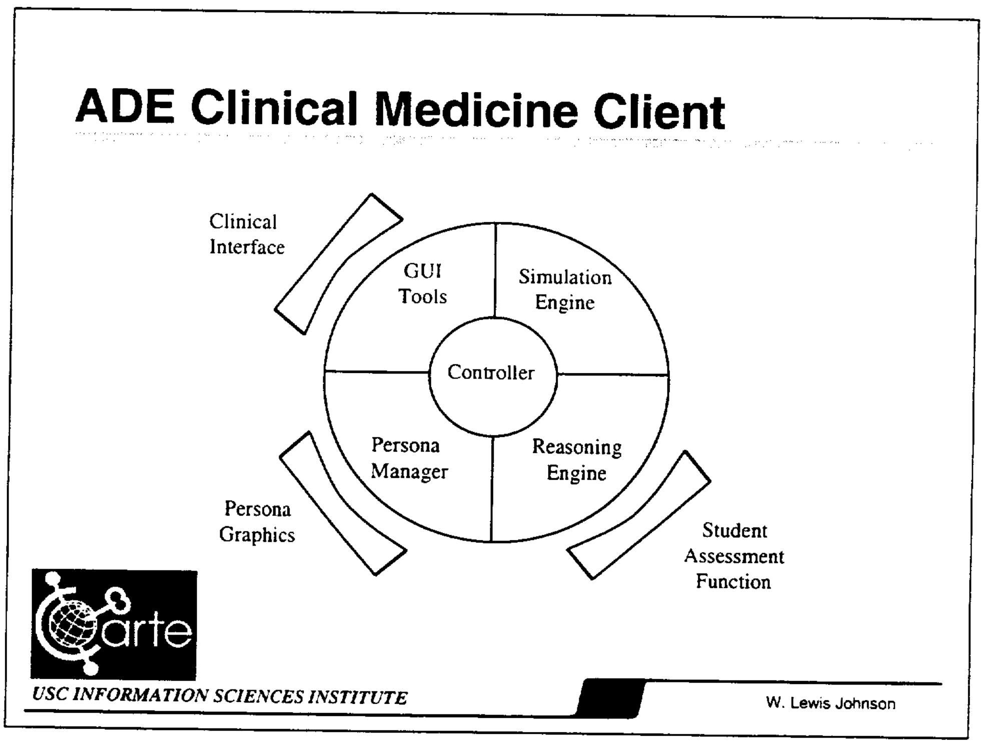
Multi-Purpose Architecture

The ADE architecture is a multi-purpose architecture. Individual applications are created in part by adding specialized extensions for individual subject matter areas. In addition to the three health science specializations that we have created, we are developing a specialization for Air Force training in the area of time-critical targeting.



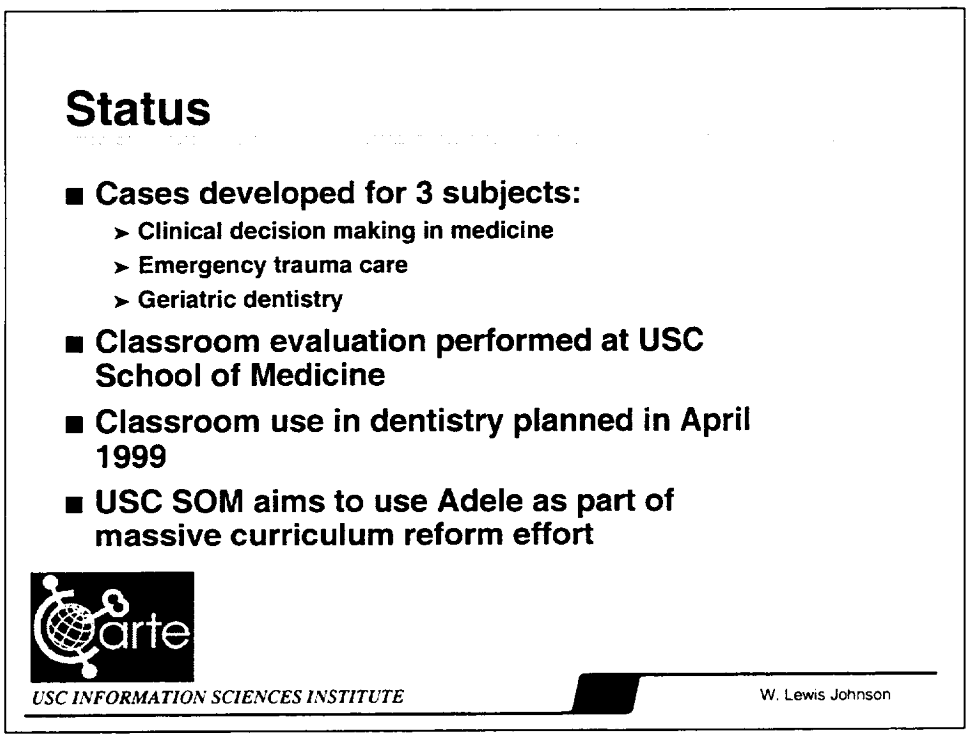
ADE Clinical Medicine Client

This diagram shows the components of the ADE client architecture in more detail as applied to clinical medicine. The main client engine is divided into the following components: a controller that coordinates the other modules, a simulation engine that simulates the state of the patient, GUI tools that are used to create the user interface for interaction with the simulation, a reasoning engine that enables Adele to monitor the student's actions, and a persona manager that controls the animations of Adele's humanlike persona. Each component is domain-independent with the exception of the reasoning engine, which is specialized to diagnostic reasoning tasks. Then specific modules are added for clinical medicine: a clinical medicine user interface, a student assessment function for evaluating the student's performance in comparison to best practice, and the particular graphical images used to give Adele the appearance of an attending physician.



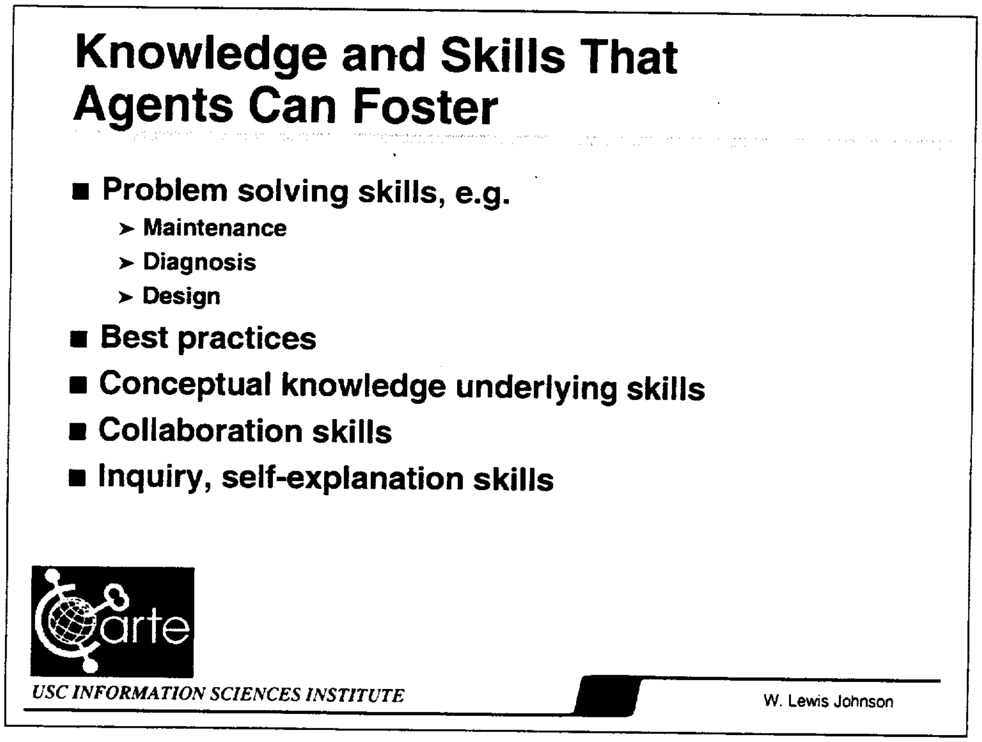
Status

As indicated above, cases have been developed for Adele in three health science subjects. A preliminary field trial was performed last Fall at the USC School of Medicine. A class of fifty second-year medical students worked through a medical case on line, assisted by Adele. They found Adele's assistance to be helpful and preferred an animated persona to text-based instruction. We have recently released an authoring tool for creating Adele cases, which is being used by the USC School of Dentistry to create teaching materials for geriatric dentistry. The USC School of Medicine has recently funded the development of additional curriculum modules for second-year and third-year medical education, and has approved additional field trials. If successful, Adele will be incorporated into a comprehensive curriculum reform effort being conducted at the USC School of Medicine. This will be an exciting opportunity for advanced technology to have a significant impact on education.



Knowledge and Skills That Agents Can Foster

In summary, pedagogical agents are of significant value in fostering lifelong learning. They can help foster a variety of problem solving skills, such as maintenance, diagnosis, or design. They help acquaint learners with best problem solving practices by means of critiques of their problem solving strategies. Agents help learners to acquire the conceptual knowledge underlying skills. By supporting team activities, agents can help foster collaboration skills. Finally, by engaging students in question-answer dialogs agents may help learners to develop the inquiry and self-explanation skills needed for successful lifelong learning.



Generating and Delivering Diagnostic Instruction

Douglas M. Towne
Behavioral Technology Laboratories
University of Southern California
St. Helena, CA

Generating and Delivering Diagnostic Instruction*

Douglas M. Towne
Behavioral Technology Laboratories
University of Southern California

Learning to diagnose faults in a complex system requires extensive troubleshooting practice on that system. The time and cost to gain any given level of proficiency can be dramatically reduced if practice is provided in a simulation environment along with individualized guidance. This presentation will describe an extended research and development program conducted to 1) facilitate the production of realistic system simulations, and 2) automatically generate intelligent troubleshooting tutoring from the simulation.

*This work was funded by the Office of Naval Research. Susan Chipman served as Scientific Officer for the IMTS and DIAG development programs; Gerald Malecki was Scientific Officer for the Profile Diagnostic Model work.

Generating and Delivering
Diagnostic Instruction

Douglas M. Towne
Behavioral Technology Laboratories
University of Southern California

Workshop on Advanced Training Technologies and Learning Environments

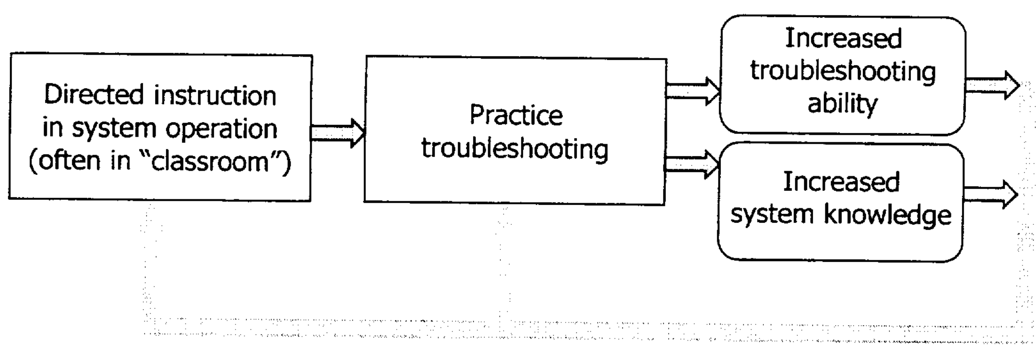
March 9-10, 1999

Acquiring Expertise in Fault Diagnosis

Ideally, troubleshooting practice is interspersed with opportunities to acquire further knowledge of the system's operation and design. This practice phase not only allows the individual to *apply* knowledge acquired through this and other means, it also enriches the troubleshooter's understanding of the system and may uncover misconceptions as well. Too often in military training, the fault diagnosis practice is provided *en masse* following a sustained period of conventional classroom instruction. This denies the learner the chance to discuss questions and discoveries evoked by the practice phase with classmates and instructors. Furthermore, it reduces the effectiveness of the practice time, as application of knowledge is far removed in time from the acquisition of that knowledge.

Acquiring Expertise in Fault Diagnosis

Following a necessary introduction, proficiency requires iteratively learning about system functionality and applying that knowledge to practice problems.



Twenty Year Overview

The capability to automatically generate intelligent tutoring in fault diagnosis stems from over two decades of research. The early work studied how experts conduct fault diagnosis and how novices differ from experts (PROFILE). Following this was an initial program to generate tutoring interactions automatically within a classical intelligent tutoring system (IMTS). Finally, we have applied the lessons learned in the IMTS work, arriving at an effective system for producing system simulations and generating intelligent tutoring (DIAG).

Twenty Years in Twenty Minutes

- PROFILE
  - a model of diagnostic reasoning
- IMTS - Intelligent Maintenance Training System
  - a classical ITS with detailed student model, etc.
- DIAG - Diagnostic Instruction and Guidance
  - an expert advisor that supports practice

Profile Development

The original motivation for modeling diagnostic behavior was to produce a system that could automatically generate repair time distributions from system designs [1]. This required an automated diagnostician that would go through the steps of troubleshooting many hundreds of simulated faults in a proposed system.

The experimentation involved observation and documentation of many hundreds of diagnostic exercises performed by both experts and novices. The first study, involving 384 exercises, formed the basis for the PROFILE model, which closely reproduced the troubleshooting sequences of 48 Navy instructors. The second study of 174 exercises permitted controlled variation in such variables as problem complexity and symptom knowledge.

The final study tested the validity and robustness of the PROFILE model using a different technician population and target system.

PROFILE Development

Initially developed as a tool for predicting maintainability from system designs.

Derived and validated with studies of 638 diagnostic problems.

- 48 Navy electronics instructors; 8 computer faults
- 29 college students; 6 logical network faults
- 10 Navy instructors; 8 transmitter/receiver faults

Profile Validation Study Findings

After achieving an excellent model of expert diagnostic decision making, we turned to the issue of novice performance. Virtually every aspect of the decision making process was systematically degraded in an attempt to generate diagnostic sequences that resembled those observed. These included less effective use of symptom information, less effective updating of suspicions, and less effective choice of next tests. All of these attempts to emulate novice performance were unsuccessful.

Finally, we degraded only the underlying knowledge of system behavior, as represented by fault effect information, leaving the (expert) fault diagnosis strategy unchanged. This produced diagnostic sequences that closely resembled those of novices.

Profile Validation Study Findings

PROFILE strategies closely approximated those of

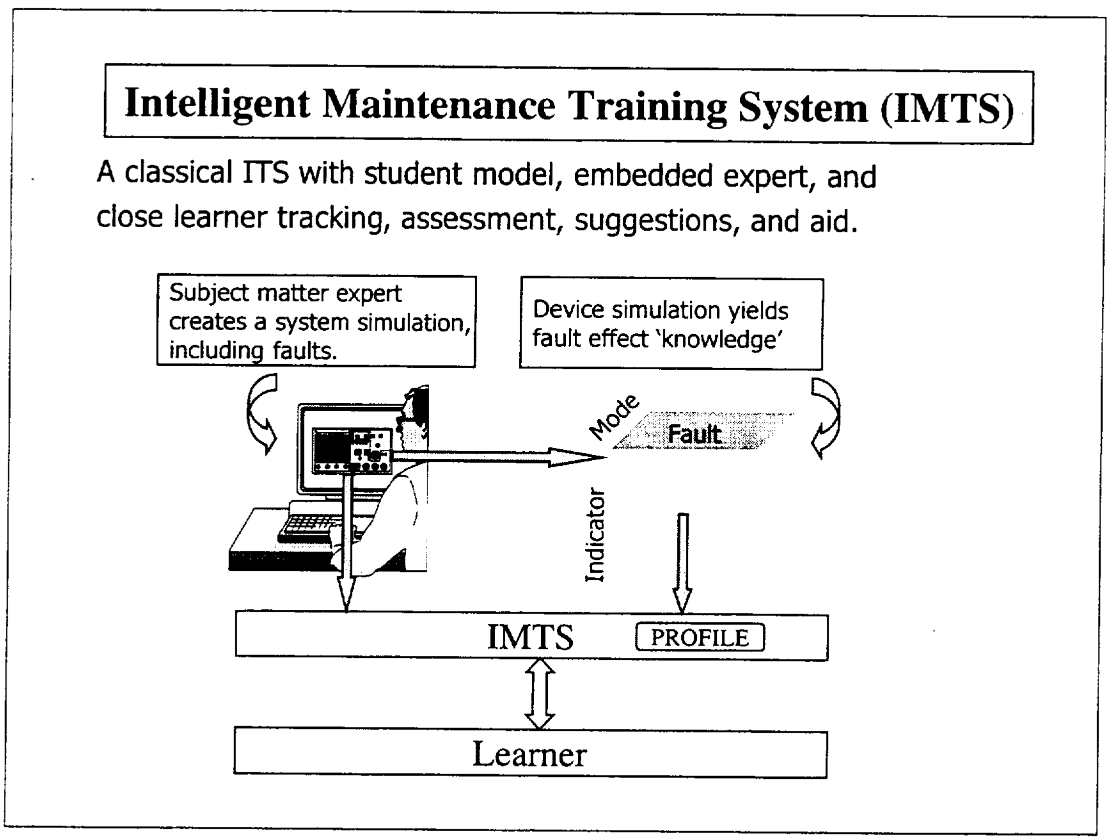
- *experts*, if PROFILE used *excellent* fault effect information.
- *novices*, if PROFILE used *weak and incomplete* fault effect information.

So, we infer that the major difference between novices and experts is NOT the quality of their diagnostic strategies, but rather the quality of their system knowledge.

Intelligent Maintenance Training System (IMTS)

Embedding the PROFILE model within a tutoring system yielded a system that could converse intelligently with a learner about his or her choice of tests, interpretation of test results, and maintenance of suspicions. This system, IMTS [2, 3], maintained a model of each learner using an overlay structure representing the conceptual elements and organization of the target system. IMTS selected problems adaptively, compared the learner's work to that of an expert (PROFILE), interjected suggestions when it felt they were necessary, and changed the course of instruction when evidence suggested the learner was encountering excessive difficulty.

All of this domain-specific processing was supported by a large and detailed array of fault effect data, generated automatically by systematically inserting faults into the simulation of the target system and recording the symptoms that resulted.



IMTS Findings

The adaptive problem selection process chose faults according to the learner's demonstrated proficiency. This process produced a reasonable sequence for individuals, but was not rigorously tested. The simulation environment was found to be highly effective, engaging, and cost effective, and the generated diagnostic advice was valid and apparently appropriate.

The less positive findings were that: 1) the assessment of individual proficiency was inaccurate if the learner did a great deal of exploration and confirmation of theories; 2) the simulation construction process required high expertise; and 3) the diagnostic advice was perfectly valid only if virtually every possible fault was simulated previously, and thus represented in the knowledge base.

IMTS Findings

- Adaptive problem selection
  - reasonable selections, but never rigorously tested
- Simulation environment for practice
  - very powerful; learners highly engaged; realism judged excellent
- Response to requests for help
  - excellent responses and demonstrations generated
- Unsolicited offers for help
  - good if learner does no exploring (or, NO LEARNING ALLOWED)
- Proficiency assessment
  - excellent in test mode; poor in learning mode
- Cost effectiveness of tutor development
  - good, but 1) faults were limited to those specifically simulated and 2) simulation authoring tools were limited

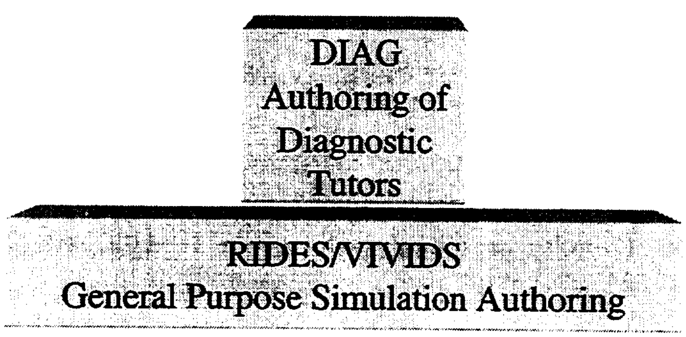
DIAG: Diagnostic Instruction and Guidance

The final research program, DIAG [6, 7], was designed and implemented to address the findings of the IMTS work. While sharing the embedded diagnostic advisor and fault effect generator of IMTS, DIAG was built upon the far more usable simulation authoring tools of RIDES [4, 5], and it emphasized cooperative guidance and tutoring in response to student requests rather than micro-analysis of the individual's diagnostic strategy. Consequently, DIAG encourages exploration and theory testing rather than interpreting these as signs of difficulty, and it supports the learner in pursuing quests for understanding.

Finally, DIAG employs a *qualitative* symptom knowledge base, as outlined below. This yields a level of robustness that will permit it to function as a diagnostic field aid, in addition to a tutor.

DIAG: Diagnostic Instruction and Guidance

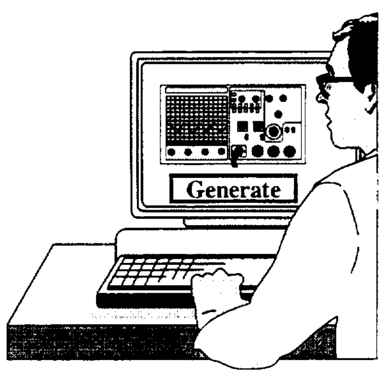
- A high level authoring system
  - for producing intelligent troubleshooting tutors
- An intelligent learning support system
  - for guiding trainees in practicing fault diagnosis
- ***Potentially*** a field diagnostic aid



Authoring a DIAG Tutor

The DIAG applicator constructs a working simulation of the target system, using the authoring tools of the RIDES/VIVIDS system. The specification of the target system behavior includes normal behavior and abnormalities resulting from faults. When the working model is complete, the developer selects a "Generate" function that automatically produces the intelligent tutor.

Authoring a DIAG Tutor



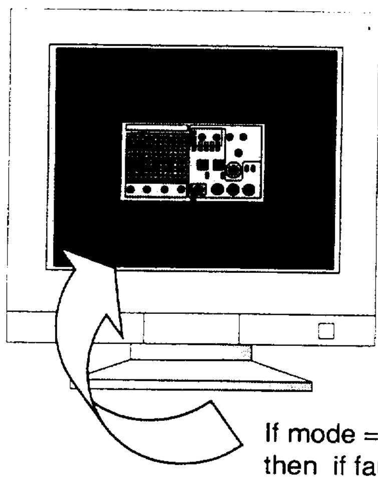
The author:

- constructs a working model of the target system, including faulty elements.
- clicks the Generate button to produce an intelligent tutor.

The Working Model

The working model provides the vehicle with which the learner practices. It behaves normally when there is no fault inserted, and it exhibits appropriate abnormalities when it is faulted. The system model is usually constructed of qualitative behavior rules for the system indicators. These rules specify what each indicator displays in various situations. Only if an indicator displays quantitative information would a rule be quantitative.

The Working Model



- Is operable by the learner
- Simulates normal and abnormal behaviors
- Is usually driven by qualitative rules rather than quantitative functions.

```
If mode = "HighGain"
then  if fault = "none"
         then frequency = 120
         else if fault = "loss"
              then ...
```

Modeling Complex Systems

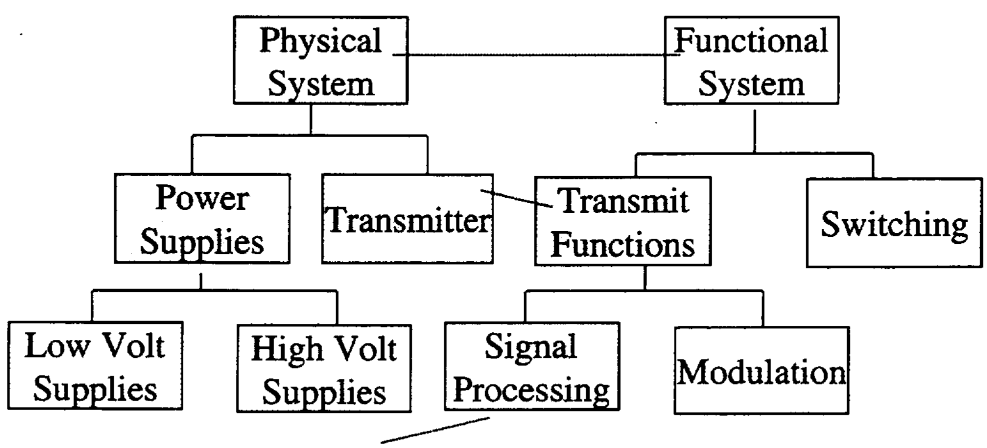
Large and complex systems cannot be represented on a single computer display screen. Consequently, the developer breaks a system down hierarchically and builds a display for each part. DIAG supports the production of a hierarchy representing a system's physical organization and an optional hierarchy of functional structure. The developer can construct links between physical and functional elements so learners can gain an understanding of the correspondence among them.

This figure illustrates how a complex system might be decomposed into individual screen views, each block in the figure being represented by one screen presentation.

Modeling Complex Systems

Complex systems are represented with many screens of graphical content.

DIAG supports the construction of a physical hierarchy, a functional hierarchy, and links between the two.



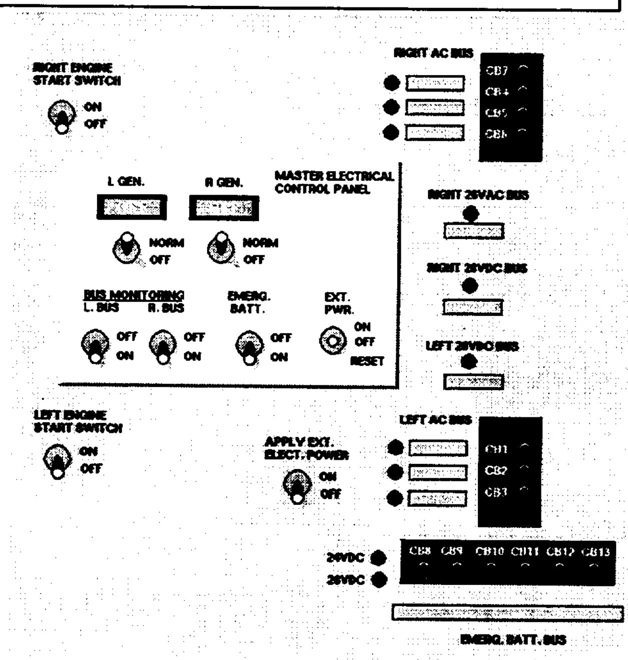
System Model: A Typical Screen View

A single block in the previous figure might be represented to the learner as shown here. This is a portion of a large front panel for an aircraft power distribution system. The learner may operate this panel at will, and may move higher or lower in the system hierarchy.

System Model: A Typical Screen View

The learner performs a checkout procedure on this panel.

The controls are operable and the indicators are responsive.



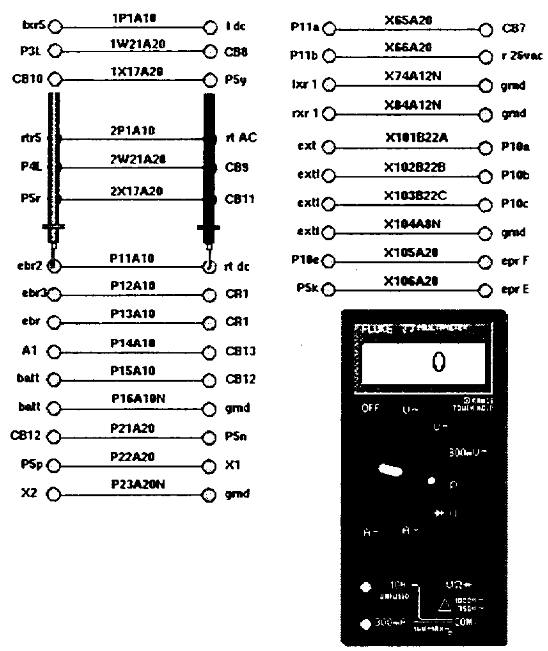
System Model: A Screen View of Test Points

In addition to checking front panel indicators in various modes, the learner can employ test equipment. Here is a screen of test points that can be probed with the multimeter. The multimeter is an object stored in the DIAG object library. A copy of that unit can be placed in any DIAG simulation and be fully operational with no further effort by the developer.

System Model: A Screen View of Test Points

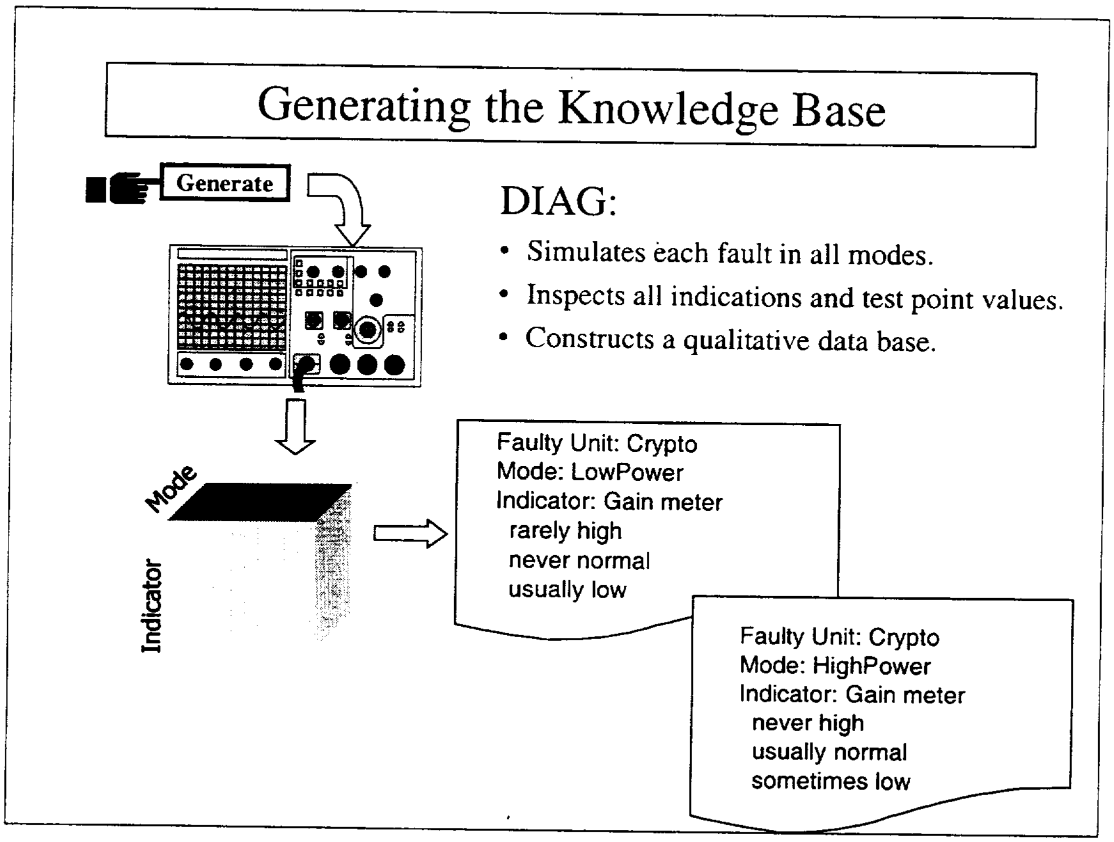
The learner may make test point readings here.

This AN77 multimeter and a T2336YA oscilloscope, developed by Ken Roberts and Steve Parchman, are available in the DIAG object library.



Generating the Knowledge Base

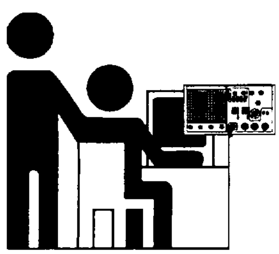
Like IMTS before it, DIAG automatically analyzes the model of the target system prior to use in training. By simulating each fault condition that has been specified, DIAG compiles a large and precise data base reflecting the effects of each fault on each indicator, in each operational mode. DIAG then collapses that data bank into a more compact, fuzzy, and qualitative form. The result is a knowledge base that states the possible consequences of faults in the larger replaceable units, rather than in individual components.



Supporting the Learner

Following generation of the knowledge base, DIAG stands ready to assist learners as they work simulated faults. DIAG selects a problem, simulates the faulted system as the learner conducts his or her own troubleshooting strategy, and responds to requests for assistance.

The Learner ...



- Practices troubleshooting on simulated faults.
- Asks for guidance when needed:
  - interpreting symptoms
  - choosing tests
  - revising suspicions
  - assessing own strategy
  - seeing expert work the problem.

The Consultation Types

The learner requests guidance by selecting from the list shown here. During an exercise, DIAG can generate responses to such questions as:

Is this symptom normal or abnormal? What does it signify?

Do the symptoms I've seen implicate or exonerate this unit?

What should I be suspecting, considering the tests I've done?

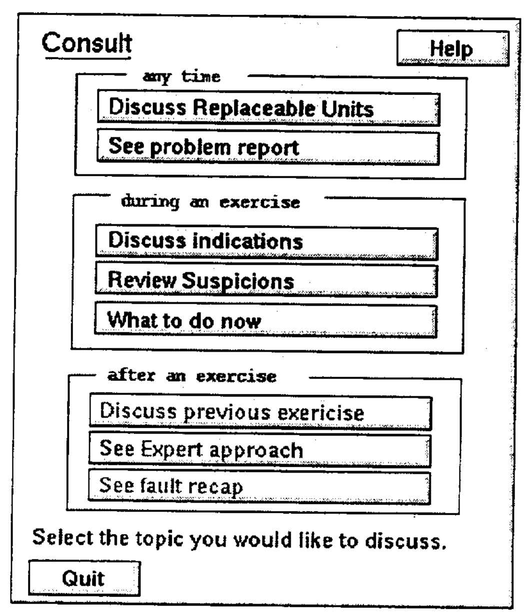
What should I do now?

After an exercise is completed, DIAG can:

- replay the individual's work, giving expert analysis of the symptoms seen
- demonstrate and explain how an expert would have worked the fault.

The Consultation Types

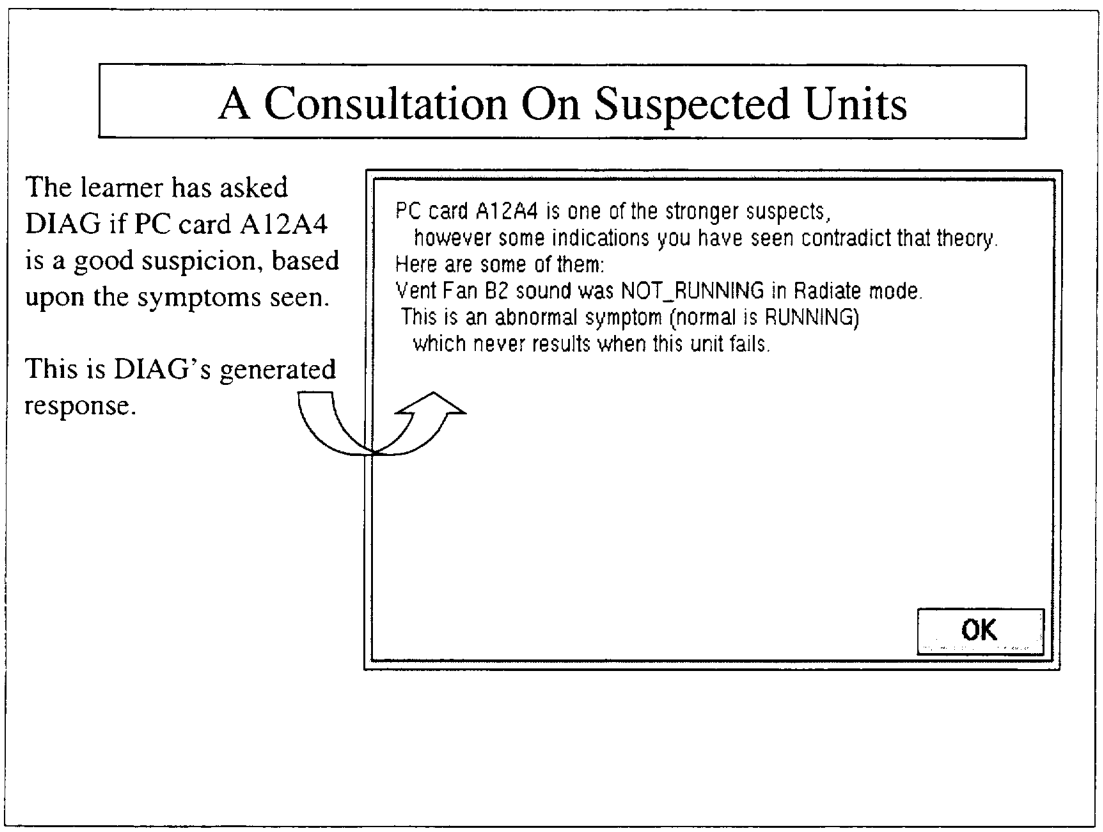
The learner requests guidance using these buttons.



A Consultation on Suspected Units

- The learner first selected Discuss Replaceable Units (see previous slide).
- DIAG then advised the learner to click the mouse on the unit in question.
- The learner selected printed circuit card A12A4, and DIAG generated the discussion shown.

To do this, DIAG had been invisibly and expertly interpreting all the symptoms seen by the learner and maintaining its own internal suspicions about the source of the fault. This particular guidance type deals with the quality of the learner's interpretations of symptoms.



Findings: In General

Four applications have now been produced using DIAG:

- R-49 Radio Trainer, developed by SSgt Brian Bagnetto, Armstrong Labs
- Home Heating System
- SPY-1B Transmitter (RF Amplifier equipment)
- Aircraft Power Distribution System

Based upon this work, we have a good understanding of the skills required to apply DIAG, and some of the benefits of the simulation-based approach.

Findings: In General

- A technician qualified to maintain a system has the domain knowledge to produce a DIAG diagnostic tutor for it.
- An experienced computer user is required to produce the system model.
- DIAG training is automatically updated if the system model is updated to reflect design modifications.
- A DIAG application could support field diagnostic aiding with a modest extension to the interface.

Findings: The Latest Tutor

The most recently developed DIAG tutor for troubleshooting aircraft power distribution systems was accomplished in just one man month. The speed with which DIAG tutors can be produced is one great advantage of this approach. In addition, a DIAG tutor is easily updated should the target system undergo field modifications, for only the system model requires modification.

Findings: Latest Tutor

Aircraft power distribution tutor:

- 107 fault exercises
- 8 operable controls; 7 operating modes
- 26 indicators; 50 test points
- developed by one person in one month
- instructional effectiveness being assessed at NATTC

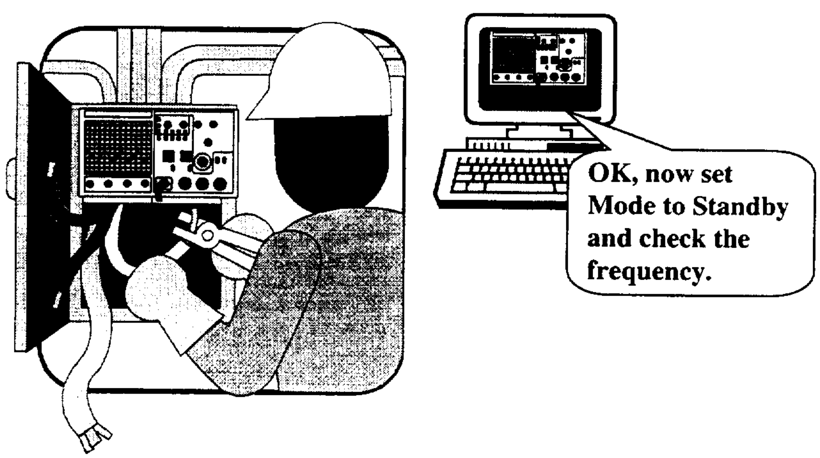
Field Aiding Potential

DIAG's diagnostic reasoning processes do not refer to the actual fault being simulated. Instead, DIAG reasons about symptom implications as any troubleshooter would. As a result, DIAG is equally capable of troubleshooting an actual fault in the field. Very modest extensions are required to allow a field technician to convey observed symptoms to DIAG, and receive guidance in isolating the fault. Two modes of interaction are planned between DIAG and the field technician:

1. Fix now, instruct later - help me fix the system as quickly as possible, but save related instructional commentary for my later review.
2. Fix and instruct now - help me fix the system, but provide explanations and rationale as we work.

Field Aiding Potential

The field technician reports observed symptoms and receives troubleshooting guidance.



The DIAG Web Site

All of the DIAG resources are available for others to use. This includes the authoring system, sample applications, a working user guide (developed in DIAG), and a storehouse of pre-built DIAG objects. This requires access to VIVIDS, and the Linux operating system.

Visit the DIAG Web Site

http://www.fcs.net/dtowne/diag/default.html

Contents:

- **DIAG Authoring and Delivery System**
- **DIAG LITE**
- **Object Library**
- **Application Guide (sample usage of DIAG objects)**
- **User Manual**

Sample applications (home heating and power distribution) can be downloaded from http://www.fcs.net/dtowne/dnload

Promoting Advanced Training Technology (From Workshop Discussions)

We know that the ***true cost*** to train to a ***given level of proficiency*** is dramatically less when using simulation and individualized tutoring than with real equipment and learners practicing in groups. However, the decision maker who is considering adopting these technologies may not realize a reduction in his or her budget, owing to the manner in which expenses are assigned in military organizations. Furthermore, current training may be extremely inexpensive, due to minimum resources allocated.

We may, therefore, be making an incorrect assumption - that these training technologies can be most effectively promoted based upon cost savings. ***Only if we address true cost and compare like quality of training can we be assured that the dramatic cost savings are evident to the decision maker.***

True Costs versus Perceived Costs

(from our workshop discussion period)

- The decision maker's budget may be unrelated to the true cost of providing quality training
  - current training may appear to be 'free' (not a budget item)
  - budgeted training may be for low quality.

Thus, we must not assume that the decision maker will necessarily be able to reduce his or her budget by adopting our methods. We must compare like qualities of training and must deal with true costs rather than budget lines.

References

1. Towne, D. M., Johnson, M. C. and Corwin, W. H., *Profile: A Technique for Projecting Maintenance Performance From Design Characteristics*, University of Southern California, Behavioral Technology Laboratories, Los Angeles, CA, Technical Report No. 100, August 1983.
2. Towne, D. M. and Munro, A., "The Intelligent Maintenance Training System," *Intelligent Tutoring Systems: Lessons Learned*, J. Psotka, L. D. Massey and S. A. Mutter (eds.), Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey, 1988.
3. Towne, D. M., Munro, A., Pizzini, Q. A., Surmon, D. S., Coller, L. D. and Wogulis, J. L., "Model-Building Tools for Simulation-Based Training," *Interactive Learning Environments*, Vol. 1, 1990, pp. 33-50.
4. Munro, A., Johnson, M. C., Surmon, D. S. and Wogulis, J. L., "Attribute-Centered Simulation Authoring for Instruction," Proc. of AI-ED '93, World Conference on Artificial Intelligence in Education, 1993.
5. Towne, D. M., "Model-Based Simulation for Instruction and Learning," Proc., Delta '94 Conference in Telematics, Dusseldorf, Nov. 1994.
6. Towne, D. M., "Approximate Reasoning Techniques for Intelligent Diagnostic Instruction," *International Journal of Artificial Intelligence and Education*, 1997.
7. Towne, D. M., "Intelligent Diagnostic Tutoring Using Qualitative Symptom Information," Proc., American Institute of Artificial Intelligence, Nov. 1997.

S13-61

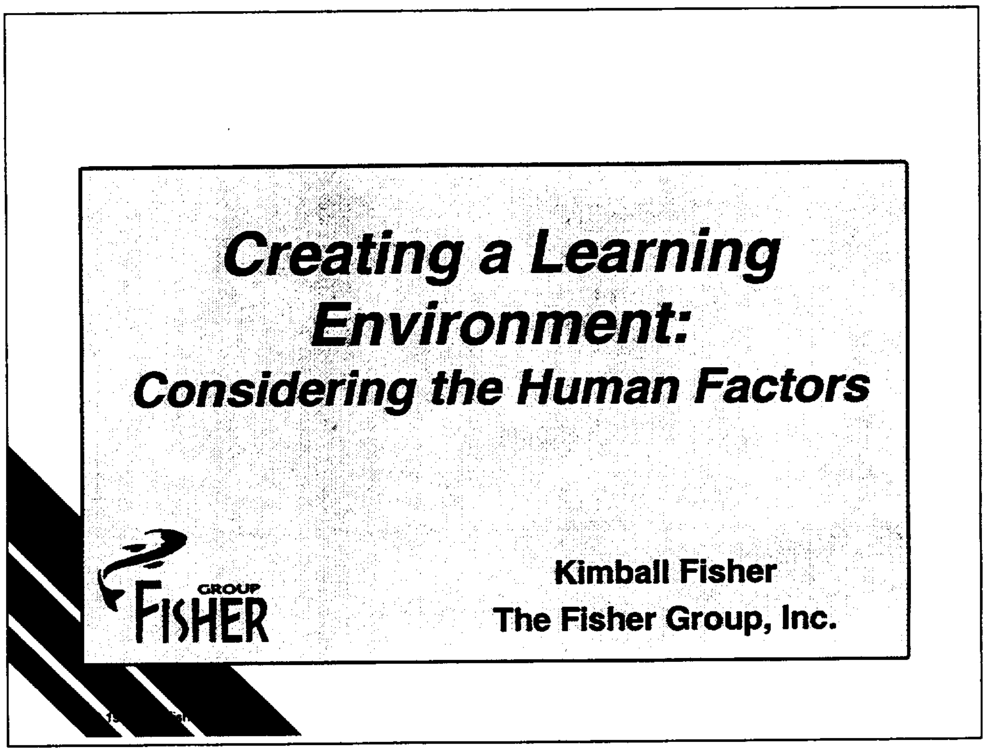
Creating a Learning Environment: Considering the Human Factors

Kimball Fisher
The Fisher Group, Inc.
Beaverton, OR

Creating a Learning Environment: Considering the Human Factors

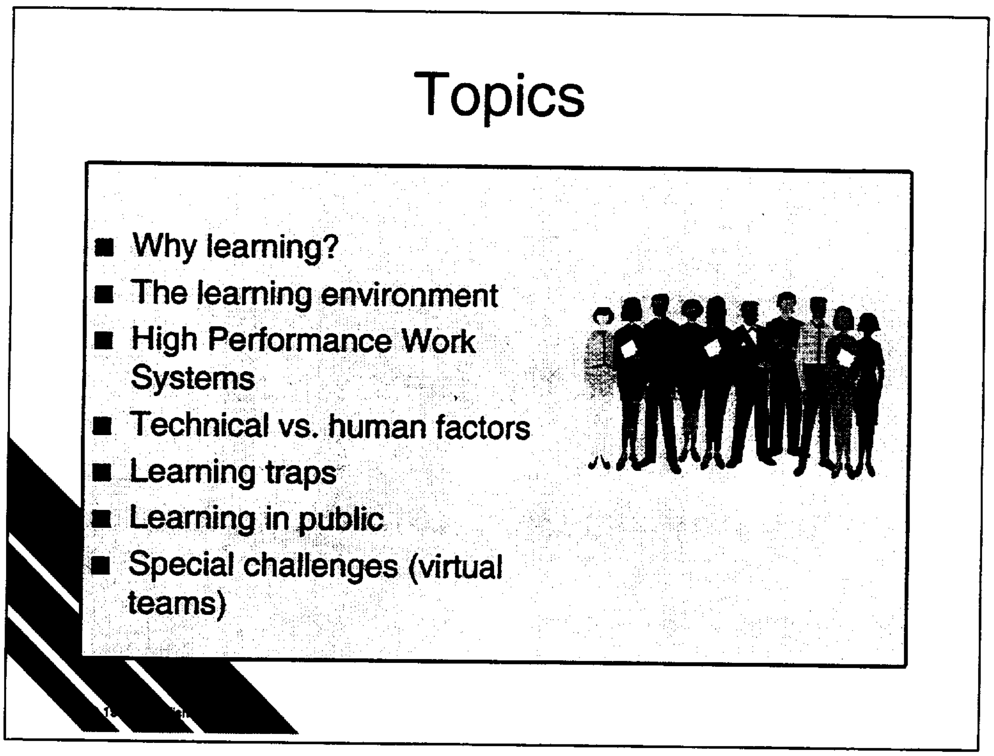
Kimball Fisher
The Fisher Group, Inc.
12550 S.W. Butner Road
Beaverton, OR 97005
(503) 641-9316; Fax (503) 641-5235

Creating an effective learning environment requires not only technology, but an understanding of a number of other factors as well.



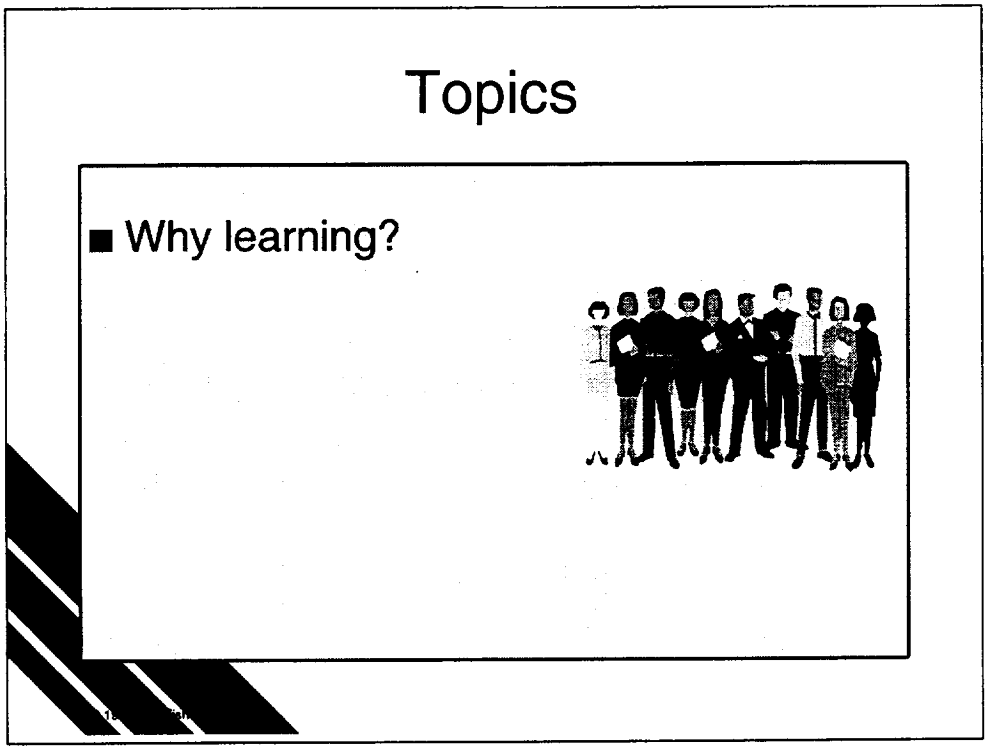
Topics

In this presentation I will discuss why we should care about creating a learning environment, what a learning environment entails, define high performance work systems, discuss the difference between technical and human factors, identify common learning traps, talk about the importance of learning in public, and mention special learning challenges associated with virtual teams.



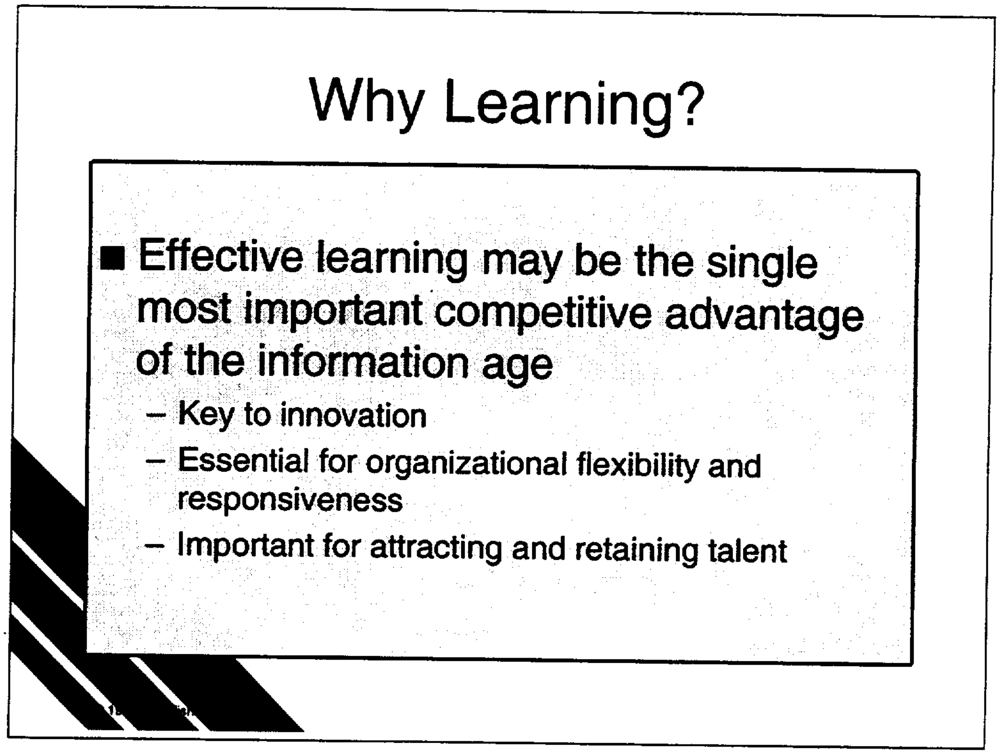
First Topic

Let's start with why learning is important.



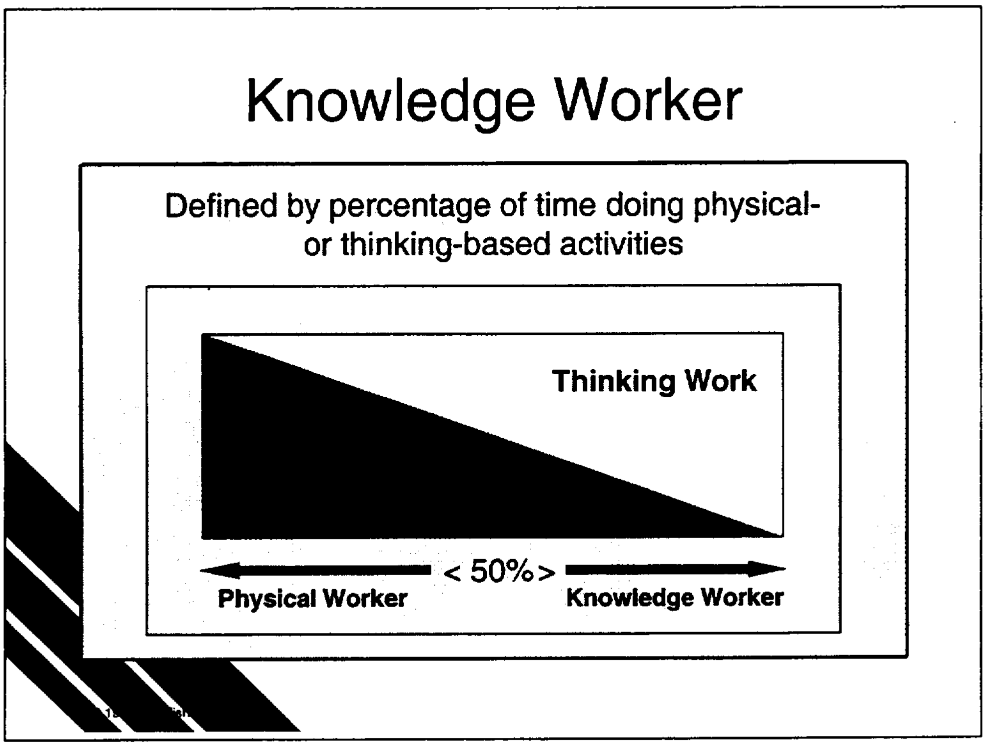
Why Learning?

Learning is the key to competitive advantage in the information age.



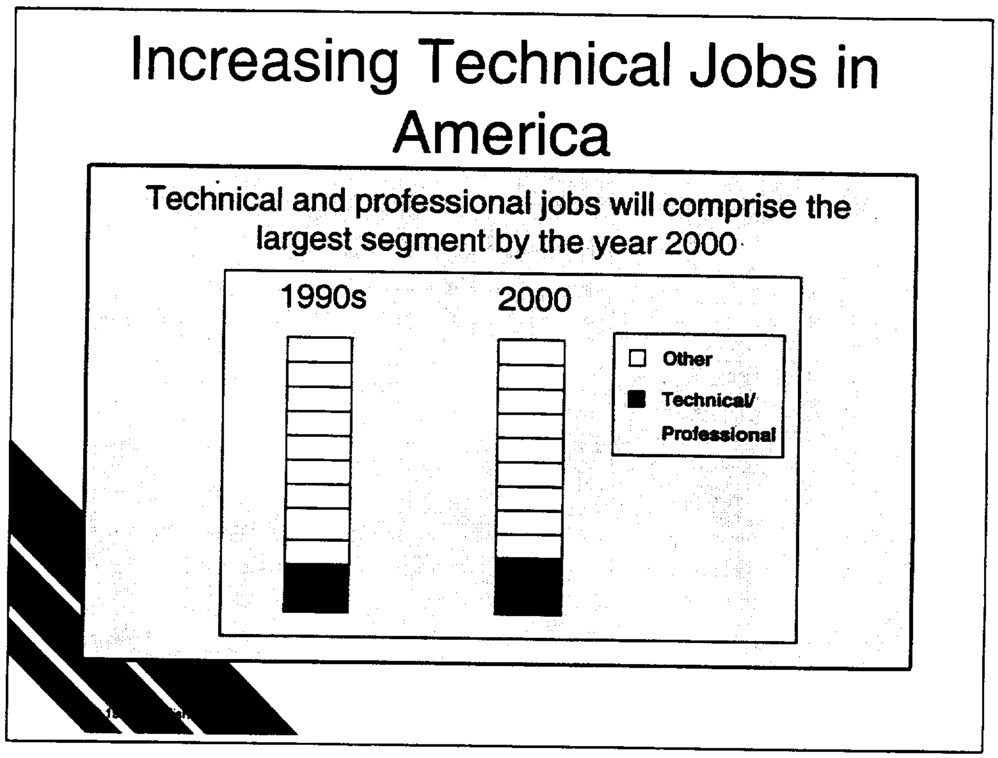
Knowledge Worker

What makes one a knowledge worker is that the majority of his/her work is knowledge work. In most jobs, the knowledge work content of our work is increasing. With the exception of highly theoretical fields, there are few knowledge workers who do only knowledge work. Conversely, there are few physical workers who do only physical work. In fact, there are few strictly physical workers at all any more.



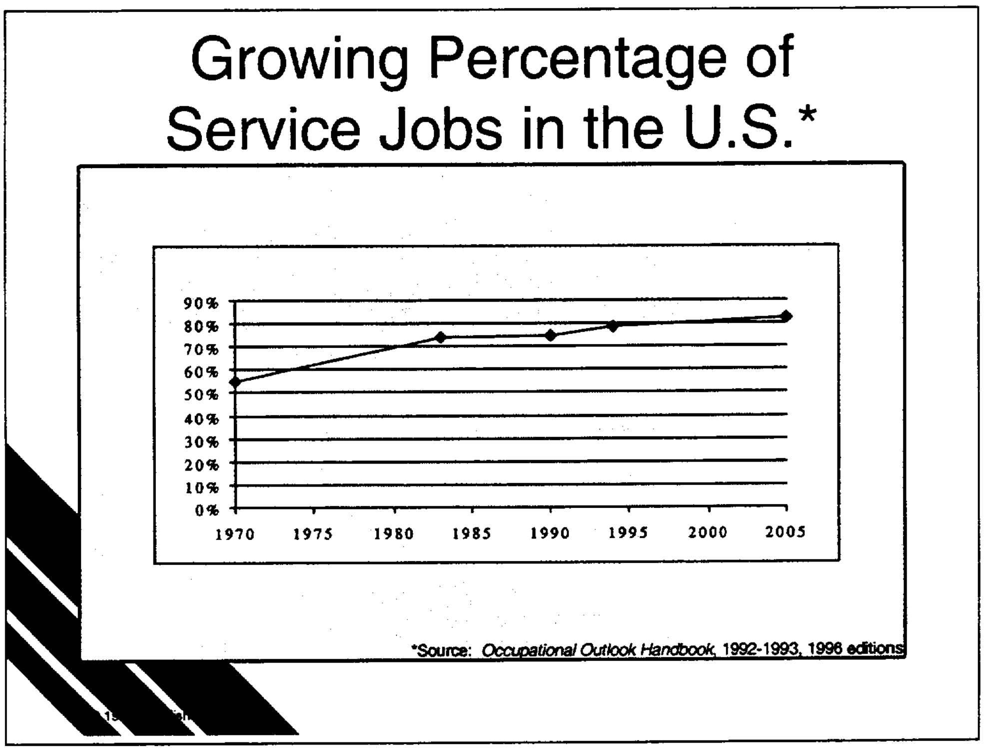
Increasing Technical Jobs in America

According to the Census Bureau, in the United States (often a forerunner for business changes across the globe), technical workers (medical technologists, technicians, paralegals, etc.) and professionals (accountants, scientists, engineers, doctors, etc.) already account for 16 percent of the American workforce. Experts project that by the year 2000, this group will be the largest single segment, with 20 percent, or over 23 million people.



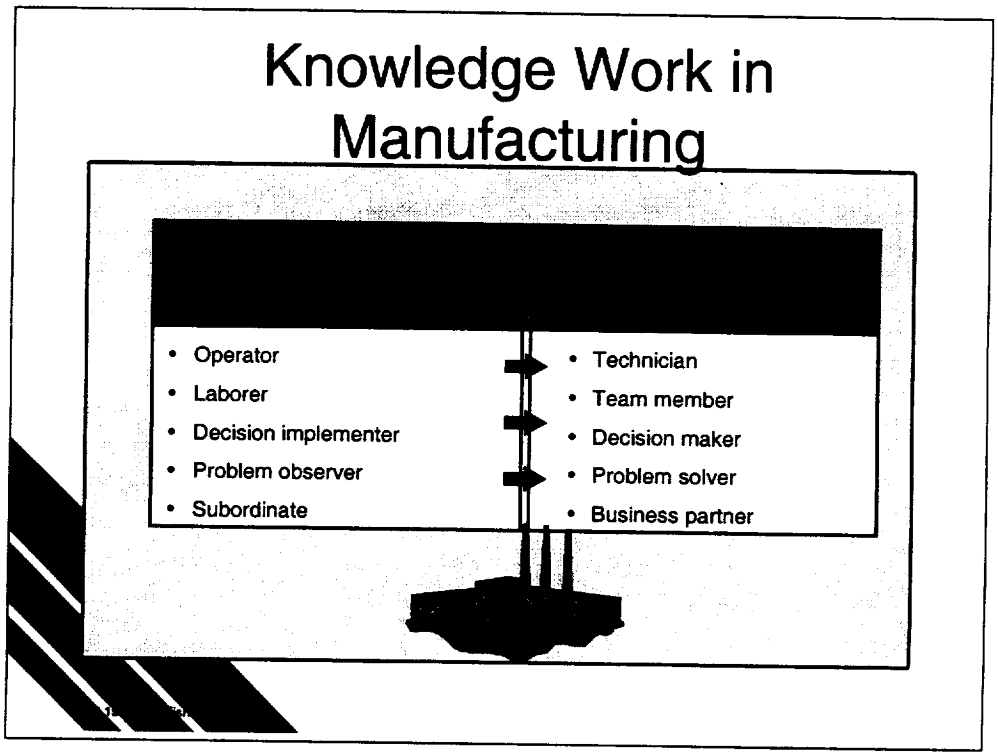
Growing Percentage of Service Jobs in the U.S.

In the 1970's, service work (which is mostly knowledge work) accounted for 55 percent of all U.S. private-sector jobs. By 1995, it accounted for 79 percent. Today the U.S. economy is dominated by these knowledge workers. The U.S. Bureau of Labor Statistics projects that industries providing services will account for about four out of five jobs by the year 2005.



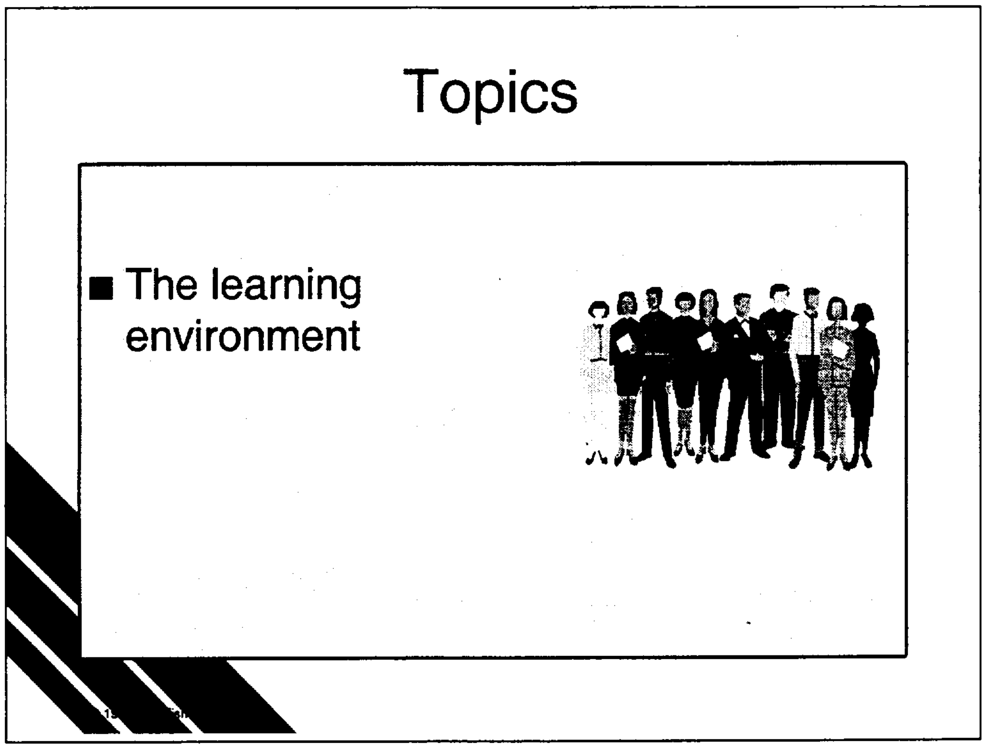
Knowledge Work in Manufacturing

Even in factories we see work shifting from physical work to mental work. These role changes require increased emphasis on developing intellectual skills such as effective decision making, problem solving, communication effectiveness, business analysis, and so forth.



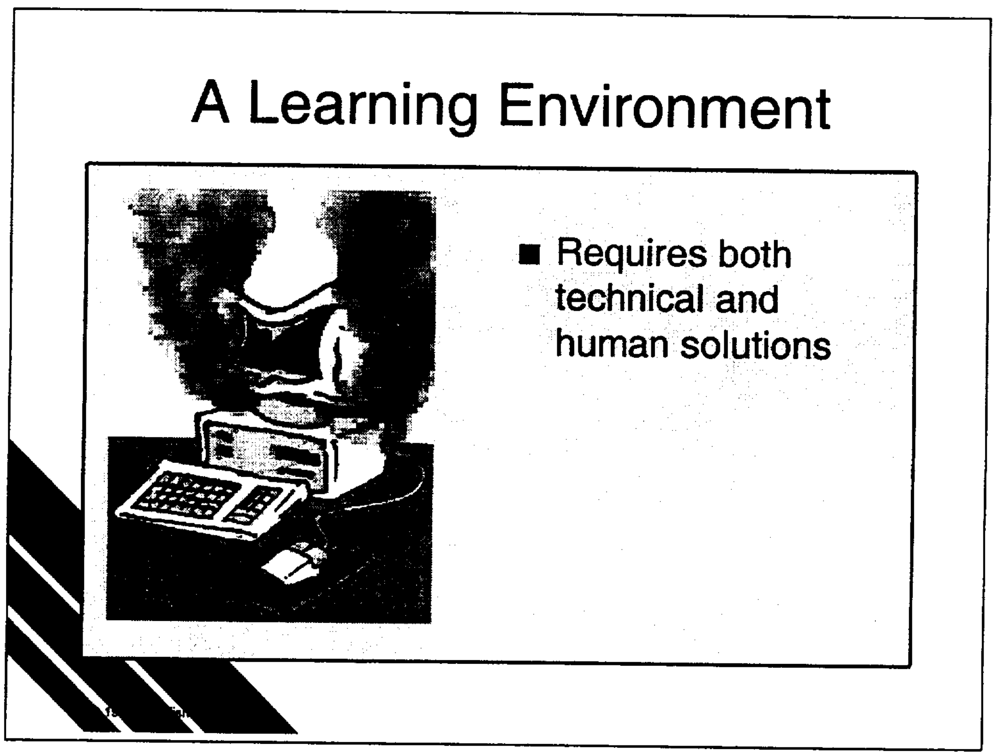
Second Topic

The second topic is the learning environment.



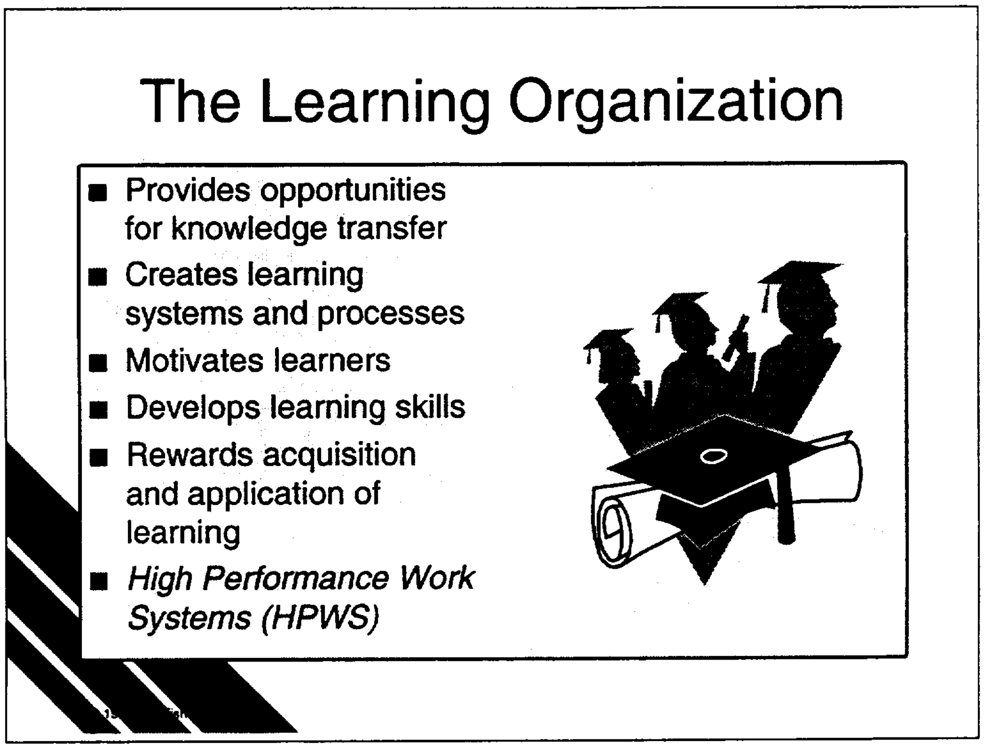
A Learning Environment

An effective learning environment requires the technical solutions we have discussed in great detail at this conference. We know, for example, that interactive, intelligent tutoring programs are significantly more effective than traditional training technologies. Training technologies alone, however, do not create learning. If, for example, the skills people learn are not reinforced in the workplace the technologies are for naught. Or, for example, if leaders disagree with a particular training method, even though people have the new skills, they may not be able to use them. Having skills without having the will to use them provides little benefit to the organization.



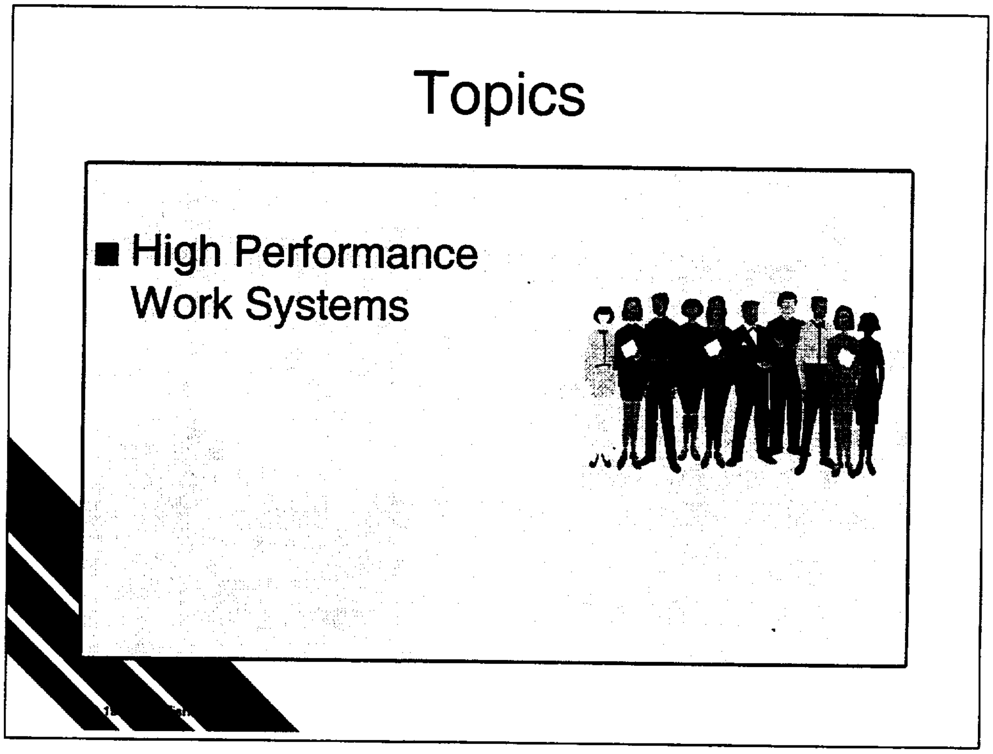
The Learning Organization

A good learning organization is composed of the following characteristics: 1) it provides opportunities for knowledge transfer, 2) it creates learning systems and processes, 3) it motivates learners to use what they know, 4) it develops learning skills, 5) it rewards both the acquisition and the application of learning, and 6) it is a high performance work system.



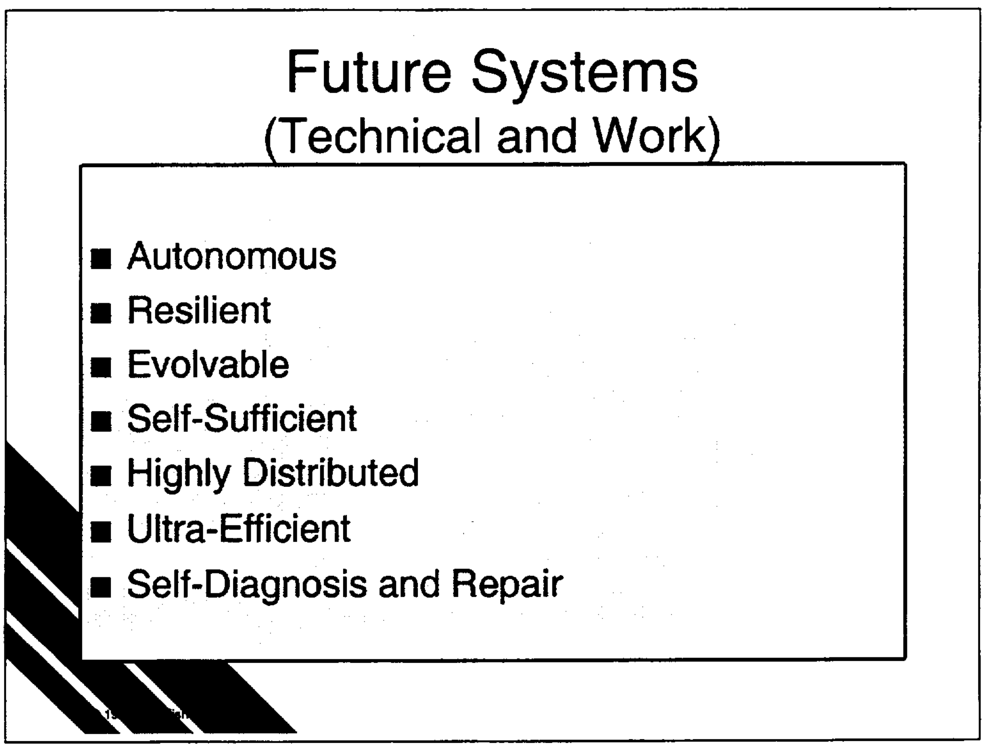
Third Topic

The next topic is high performance work systems.



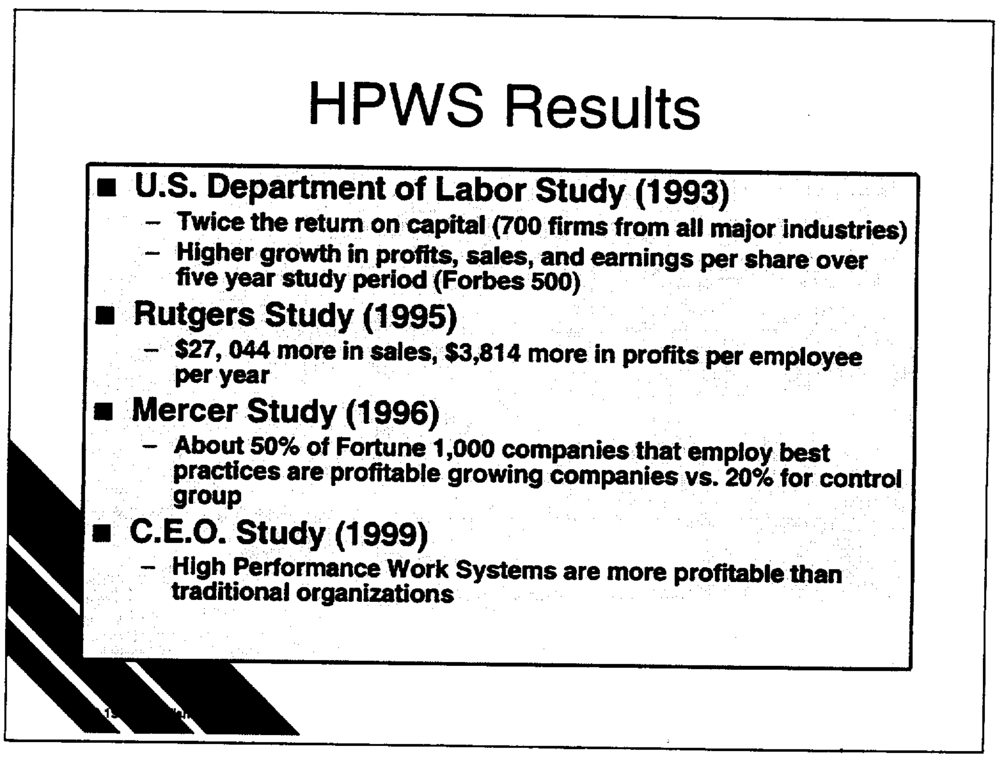
Future Systems (Technical and Work)

It has been suggested that the technologies of the future would be autonomous, resilient, evolvable, self-sufficient, highly distributed, ultra-efficient, and be capable of self-diagnosis and repair. This is as good a definition of a high performance work system as I have seen. Would we want any less of our work systems than we want of our technologies? Too many organizations use leading edge technologies in obsolete work systems. What is the benefit of that?



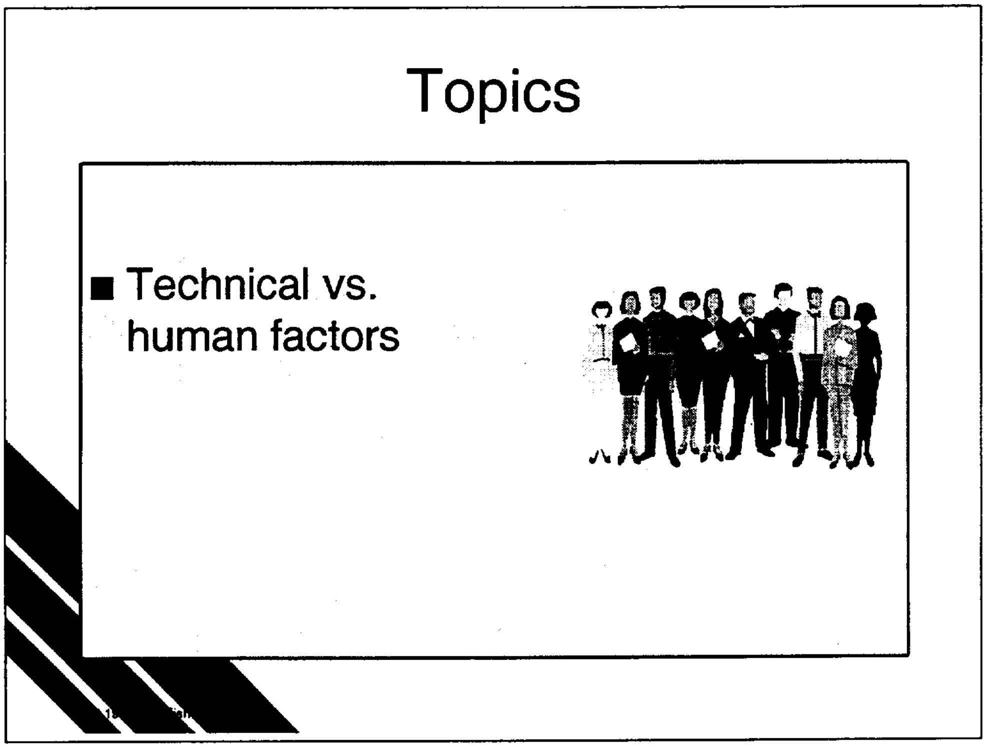
HPWS Results

There is clear evidence that high performance work systems outperform traditional organizations. A 1993 study conducted by the U.S. Department of Labor, for example, showed that these operations had twice the return on capital, and higher growth in profit, sales and earnings. A Rutgers University study in 1995 showed that these operations had more than $3,800 profit per employee per year than their traditional counterparts. A study conducted by the Mercer Consulting Group demonstrated that high performance work systems are more likely to be profitable, growing companies than traditional, bureaucratic and hierarchical organizations. The Center for Effective Organizations at USC determined that these organizations are more effective than traditional operations.



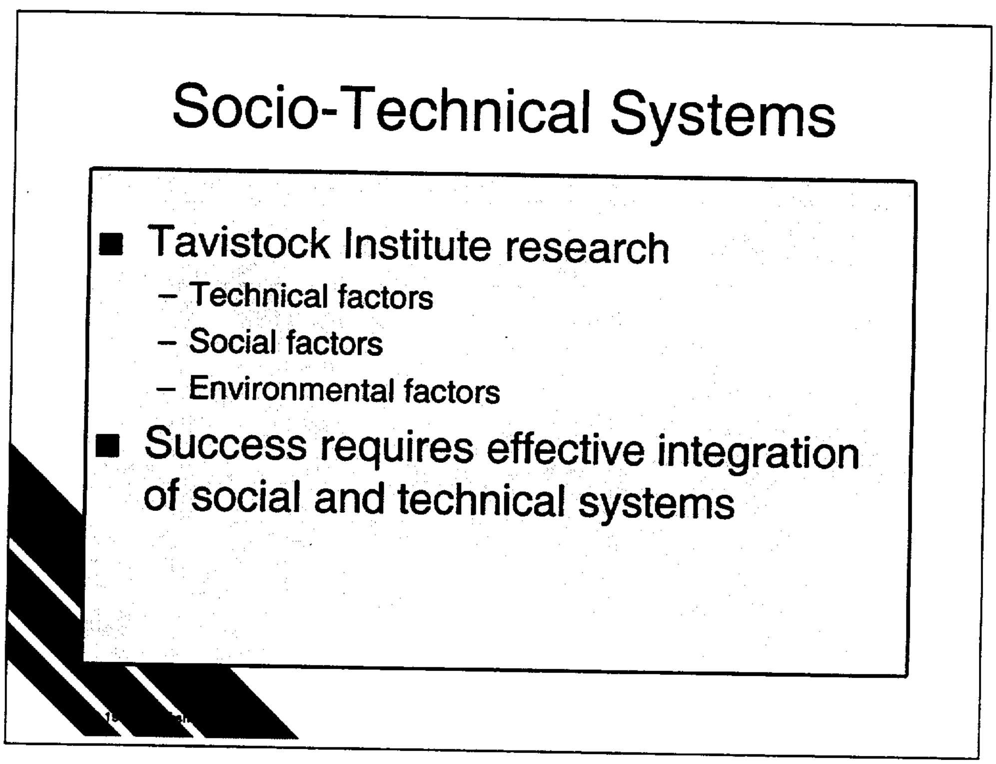
Fourth Topic

The next topic is technical versus human factors.



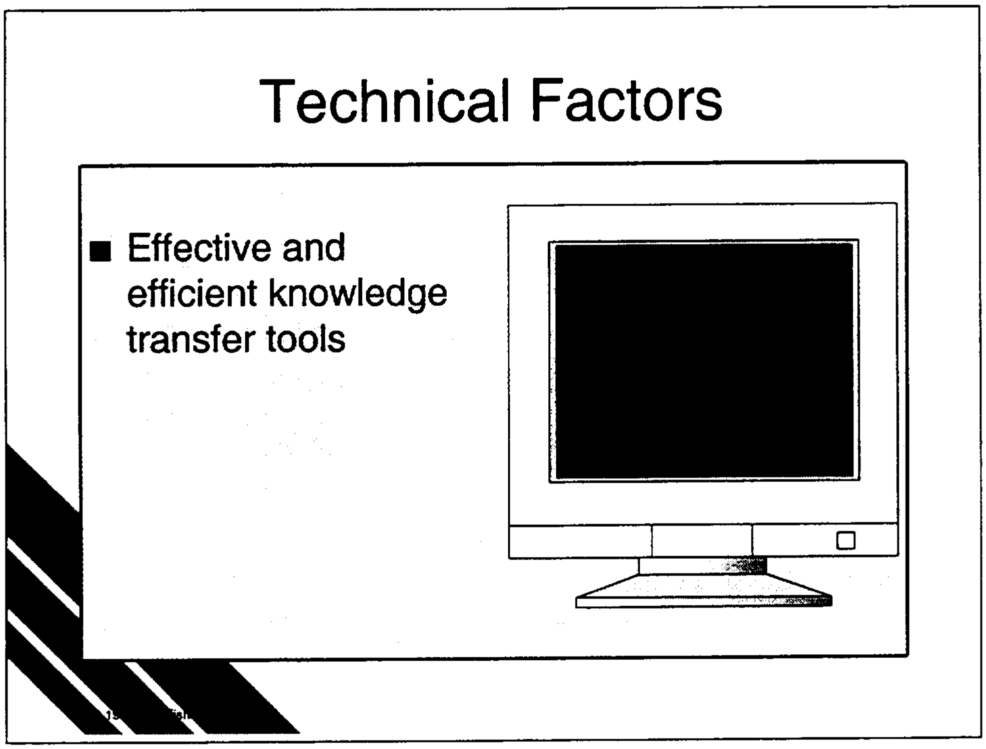
Socio-Technical Systems

Research done by the Tavistock Institute directly after World War II showed clearly that organizational effectiveness required excellence in three areas: 1) technical, 2) social (people), and 3) environmental (business). Commissioned by the government of the United Kingdom, this institute sought answers to the question, "How can we rebuild industry and government in a post-war economy without large capital outlays?" The answer to the question was found in a coal mine where miners had developed a special work system to complement their technologies. This coal mine significantly outperformed other coal mines that had exactly the same, or in some cases, superior technology. This work system, what they called "socio-technical," and what we now call High Performance Work Systems, was a non-bureaucratic, team-based structure with minimal hierarchical supervision.



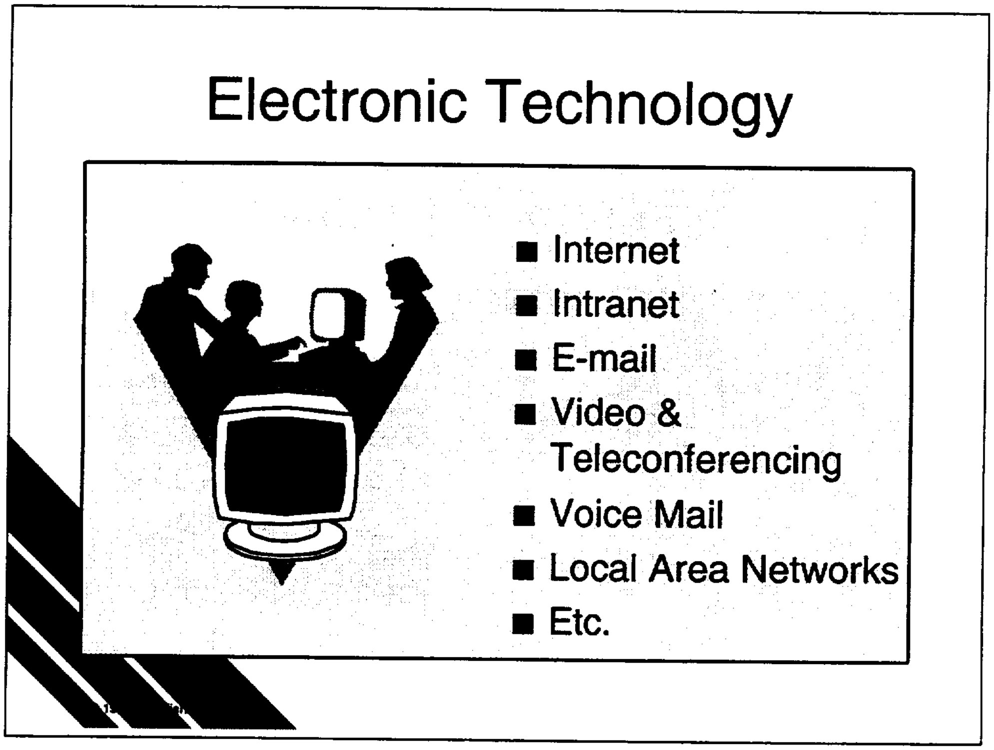
Technical Factors

Although the Tavistock research dealt primarily with industrial technology, we believe the findings are appropriate to advanced training technologies as well. The purpose of advanced training technologies is the effective and efficient transfer of knowledge. When these technologies are complemented with effective work systems, the skills and knowledge have a much higher probability of being applied and reinforced in the workplace.



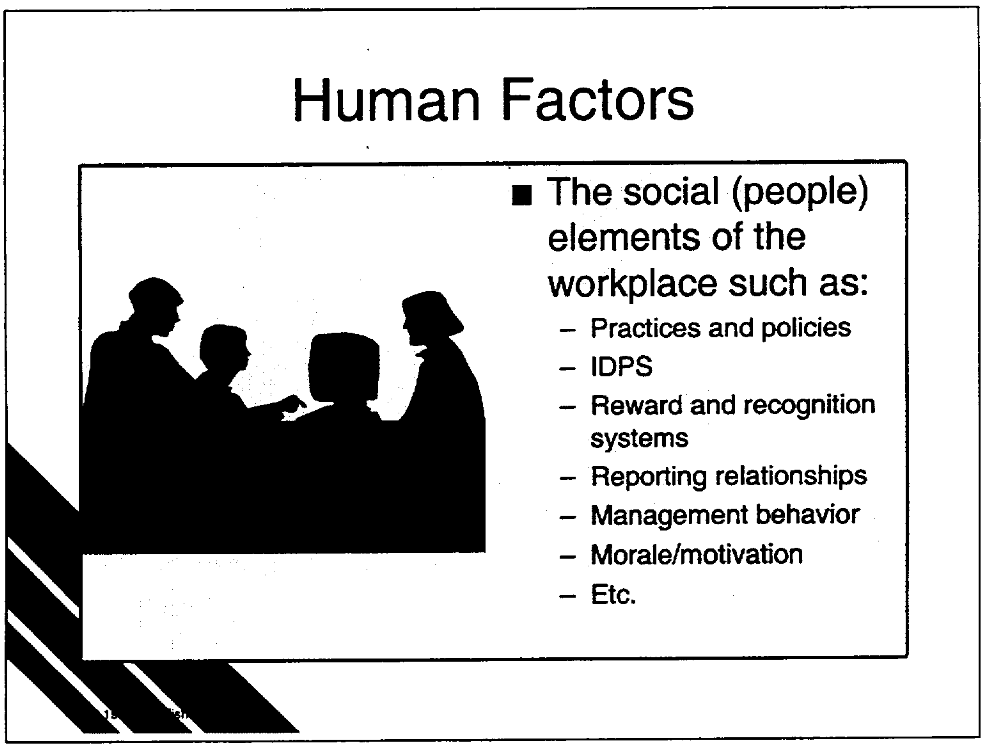
Electronic Technology

Examples of common technologies used for training include those listed on this slide.



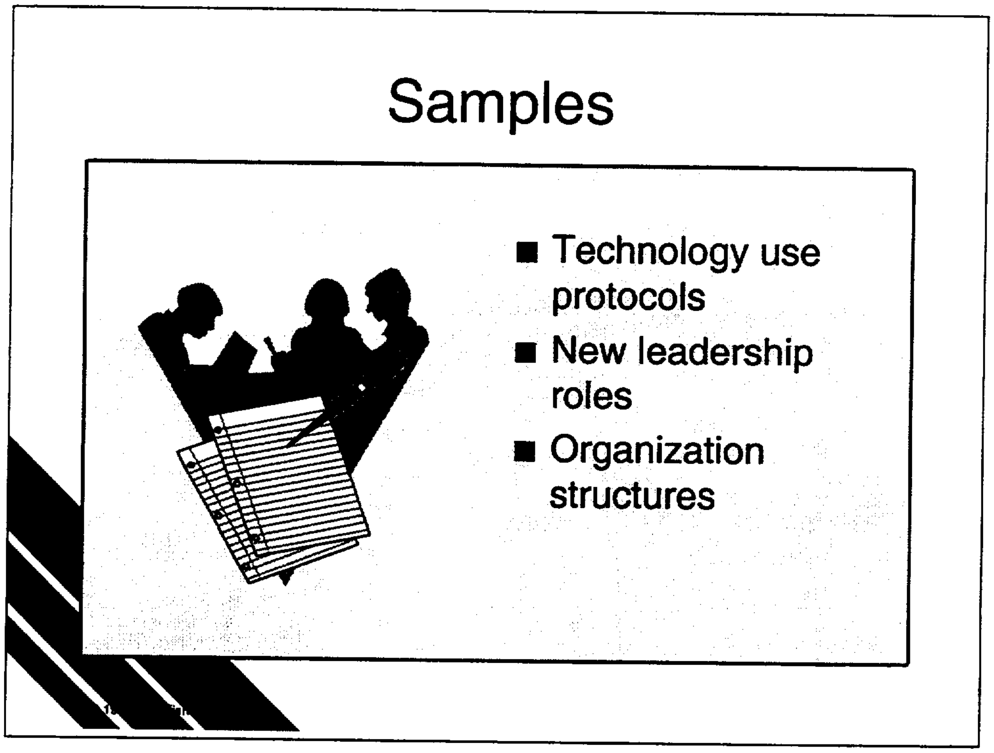
Human Factors

The human factors to be considered for developing a culture of learning and knowledge transfer include things such as the practices and policies of the workplace, the use of individual development planning activities, appropriate reward and recognition systems, reporting relationships and work structures, management behavior, and morale and motivation. Without these factors supporting knowledge transfer and application, learning technologies alone are not likely to be successful.



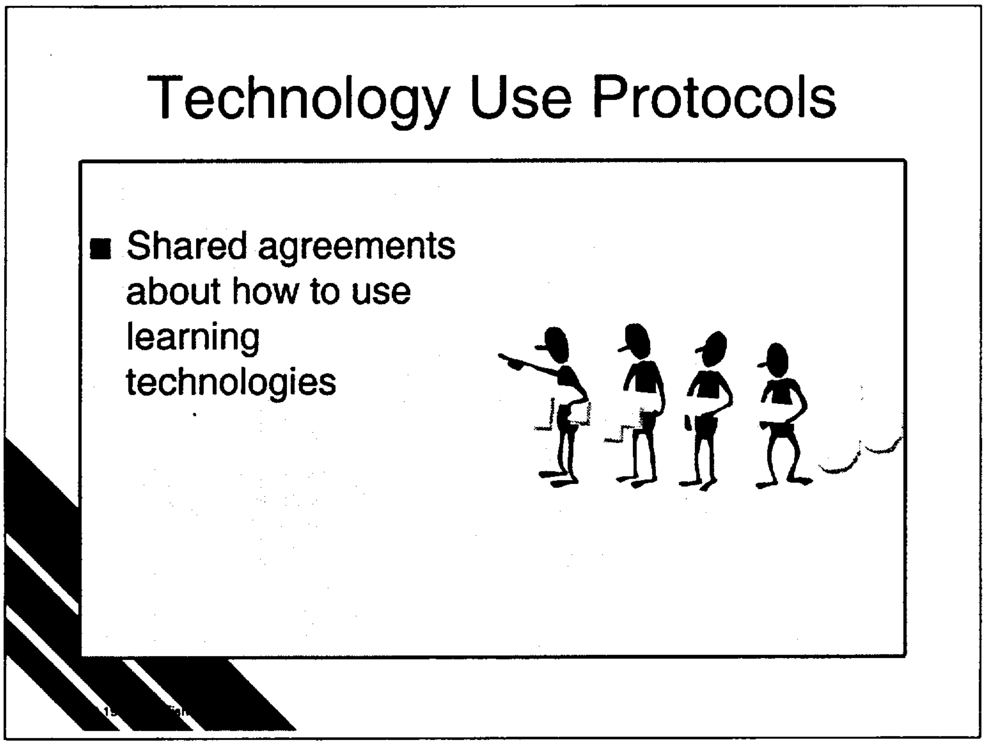
Samples

Here are some samples of the types of things companies have done to address the human factors in creating a learning environment. We will look first at technology use protocols, second at new leadership roles, and third at organization structures that reinforce learning.



Technology Use Protocols

Technology use protocols are shared agreements about how to use learning technologies. To be effective they must be created by the people who are using the technologies rather than being dictated by organization leadership or technical specialists.



Electronic Communication Protocol Example

This is a sample of a protocol developed to facilitate the use of electronic communication systems. We believe this technique applies to training technologies as well. Without such protocols people often find that the technology controls their time and behavior rather than being a tool of knowledge transfer. When these protocols are developed and administered by the people using the technologies, our experience suggests that they are useful ways to regulate team behavior without relying on hierarchical methods.

Electronic Communication Protocol

- All messages will be short and to the point.
- E-mail will not be used for urgent messages.
- We accept responsibility for a personal delivery (face to face or voice to voice) of any urgent message.
- The message is not delivered until the recipient understands it.
- We will use group replies or mass mailings only when necessary.
- We will treat people electronically the same way as we would in person.
- We agree to keep all distribution lists current.

Electronic Communication Protocol Example (Cont'd.)

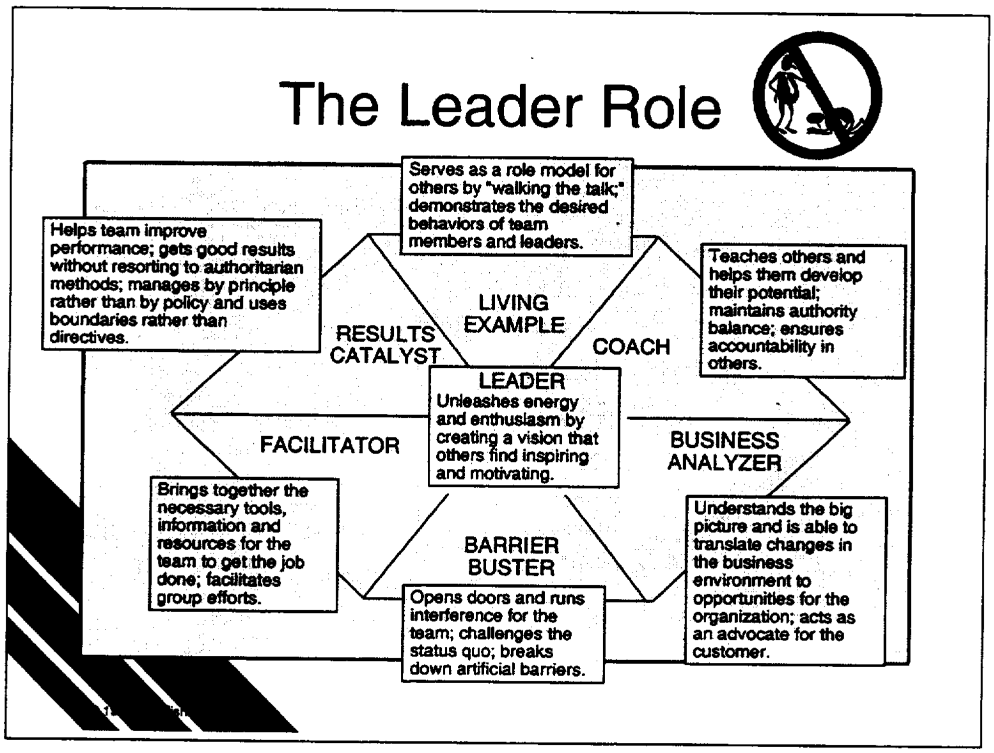
Sample continued.

Electronic Communication Protocol (Cont'd.)

- We will use mail primarily for things that require documentation.
- E-mail is a supplement to, not a substitute for, personal interaction.
- We will respect one another's time and privacy.
- We will not allow others to read messages not pertaining to them.
- To enable message prioritization we will label each with either "requires action" or "for your information."
- We will sign all messages.
- We will summarize long quotes, not copy them.

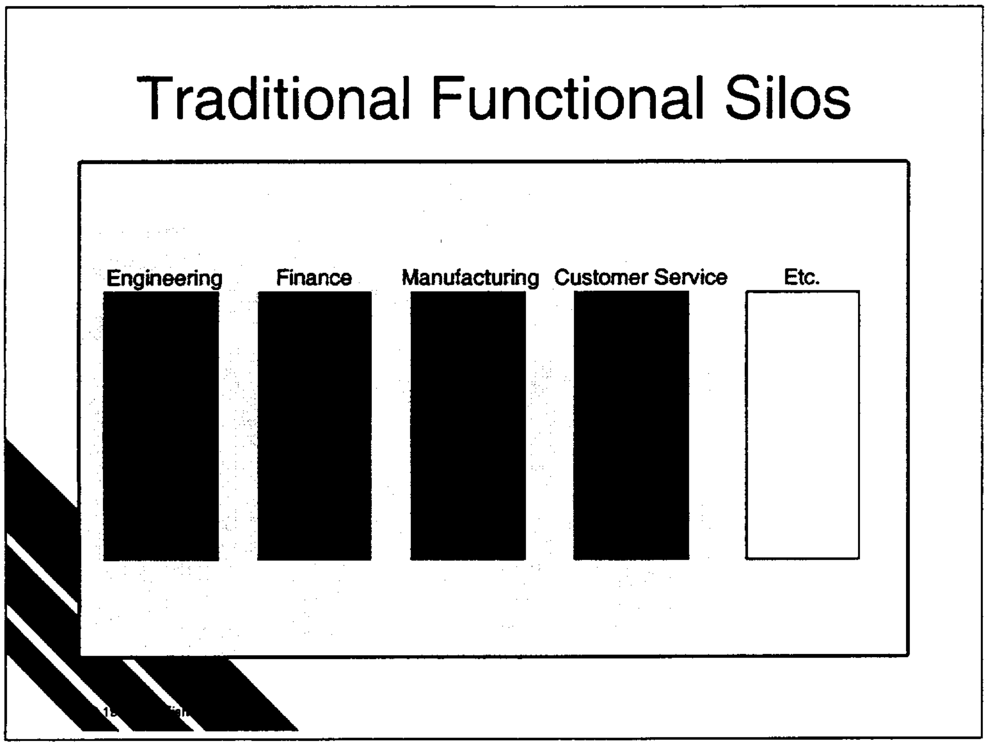
The Leader Role

This is an example of the new role of leaders and managers. It comes from research for the book *Leading Self-Directed Work Teams* (McGraw-Hill 1993). In talking with leaders of high performance work systems, we learned that management behavior is a key variable to organizational effectiveness. These seven competencies facilitate a learning culture by increasing empowerment, accountability, and focus on improved results. They also reduce fear and dependence on direct supervision.



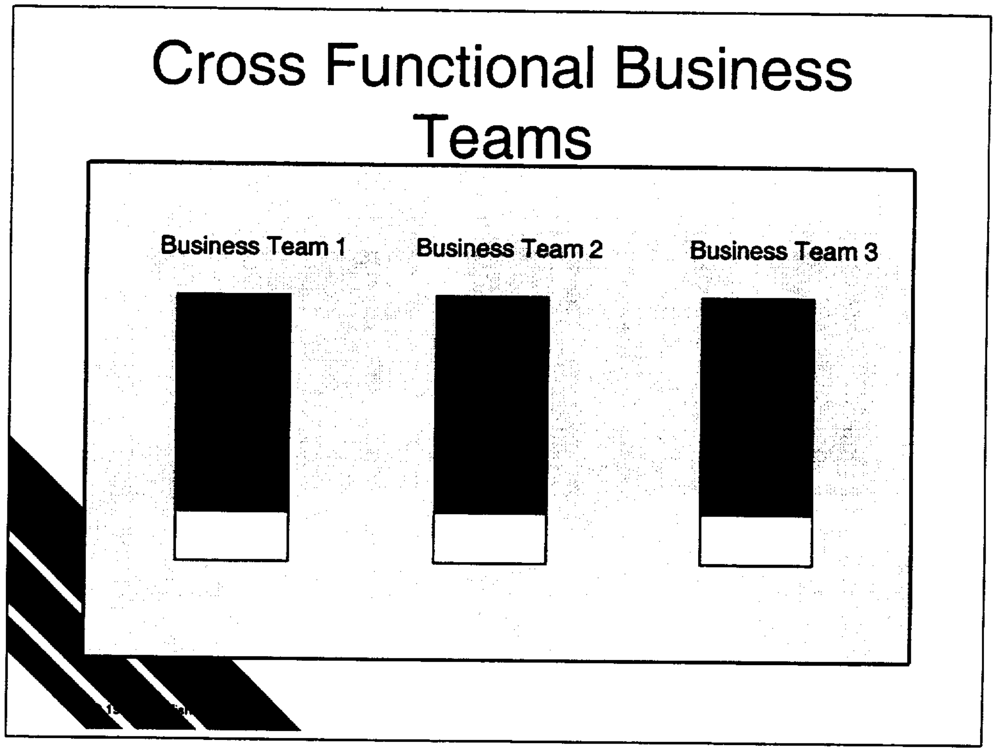
Traditional Functional Silos

A traditional organization is often organized in functional silos such as those listed on this slide. Many organizations find that although this structure is efficient for learning transfer inside the silos, cross-organization learning is difficult.



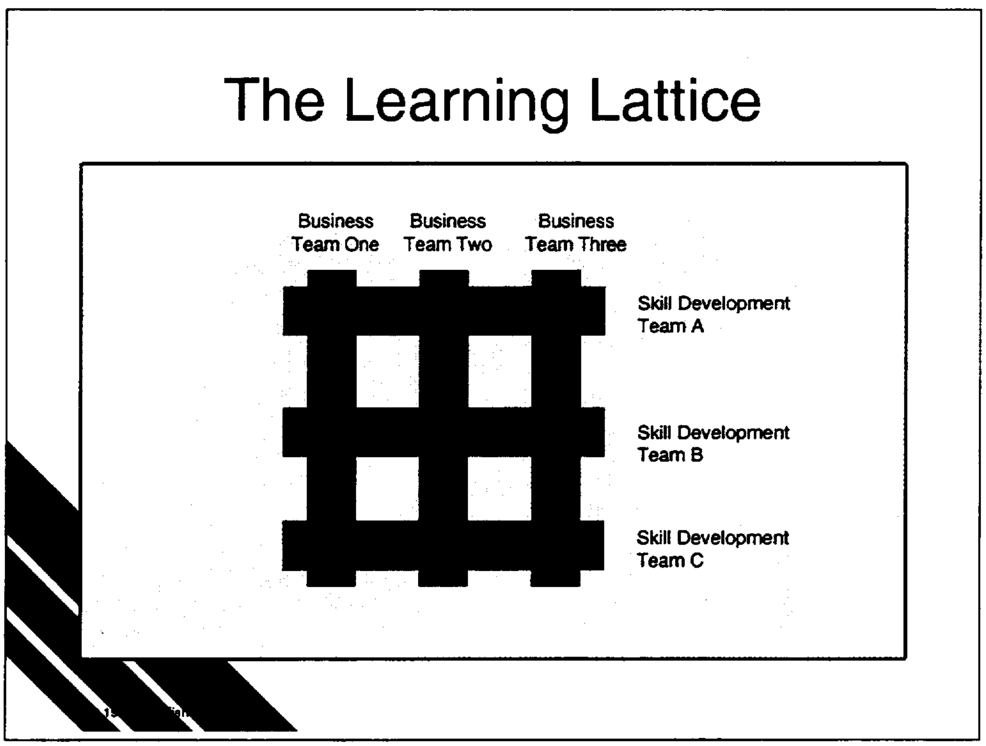
Cross Functional Business Teams

To improve cross-organizational knowledge transfer, many organizations have created cross-functional business teams as illustrated on this slide. A business team may be composed of representatives from several different functions who must work together to launch a new product, manage a project, or serve a particular customer base, etc.



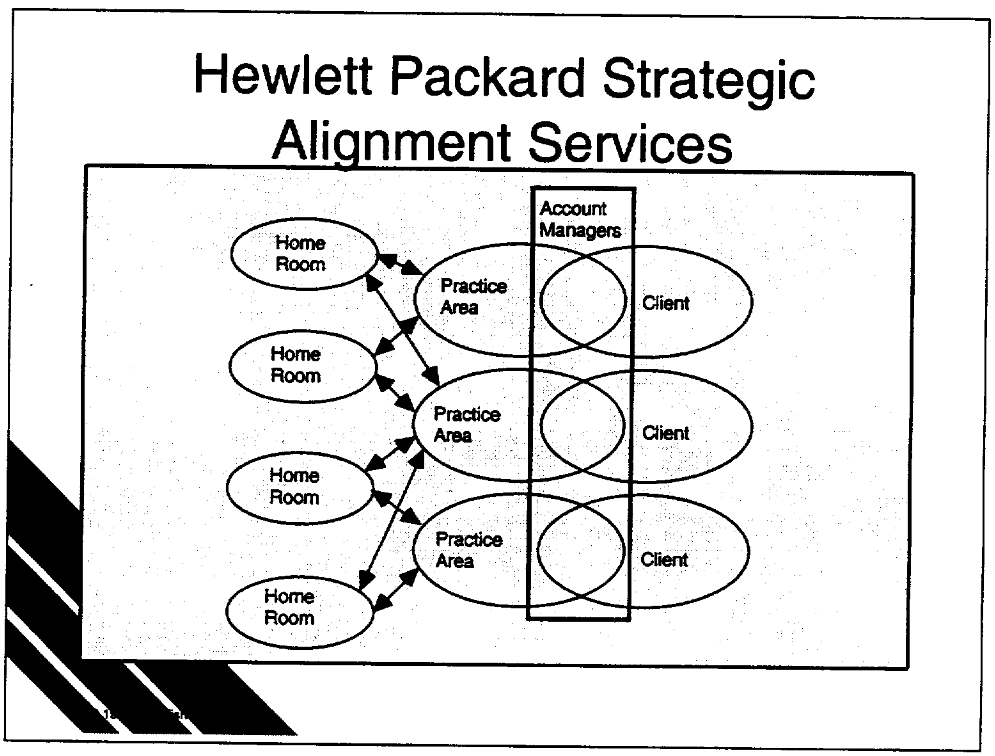
The Learning Lattice

In our research for the book *The Distributed Mind* (AMACOM 1998), however, we found that a number of organizations such as Hewlett Packard and The Port of Seattle had developed an organizational structure that went beyond cross-functional work teams. This structure, what we call "The Learning Lattice," integrates representatives from the business teams into skill development teams for the specific purpose of learning and maintaining skills. This maintains the learning efficiency between people with similar responsibilities while assuring the integrity of the business team model. Unlike the matrix model of the past, the learning lattice does not have dual reporting relationships which typically results in confusion.



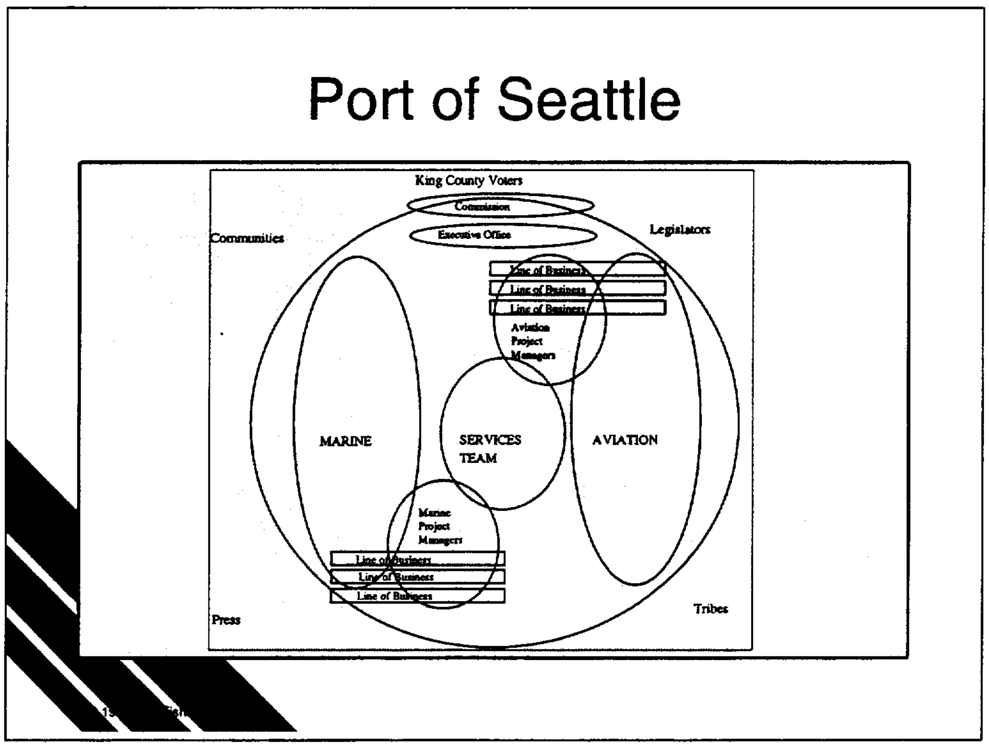
Hewlett Packard Strategic Alignment Services

This is an example of how Hewlett Packard modified the learning lattice structure for one of their internal consulting groups. Practice areas are cross-functional consulting teams composed of statisticians, organizational development practitioners, technicians and mathematical modelers who interface with Hewlett Packard clients through account managers. The home room is established for people with similar technical disciplines (mathematical modelers, for example) to learn about new technologies and share learning across the practice areas. In organizations such as this, learning transfer is more rapid and effective as empowered consultants can immediately use what they learn in the client system.



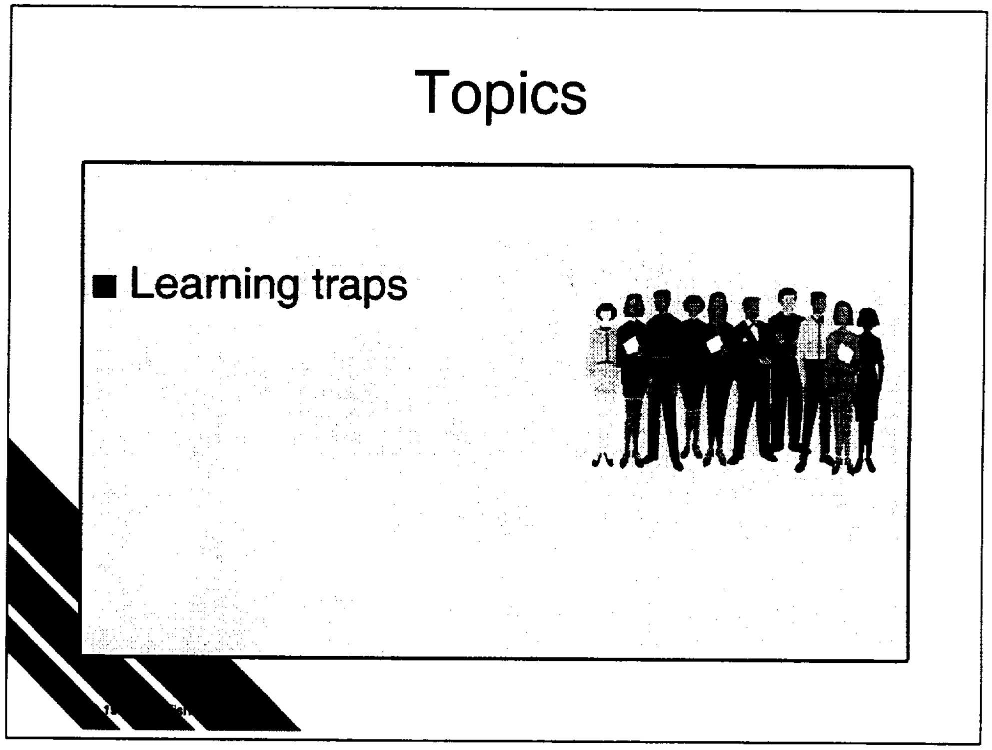
Port of Seattle

This is an example of how The Port of Seattle has used the learning lattice. Lines of business are cross-functional organizations with project managers, engineers, accountants, attorneys and other specialists. Each technical specialty meets either in their service team or with other project managers for technical knowledge transfer activities such as training.



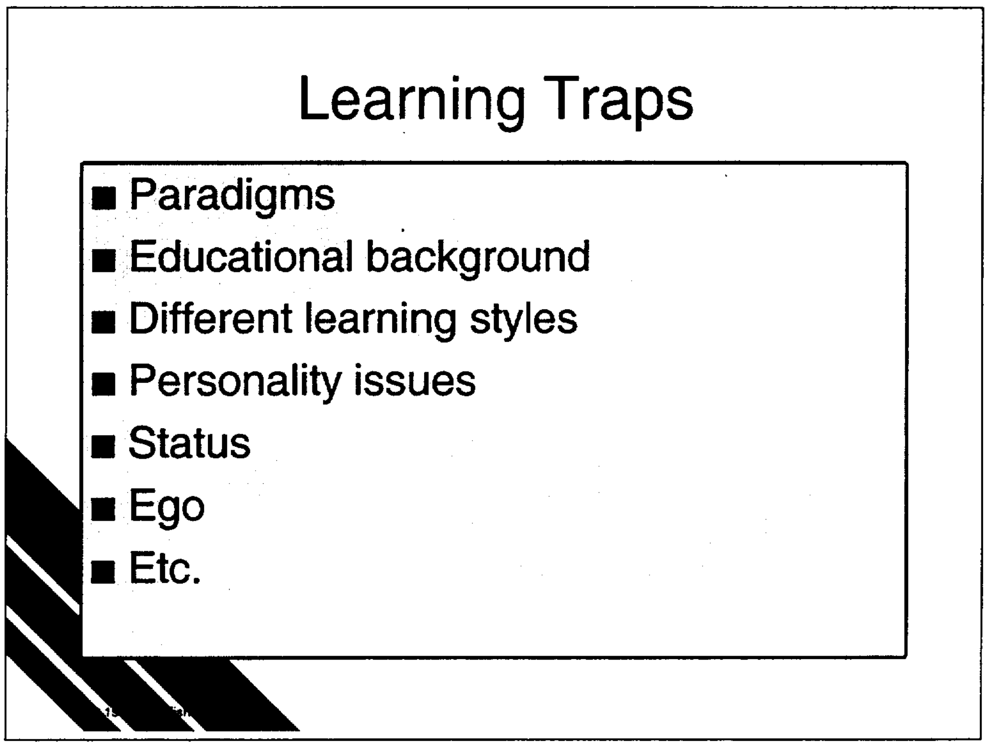
Fifth Topic

The next topic is learning traps.



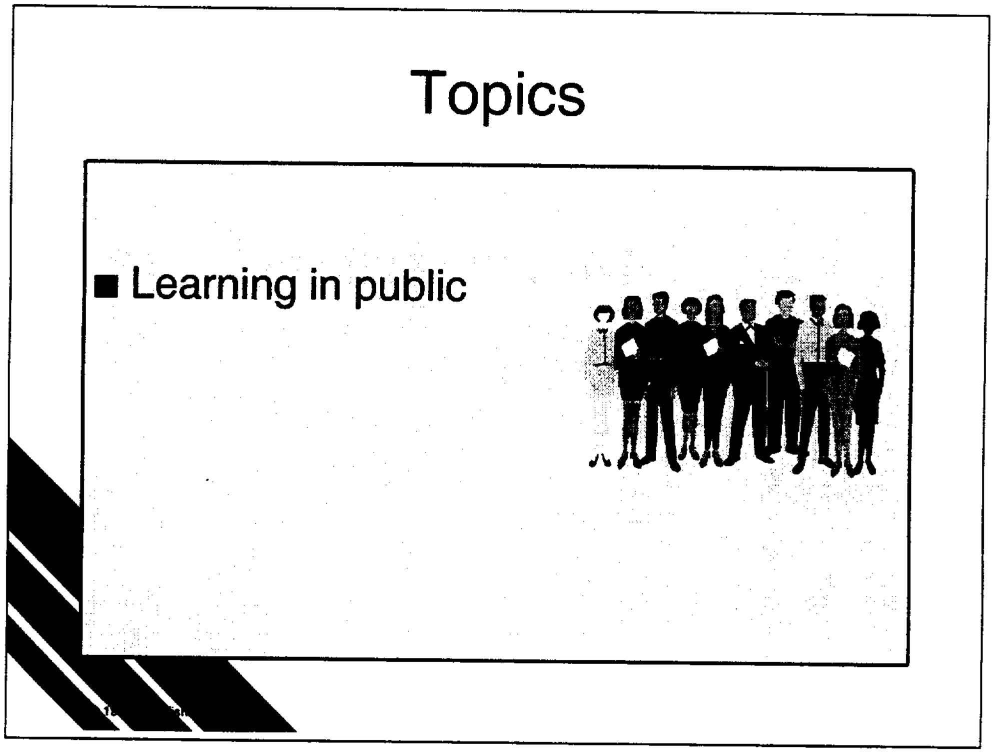
Learning Traps

In addition to organization structure, leadership behaviors and learning protocols, there are a number of other things that affect the human factors associated with the learning environment. For example, differing paradigms about learning, different education backgrounds, learning styles, cultures, personality, status and ego issues, all affect learning transfer and can create noise in the transfer system.



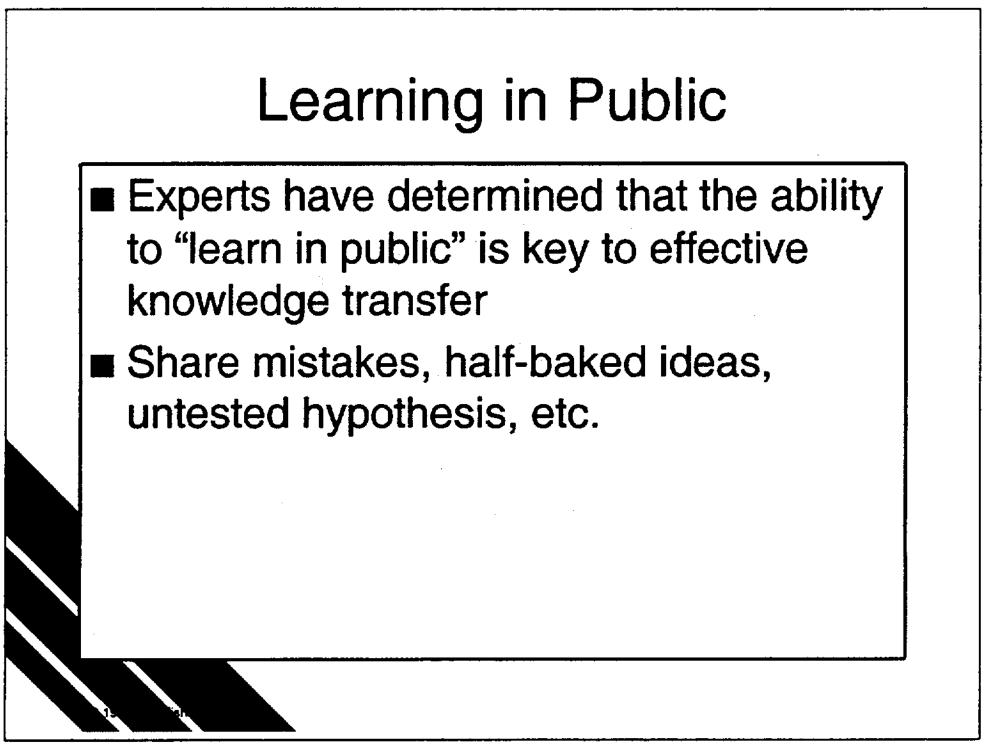
Sixth Topic

The next topic is learning in public, a very common learning trap.



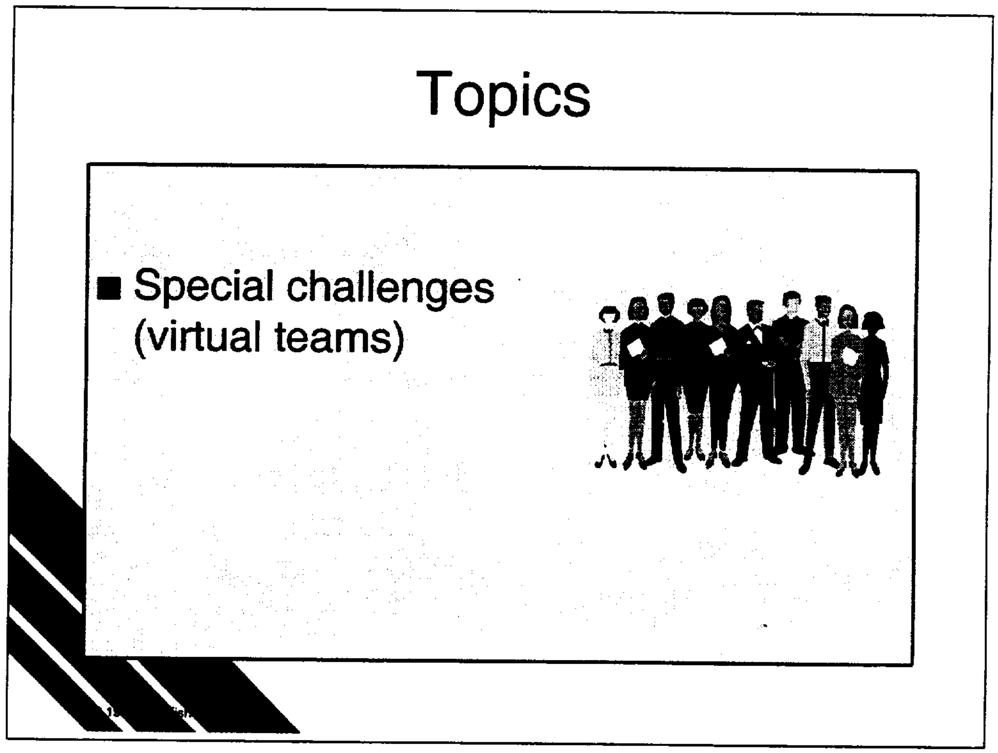
Learning in Public

In high tech organizations or other operations using a variety of specialists, a common learning trap is the inability of people to "learn in public." Many engineers, scientists, and other professionals have been taught over their careers not to share incomplete thoughts, untested hypotheses, or mistakes with their colleagues. This severely limits the real-time learning transfer between these people. Some organizations have found that they have to create forums for people to share half-baked ideas in a non-critical environment. This facilitates the generation of new ideas and accelerates the learning process, even though it is often inconsistent with the existing paradigm of learning.



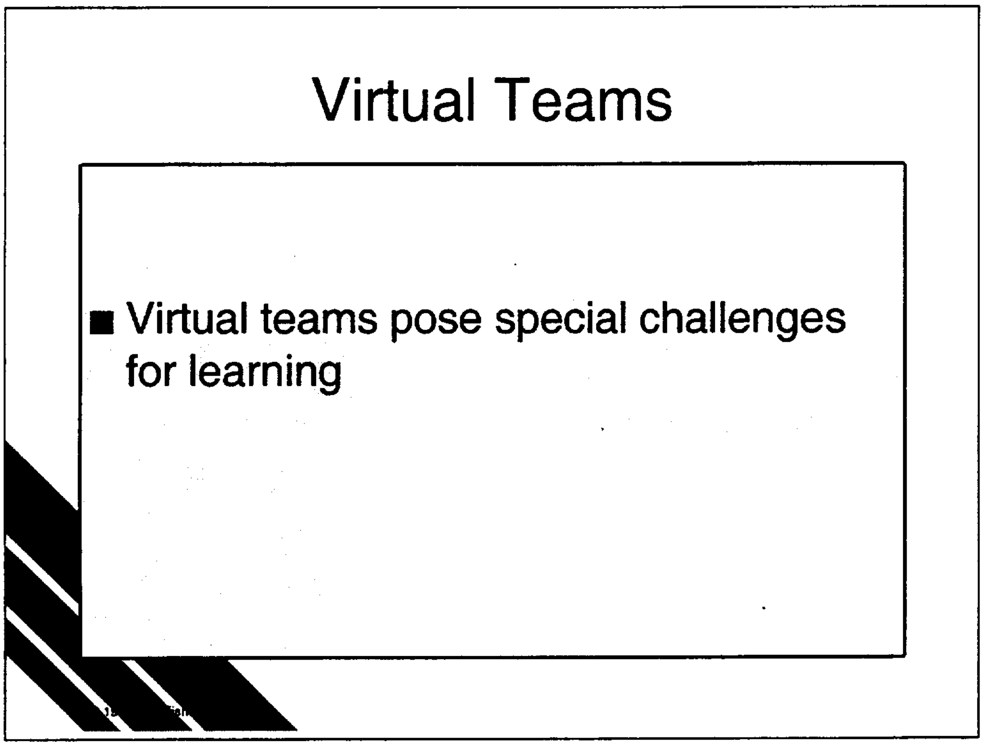
Seventh Topic

The final topic deals with the special challenges of virtual teams.



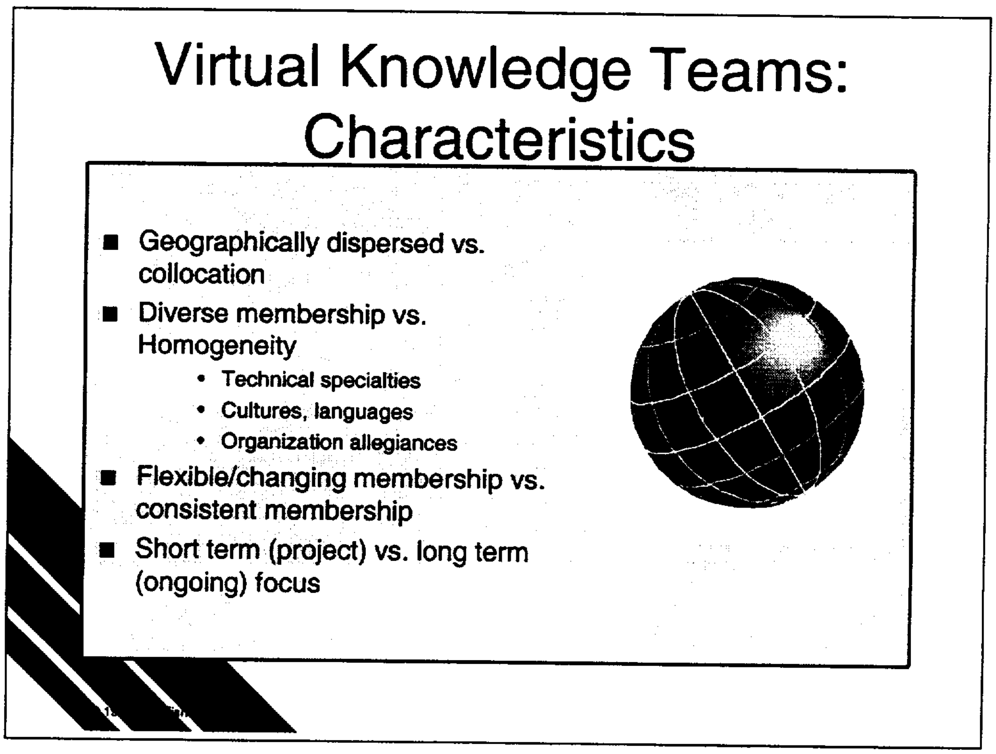
Virtual Teams

Virtual teams (geographically dispersed) have special knowledge transfer challenges.



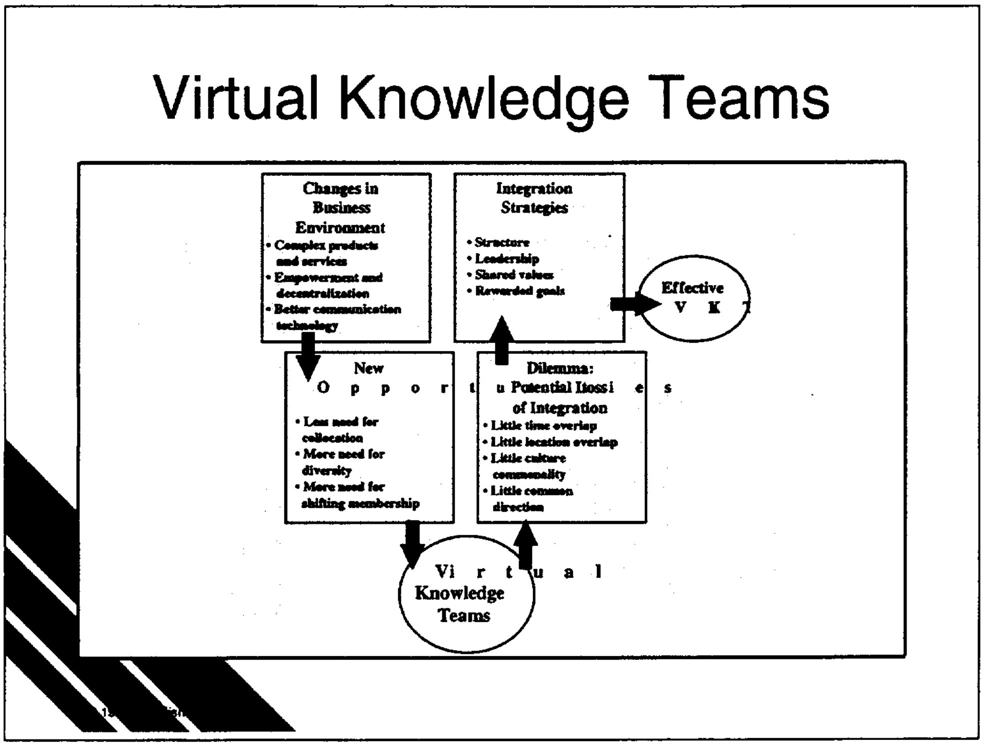
Virtual Knowledge Teams: Characteristics

This slide shows the defining characteristics of a virtual knowledge team, including geographical dispersion, diverse membership, inconsistent membership, and short-term focus. These characteristics make it very difficult for virtual knowledge teams to incorporate things like the learning lattice organization structure or other knowledge transfer techniques more common in collocated operations.



Virtual Knowledge Teams

Virtual knowledge teams are expanding rapidly. Changes in many business environments require it and we expect this trend to continue well into the future. Therefore, our challenge is to determine how to make these organizations more effective through some of the strategies we have discussed.

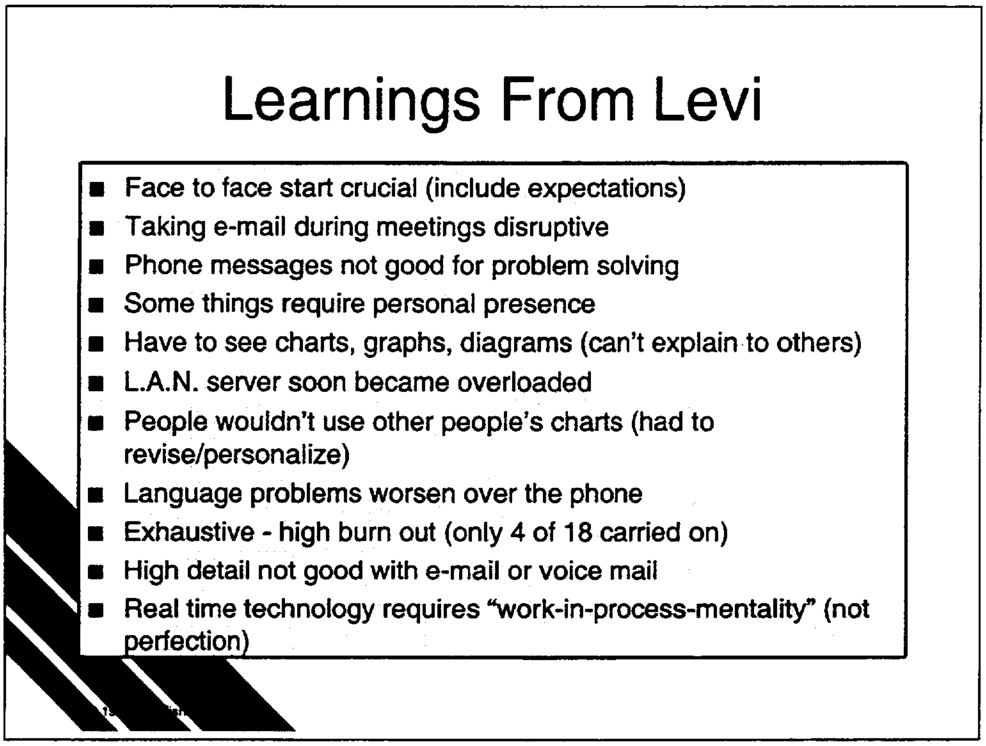


Levi-Asia Pacific Marketing Operation Design

Here is an example of a Levi virtual knowledge team established to determine marketing strategy for the Asia Pacific territory. As you will note, it has dispersed membership and the other characteristics of virtual knowledge teams.

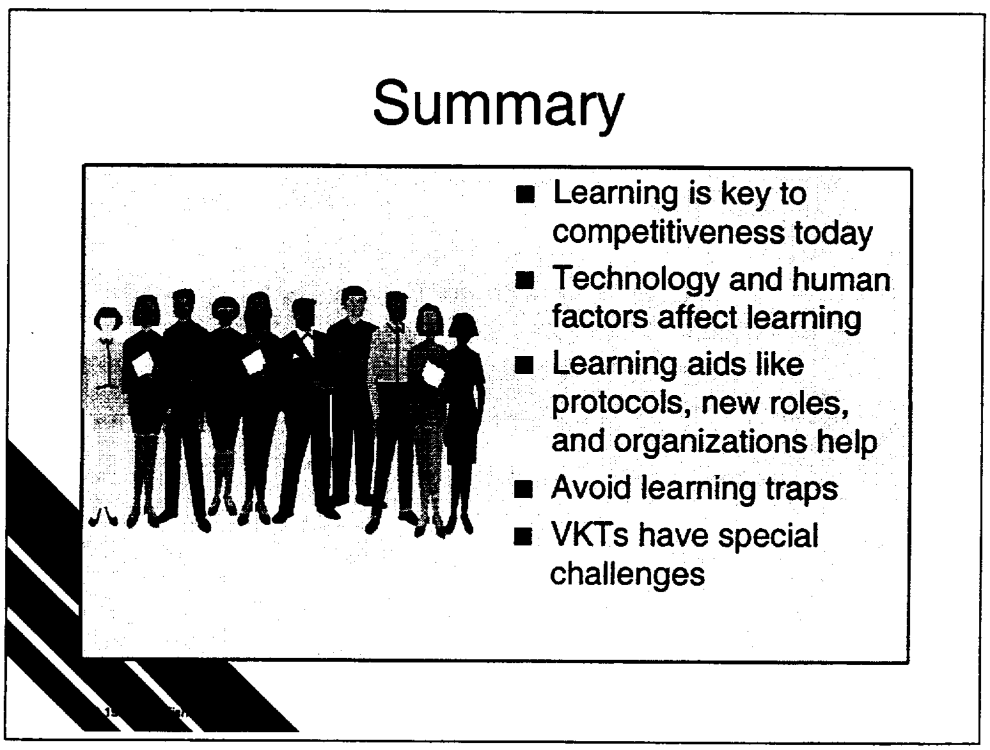
Learning From Levi

Levi learned that there were a number of things critical to making virtual knowledge teams effective. One was a face to face start-up. Others included appropriate use of technologies and special paradigm shifts required of participants.



Summary

We know that learning is key to competitiveness today and that technology and human factors affect the ability of an organization to learn. Advanced training technologies alone are insufficient. Learning aids like technology protocols, new leadership roles, and organization structures like the learning lattice, help. Learning traps must be avoided. In particular, the ability to learn in public seems to be important. Virtual knowledge teams are becoming more common, but have special challenges that we must learn to deal with. If we are to be effective in creating cultures of learning, we must look at human factors above and beyond the advancement of leading edge training technologies.



Visit Us

Please visit us at our website: www.thefishergroup.com.

Human-Systems Interaction for Immersed Training

Randy Stiles
Advanced Technology Center
Lockheed Martin Missiles & Space
Palo Alto, CA

Human-Systems Interaction for Immersed Training

Randy Stiles
Advanced Technology Center
Lockheed Martin Missiles and Space
Palo Alto, CA 94304-1191

Human-systems interaction is the means by which a human interacts with computing and other hardware systems to accomplish tasks. Included as human-systems interaction tools are multiple modes of interaction that address human senses, such as speech dialog capability, real-time visualization, haptic feedback, etc. More important to effective interaction is a design based on the planned human tasks. This presentation relates the interaction design established for our Virtual Environments for Training (VET) program, covering the classes of interaction, tradeoffs, and sets of training tasks for our VE software. The VET program, funded by the Office of Naval Research, has developed a virtual environment architecture focused on authoring instruction for virtual environments. The VET instructional simulation, pedagogical agents, and VRML world models can be authored to provide immersive team training. The Lockheed Martin Advanced Technology Center in Palo Alto, CA is the prime contractor for VET, with USC Information Sciences Institute and USC Behavioral Technology Labs as collaborative partners. See http://vet.parl.com/~vet/

Human-Systems Interaction for Immersed Training*

NASA Workshop on Advanced Training Technologies and Learning Environments
March 9-10, 1999

Randy Stiles
Advanced Technology Center
Lockheed Martin Missiles & Space
O/L922 B/255 3251 Hanover St.
Palo Alto, CA 94304-1191
(650) 354-5256 Email: randy.stiles@lmco.com
see http://vet.parl.com/~vet/

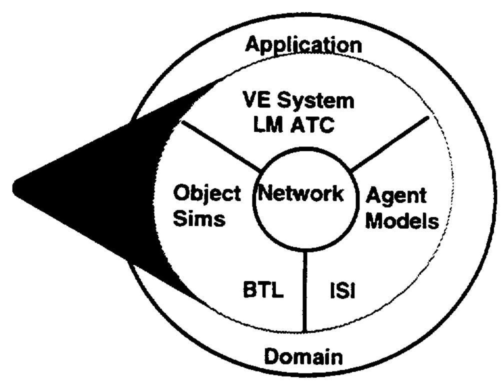
*** Funded by Office of Naval Research**
Contract No. *N00014-95-C-0179*

Virtual Environments for Training

- The VET program is sponsored by the Office of Naval Research, Dr. Helen Gigley, COTR. Some of the concepts are derived from earlier work funded by James Fleming at the USAF Armstrong Labs, Brooks AFB.
- According to Durlach, et al., in the 1995 National Research Council study on virtual environment technology[2], *the most immediate challenge at hand is one of integrating the existing technology into a working system, along with other elements of VE construction software.* With the VET software, we have addressed this challenge for VE training, developing a prototype system that allows development and delivery of instruction in a virtual environment.
- The Lockheed Martin Advanced Technology Center (ATC), as prime integrator, directed design, integration and testing of the software systems, and developed the virtual environment display and interaction capability.
- The USC Information Sciences Institute (ISI) developed our pedagogical agent capability and intelligent agent system that uses multi-modal input and output (speech and visual) to instruct students or act as a missing team member.
- The USC Behavioral Technology Labs (BTL) developed our intelligent simulation-based training capability. BTL software supports authoring and delivering object behaviors for the virtual environment.

Virtual Environments for Training

- **Sponsored by Office of Naval Research, Dr. Helen Gigley COTR, derived from initial work funded by USAF Armstrong Labs**
- **Lockheed Martin Advanced Technology Center (ATC)**
  - **Virtual environment interaction, integration**
- **USC Information Sciences Institute (ISI)**
  - **Cognitive modeling for agent mentors, team members**
- **USC Behavioral Technology Laboratories (BTL)**
  - **Simulation-based training, simulation authoring, behavior**



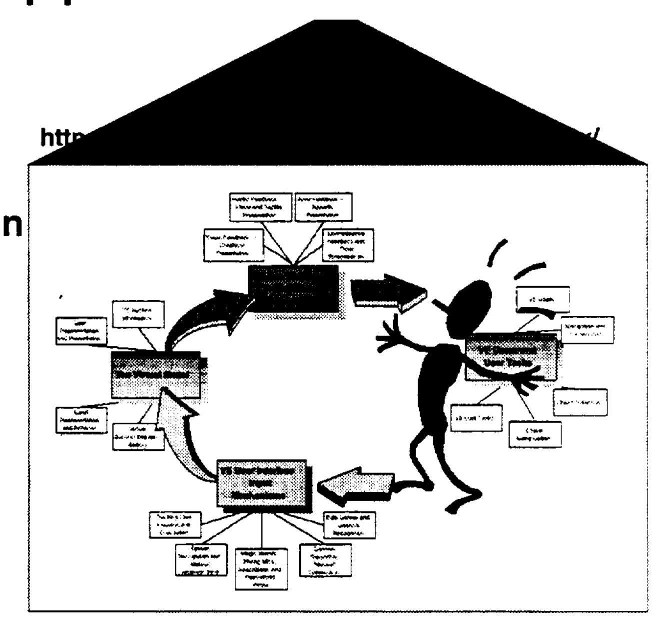
OFFICE OF ONR NAVAL RESEARCH

Interaction Approach

- Our immersed interaction approach for the VET project is derived from the tasks for delivery of equipment operation instruction, and the task of authoring instruction in a virtual environment. This resulted in a requirement to manipulate objects with both hands, and to author (specify) this manipulation. Our domain, that of engine room operations and maintenance for ship-based gas turbine engines, requires fine manipulation of controls, such as throttles, valves, switches, and equipment parts.
- VRML proved to be a good substrate for specifying the manipulation of objects - its sensor approach allows manipulation to be constrained and specified using COTS authoring tools, and the sensors cover a wide range of motions needed. We did extend VRML to allow the specification of 6DOF manipulation.
- About mid-course in the VET project, ONR funded a useful VE interaction taxonomy[3] that itemizes the types and forms of interaction that have become useful in virtual environments. It is the result of visits to a number of VE labs, and a survey of papers on HCI inside of VE systems. We have made use of the categorizations to describe our VE software. This led us to use snapping in the assembly of parts, and to develop two-handed manipulation and pinch glove mappings that are intuitive for manipulating objects.

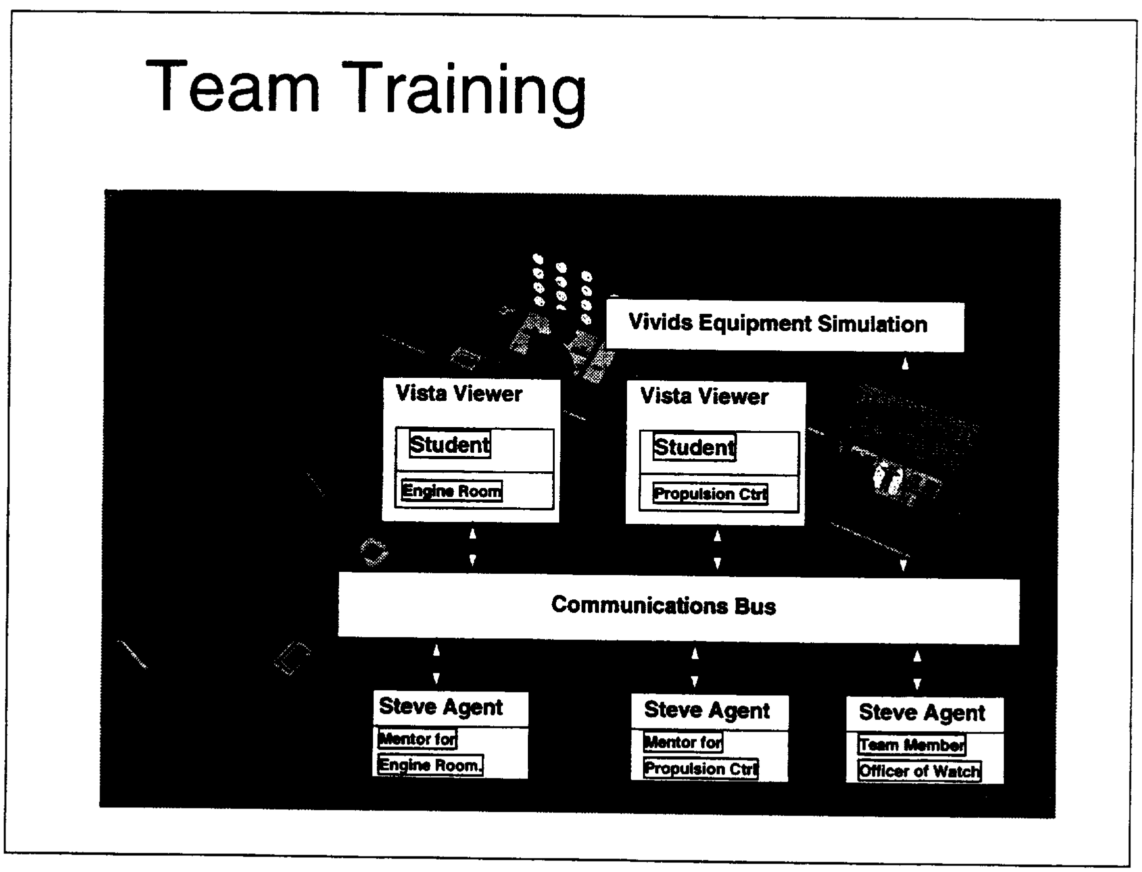
Interaction Approach

- **Derived from tasks for equipment operation and maintenance domain**
- **Derived from tasks for authoring instruction**
- **Derived from VRML Standards**
- **Derived from ONR VE Usability Taxonomy**



Team Training

- Most maintenance and operations tasks were conducted while working with others. We needed to realize a Team Training Testbed to test VE training approaches, as well as an instance of team training for the GTE domain.
- We developed a component-based architecture, called the Training Studio, that has three main components: the Vista virtual environment display, the VIVIDS equipment simulation, and the STEVE pedagogical agents.
- The Training Studio is designed to support training activities for any number of participants. Each participant has an associated Vista Viewer display environment, where all interaction with the student in the 3D world takes place. Equipment and other objects in the environment are simulated by VIVIDS, a descendent of BTL's RIDES simulation authoring system. Multiple team members and mentors are simulated using Steve, the Soar Training Expert for Virtual Environments. One Steve agent is assigned to each participant as a personal mentor. As students manipulate objects in the world, VIVIDS simulations change the world accordingly. These changes are sensed by Steve agents, which can intervene, explain or demonstrate tasks[4].

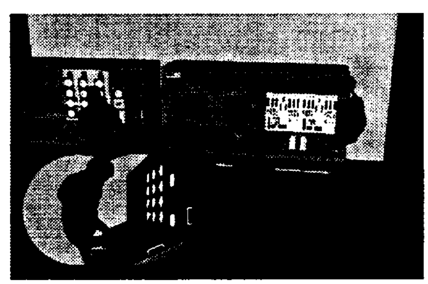


VE User Tasks

- There are really two user categories for the VET system - student users and instructional development users. Because of this, VET did not simply address the GTE domain only, but generalized to a level needed for authoring instruction in a VE. This required more forethought and underlying development.
- A common user task for students that emerged was manipulation of equipment, so we developed both a means to author manipulation and to actually manipulate equipment using immersed VRML.
- Casualty control procedures in the GTE domain involved a team of people carrying out actions on equipment and talking to each other. This influenced our entry into spatial dialog approaches, where the visual context is used with the spoken context, and influenced us to characterize and model team objectives and tasks in the VET pedagogical agents. We proceeded to support dialog for individual tutoring, where the student can ask for explanation, demonstration, and identification, and since speech acts were part of the team interaction, we supported mixed initiative dialog, where questions can be asked of the mentor and speech acts between team members can be recognized.

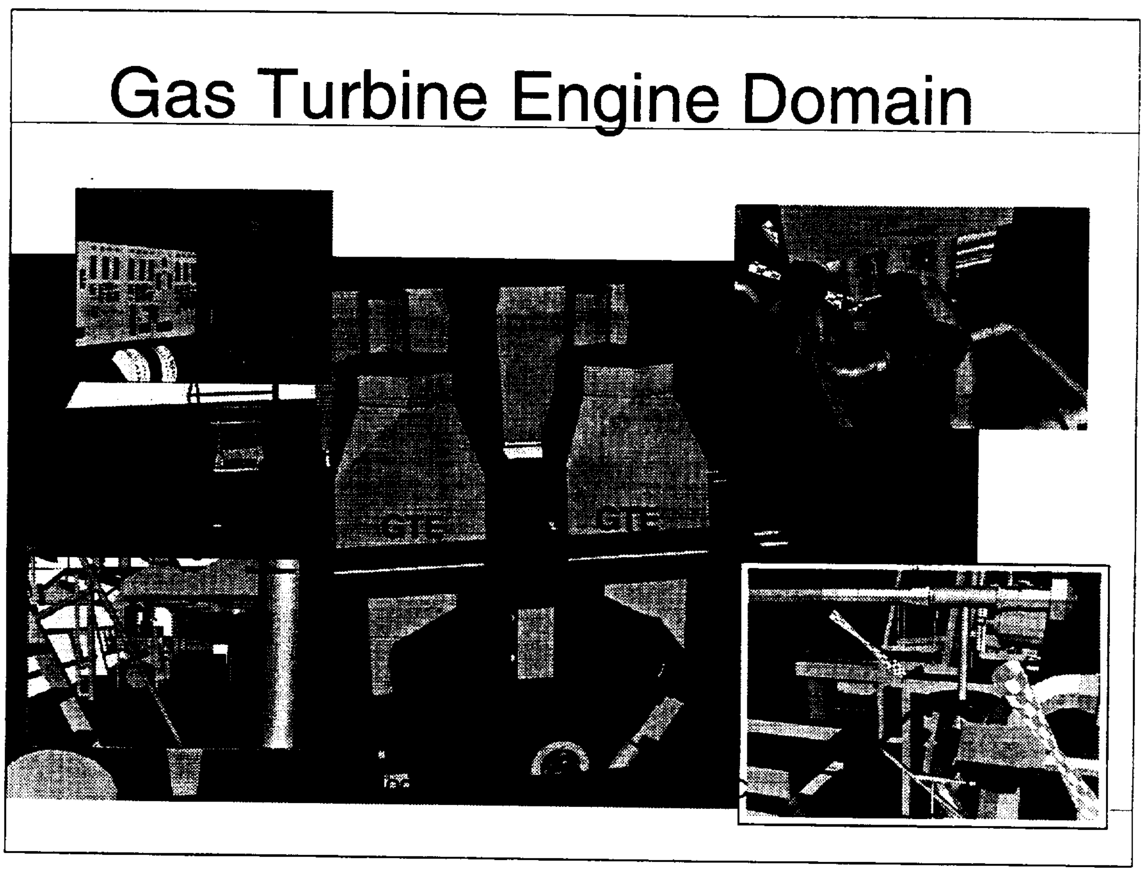
VE User Tasks

- **Authoring across domains for VE Instruction**
- **Student manipulation of equipment**
- **Team tasks with real or synthetic team members**
  - **Individual Tutoring**
  - **Mixed Initiative Dialog**
- **Voice and visual communication**
  - **Mixed initiative dialog**
  - **Task-oriented dialog**



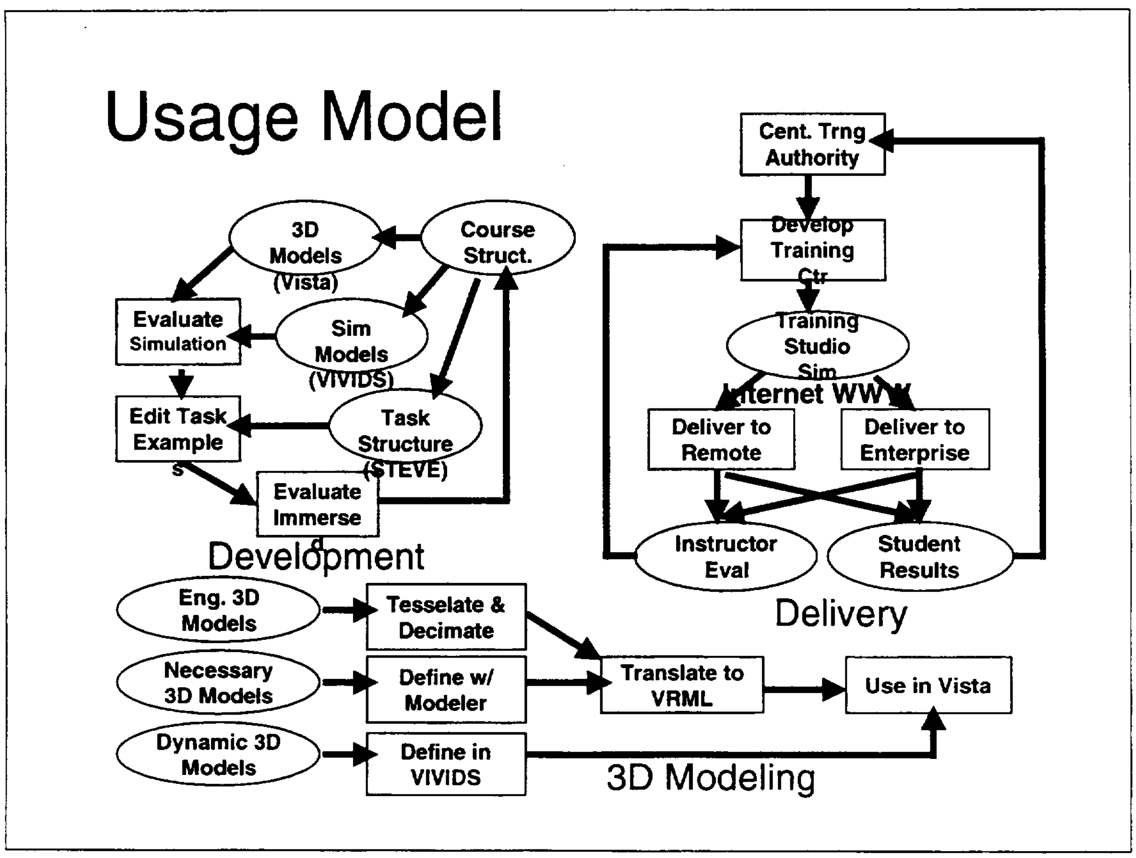
Gas Turbine Engine Domain

- The Gas Turbine Engine domain influenced our choices for interaction. The Navy's Arleigh Burke Main Engine room became the spatial setting for our instruction on casualty control, and issues of student navigation, view transitions and equipment manipulation became important.
- The GTE systems are surrounded by a large amount of piping and metal framework, so having agents move around to instruct students on the use of equipment had to be addressed for the agent, as well as the student.
- We built upon the concept of VRML viewpoints to arrive at immersed viewpoint transitions that move the viewer from the current view, orient the viewer down the path to be traveled, move him along the path, then orient him in the desired way at the end of traveling the path.
- For agent movement, we created agent waypoints away from equipment, and the agents searched for the shortest path on safe waypoints to get to the equipment where they are to give instruction.



Usage Model

- In the delivery usage model, a demand for training in a particular topic comes from a central training authority. This request is handled at a Training Center which has the appropriate hardware, software and personnel to develop the training material. A course using the Training Studio is developed and fielded to remote locations. Development of a course for a given domain starts at a Training Center by defining the course structure. Here the instructional developers get an idea of what material needs to be covered, what simulations may be helpful, what the tasks in the domain may be, and which 3D models will be needed.
- For the course development usage model, initial training simulations are developed in 2D using VIVIDS and at the same time members of the development group assess existing 3D model resources and obtain these. Soon after initial simulations are developed, they are augmented to update the 3D scenes in Vista. At each stage, the simulation is evaluated by the development group to see if it matches real system behavior in the areas significant to training transfer.
- The usage model for using 3D models recognizes three forms of models: engineering (CAD) models and already existing visual simulation models which can be purchased, the set of necessary 3D models not already existing, and those 3D models which, because of the nature of the domain or its explanation, must be dynamic (mutable while using the virtual environment).



Interaction for Training

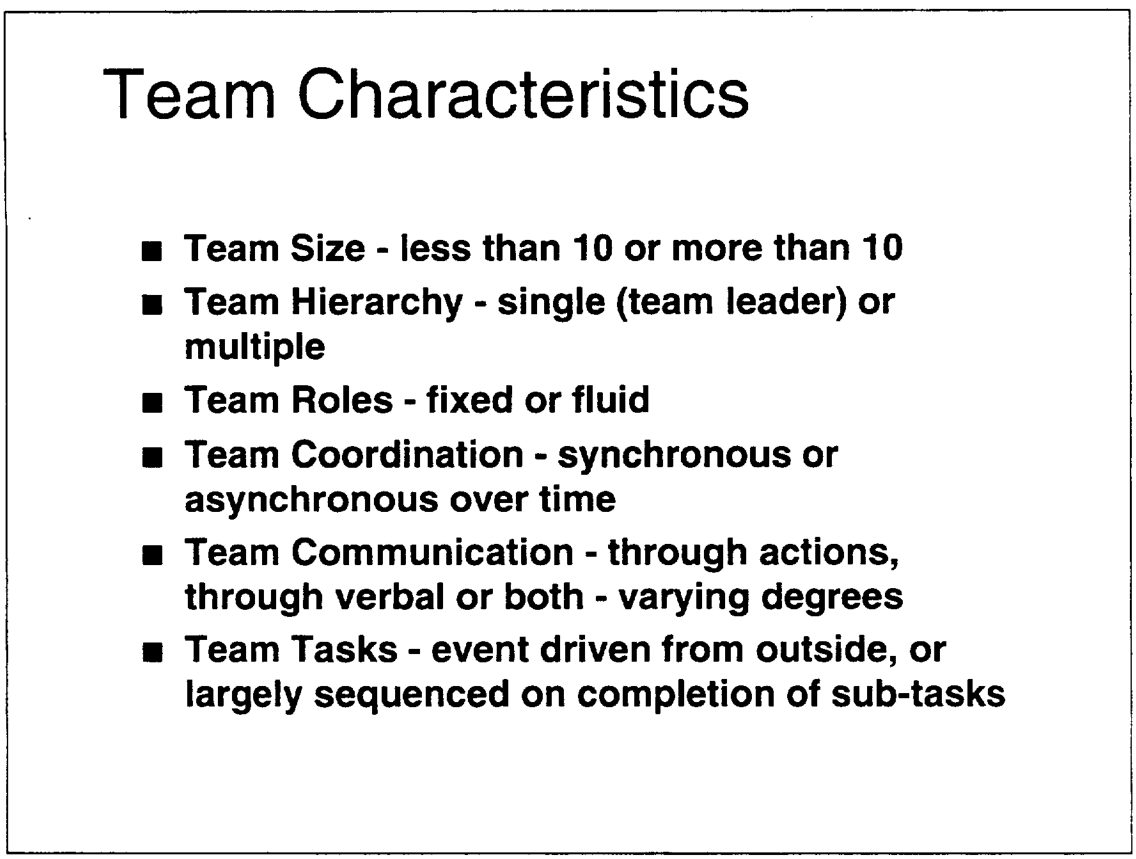
- In designing the Training Studio, we have considered the dynamics of both mentor-student and team member to team member interactions, which are important aspects of team training. Because any task role can be undertaken by an agent or a human, the system must accommodate the following communication modes: human to human, human to agent, agent to human, agent to agent, and either agent or human to all. Since the team works to affect their environment, this in turn involves identifying and resolving issues surrounding human to virtual scene, human to human and human to agent interactions within the immersed environment.
- Human-to-scene interactions are those that support working with objects in the scene carry out tasks. We support events for manipulation of objects, and found it necessary in a team setting to support networked update of the visual scene, as a form of human-to human interaction using the scene.
- Human-agent interactions are those that involve communication between the human and agents. We support spoken dialog with agents, as well as giving agents information about object selection and manipulation, and the objects that the student is viewing. The agents in turn visually show students where objects are, how to use them, and verbally explain their actions.

Interaction for Training

Networked Interaction	Qualities supported by Training Studio
	Visual Communication (Vista) (networked VRML scenes) Verbal Communication (radio, phone) (free-form dialog)
	Visual Communication (Steve) (gaze, gesture, focus) Verbal Communication (Steve) (task-oriented dialog)
	Object Simulation (VIVIDS) (Instructional Simulation) Discrete Manipulation (Vista) (Immersed selection of objects) Continuous Manipulation (Vista) (Immersed VRML, 6DOF object movement, constrained: plane, cylinder, sphere)

Team Characteristics

- There are a variety of team types, and we worked to categorize the problem space as part of our design for interaction. Team size of less than ten, with a single leader, was selected because that is the typical small-team configuration for the military (and space missions). We resolved to address fixed team roles during synchronous team actions rather than more complex asynchronous team interactions, and fluid roles because this is the typical team type for dealing with emergencies and other critical situations.
- We chose to address multi-modal team communications where there may be speech acts, visual acts to modify the scene, or visual acts to inform other people, since this is closer to the way teams act in synchronous situations.
- We also chose team tasks that are previously defined and task based because these are the types of approaches used for critical situations.



Human-Human Interactions

- Based on the type of team training we chose to address, it became necessary to enable visual communication between participants and to show the location of participants.
- To remediate during team training, it was evident that we should allow scene changes scoped per participant for cases of individual instruction on a team role without interfering with the rest of the team. We also support cases where instruction happens "in front of" the rest of the team since this can lead to learning by the rest of the team members.

Human - Human Interactions

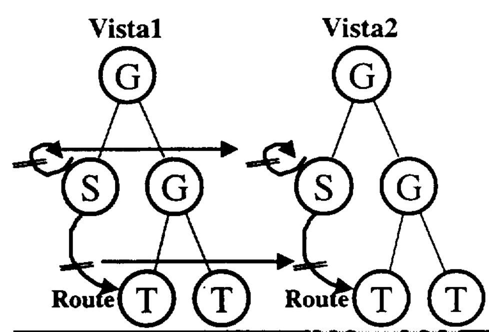
- Task - allow team of people to use VE for instruction
- Enable Visual Communications
  - Visual scene representations are updated between participants
  - Participant representations are shown for each person
- Multiple viewer coordination and consistent updates

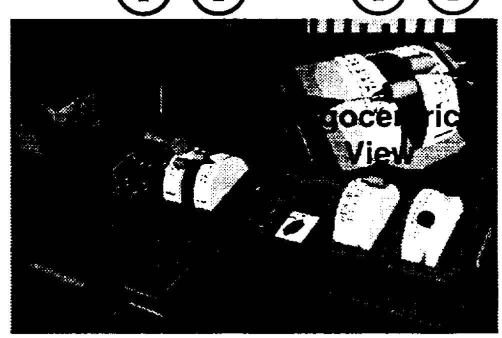
Shared VRML

- Shared update of VRML scenes are important for visual communication between participants and with agents or the simulations.
- There are a number of elaborate approaches to enable shared update of VRML scenes, most of which require modification of VRML files to support updating objects across the network. Our approach is sensor based and allows the use of unmodified VRML files in a shared setting.
- The only way a human can interact with the VRML scene is through the sensors. When we load a VRML scene into more than one Vista virtual environment display, we enable routing sensor outputs and histories across the network.
- So when Student 1 moves a sensor, such as a door, the sensor output is routed to Student 1's VRML nodes, as expected, and the update is sent across the network to another Vista display for Student 2, where it affects the VRML nodes in that scene. To allow Student 2 to manipulate the same object without jumping, we also send the sensor history across the network to all sensors of the same name. By doing this, we ensure that whoever touches the sensor next will be able to move the sensor based on its last shared position.

Shared VRML

- **Multi-user, human-human visual interaction through sensors**
- **Unmodified VRML can be updated over network**
- **Sensors output along routes in scene graph, changes are routed across network**
- **Destination of sensor route receives changes**
- **Sensor histories are updated with network data**





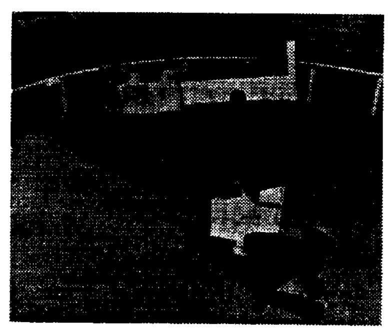
Networked Distribution

- VRML is a 3D exchange format that is an analog to HTML - it was designed for transmission across the Internet. Updates to 3D models used in instruction can be propagated across the enterprise using the Intranet or the Internet.
- In this way, the usage model for just-in-time distribution and update of training across the enterprise can be realized.
- Most models from CAD systems can be translated to VRML. There are tools to translate CATIA, IDEAS and ProEngineer files into VRML.
- VRML not only expresses 3D geometry, it also allows specifying user input and behavior through script nodes, so if changes are made to the types of interaction associated with a CAD file, this can be propagated on the net as well.
- For testing of the VRML scenes, the same models you interact with using a browser can be used immersed.
- Vista downloads models over the Internet, allowing maintenance at a central site.

Networked Distribution

- **VRML is analog to HTML which supports exchanging, interacting and modifying 3D scenes over the Internet**
- **Vista Supports VRML 97 Standard**
- **Developed Immersive Use of VRML 97 for Sensors**

COTS Browsing of VE Scenes



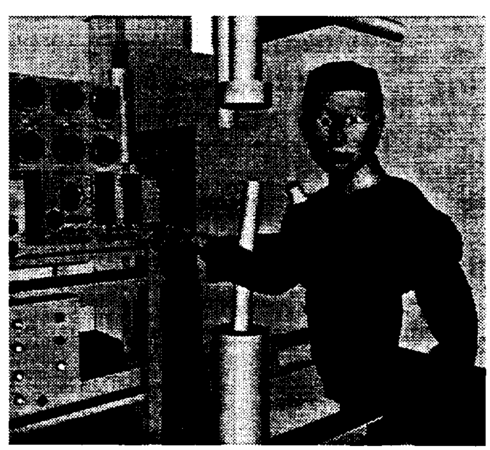
Immersed Viewing of Manufacturing Animations

Human-Agent Interactions

- Human-agent interactions are a critical part of delivering team training inside a virtual environment; more so than in conventional 2D software where the agent has the potential to show how a task is really carried out, and to spatially locate objects for the student.
- The Steve pedagogical agent uses interaction events created by Vista when the student selects objects or changes their viewpoint. For the agent to talk about a control or to visually show the use of a control, the student must be viewing the object. Using visibility events, Steve can perceive if the student sees the object he is using in instruction. Similarly, Steve can perceive the use of objects by the student.
- The human and pedagogical agent also interact using speech. The student can carry out speech acts during team activities, and the Steve agent will recognize these. Also, the student can ask meta questions to aid learning, such as asking the agent to demonstrate a task, asking why a task is being done, and asking to be allowed to complete a task.
- The Steve agent also interacts with the student by gazing at objects or pointing at objects to direct the student's attention, as well as twisting its torso to properly manipulate controls.

Human - Agent Interactions

- **Agent demonstrates tasks for students visually in spatial setting, directing attention and allowing location of items**
- **Agent uses VE interaction services**
  - **Selection and manipulation of objects by human**
  - **Identify when objects of interest go in or out of field of view**
- **Verbal communication**
  - **Important capability for surrogate team members**
  - **Verbal cues for mentoring students**
- **Non-verbal attention cues**
  - **Gesture**
  - **Gaze**
  - **Head/body positioning**

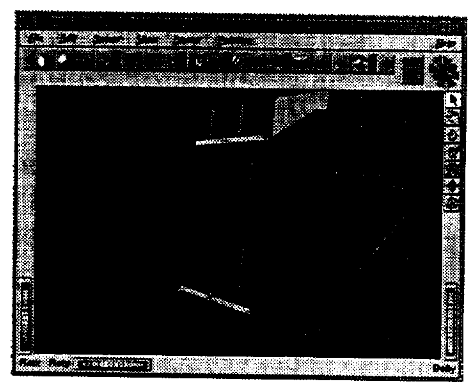


Human-Scene Interactions (Development)

- To support the task of authoring human-scene interactions, we chose VRML as a standard 3D authoring format.
- Human-scene interactions are specified and constrained ahead of time (during instructional authoring) by using COTS VRML authoring tools.
- VRML provides geometric sensors that allow specification of how geometry can be manipulated.
- VRML provides the means to specify viewpoints onto the scene so that important views can be used for instruction.
- VRML provides a way to test the visibility of objects using visibility sensors, and to sense where the viewer is, using proximity sensors.

Human - Scene Interactions
(Development)

- **Task - developing VEs for instruction**
- **Generic scenes composed and modified live with VRML**
- **VRML authors can insert interaction constraints into VRML models so that the objects can be re-used**
- **Viewpoint transitions provide instructor control of student view while limiting disorientation effects**
- **Proximity sensors and visibility ensure events are viewable by student**

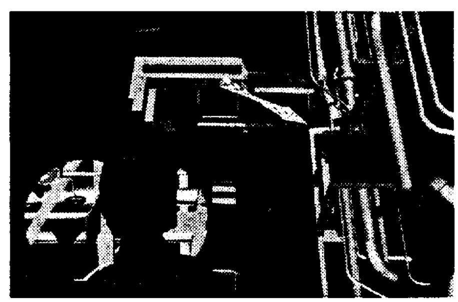


Human-Scene Interactions (Delivery)

- To support the task of delivering VE for instruction to one or more people in a dynamic scene, we needed to provide a way to find out an object's bounding sphere, world location, and to change its color state.
- Using a networked update message for VRML fields, it is possible to change color, transform and activity of named VRML nodes in the scene during delivery of instruction.
- The modifications of object state can be scoped to particular participants during delivery of instruction.

Human - Scene Interactions
(Delivery)

- **Task - delivering VE for instruction to one or more people**
- **Vista services enable agent and simulation to support human-scene human-human interactions**
  - **Supports queries regarding participant and object states and relative positions**
  - **Support changing, constraining objects**
- **Per participant control and interaction for training, including networked update of VRML**

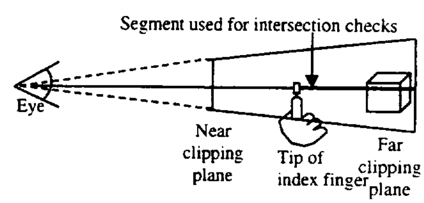


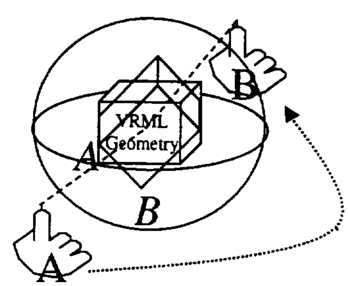
Exocentric and Egocentric Manipulations

- For manipulation of VRML sensors, we support two modes of manipulation - projected and direct. Direct requires initial intersection and selection of the object with the hand cursor. Projected requires intersection of a line from the eye to the finger to the object during selection using a pinch glove. It is similar to selection using a conventional mouse, except this is done while immersed.
- Projected mode is exocentric in that selection takes place distant from the body. It is suitable for coarse manipulation of an object in a larger scene context. Oftentimes, familiarization tasks fall into this category.
- Direct mode is egocentric in that selection takes place near the body using the arms and hands. It is more suited to fine manipulation of objects and detailed placement of objects, preferably using the depth perception possible with stereopsis. Most equipment manipulation tasks fall in this category.

Exocentric and Egocentric Manipulations

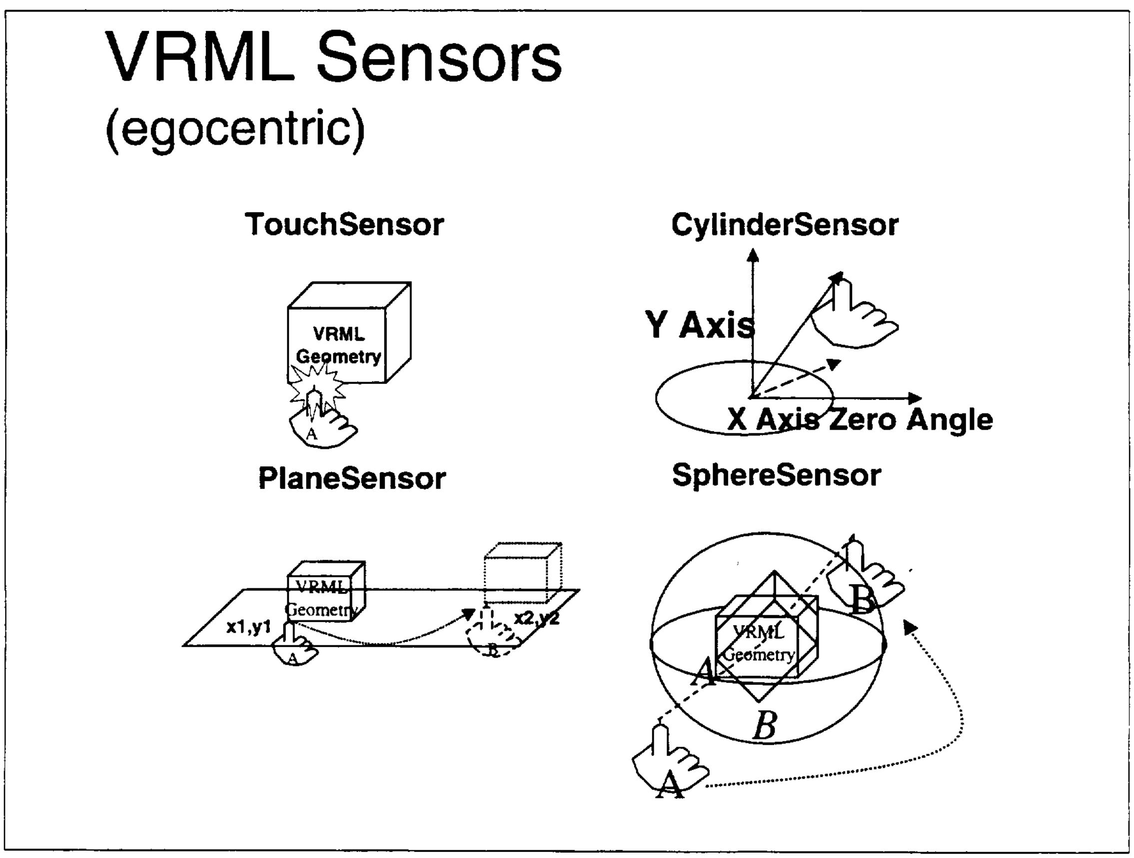
- **Projected mode is exocentric**
  - **From eye to finger to object defines intersection**
  - **Suitable for coarse, distant manipulations relative to other objects**
- **Direct mode is egocentric**
  - **Direct mapping of 6 DOF data to sensor**
  - **Suitable for fine manipulations using stereopsis**





VRML Sensors (Egocentric)

- TouchSensor - Determine if selector is over objects associated with sensor. Determine if selector used on objects is associated with sensor.
- Anchor Node Similar, except action always routed to file load with URL.
- Implemented with simple intersection for Projected, Direct.
- PlaneSensor - Projected: Intersection point derived from eye to finger to plane defined by sensor is used Direct: Translation in world is mapped onto closest point on plane defined by sensor, allows precise movement on plane.
- CylinderSensor - Projected: Intersection of line defined by casting a line from the eye to the index finger onto cylinder, and this determines radius, intersection onto plane determines rotation Direct: Translation of fingertip in CylinderSensor frame determines vector from center of cylinder to fingertip, and XZ component vector determines rotation about Y axis.
- SphereSensor - Projected: Intersection of line defined by wand onto sphere determines radius, point on plane determines rotation Direct: Translation of fingertip in world is used to get vector for quaternion, and quaternion defines rotation about Sphere Center.



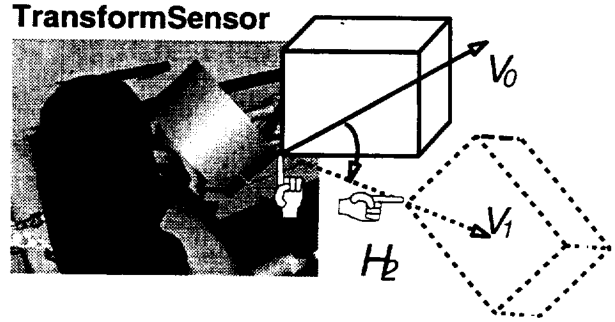
VRML 6DOF Sensors (Egocentric)

- The VRML TransformSensor is a way to specify full movement while immersed. In the VRML standard, there previously was no sensor that allowed full motion. During training, the VRML scene can be authored so that a few designated objects can be picked up and manipulated, and the rest are by default inert to manipulation. By specifying this interaction in the 3D file format, one begins to get composability and re-use of objects.
- The SnapSensor is used in tandem with the TransformSensor. When the immersed person moves an object into the range of the SnapSensor, the SnapSensor updates the object's TransformSensor, thus changing the Transform to snap into place. This is useful when assembling or dis-assembling equipment immersed because it is forgiving of small errors in motion when there is no force feedback. Snapping objects is a common approach even in the 2D interfaces, such as snapping to a grid, because it allows correct placement.

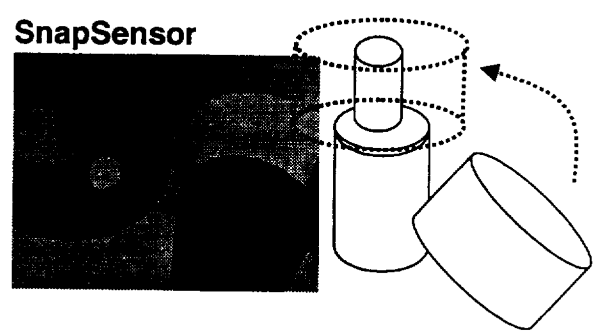
VRML 6DOF Sensors
(Egocentric)

- **TransformSensor**
  - **Designate objects for full translation and orientation**
  - **No 2D mapping from mouse, full use of 6 DOF**
  - **Useful in cases without force-feedback**
- **SnapSensor**
  - **Designate 6 DOF regions that types of objects snap into in 6 DOF**
  - **Useful in assembly tasks where parts would snap in place, or extreme precision needed**

TransformSensor



SnapSensor



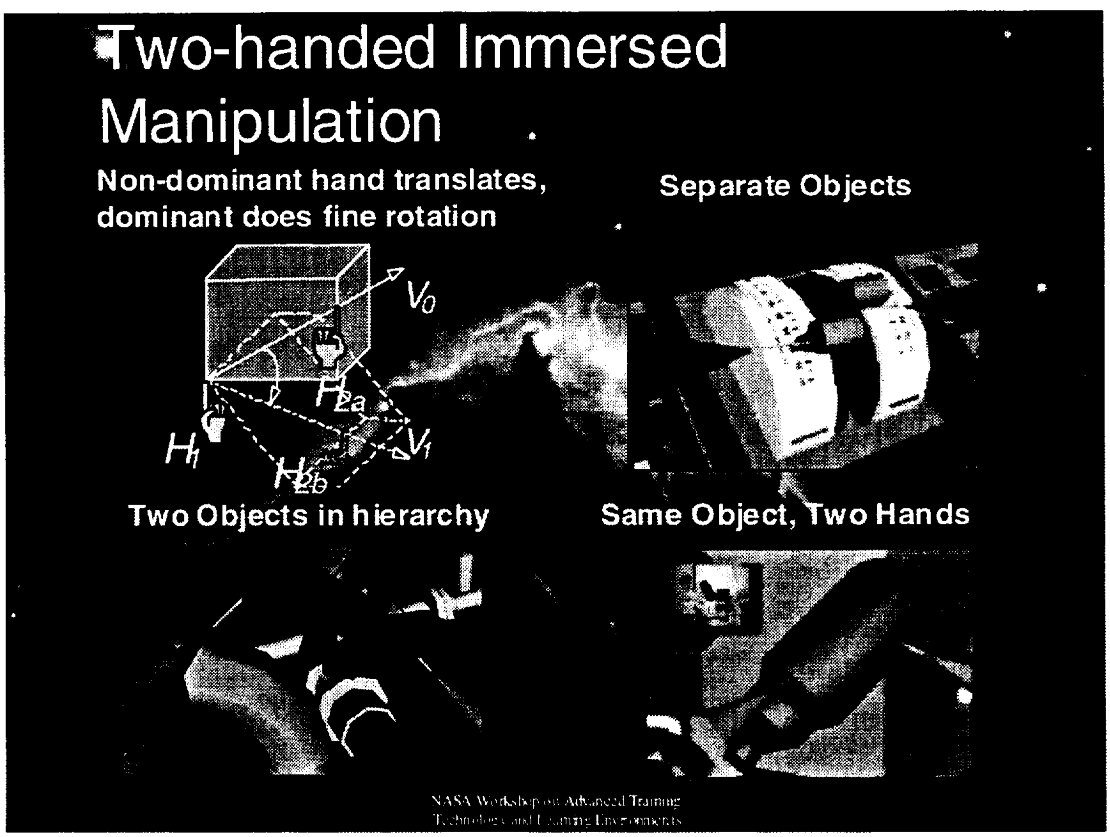
Two-Handed Immersed Manipulation

Two-handed manipulation of objects is often required for maintenance. We have developed extensions to VRML that allow full 6DOF manipulation with two hands based on the workbench example at Stanford, where the non-dominant hand holds the object, and the dominant hand is used for fine orientation of the object.

Two-handed - objects unrelated by hierarchy: The manipulation of one object does not affect the other. Two completely different sensors could be driving the two objects, and this involves the notion of more than one cursor.

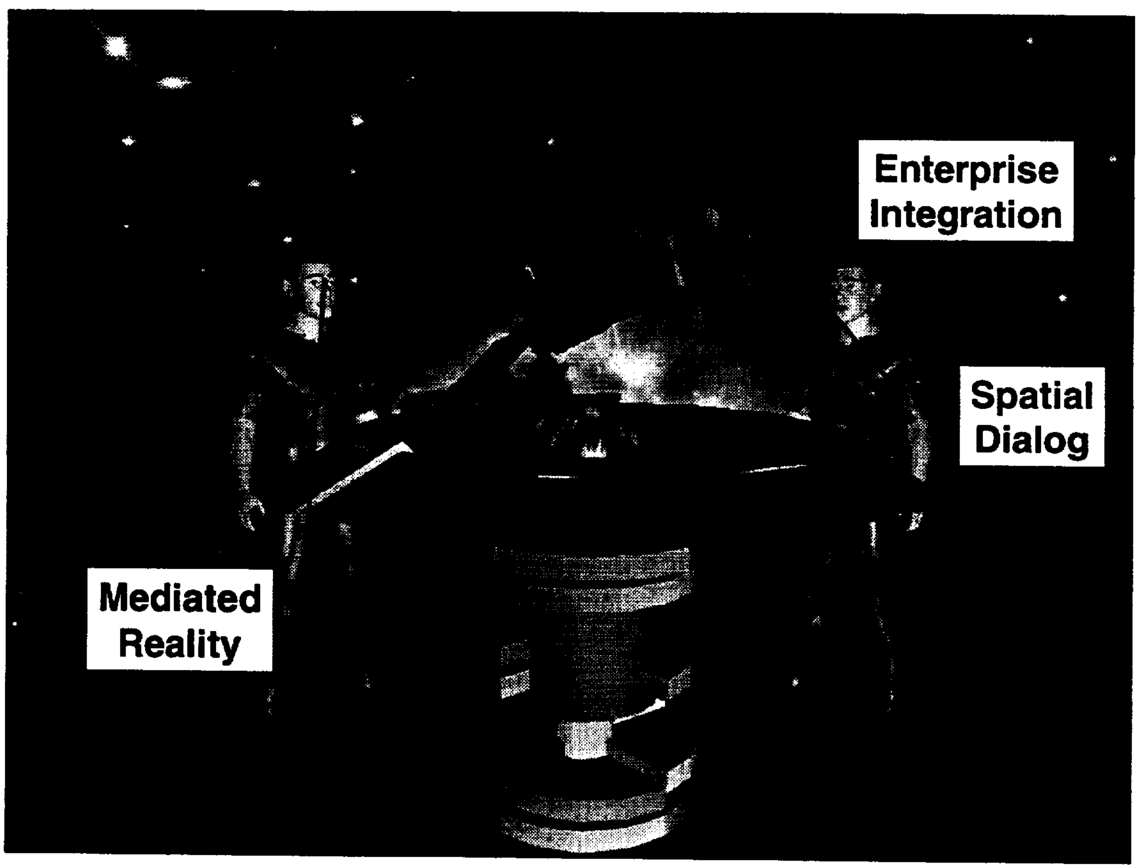
Manipulating objects in same hierarchy: Manipulation of one object affects the other. Although the objects are related to each other in a hierarchy they should follow the user's hand. This involves hierarchical updating of sensors. The sensor lower down in the hierarchy compensates for sensor higher up. The non-dominant hand is used for trivial task and the primary task is done by the dominant hand

Manipulation of same object: Involves the notion of two cursors on the same object. There is smooth switching between two-handed and one-handed manipulation. Some tasks require the use of both hands, e.g., certain equipment may require that both hands be used when moving it.



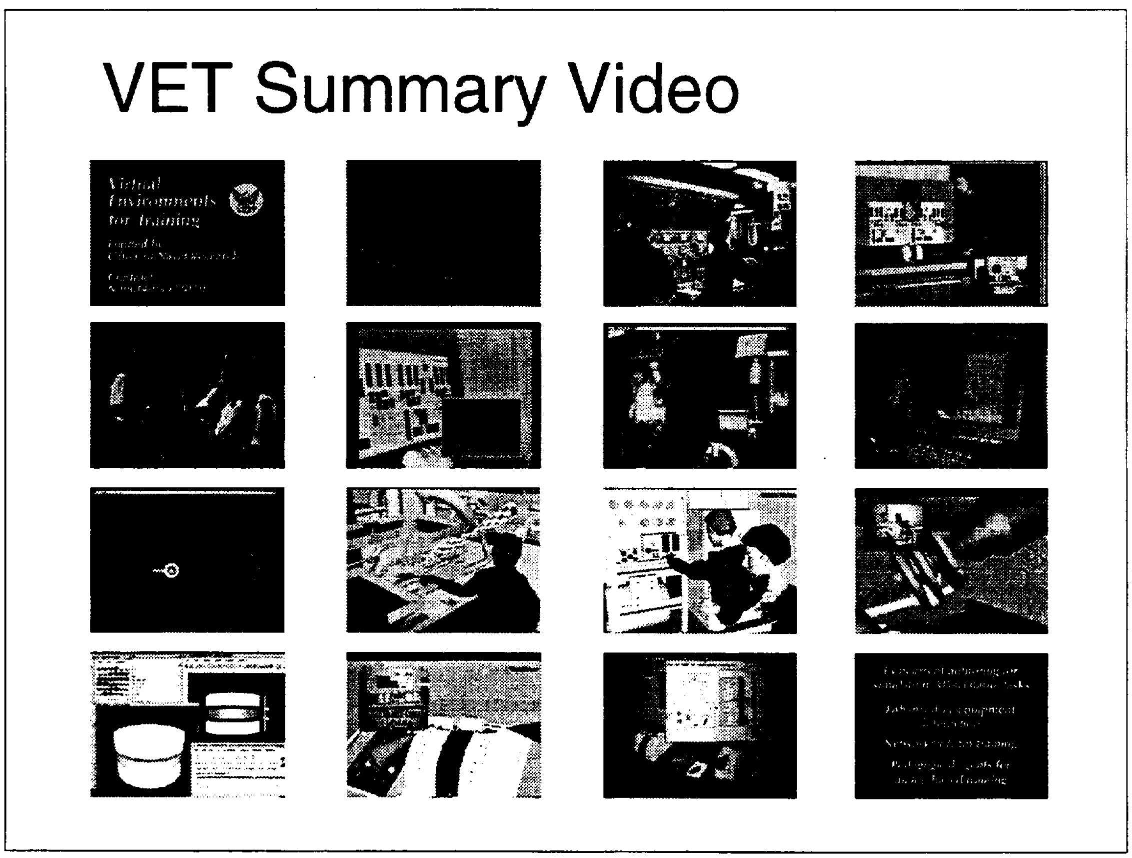
Future Directions

- The future directions for our VET project include mediated reality, enterprise integration and spatial dialog. These technology capabilities can lead to a VE training system that connects to the enterprise through CAD and simulation data, and engages the student in a natural manner.
- Mediated Reality. Our plans include overlaying 3D instruction onto a video background so that the student is not cut off from his real surrounding environment.
- Enterprise Integration. More work is needed to make the translation of CAD models and behaviors into a VE for instruction more seamless. If this goal is realized there will be less impediments to take a modeled system and create detailed training to use the system.
- Spatial Dialog. Spatial cues, such as which objects are in a person's view and which objects they are touching, can add a great deal of context to spoken dialog with the computing system. The selection and viewing of objects provides the "pronouns" in spoken sentences when talking with the agents in the virtual environment. This is a promising research area that could allow much improvement in the human-computer interface for VE systems.



The VET Summary Video

- The VET Summary Video summarizes the project, covering project motivation, project demonstration and project significance.
- The project motivation explains that there are a wide variety of applications that need training without being away from station, and without the large expense of a dedicated simulator.
- The project demonstration shows a casualty control procedure in a Navy engine room dealing with loss of fuel oil pressure. It involves a team onboard the ship that must handle the casualty and shows a student being instructed as part of this team.
- Project significance is summarized by these qualities present in the system; concurrent authoring for simulation, 3D scenario, and tasks, enhanced VE equipment interaction, networked team training, and pedagogical agents for dialog-based training. According to Barfield and Furness, *there is a need for software infrastructure and tools for constructing, managing and interacting within virtual environments*[1]. The VET Training Studio is an example of a VE testbed that does address constructing, managing, and interacting within virtual environments.



Virtual Environments for Training

The VET web page has technical papers on some of the topics in this briefing in PDF format. See http://vet.parl.com/~vet/

References:

1. Barfield, W., Furness, T. A (eds.), *Virtual Environments and Advanced Interface Design*, Oxford University Press, New York, 1995.
2. Durlach, N. I., Mavor, A. S. (eds.), *Virtual Reality: Scientific and Technological Challenges*, National Academy Press, Washington, D.C., 1995.
3. Gabbard, J. L., Hix, D., *A Taxonomy of Usability Characteristics in Virtual Environments*, Technical Report, Dept. Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, 1997. See http://csgrad.cs.vt.edu/~jgabbard/ve/taxonomy/
4. Johnson, W. L., Rickel, J., Stiles, R., Munro, A., "Integrating Pedagogical Agents Into Virtual Environments," *Presence Journal*, Vol. 7, No. 6, MIT Press, Dec. 1998.

Shown are some of the developers of the VET system, from left to right; Sandeep Tewari, Craig Hall, Rich Angros, Mihir Mehta, Allen Munro, Randy Stiles, Marcus Thiebaux, Lewis Johnson, Quentin Pizzini, Jeff Rickel. Not shown; Laurie McCarthy, Carol Horwitz, Erin Shaw.

Virtual Environments for Training

Lockheed Martin Advanced Technology Center

USC Information Sciences Institute

USC Behavioral Technology Laboratory

See http://vet.parl.com/~vet/

for more information

Training Dismounted Soldiers in Virtual Environments

Stephen L. Goldberg and Bruce W. Knerr
Simulator Systems Research Unit
U.S. Army Research Institute
Orlando, FL

Training Dismounted Soldiers in Virtual Environments

Stephen L. Goldberg and Bruce W. Knerr
U.S. Army Research Institute
Simulator Systems Research Unit
Orlando, Florida

Training Dismounted Soldiers
in Virtual Environments

Stephen L. Goldberg and Bruce W. Knerr
U.S. Army Research Institute
Simulator Systems Research Unit
Orlando, Florida

Simulation-Based Army Training

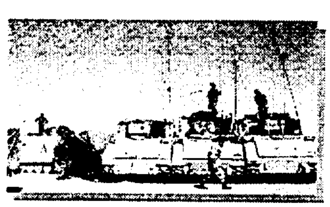
The Army has traditionally trained its combat units through field training exercises. However, the end of the cold war has led to a substantial reduction in military budgets. Cuts in training funds have limited the amount of field training possible. Not only are modern Army weapon systems expensive to acquire and operate, their increased range and speed, relative to their predecessors, has made many existing training facilities too small to be used for realistic field training. Political and ecological considerations have made it difficult to enlarge the training areas.

On a more positive note, during the same period great strides have been made in the development of simulation technology. Particularly important to Army collective training has been the growth of computer networking technology which led to distributed interactive simulations. Beginning with SIMNET, a prototype system fielded in the 1980's, distributed simulations have allowed Army unit training to be brought indoors. The Army has realized that it can no longer rely on field training as its primary mode of training. Instead it is opting for a simulation based approach that incorporates field training but relies heavily on simulation systems to carry much of the training load.

Simulation-Based Army Training

- **Fewer dollars for military training**
- **Increased cost and capabilities of weapon systems**
- **Changes in political and environmental realities**
- **Increased capabilities of simulators**

Switch From Field-Based Training to Simulation-Based



Collective Training in Virtual Environments

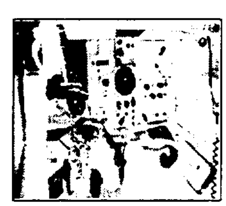
SIMNET, developed by DARPA for the Army, was the first example of a large scale simulation network. Simulation networks distribute the computing load across the various computers on the network. There is no central computer that is controlling the simulation. Standard communication protocols update each of the computers regarding changing events. Each computer maintains its own ground truth. Some of these computers are in manned simulators, others guide computer generated forces that play enemy and supporting friendly forces. Still other computers emulate battlefield systems such as artillery. Distributed simulations have developed terrain databases that can represent any location in the world.

The SIMNET experiment in networking and training has now been replaced by the Close Combat Tactical Trainer (CCTT). CCTT provides a company team training environment that has higher fidelity than SIMNET's. CCTT is currently being fielded throughout the Army. The first sites are at Fort Hood,TX and Fort Knox, KY.

Collective Training in Virtual Environments

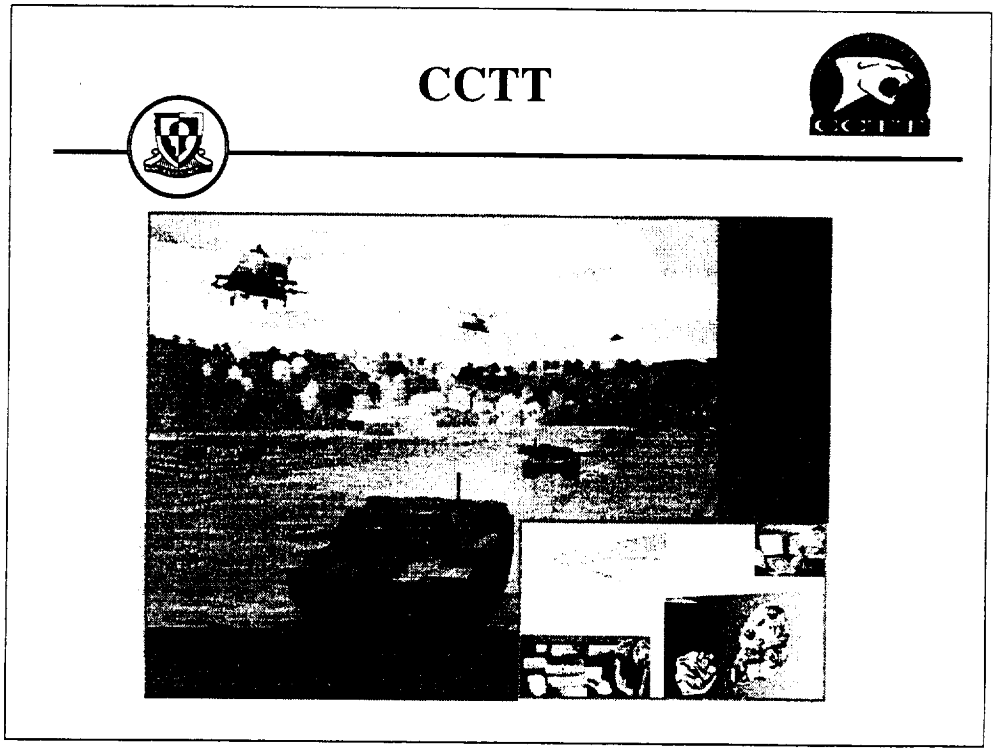
- **Large-Scale Distributed Simulation Networks**
  - **No Central Computer**
  - **Standard Communication Protocols**
- **Terrain Databases**
- **Vehicle Simulations**
- **Computer Generated Forces**
- **SIMNET followed by the Close Combat Tactical Trainer (CCTT)**

→



CCTT

This slide gives you an idea of the level of realism attained by CCTT graphics.



Training Dismounted Soldiers in Virtual Environments

SIMNET and CCTT were designed to train mounted forces, soldiers who fight from within vehicles. As such they have large numbers of vehicle simulators. These tanks and infantry fighting vehicles are only part of the combined arms team the Army employs. Dismounted Infantry play an important role as well. Early experience with SIMNET showed that failure to represent infantry resulted in unrealistic battles. CCTT has a crude Dismounted Infantry Module that permits infantry forces to be represented on the virtual battlefield, but it is difficult to operate and provides little training value. Better ways are needed to bring the dismounted soldier into CCTT.

While it may seem that training dismounted soldiers could best be done on the ground, there are a number of good reasons for developing a dismounted soldier simulation capability. First, dismounted soldiers and mounted soldiers need to be trained to fight together. Second, dismounted soldier simulation would provide a realistic mission rehearsal capability. It would provide a flexible means for elite units to plan and train. A third use for this technology would be as a small unit leader trainer. It could provide soldiers with a considerable amount of practice in a relatively short period of time. Finally, dismounted soldier simulation can be used to develop new equipment and doctrine concepts through virtual prototyping.

Placing dismounted soldiers into simulated environments is a challenge. Virtual reality (VR) technology provides the tools for immersing the individual into virtual environments.

Training Dismounted Soldiers in Virtual Environments

- **CCTT and other Army Simulation Systems vehicle dominated**
- **Inadequate capability to represent or train dismounted soldiers**
- **Dismounted soldier simulation capability needed to:**
  - **Allow infantry to train within combined arms team**
  - **Provide a means for small unit mission rehearsal**
  - **Train small unit leaders**
  - **Train soldiers to conduct operations in urban terrain**
  - **Develop and test new concepts for doctrine and equipment**

Dismounted Soldier Simulation

The Army Research Institute recognized the need for a dismounted soldier simulation capability and began a program of research in this area in 1992. The research has focused on three areas: determining what VR characteristics are necessary to train military tasks; the transfer of knowledge gained in the virtual world to the real world; and determining how best to utilize VR technology for training through testing of possible training approaches.

Dismounted Soldier Simulation

ARI Research Program to:

- **Focus on requirements for individual soldier and leader accomplishment of unit/team tasks.**
- **Determine the necessary characteristics of Virtual Environments technology, including fidelity requirements.**
- **Evaluate transfer of training and performance to the real world.**
- **Consider alternative training features and approaches.**

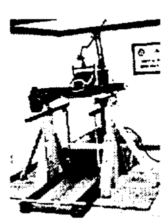
Human Performance in Virtual Environments

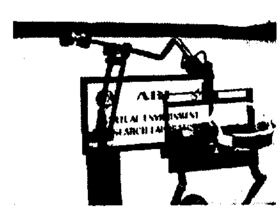
We have completed a significant amount of research on human performance issues in virtual environments. I will not be talking about this research today. We have conducted experiments in each of the areas listed below. If you have an interest in any of these topics I would be happy to provide you with the relevant reports. The last slide in this presentation will contain email and phone information if you would like to contact me. The remainder of the talk will focus on the training and training transfer research we have completed and are contemplating.

Behavioral Research Issues
Dismounted Soldier Simulation

Human Performance in Virtual Environments

- **Determine ability to judge distance**
- **Requirements for stereo vision**
- **Is knowledge of body orientation necessary or helpful**
- **Assess visual system support to task performance**
- **How does method of movement effect performance**





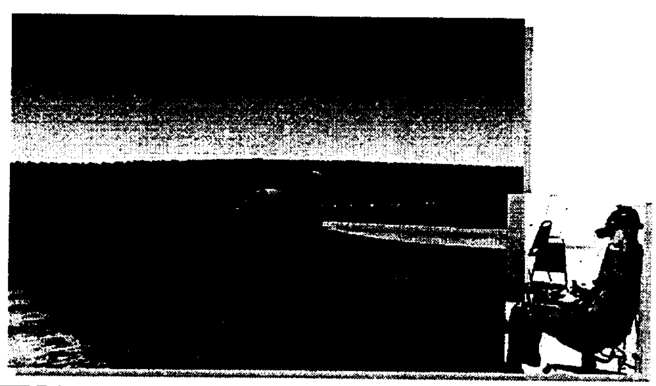
Training in Virtual Environments

Today, virtual environments cannot provide a realistic substitute for the real world. Many VR technologies are immature. Consequently, there are a substantial number of tasks that soldiers cannot perform in a virtual environment that they could in the real world. Movement can be represented but usually requires more attention than in the real world. Some types of locomotion, such as crawling, are highly unrealistic. Also, current tracking systems are not precise enough for marksmanship training. The strengths of virtual environments are their ability to accurately represent spatial relationships, and to allow for interactions between individuals. Our training research projects have investigated questions in spatial knowledge acquisition and transfer, situational awareness and team training. Our initial experiment was aimed at demonstrating that knowledge trained in the virtual world could be applied in the real one.

Behavioral Research Issues
Dismounted Soldier Simulation

Training in Virtual Environments

- **Ability to transfer knowledge from VE to real world**
- **Ability to gain and maintain situational awareness**
- **Applicability for training leaders and teams**



Learning Spatial Relationships

Subjects in the experiment were asked to learn a route through an office building they had never seen before. We used three training conditions. Following an initial period of route study for all subjects, subjects in the symbolic condition used verbal descriptions and photographs to practice the route and learn the landmarks along the way. The building condition group walked through the actual building. The third group navigated through a virtual model of the building. The picture on this slide was taken from the virtual database representing the building. After the rehearsal trials all the subjects were tested on the route in the real building.

Learning Spatial Relationships

- **Route Learning in an Office Building**
  - **Three Training Conditions**
    - **Rehearsal Using Verbal Descriptions, Pictures and Maps**
    - **Rehearsal in Actual Building**
    - **Rehearsal in Virtual Building**
- **Evaluate Transfer of Training from Virtual Environment to Real World**
  - **Transfer Condition - Walk Route in Building**

Transfer Test Performance as a Function of Training Mode

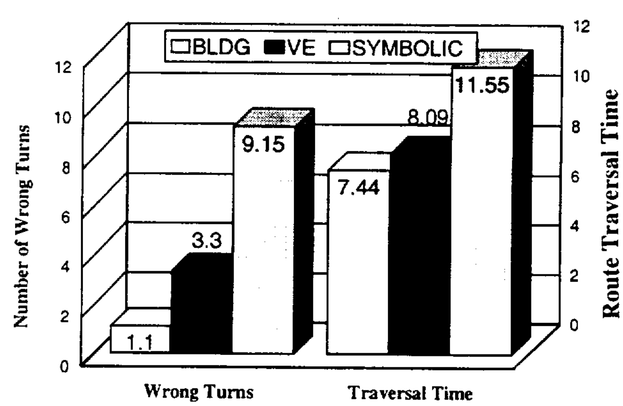
The two graphs represent the number of wrong turns that subjects made on the test and the amount of time it took them to traverse the route. Those who rehearsed the task in the actual building demonstrated the best test performance followed by the Virtual Environments trained group.

The building group made significantly fewer errors than the VE trained group. While statistically significant, the result has little practical significance. In a route that had fifty turns, the difference between one and three errors is not that great.

The traversal times for the building and VE trained groups were not significantly different, but both were significantly faster than the group that used verbal descriptions and pictures. We took these results as a positive demonstration of transfer of training from a VE to a real world environment.

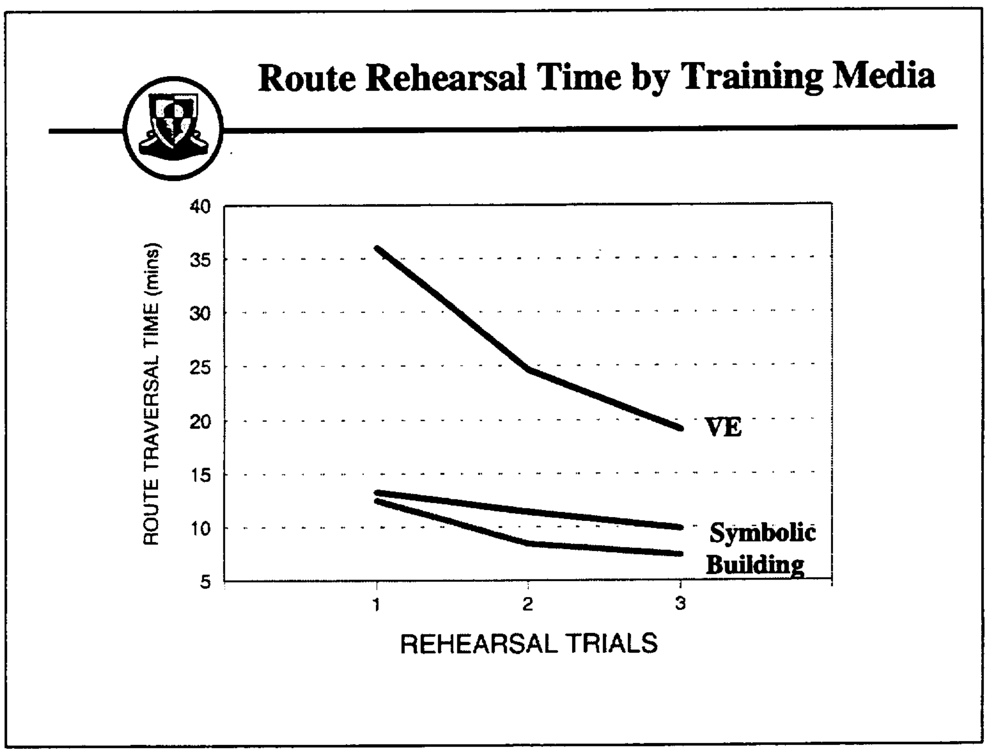
Learning Spatial Relationships

Transfer Test Performance as a Function of Training Mode



Route Rehearsal Time by Training Media

Another interesting finding from this research is shown in this graph. Rehearsal times in the VE training group improved dramatically relative to changes in rehearsal times of the other two groups. I believe this means that subjects in the VE group were not just learning the route. They were also learning to deal with the differences in perception and movement required to negotiate the route in a VE. Virtual reality is not reality and people definitely have to learn to deal with problems like getting unstuck from walls or determining when they are in the center of a doorway. Given the split attention of the VE groups their performance in the test condition is more impressive.



Team Training

We recently completed our first experiment in which we immersed more than one individual in a VE at the same time. Our objective was to evaluate the applicability of Virtual Environments for teaching decision making and teamwork skills. Located in Orlando, we do not have a ready source of military personnel. We had to design team tasks that had many of the characteristics of a military task but could be learned relatively quickly by civilians.

Team Training

Can we use VE to train team and leader skills necessary for dismounted soldiers?

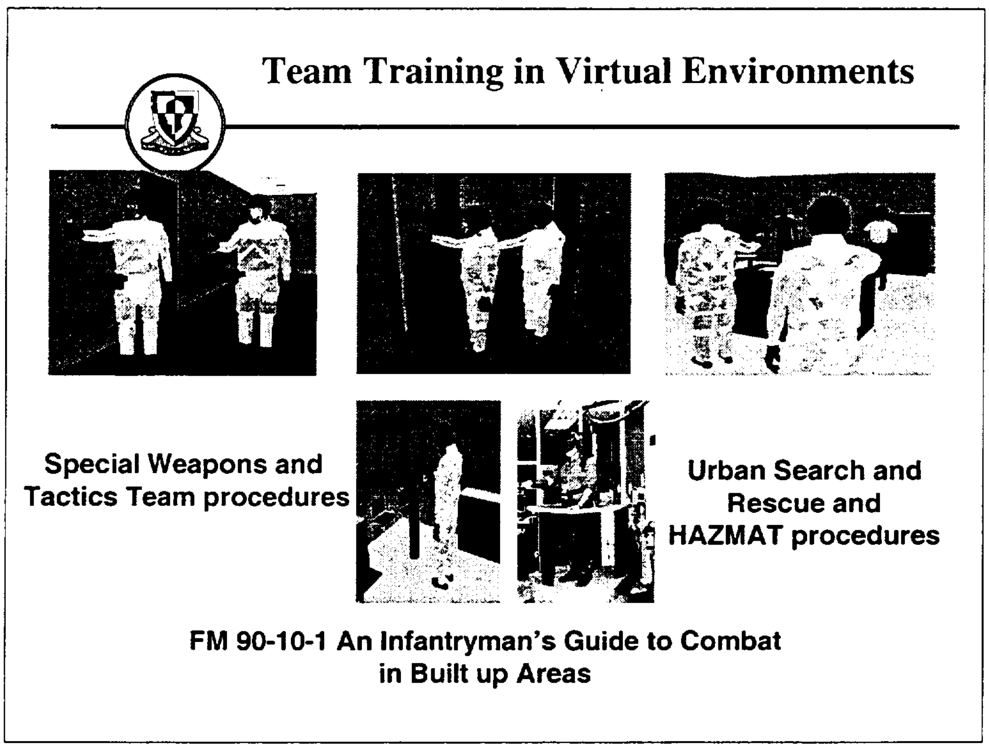
- **Decision making in complex, dynamic environments**
- **Acquire and maintain situational awareness**
- **Coordination**
- **Communication**

Challenge: Creating tasks and environments that

- **Can be implemented with current VE technology**
- **Require limited prerequisite skills**
- **Represent the critical elements of military tasks.**

Team Training in Virtual Environments

The mission we developed involved a two-person team of hazardous material handlers searching a building for leaking gas canisters. While in the building they could encounter enemies, looters or friendly civilians that they have to deal with. The team members wear virtual protective garments, and there is strong time pressure on the teams to complete their tasks before their oxygen runs out. We developed a training manual for the mission based on procedures used by SWAT teams, the infantry in built up areas, and urban search and rescue teams. Within the two person teams, one team member is designated as the leader and the other is responsible for specialty jobs like turning off the leaking gas canisters.



Instructional Strategies

We ran teams under four conditions. Three of the conditions utilized different instructional strategies. The fourth condition was a control condition. All of the teams first studied the mission manual which instructed them on the procedures and methods they were to use.

The Control Group practiced the mission twice without feedback prior to a test mission. Each mission was conducted in a different building, although all of the buildings were similar in complexity.

Teams in the demonstration condition watched a virtual demonstration of an expert team performing the mission followed by a practice session and the test.

The coaching group practiced the mission twice with a trainer providing them instruction and feedback as they performed the mission. The practice sessions were again followed by the test.

And, finally, the After Action Critique condition held a practice session that was followed by a replay of their performance, with a detailed debriefing and discussion of the team's performance. The critique was followed by the test.

As I mentioned earlier, the experiment was completed recently and we are just beginning to analyze the data. I can show you some preliminary results.

Instructional Strategies

Timing of Training Intervention

Before: Demonstration (replay of "expert" performance)

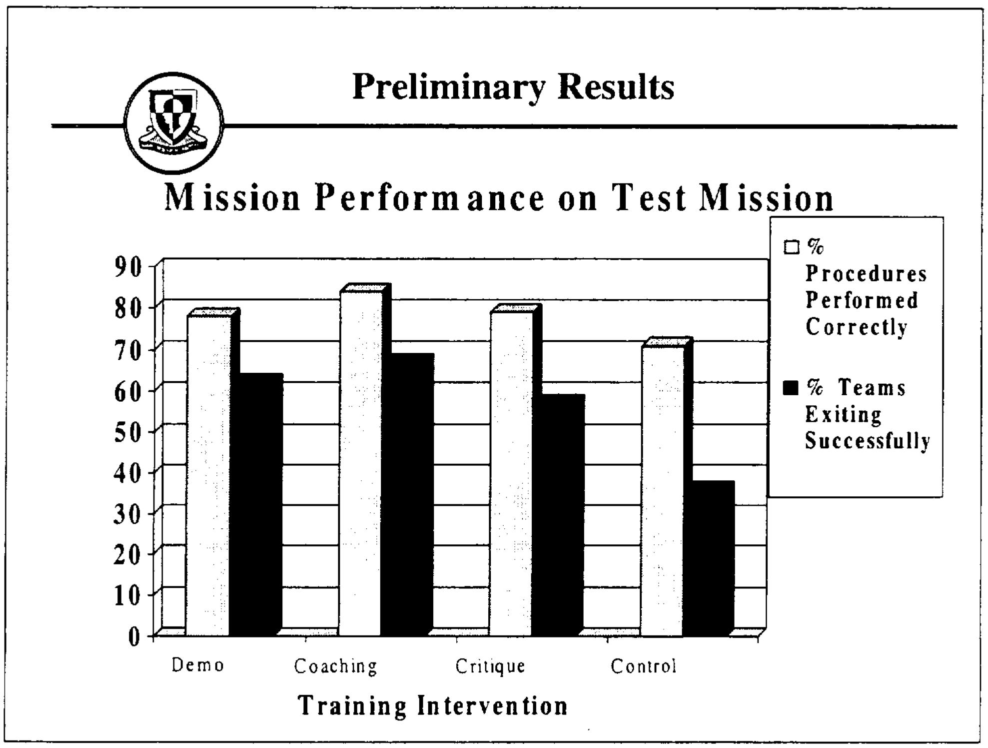
During: Coaching (by mission commander)

•After: After-Action Critique during replay

•Never: Control (practice without external feedback

Preliminary Results

The graphs represent the percentage of teams on the test trial that exited the building prior to their oxygen running out, and the percentage of correct procedures followed by the teams. Each of the instructional strategy conditions beat the control group, and the coaching group had the highest overall means. There was a lot of variance in the performance of the teams both within and across conditions.data. As I mentioned, further analyses are being conducted. It is too early to make any definitive statements about the results.



Future Directions

We are currently working on two other training projects. The first is another team training effort sponsored by The Technical Cooperation Program. TTCP is a technology sharing agreement between the US, UK, Canada, Australia and New Zealand. In this experiment we will be training team members who are either co-located in the same location or are in different locations. In this case the different locations are Toronto and Houston. We can envision situations in the military and at NASA where teams needed to be formed and trained, but the teammates are not co-located. We are well along in the planning of this research and hope to begin collecting data later this Spring.

The second project is a long-term development effort to create a capability to train small unit leaders at the fire team and squad level. The training would take place in a virtual environment with the rest of the unit being represented by computer-generated forces. This trainer would allow small unit leaders to train on numerous tactical situations across a range of conditions. In this research we will be using voice and gesture commands to guide the performance of computer generated dismounted soldiers. We plan to incorporate an intelligent tutoring system in the design to provide feedback and sequence exercises. Success in this project would be directly applicable to solving the CCTT dismounted soldier problem. We are working with the Army's Simulation Training and Instrumentation Command (STRICOM) on this effort. It is just getting underway and is in the early planning phase.

Future Directions

- **Distributed Team Training**
  - **Networking sites in Toronto and Houston to Orlando**
  - **Investigating possible differences in performance of teams trained apart vs. at the same geographical location**
- **Small Unit Leader Trainer**
  - **Provide a means to train key leaders in tactical skills**
  - **Incorporate Dismounted Infantry Semi-Automated Forces**
  - **Utilize speech recognition, gesture recognition**
  - **Apply intelligent tutor technology**

Further Information

ARI has a number of publications in this area and we have recently published a technical report which summarizes our research and discusses the recommendations we feel we can make about training with VR technology at this time. If you are interested in this or any of our other reports, please contact me at the following phone number or email address.

Further Information

Dr. Stephen Goldberg

(407) 384-3980

Stephen_Goldberg@stricom.army.mil

Government Education and Training Network (GETN)

Philip J.-L. Westfall
ATN Program Manager
Center for Distance Education
Air Force Institute of Technology
Department of the Air Force
Wright Patterson Air Force Base, OH

Government Education and Training Network (GETN)

Philip J.-L. Westfall
ATN Program Manager
AFIT Center for Distance Education
Department of the Air Force
Wright Patterson Air Force Base, OH

The Center for Distance Education (CDE), located at the Air Force Institute of Technology (AFIT), Wright-Patterson AFB, OH, conceived and developed a satellite-based, interactive television (ITV) network to meet distance learning needs of AFIT, but with the vision of creating a government-wide distance learning network, which the CDE dubbed the *Government Education and Training Network (GETN).*

CDE's network, called *Air Technology Network (ATN),* consists of a one way, digital-video up-link earthstation, reaching receive-only, down-link earthstations, but with two way audio interaction. ATN uses cost effective compressed digital video (CDV) technology for video and a terrestrial, push-to-talk audio-conferencing system manufactured by AT Products, Inc., for two-way audio interaction .

- CDE developed the network in 1992 with the help of the National Technological University, a non-profit, private satellite network for engineering education that began using CDV in 1991.

- ATN now reaches 76 Air Force bases within the United States, ten locations in Europe, and one in the Pacific, with education and training programs broadcast from up-links at Wright-Patterson (the first up-link on ATN), Maxwell, Sheppard and Keesler Air Force Bases.

- AFIT has graduated over 17,000 students from its continuing education courses via satellite since 1991, and yearly throughput of students is increasing dramatically.

CDE made GETN possible by requesting that the Defense Information Systems Agency and the General Services Administration modify their existing long-haul telecommunications contracts to include CDV. Both contracts were operated by AT&T. Satellite management on GE-3, however, is subcontracted to Spacenet, Inc., a U.S. based company and wholly owned subsidiary of Gilat Satellite, Inc.

- DISA (in 1993) and GSA (in 1994) began offering the CDV satellite service to the federal government, allowing for the first time, a single satellite network dedicated to education and training. CDE is continuing negotiations to reach more DOD sites in the Pacific (Alaska and Hawaii are already on GETN). The connection to Europe is being made through the Joint Broadcast Service through an up-link earthstation at the Pentagon.

- CDE promoted interagency sharing of distance learning programs using ITV and quickly gained support from the U.S. Army. Other agencies joined the growing GETN community of ITV users. For example:

  - U.S. Army at Fort Lee, VA, retrofitted its existing up-link and 80-site Satellite Education Network to CDV.

  - Air National Guard's (ANG) Warrior Network has three up-links and 198 down-links installed with over 290 additional classrooms.

  - Federal Aviation Administration has one up-link going to 60 down-links with 32 additional drops.

  - Department of Energy has one up-link and 24 down-links.

  - Environmental Protection Agency has a shared up-link with North Carolina State University reaching 118 down-links.

  - Veteran Benefits Administration has a T-1 connection to the FAA up-link and has 63 down-links of its own.

  - Defense Information Systems Agency has 22 sites.

  - The Defense Equal Opportunity Management Institute (DEOMI) has an up-link at Patrick AFB, FL.

  - The U.S. Courts' Federal Judicial Television Network (FJTN) connects to an ANG up-link and is currently installing 250 down-links.

  - The Department of Interior's Fish and Wildlife Service National Conservation Training Center has its own up-link with ten down-link sites and is growing.

  - The Defense Logistics Agency is currently installing 60 down-link sites and is originating its programming from six remote sites connected to a VTC hub at Ft. Belvoir, which in turn is connected to the ATN and Army at Ft. Lee.

  - The U.S. Coast Guard utilizes an ISDN connection to DEOMI's up-link to reach 12 of its own sites.

  - The Internal Revenue Service's Corporate Education Interactive Video Tele-training (IVT) Network has its own up-link and 130 down-link sites and is expanding to 230 sites.

- Other government agencies using the GETN are GSA, Nuclear Regulatory Commission, U.S. Customs, Naval Undersea Warfare Center, NAVAIR, et al.

Agencies within the federal government have formed committees and action teams to coordinate, implement and expand the use of GETN and other distance learning technologies.

- Air Education and Training Command, the lead command for distance learning, has created a single organization, the Air Force Distance Learning Office, to provide leadership and develop procedures to promote distance learning within the Air Force, and is located at Maxwell AFB, AL (see www.au.af.mil/afdlo). Management of ATN comes under the ATN Program Management Office, which is under the AFDLO, but located at AFIT. Currently, the ATN Program Management Office coordinates the schedule for GETN (see atn.afit.af.mil).

- Total Force Distributed Learning Action Team (TFDLAT), created by the Office of the Undersecretary of Defense for Reserve Affairs, and jointly chaired by staff from the Office of the Deputy Undersecretary of Defense for Readiness, and from DISA, has a vision to expand interoperable distance learning technologies, and maximize use to reduce overall education and training budgets (see www.adlnet.org/tfdlat/index.cfm).

- Government Alliance for Training and Education (GATE), formed by GETN Network managers from USAF, FAA and DOE, has been joined by nine other government agencies to promote the joint use of GETN facilities and shared programming throughout the federal government. Human Resources Development Council, under the OPM, is the sponsor. The Director of Training for the Office of Health and Human Services is currently the President.

- The Federal Government Distance Learning Association (FGDLA) is a nonprofit, professional organization formed to promote development and application of distance learning to education and training within the federal government, as well as promoting the use of existing government resources to promote retraining of the workforce in the private/commercial sector. Association members come from industry and government sectors. The Association is affiliated with the United States Distance Learning Association (see www.fgdla.org).

With over 9,000 scheduled hours of broadcasting in 1999 (4,500 from ATN) and from 13 up-links reaching out to over 1,000 receive sites, GETN offers high-quality, low-cost ITV with virtually limitless distribution capability within the U.S. GETN has set the standard for instructional video communications within the federal government.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 07704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
July 1999

3. REPORT TYPE AND DATES COVERED
Conference Publication

4. TITLE AND SUBTITLE
Advanced Training Technologies and Learning Environments

5. FUNDING NUMBERS
WU 282-10-01-42

6. AUTHOR(S)
Ahmed K. Noor and John B. Malone, Compilers

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER
L-17860

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA/CP-1999-209339

11. SUPPLEMENTARY NOTES
Noor: University of Virginia Center for Advanced Computational Technology, Hampton, VA
Malone: Langley Research Center, Hampton, VA

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified–Unlimited
Subject Category 61 Distribution: Standard
Availability: NASA CASI (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This document contains the proceedings of the Workshop on Advanced Training Technologies and Learning Environments held at NASA Langley Research Center, Hampton, Virginia, March 9-10, 1999. The workshop was jointly sponsored by the University of Virginia's Center for Advanced Computational Technology and NASA. Workshop attendees were from NASA, other government agencies, industry, and universities. The objective of the workshop was to assess the status and effectiveness of different advanced training technologies and learning environments.

14. SUBJECT TERMS
Engineering education; Instructional technology; Advanced training; Curriculum reform

15. NUMBER OF PAGES
451

16. PRICE CODE
A20

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102