COPY 1

UNITED STATES AIR FORCE

SUMMER RESEARCH PROGRAM -- 1992

SUMMER RESEARCH EXTENSION PROGRAM FINAL REPORTS

VOLUME 1A

ARMSTRONG LABORATORY

RESEARCH & DEVELOPMENT LABORATORIES

5800 UPLANDER WAY

CULVER CITY, CA 90230-6608

SUBMITTED TO:

LT. COL. CLAUDE CAVENDER
PROGRAM MANAGER

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

BOLLING AIR FORCE BASE

WASHINGTON, D.C.

MAY 1993

REPOR, DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
28 Dec 92

3. REPORT TYPE AND DATES COVERED
Annual 1 Sep 91 - 31 Aug 92

4. TITLE AND SUBTITLE
1992 Summer Faculty Research Program (SFRP)
Volume: 1A. CSREPI

5. FUNDING NUMBERS
F49620-90-C-0076

6. AUTHOR(S)
Mr Gary Moore

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Research & Development Laboratoreis (RDL)
5800 Uplander Way
Culver City CA 90230-6600

AFOSR-TR-95
0728

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NI
110 Duncan Ave., Suite B115
Bldg 410
Bolling AFB DC 20332-0001
Lt Col Claude Cavender

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
UNLIMITED

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The purpose of this program is to develop the basis for cintinuing research of interest to the Air Force at the institution of the faculty member; to stiumlate continuing relations among faculty members and professional peers in the Air Force to enhance the research interests and capabilities of scientific and engineering educators; and to provide follow-on funding for research of particular promise that was started at an Air Force laboratory under the Summer Faculty Research Program.

During the summer of 1992 185 university faculty conducted research at Air Force laboratories for a period of 10 weeks. Each participant provided a report of their research, and these reports are consolidated into this annual report.

19951127 028

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

UNITED STATES AIR FORCE

SUMMER RESEARCH PROGRAM -- 1992

SUMMER RESEARCH EXTENSION PROGRAM FINAL REPORTS

VOLUME 1A

ARMSTRONG LABORATORY

RESEARCH & DEVELOPMENT LABORATORIES

5800 Uplander Way

Culver City, CA 90230-6608

Program Director, RDL
Gary Moore

Program Manager, AFOSR
Lt. Col. Claude Cavender

Program Manager, RDL
Scott Licoscos

Program Administrator, RDL
Gwendolyn Smith

Submitted to:

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

Bolling Air Force Base

Washington, D.C.

May 1993

PREFACE

This volume is part of a five-volume set that summarizes the research of participants in the 1992 AFOSR Summer Research Extension Program (SREP). The current volume, Volume 1A of 5, presents the final reports of SREP participants at Armstrong Laboratory.

Reports presented in this volume are arranged alphabetically by author and are numbered consecutively -- e.g., 1-1, 1-2, 1-3; 2-1, 2-2, 2-3, with each series of reports preceded by a 22-page management summary. Reports in the five-volume set are organized as follows:

VOLUME	TITLE
1A	Armstrong Laboratory (part one)
1B	Armstrong Laboratory (part two)
2	Phillips Laboratory
3	Rome Laboratory
4A	Wright Laboratory (part one)
4B	Wright Laboratory (part two)
5	Air Force Civil Engineering Laboratory, Arnold Engineering Development Center, Frank J. Seiler Research Laboratory, Wilford Hall Medical Center

1992 SUMMER RESEARCH EXTENSION PROGRAM FINAL REPORTS

Armstrong Laboratory

Armstrong Laboratory (cont'd)

1992 SUMMER RESEARCH EXTENSION PROGRAM (SREP) MANAGEMENT REPORT

1.0 BACKGROUND

Under the provisions of Air Force Office of Scientific Research (AFOSR) contract F49620-90-C-0076, September 1990, Research & Development Laboratories (RDL), an 8(a) contractor in Culver City, CA, manages AFOSR's Summer Research Program. This report is issued in partial fulfillment of that contract (CLIN 0003AC).

The name of this program was changed during this year's period of performance. For that reason, participants' cover sheets are captioned "Research Initiation Program" (RIP), while the covers of the comprehensive volumes are titled "Summer Research Extension Program" (SREP). The program's sponsor, the Air Force Office of Scientific Research (AFOSR), changed the name to differentiate this program from another which also bore its original name.

Apart from this name change, however, the program remained as it has been since its initiation as the Mini-Grant Program in 1983. The SREP is one of four programs AFOSR manages under the Summer Research Program. The Summer Faculty Research Program (SFRP) and the Graduate Student Research Program (GSRP) place college-level research associates in Air Force research laboratories around the United States for 8 to 12 weeks of research with Air Force scientists. The High School Apprenticeship Program (HSAP) is the fourth element of the Summer Research Program, allowing promising mathematics and science students to spend two months of their summer vacations at Air Force laboratories within commuting distance from their homes.

SFRP associates and exceptional GSRP associates are encouraged, at the end of their summer tours, to write proposals to extend their summer research during the following calendar year at their home institutions. AFOSR provides funds adequate to pay for 75 SREP subcontracts. In addition, AFOSR has traditionally provided further funding, when available, to pay for additional SREP proposals, including those submitted by associates from Historically Black Colleges and Universities (HBCUs) and Minority Institutions (MIs). Finally, laboratories may transfer internal funds to AFOSR to fund additional SREPs. Ultimately the laboratories inform RDL of their SREP choices, RDL gets AFOSR approval, and RDL forwards a subcontract to the institution where the SREP associate is employed. The subcontract (see Attachment 1 for a sample) cites the SREP associate as the principal investigator and requires submission of a report at the end of the subcontract period.

Institutions are encouraged to share costs of the SREP research, and many do so. The most common cost-sharing arrangement is reduction in the overhead, fringes, or administrative changes institutions would normally add on to the principal investigator's or research associate's labor. Some institutions also provide other support (e.g., computer run time, administrative assistance, facilities and equipment or research assistants) at reduced or no cost.

When RDL receives the signed subcontract, we fund the effort initially by providing 90% of the subcontract amount to the institution (normally $18,000 for a $20,000 SREP). When we receive the end-of-research report, we evaluate it administratively and send a copy to the laboratory for a technical evaluation. When the laboratory notifies us the SREP report is acceptable, we release the remaining funds to the institution.

2.0 THE 1992 SREP PROGRAM

SELECTION DATA: In the summer of 1991, 170 faculty members (SFRP associates) and 142 graduate students (GSRP associates) participated in the summer program. Of those, 147 SFRPs and 10 GSRPs submitted SREP proposals; 88 SFRP SREPs and 7 GSRP SREPs were selected for funding (total: 95).

	Summer 1991 Participants	Submitted SREP Proposals	SREPs Funded
SFRP	170	147	88
GSRP	142	10	7

The funding was provided as follows:

Contractual slots funded by AFOSR	75
Laboratory-funded	13
Additional funding from AFOSR	7
Total	95

Seven HBCU/MI associates from the 1991 summer program submitted SREP proposals; five were selected (one was lab-funded; four were funded by additional AFOSR funds).

By laboratory, the applications submitted and selected show in the following table:

	Applied	Selected
Air Force Civil Engineering Laboratory	6	4
Armstrong Laboratory	34	20
Arnold Engineering Development Center	12	2
Frank J. Seiler Research Laboratory	5	3
Phillips Laboratory	30	18
Rome Laboratory	16	11
Wilford Hall Medical Center	1	1
Wright Laboratory	53	36
TOTAL	157	95

Note: Phillips Laboratory funded 2 SREPs; Wright Laboratory funded 11; and AFOSR funded 7 beyond its contractual 75.

ADMINISTRATIVE EVALUATION: The administrative quality of the SREP associates' final reports was satisfactory. Most complied with the formatting and other instructions RDL provided to them. In the final days of December 1992 and in the first two months of 1993, several associates called and requested no-cost extensions of up to six months. After consultation with our AFOSR Contracting Officer's Representative, RDL approved the requests but asked that all such associates provide an interim report to be included in this volume. That caused an AFOSR-approved delay beyond the 1 April 1993 submission of this report. The subcontracts were funded by $1,893,616 of Air Force money. Institutions' cost sharing amounted to $948,686.

TECHNICAL EVALUATION: The form we used to gather data for technical evaluation and the technical evaluations of the SREP reports are provided as Attachment 2. This summary evaluation is shown by SREP number. The average rating range was from 3.1 to 5.0. The overall average for those evaluated was 4.6 out of 5.00. The three rating factors with the highest average scores were:

- The USAF should continue to pursue the research in this RIP report.
- The money spent on this RIP report was well worth it.
- I'll be eager to be a focal point for summer and RIP associates in the future.

Thus it is clear that the laboratories place a high value on AFOSR's Summer Research Program: SFRP, GSRP, and SREP.

3.0 SUBCONTRACTS SUMMARY

Table 1 lists contractually required information on each SREP subcontract. The individual reports are published in volumes as follows:

Laboratory	Volume
Air Force Civil Engineering Laboratory	5
Armstrong Laboratory	1
Arnold Engineering Development Center	5
Frank J. Seiler Research Laboratory	5
Phillips Laboratory	2
Rome Laboratory	3
Wilford Hall Medical Center	5
Wright Laboratory	4

TABLE 1: SUBCONTRACTS SUMMARY

Researcher's name Institution Location	Highest Degree	Subcontract Number Department Amount	Duration Sharing
Abbott, Ben A Vanderbilt University Nashville, TN 37235	MS	135 Dept of Electrical Engineering 19966.00	01/01/92-12/31/92 0.00
Acharya, Raj State University of New York, Buffalo Buffalo, NY 14260	PhD	151 Dept of Electrical & Comp Engrg 20000.00	01/01/92-12/31/92 0.00
Adams, Christopher M Oklahoma State University Stillwater, OK 74078	PhD	68 Dept of Chemistry 20000.00	01/01/92-12/31/92 0.00
Anderson, Richard A University of Missouri, Rolla Rolla, MO 65401	PhD	50 Dept of Physics 20000.00	01/01/92-12/31/92 5000.00
Arora, Vijay K Wilkes University Wilkes-Barre, PA 18766	PhD	3 Dept of Electrical & Comp Engrg 19996.00	10/01/91-09/30/92 36208.00
Ball, William P Duke University Durham, NC 27706	PhD	71 Dept of Civil & Environmental Eng 20000.00	01/01/92-12/31/92 26747.00
Battles, Frank P Massachusetts Maritime Academy Buzzard's Bay, MA 025321803	PhD	152 Dept of Basic Sciences 20000.00	01/01/92-12/31/92 22000.00
Bieniek, Ronald J University of Missouri, Rolla Rolla, MO 65401	PhD	147 Dept of Physics 19945.00	01/01/92-12/31/92 4000.00
Blystone, Robert V Trinity University San Antonio, TX 78212	PhD	127 Dept of Biology 20000.00	01/01/92-12/31/92 14783.00
Cha, Soyoung S University of Illinois, Chicago Chicago, IL 60680	PhD	011 Dept of Mechanical Engineering 20000.00	01/01/92-12/31/92 3842.00
Chandra, D. V. Satish Kansas State University Manhattan, KS 66506	PhD	89 Dept of Electrical Engineering 20000.00	01/18/92-10/17/92 11170.00
Chenette, Eugene R University of Florida Gainesville, FL 32611	PhD	106 Dept of Electrical Engineering 20000.00	01/01/92-12/31/92 0.00
Christensen, Douglas A University of Utah Salt Lake City, UT 84112	PhD	83 Dept of Electrical Engineering 19999.00	01/01/92-12/31/92 5000.00

Chubb, Gerald P Ohio State University Columbus, OH 43235	PhD 26 Dept of Aviation 20000.00		01/01/92-12/31/92 7600.00
Courter, Robert W Louisiana State University Baton Rouge, LA 70803	PhD 8 Dept of Mechanical Engineering 20000.00		10/01/91-09/30/92 445.00
Dey, Pradip P Hampton University Hampton, VA 23668	PhD 120 Computer Science Department 19921.00		01/01/92-12/31/92 0.00
Draut, Arthur W Embry Riddle Aeronautical University Prescott, AZ 86301	PhD 133 Computer Science Dept 19431.00		01/06/92-05/08/92 0.00
Dreisbach, Joseph University of Scranton Scranton, PA 185104626	PhD 108 Dept of Chemistry 20000.00		12/01/91-12/01/92 4000.00
Dror, Itiel Harvard University Cambridge, MA 02138	BS 76 Dept of Psychology 20000.00		01/01/92-12/31/92 0.00
Drost-Hansen, W. University of Miami Coral Gables, FL 33124	PhD 124 Dept of Chemistry 20000.00		12/01/91-12/01/92 12000.00
Dunleavy, Lawrence P University of South Florida Tampa, FL 33620	PhD 41 Dept of Electrical Engineering 20000.00		01/01/92-12/31/92 6463.00
Evans, Joseph B University of Kansas Lawrence, KS 66045	PhD 96 Dept of Electrical & Comp Engrg 20000.00		01/01/92-12/31/92 0.00
Flowers, George T Auburn University Auburn, AL 368495341	PhD 73 Dept of Mechanical Engineering 19986.00		01/01/92-12/30/92 12121.00
Gantenbein, Rex E University of Wyoming Laramie, WY 82071	PhD 22 Dept of Computer Science 20000.00		01/01/91-12/31/92 26643.00
Garcia, Ephrarim Vanderbilt University Nashville, TN 37235	PhD 32 Dept of Mechanical Engineering 20000.00		12/01/91-11/30/92 9659.00
German, Fred J Auburn University Auburn University, AL 36830	PhD 49 Dept of Electrical Engineering 20000.00		01/01/92-12/31/92 0.00
Gould, Richard D North Carolina State University Raleigh, NC 276957910	PhD 87 Dept of Mech and Aerospace Engrg 20000.00		01/01/92-12/31/92 14424.00
Gove, Randy L University of Alabama, Huntsville Huntsville, AL 35899	MS 122 Dept of Physics 20000.00		01/01/92-12/31/92 3469.00
Grabowski, Marek University of Colorado, Colorado Springs Colorado Springs, CO 809337150	PhD 92 Dept of Physics 19700.00		01/01/92-12/31/92 0.00

Gunaratne, Manjriker University of South Florida Tampa, FL 33620	PhD	90	01/01/92-12/31/92	Dept of Civil Engrg & Mechanics 19994.00 10062.00
Hall, Ernest L University of Cincinnati Cincinnati, OH 452210072	PhD	134	01/01/92-12/31/92	Dept of Robotics Research 19975.00 0.00
Hamilton, William L Salem State College Salem, MA 01970	PhD	47	01/01/92-12/31/92	Dept of Geography 20000.00 32000.00
Hamilton, Kirk L Xavier University of Louisiana New Orleans, LA 70125	PhD	57	01/01/92-12/31/92	Dept of Biology 20000.00 16100.00
Harris, Harold H University of Missouri, St.Louis St. Louis, MO 63121	PhD	94	01/01/92-12/31/92	Dept of Chemistry 19300.00 8600.00
Hartung, George H University of Hawaii Honolulu, HI 96822	PhD	46	01/01/92-12/31/92	Dept of Physiology 20000.00 7530.00
Hatfield, Steven L University of Kentucky Lexington, KY 40506	BS	23	01/01/92-12/31/92	Dept of Materials Science & Engrg 20000.00 28625.00
Hedman, Paul O'Dell Brigham Young University Provo, UT 84602	PhD	17	01/01/92-12/31/92	Dept of Chemical Engineering 19999.00 6928.00
Heister, Stephen D Purdue University West Lafayette, IN 47907	PhD	5	01/01/92-12/31/92	School of Aero & Astronautics 20000.00 4419.00
Hess, David J University of Texas, Austin Austin, TX 78713	BA	149	01/01/92-12/31/92	Dept of Psychology 19914.00 8784.00
Hoffman, R. W Case Western Reserve University Cleveland, OH 44106	PhD	99	01/01/92-12/31/92	Dept of Physics 19770.00 0.00
Huerta, Manuel A University of Miami Coral Gables, FL 33124	PhD	62	01/01/92-12/31/92	Dept of Physics 20000.00 1207.00
Hui, David University of New Orleans New Orleans, LA 70148	PhD	116	01/01/92-12/31/92	Dept of Mechanical Engineering 20000.00 0.00
Iyer, Ashok University of Nevada, Las Vegas Las Vegas, NV 89154	PhD	74	01/01/92-12/31/92	Dept of Electrical & Comp Engrg 20000.00 18549.00
Khonsari, Michael M University of Pittsburgh Pittsburgh, PA 15260	PhD	53	01/01/92-12/31/92	Dept of Mechanical Engineering 20000.00 32958.00
Kibert, Charles J University of Florida Gainesville, FL 32611	PhD	2	01/01/92-12/31/92	Dept of Fire Testing & Research 20000.00 6928.00

Klarup, Douglas G University of Montana Missoula, MT 59812	PhD	84	01/01/92-12/31/92	Dept of Chemistry 20000.00 0.00
Koblasz, Arthur J Georgia Institute of Technology Atlanta, GA 30332	PhD	145	01/01/92-09/30/92	Dept of Civil Engineering 19956.00 0.00
Kornreich, Philipp Syracuse University Syracuse, NY 13244	PhD	35	10/01/91-09/30/92	Dept of Electrical & Comp Engrg 20000.00 0.00
Kuo, Spencer P Polytechnic University Farmingdale, NY 11735	PhD	59	01/01/92-12/31/92	Dept of Electrical Engineering 20000.00 9916.00
Langhoff, Peter W Indiana University Bloomington, IN 47402	PhD	115	01/01/92-12/31/92	Dept of Chemistry 20000.00 35407.00
Lee, Byung-Lip Pennsylvania State University University Park, PA 16802	PhD	93	01/01/92-12/31/92	Dept of Engrg Science & Mechanics 20000.00 8173.00
Leigh, Wallace B Alfred University Alfred, NY 14802	PhD	118	01/01/92-12/31/92	Dept of Electrical Engineering 19767.00 18770.00
Liddy, Elizabeth Syracuse University Syracuse, NY 132444100	PhD	104	01/01/92-12/31/92	Dept of Information Studies 20000.00 0.00
Liu, Cheng University of North Carolina, Charlotte Charlotte, NC 28270	PhD	6	11/01/99-12/31/92	Dept of Engineering Technology 20000.00 0.00
Main, Robert G California State University, Chico Chico, CA 959290504	PhD	28	01/01/92-06/30/92	Dept of Communication Design 20000.00 7672.00
Mains, Gilbert J Oklahoma State University Stillwater, OK 74078	PhD	52	01/01/92-12/31/92	Dept of Chemistry 19071.00 8746.00
Marathay, Arvind S University of Arizona Tucson, AZ 85721	PhD	51	01/01/92-12/31/92	Dept of Optical Sciences 20000.00 0.00
Martin, Charlesworth R Norfolk State University Norfolk, VA 23504	PhD	125	01/01/92-12/31/92	Dept of Physics & Engineering 20000.00 0.00
Mayes, Jessica L University of Kentucky Lexington, KY 405034203	BS	16	01/01/92-12/31/92	Dept of Material Science & Engrng 20000.00 28625.00
Mulligan, Benjamin E University of Georgia Athens, GA 30602	PhD	54	01/01/92-12/31/92	Dept of Psychology 19895.00 13677.00
Munday, Edgar G University of North Carolina, Charlotte Charlotte, NC 28223	PhD	38	10/01/91-10/30/92	Dept of Mechanical Engineering 20000.00 11638.00

Nurre, Joseph H Ohio University Athens, OH 45701	PhD 56 01/01/92-12/31/92 Dept of Electrical & Comp Engrg 19842.00 15135.00
Orkwis, Paul D University of Cincinnati Cincinnati, OH 452210070	PhD 14 10/01/91-10/30/92 Dept of Engineering Mechanics 19966.00 23017.00
Patra, Amit L University of Puerto Rico Mayaquez, PR 00681	PhD 69 01/01/92-12/31/92 Dept of General Engineering 20000.00 2750.00
Peters II, Richard A Vanderbilt University Nashville, TN 37235	PhD 160 01/01/92-12/31/92 Dept of Electrical Engineering 20000.00 0.00
Pollack, Steven K University of Cincinnati Cincinnati, OH 452200012	PhD 31 01/01/92-12/31/92 Dept of Materials Sci & Engrg 20000.00 14877.00
Prescott, Glenn E University of Kansas Lawrence, KS 66045	PhD 72 01/01/92-12/31/92 Dept of Electrical Engineering 20000.00 8000.00
Price, James L University of Iowa Iowa City, IA 52242	PhD 48 01/01/92-12/30/92 Dept of Sociology 20000.00 8600.00
Qazi, Salahuddin SUNY, Utica Utica, NY 13504	PhD 129 01/01/92-12/31/92 Dept of Electrical Engineering 20000.00 25000.00
Rappaport, Carey M Northeastern University Boston, MA 02115	PhD 58 01/01/92-06/30/92 Dept of Electrical & Comp Engrng 19999.00 0.00
Rawson, Jenny L North Dakota State University Fargo, ND 58105	PhD 144 01/01/92-12/31/92 Dept of Electrical Engineering 19997.00 19826.00
Riccio, Gary E University of Illinois, Urbana Urbana, IL 61821	PhD 80 01/01/92-12/31/92 Dept of Human Perception 20000.00 0.00
Rotz, Christopher A Brigham Young University Provo, UT 84602	PhD 136 12/01/91-12/31/92 Dept of Manufacturing Engineering 20000.00 11814.00
Schwartz, Martin University of North Texas Denton, TX 762035068	PhD 55 01/01/92-12/31/92 Dept of Chemistry 20000.00 18918.00
Senseman, David M University of Texas, San Antonio San Antonio, TX 78285	PhD 77 12/01/91-11/30/92 Dept of Information 20000.00 19935.00
Sensiper, Martin University of Central Florida Orlando, FL 32816	BS 15 11/01/91-05/31/92 Dept of Electrical Engineering 20000.00 0.00
Shamma, Jeff S University of Texas, Austin Austin, TX 78713	PhD 70 01/01/92-12/31/92 Dept of Electrical Engineering 20000.00 0.00

Shively, Jon H California State University, Northridge Northridge, CA 91330	PhD	140	01/01/92-12/31/92	
	Dept of CIAM			
	20000.00		14553.00	
Singh, Sahjendra N University of Nevada, Las Vegas Las Vegas, NV 89014	PhD	79	01/01/92-12/31/92	
	Dept of Electrical Engineering			
	20000.00		20595.00	
Smith, Gerald A Pennsylvania State University University Park, PA 16802	PhD	63	07/01/92-07/01/93	
	Dept of Physics			
	20000.00		0.00	
Stephens, Benjamin R Clemson University Clemson, SC 29634	PhD	114	01/01/92-12/31/92	
	Dept of Psycology			
	19988.00		4250.00	
Sudkamp, Thomas Wright State University Dayton, OH 45435	PhD	97	01/01/92-08/31/92	
	Dept of Computer Science			
	20000.00		18739.00	
Sydor, Michael University of Minnesota, Duluth Duluth, MN 55804	PhD	11	01/01/92-12/31/92	
	Dept of Physics			
	20000.00		0.00	
Tankin, Richard S Northwestern University Evanston, IL 60208	PhD	44	01/01/92-12/31/92	
	Dept of Mechanical Engineering			
	20000.00		29103.00	
Taylor, Michael D University of Central Florida Orlando, FL 32816	PhD	141	05/01/92-07/31/92	
	Dept of Mathematics			
	20000.00		1587.00	
Teegarden, Kenneth J University of Rochester Rochester, NY 14627	PhD	98	01/01/92-12/31/92	
	Dept of Optics			
	20250.00		60600.00	
Tew, Jeffrey D Virginia Polytech Instit and State Univ Blacksburg, VA 24061	PhD	137	03/01/92-09/30/92	
	Dept of Industrial Engineering			
	17008.00		4564.00	
Tipping, Richard H University of Alabama Tuscaloosa, AL 35487	PhD	81	01/01/92-05/31/92	
	Dept of Physics & Astronomy			
	20000.00		15000.00	
Tripathi, Ram C University of Texas, San Antonio San Antonio, TX 78249	PhD	105	01/01/92-12/31/92	
	Dept of Mathematics			
	20000.00		2274.00	
Wells, Fred V Idaho State University Pocatello, ID 83209	PhD	155	01/01/92-12/31/92	
	Dept of Chemistry			
	20000.00		8000.00	
Whitefield, Phillip D University of Missouri, Rolla Rolla, MO 65401	PhD	25	01/01/92-12/31/92	
	Dept of Chemistry			
	19991.00		25448.00	
Wolfenstine, Jeffrey B University California, Irvine Irvine, CA 92717	PhD	18	01/01/92-12/31/92	
	Dept of Mechanical Engineering			
	20000.00		11485.00	
Wolper, James S Idaho State University Pocatello, ID 83209	PhD	138	01/15/92-09/30/92	
	Dept of Mathematics			
	20000.00		4828.00	

Zavodney, Lawrence D Ohio State University Columbus, OH 43210	PhD	148	01/01/92-12/31/92
	Dept of Engineering Mechanics		
	20000.00		0.00
Zimmerman, Wayne J Texas Women University Denton, TX 76204	PhD	111	01/01/92-12/31/92
	Dept of Mathematics		
	19990.00		8900.00

ATTACHMENT 1:

SAMPLE SREP SUBCONTRACT

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
1993 SUMMER RESEARCH EXTENSION PROGRAM SUBCONTRACT 93-36

BETWEEN

Research & Development Laboratories
5800 Uplander Way
Culver City, CA 90230-6608

AND

University of Delaware
Sponsored Programs Admin.
Newark, DE 19716

REFERENCE: Summer Research Extension Program Proposal 93-36
Start Date: 01/01/93 End Date: 12/31/93
Proposal amount: $20000.00

(1) PRINCIPAL INVESTIGATOR: Dr. Ian W. Hall
Materials Science
University of Delaware
Newark, DE 19716

(2) UNITED STATES AFOSR CONTRACT NUMBER: F49620-90-C-09076

(3) CATALOG OF FEDERAL DOMESTIC ASSISTANCE NUMBER (CFDA): 12.800
PROJECT TITLE: AIR FORCE DEFENSE RESEARCH SOURCES PROGRAM

(4) ATTACHMENTS 1 AND 2: SREP REPORT INSTRUCTIONS

*** SIGN SREP SUBCONTRACT AND RETURN TO RDL ***

1. BACKGROUND: Research & Development Laboratories (RDL) is under contract (F49620-90-C-0076) to the United States Air Force to administer the Summer Research Programs (SRP), sponsored by the Air Force Office of Scientific Research (AFOSR), Bolling Air Force Base, D.C. Under the SRP, a selected number of college faculty members and graduate students spend part of the summer conducting research in Air Force laboratories. After completion of the summer tour participants may submit, through their home institutions, proposals for follow-on research. The follow-on research is known as the Research Initiation Program (RIP). Approximately 75 RIP proposals annually will be selected by the Air Force for funding of up to $20,000; shared funding by the academic institution is encouraged. RIP efforts selected for funding are administered by RDL through subcontracts with the institutions. This subcontract represents such an agreement between RDL and the institution designated in Section 5 below.

2. RDL PAYMENTS: RDL will provide the following payments to RIP institutions:

- 90 percent of the negotiated RIP dollar amount at the start of the RIP Research period.
- the remainder of the funds within 30 days after receipt at RDL of the acceptable written final report for the RIP research.

3. INSTITUTION'S RESPONSIBILITIES: As a subcontractor to RDL, the institution designated on the title page will:

a. Assure that the research performed and the resources utilized adhere to those defined in the RIP proposal.
b. Provide the level and amounts of institutional support specified in the RIP proposal.
c. Notify RDL as soon as possible, but not later than 30 days, of any changes in 3a or 3b above, or any change to the assignment or amount of participation of the Principal Investigator designated on the title page.
d. Assure that the research is completed and the final report is delivered to RDL not later than twelve months from the effective date of this subcontract. The effective date of the subcontract is one week after the date that the institution's contracting representative signs this subcontract, but no later than January 15, 1992.
e. Assure that the final report is submitted in the format shown in Attachment 1.

f. Agree that any release of information relating to this subcontract (news releases, articles, manuscripts, brochures, advertisements, still and motion pictures, speeches, trade association meetings, symposia, etc.) will include a statement that the project or effort depicted was or is sponsored by: Air Force Office of Scientific Research, Bolling AFB, D.C.

g. Notify RDL of inventions or patents claimed as the result of this research in a format specified in Attachment 1.

h. RDL is required by the prime contract to flow down patent rights and technical data requirements in this subcontract. Attachment 2 to this subcontract contains a list of contract clauses incorporated by reference in the prime contract.

4. All notices to RDL shall be addressed to:

RDL Summer Research Program Office
5800 Uplander Way
Culver City, CA 90230-6608

5. By their signatures below, the parties agree to the provisions of this subcontract.

____________________ Abe S. Sopher RDL Contracts Manager	____________________ Signature of Institution Contracting Official
	____________________ Typed/Printed Name
____________________ Date	____________________ Title
	____________________ Institution
	____________________ Date/Phone

Attachment 1

Final Report Format

1. All RIP Principal Investigators will submit a final report of the research conducted.

2. One copy of the report is due to RDL no later than twelve months after the effective date of the RIP subcontract. At the same time, submit one copy to the Air Force laboratory focal point.

3. The title page should contain the title of the research, the Principal Investigator and or other co-investigators, the month and year of issue, the university with department and address, and acknowledgement of sponsorship by AFOSR (see clause 3f of this subcontract).

4. For text, use a font that is 12 characters per inch (elite) and as close to letter quality as possible. Start with the title in all caps one and one-half inches from the top of the first page; if the title requires two or more lines, single space it. Double space below the title, and then center and type the researcher's title and name. Then space twice and begin the double-spaced text.

   Use a one-and-one-half-inch left margin and a one-inch right margin for the body of the text. Center page numbers at the foot of each page, one inch from the bottom. Each page should have a one-inch margin at the top. The format should be that of a standard research paper: it should begin with a one-paragraph abstract (on its own page) summarizing your work and should be followed by an introduction, a discussion of the problem, a results section, and a conclusion. Since multiple copies of your report may be required, assure that all pages can be readily copied to a black-and-white 8 1/2" by 11" page. (No colors, such as blue or green, that don't photocopy well, and no foldouts, please.)

5. The report must be accompanied by a separate statement on whether or not any inventions or patents have resulted from this research. If yes, use a DD Form 882 (supplied by RDL on request) to indicate the patent filing date, serial number, title, and a copy of the patent application, and patent number and issue date for any subject invention in any country in which the subcontractor has applied for patents.

Attachment 2

Contract Clauses

This contract incorporates by reference the following clauses of the Federal Acquisition Regulations (FAR), with the same force and effect as if they were given in full text. Upon request, the Contracting Officer or RDL will make their full text available (FAR 52.252-2).

FAR CLAUSES	TITLE AND DATE
52.202-1	DEFINITIONS (APR 1984)
52.203-1	OFFICIALS NOT TO BENEFIT (APR 1984)
52.203-3	GRATUITIES (APR 1984)
52.203-5	COVENANT AGAINST CONTINGENT FEES (APR 1984)
52.304-6	RESTRICTIONS ON SUBCONTRACTOR SALES TO THE GOVERNMENT (JUL 1985)
52.203-7	ANTI-KICKBACK PROCEDURES (OCT 1988)
52.203-12	LIMITATION ON PAYMENTS TO INFLUENCE CERTAIN FEDERAL TRANSACTIONS (JAN 1990)
52.204-2	SECURITY REQUIREMENTS (APR 1984)
52.209-6	PROTECTING THE GOVERNMENT'S INTEREST WHEN SUBCONTRACTING WITH CONTRACTORS DEBARRED, SUSPENDED, OR PROPOSED FOR DEBARMENT (MAY 1989)
52.212-8	DEFENSE PRIORITY AND ALLOCATION REQUIREMENTS (MAY 1986)
52.215-1	EXAMINATION OF RECORDS BY COMPTROLLER GENERAL (APR 1984)
52.215-2	AUDIT - NEGOTIATION (DEC 1989)
52.222-26	EQUAL OPPORTUNITY (APR 1984)
52.222-28	EQUAL OPPORTUNITY PREAWARD CLEARANCE OF SUBCONTRACTS (APR 1984)
52.222-35	AFFIRMATIVE ACTION FOR SPECIAL DISABLED AND VIETNAM ERA VETERANS (APR 1984)
52.222-36	AFFIRMATIVE ACTION FOR HANDICAPPED WORKERS (APR 1984)

52.222-37	EMPLOYMENT REPORTS ON SPECIAL DISABLED VETERANS AND VETERANS OF THE VIETNAM ERA (JAN 1988)
52.223-2	CLEAN AIR AND WATER (APR 1984)
52.232-6	DRUG-FREE WORKPLACE (MAR 1989)
52.224-1	PRIVACY ACT NOTIFICATION (APR 1984)
52.224-2	PRIVACY ACT (APR 1984)
52.225-13	RESTRICTIONS ON CONTRACTING WITH SANCTIONED PERSONS (MAY 1989)
52.227-1	AUTHORIZATION AND CONSENT (APR 1984)
52.227-2	NOTICE AND ASSISTANCE REGARDING PATENT AND COPYRIGHT INFRINGEMENT (APR 1984)
52.227-10	FILING OF PATENT APPLICATIONS - CLASSIFIED SUBJECT MATTER (APR 1984)
52.227-11	PATENT RIGHTS - RETENTION BY THE CONTRACTOR (SHORT FORM) (JUN 1989)
52.228-6	INSURANCE - IMMUNITY FROM TORT LIABILITY (APR 1984)
52.228-7	INSURANCE - LIABILITY TO THIRD PERSONS (APR 1984)
52.230-5	DISCLOSURE AND CONSISTENCY OF COST ACCOUNTING PRACTICES (SEP 1987)
52.232-23	ASSIGNMENT OF CLAIMS (JAN 1986)
52.237-3	CONTINUITY OF SERVICES (APR 1984)
52.246-25	LIMITATION OF LIABILITY - SERVICES (APR 1984)
52.249-6	TERMINATION (COST-REIMBURSEMENT) (MAY 1986)
52.249-14	EXCUSABLE DELAYS (APR 1984)
52.251-1	GOVERNMENT SUPPLY SOURCES (APR 1984)

DoD FAR CLAUSES	TITLE AND DATE
252.203-7001	SPECIAL PROHIBITION ON EMPLOYMENT (MAR 1989)
252.203-7002	STATUTORY COMPENSATION PROHIBITIONS AND REPORTING REQUIREMENTS RELATING TO CERTAIN FORMER DEPARTMENT OF DEFENSE (DoD) EMPLOYEES (APR 1988)
252.223-7500	DRUG-FREE WORK FORCE (SEP 1988)
252.225-7001	BUY AMERICAN ACT AND BALANCE OF PAYMENTS PROGRAM (APR 1985)
252-225-7023	RESTRICTION ON ACQUISITION OF FOREIGN MACHINE TOOLS (JAN 1989)
252.227-7013	RIGHTS IN TECHNICAL DATA AND COMPUTER SOFTWARE (OCT 1988)
252.227-7018	RESTRICTIVE MARKINGS ON TECHNICAL DATA (OCT 1988)
252.227-7029	IDENTIFICATION OF TECHNICAL DATA (APR 1988)
252.227-7034	PATENTS - SUBCONTRACTS (APR 1984)
252.227-7037	VALIDATION OF RESTRICTIVE MARKINGS ON TECHNICAL DATA (APR 1988)
252.231-7000	SUPPLEMENTAL COST PRINCIPLES (APR 1984)
252.231-7001	PENALTIES FOR UNALLOWABLE COSTS (APR 1988)
252.231-7003	CERTIFICATION OF INDIRECT COSTS (APR 1986)
252.251-7000	ORDERING FROM GOVERNMENT SUPPLY SOURCES (APR 1984)
252.271-7001	RECOVERY OF NONRECURRING COSTS ON COMMERCIAL SALES OF DEFENSE PRODUCTS AND TECHNOLOGY AND OF ROYALTY FEES FOR USE OF DoD TECHNICAL DATA (FEB 1989)

7 November 1991

AFOSR/PKO
Bldg. 410, Room C-124
Bolling AFB, DC 20332-6448

Attn: Ms. Kathleen Wetherell

Dear Ms. Wetherell:

Enclosed for your approval is the model subcontract for the Research Initiation Program under the Summer Research Programs (Contract F9620-90-C-0076). The blanks will be filled by merging information from our dBase IV database.

Sincerely,

Abe S. Sopher
Contracts Manager

cc: AFOSR/NI (Lt. Col. Cavendar)

ATTACHMENT 2:

SAMPLE TECHNICAL EVALUATION FORM AND TECHNICAL EVALUATION SUMMARY

1992 RESEARCH INITIATION PROGRAM TECHNICAL EVALUATION

RIP NO: 92-2
RIP ASSOCIATE: Dr. Charles Kibert

Provided are several evaluation statements followed by ratings of (1) through (5). A rating of (1) is the lowest and (5) is the highest. Circle the rating level number you best feel rates the statement. Document additional comments on the back of this evaluation form.

Mail or fax the completed form to:

RDL
Attn: 1992 RIP TECH EVALS
5800 Uplander Way
Culver City, CA 90230-6608
(Fax: 310 216-5940)

1.	This RIP report has a high level of technical merit	1 2 3 4 5
2.	The RIP program is important to accomplishing the lab's mission	1 2 3 4 5
3.	This RIP report accomplished what the associate's proposal promised	1 2 3 4 5
4.	This RIP report addresses area(s) important to the USAF	1 2 3 4 5
5.	The USAF should continue to pursue the research in this RIP report	1 2 3 4 5
6.	The USAF should maintain research relationships with this RIP associate	1 2 3 4 5
7.	The money spent on this RIP effort was well worth it	1 2 3 4 5
8.	This RIP report is well organized and well written	1 2 3 4 5
9.	I'll be eager to be a focal point for summer and RIP associates in the future	1 2 3 4 5
10.	The one-year period for complete RIP research is about right	1 2 3 4 5

******USE THE BACK OF THIS FORM FOR ADDITIONAL COMMENTS******

LAB FOCAL POINT'S NAME (PRINT): ______________________________

OFFICE SYMBOL: ____________ PHONE: ________________

TECHNICAL EVALUATION SUMMARY

Subcontract no.	Technical Evaluation Questionnaire Rating Factors										
	1	2	3	4	5	6	7	8	9	10	Average
135	5	4	5	4	4	4	4	4	5	5	4.4
50	4	4	5	4	4	4	4	3	5	5	4.2
3	4	3	3	3	3	3	3	3	3	4	3.2
71	4	4	4	4	3	5	5	4	5	5	4.3
152	3	4	3	4	4	3	4	3	4	5	3.7
147	5	5	5	5	5	5	5	5	5	4	4.9
011	4	4	5	4	5	5	5	4	5	4	4.5
106	5	5	4	5	5	5	5	5	5	5	4.9
83	5	4	5	5	5	5	5	5	5	4	4.8
26	5	4	4	5	5	5	5	5	4	4	4.6
8	5	3	4	4	5	5	5	3	5	5	4.4
120	1	5	2	4	5	3	2	1	4	4	3.1
133	3	2	4	5	5	4	3	4	3	5	3.8
108	5	4	4	5	5	5	5	5	5	5	4.8
76	5	5	5	5	5	5	5	5	5	3	4.8
122	5	5	5	5	5	4	5	5	5	5	4.9
92	4	5	5	5	5	5	5	5	5	5	4.9
47	5	5	5	5	5	4	4	5	5	5	4.8
57	4	4	4	5	5	4	4	4	4	2	4.0
17	5	5	5	5	5	5	5	5	5	5	5.0
5	5	3	4	4	4	5	5	5	4	3	4.2
62	5	4	5	4	4	5	5	5	5	5	4.7
74	4	3	4	4	4	4	5	4	4	5	4.1
53	4	3	4	4	3	4	3	5	3	4	3.7
84	5	4	4	5	5	5	5	5	5	4	4.7
145	4	4	5	4	5	5	5	5	5	4	4.6
35	5	5	5	5	5	5	5	5	5	5	5.0

Subcontract no.	1	2	3	4	5	6	7	8	9	10	Average
	Technical Evaluation Questionnaire Rating Factors										
59	5	4	5	5	5	5	5	5	5	5	4.9
115	5	5	5	5	5	5	5	5	5	5	5.0
118	4	5	5	5	5	5	5	4	5	4	4.7
104	5	3	4	3	5	4	5	5	4	5	4.3
6	3	5	5	5	3	5	5	4	5	3	4.3
28	5	4	5	5	5	4	5	4	4	4	4.5
51	5	5	4	5	5	5	5	5	5	4	4.8
16	5	5	5	5	5	4	5	5	5	5	4.9
54	5	4	5	4	5	4	5	5	5	5	4.7
56	3	3	5	4	5	3	4	5	5	5	4.2
69	4	5	4	5	5	4	5	5	5	5	4.7
72	5	5	5	5	5	5	5	5	5	5	5.0
129	5	5	5	5	5	5	5	5	5	5	5.0
58	3	4	5	4	3	4	5	4	4	4	4.0
144	5	5	5	5	5	5	5	5	5	5	5.0
80	5	5	5	5	5	5	5	5	4	4	4.8
136	5	4	5	5	5	5	5	5	5	4	4.8
55	5	5	5	5	5	5	5	5	5	4	4.9
77	5	4	3	4	3	4	4	4	5	4	4.0
15	5	4	5	5	5	5	5	4	5	5	4.8
70	5	4	4	5	5	5	5	5	5	4	4.7
140	5	5	5	5	5	5	5	5	5	5	5.0
79	4	3	5	4	5	4	5	5	4	5	4.4
63	5	5	5	5	5	5	5	5	5	5	5.0
97	5	4	4	5	5	5	5	5	5	5	4.8
11	5	4	4	4	4	5	4	4	5	3	4.2
44	5	5	5	5	5	5	5	5	5	5	5.0
141	5	4	5	4	4	5	5	5	5	4	4.6
98	5	5	5	5	5	5	5	5	5	5	5.0

Technical Evaluation Questionnaire Rating Factors

Subcontract no.	1	2	3	4	5	6	7	8	9	10	Average
81	4	4	3	4	4	4	4	5	5	4	4.1
105	5	5	5	5	5	5	5	5	5	5	5.0
25	4	4	4	5	5	5	4	5	4	2	4.2
18	5	3	5	5	5	3	5	5	5	4	4.5
138	5	4	5	5	5	5	5	3	5	3	4.5
111	5	5	5	5	5	5	5	5	5	5	5.0
Avg by factor:	4.5	4.2	4.5	4.6	4.7	4.6	4.7	4.6	4.7	4.4	4.6

VISUALIZATION OF MIXED AGED MACROPHAGE RESPONSE TO LPS CHALLENGE

Robert V. Blystone
Professor
Department of Biology

Trinity University
715 Stadium Drive
San Antonio, Texas 78212

Final Report for:
Research Initiation Program
Armstrong Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, D.C.

and

Trinity University

March 1993

VISUALIZATION OF MIXED AGED MACROPHAGE RESPONSE TO LPS CHALLENGE

Robert V. Blystone
Professor
Department of Biology
Trinity University

Abstract

Co-cultivation experiments with RAW 264.7 and 3T3 cells proved problematic. Vital staining with 800 parts per million of methylene blue allowed the following of stained and unstained RAW macrophage cells for periods up to 24 hours. Experiments where RAW cells were treated with Lipopolysaccharide proved unsuccessful. Computer modeling cell growth within culture chambers proved very successful. Growth in multiple cell chambers were computer averaged into "mean" image sets and demonstrated effects of media meniscus and slope of chamber wall on cell growth. Assistance was provided to three Armstrong Lab personnel in protocols not included in the original research proposal. These experiments dealt with electron microscopy of decompressed guinea pig lung and of additional cell cultures under investigation at Brooks AFB. Computer reconstructed, three dimensionally modeled, microwave -treated rat hypothalamus was also undertaken during the research period.

VISUALIZATION OF MIXED AGED MACROPHAGE RESPONSE TO LPS CHALLENGE

Robert V. Blystone

INTRODUCTION

The primary thrust of the proposed research was to discover whether RAW 264.7 cells in culture could respond differently to stimuli based on their age in culture. The speculation was that as RAW cells age they accumulate extracellular matrix on their surface. This matrix material may influence how the cells interact with their environment and thus to stimuli. The experimental approach was to incubate two different aged RAW cultures which were cytologically marked with a vital dye. Using computer imaging techniques, the co-cultivated and marked cells could be followed and differences in response followed.

In short, the objectives of the experiment were not met. The project proved to be too ambitious for the resources available. Portions of the experimental protocol were achieved; however, the question posed concerning the age-in-culture dependent response to stimuli was not answered.

The funded period did achieve other objectives not envisioned in the original proposal. Microscopy and computer support was given to four Armstrong laboratory projects not covered in the original research protocol. Five students were involved in these projects which resulted in four student presentations. Two professional presentations were given and one published, referred abstract appeared during the supported period. These additional projects will be reviewed here along with those results that were directed to the proposed research protocol.

METHODOLOGY

Phase one: As directed towards the original protocol.

Two cell lines were established in the lab at Trinity University for the purpose of following co-cultured cells with digital techniques. RAW 264.7 cells were cultured in RPMI 1640 medium with Penn/Strep added and 10% Fetal Bovine Serum. The cultures were maintained at 37°C with 5% CO_2. A second culture with 3T3 cells were grown in DMEM high glucose media with calf serum, also maintained at 37°C with 5% CO_2. These two cells types were co-cultured together in either RPMI or DMEM. The two distinctively different cell types would facilitate developing computer techniques to follow simultaneously two types of cells in a culture at one time.

A time lapse digital video system was assembled to follow the cultured cells for short periods of time. At Trinity, the means to monitor cells under magnification for periods of time in excess of one hour was not possible. Using a Nikon TMS inverted microscope with a Javelin 3462RGB CCD camera coupled to a Macintosh IIci computer with a Data Translation DT2255 frame grabber board, up to 200 time lapse frames could be recorded digitally. With this system it was possible to watch over short duration the behavior of cells either in monoculture and co-culture. Individual frames could be analyzed and image processed using NIH Image software and/or other software such as Adobe Photoshop.

Three vital stains were explored: Janus green B, methylene blue, and trypan blue. Based on cell responses, most of the work settled on using methylene blue as the principal vital stain. Experiments were performed using 100, 200, 400, 800, 1200, or 1600 parts per million of this vital stain. It was determined that in concentrations above 800 parts per million, methylene blue would inhibit growth. At levels below 200 parts per million, not enough vital staining took place for cells to be followed over any period of time.

Cells were grown in a variety of containers: standard T-25 flasks, T-75 flasks, and NUNC Labtek microscope slide mini-culture chambers. The Labtek chambers were quite useful and had been used in previous experiments. The decision was made to continue their use.

LPS was added to several cultures in an effort to monitor its affect on the growth of RAW cells. These experiments were not successful.

Phase two: collaborative experiments not included in the original protocol. Transmission microscopy and scanning microscopy were performed on both cells and tissues. Using either an Hitachi HS-8 or HU-11E transmission electron microscope or and ISI-40 scanning electron microscope, specimens provided by Dr. Mitch Garber or Dr. John Bruno were examined.

Dr. Garber was interested in guinea pig lungs that had been exposed to severe decompression. These lungs were fixed in a standard 0.1M phosphate buffer, 3% glutaraldehyde solution. The tissue was subsequently stained with a 1% OsO_4 solution and then dehydrated. Spurr plastic sections were made and the tissue examined. Some of the tissue was held from Spurr infiltration and prepared for scanning microscopy. These specimens were coated with gold and photographed with the ISI-40 SEM.

Photographic images were optically scanned into a Macintosh IIci computer and data concerning alveolar cells and capillaries were collected. This information was output to printers and photographs of digital images taken.

Dr. Bruno expressed interest in viewing some of the cell cultures he was working with at Armstrong Labs. His cell cultures were spun down and pelleted into 2% agar. The agar was hardened and then handled like tissue. The cells were fixed in a manner similar to Dr. Garber's lung tissue. Photographs were made by means of transmission electron microscopy. Several SEM preparations were made based on filter paper collected cells. These SEM preparations were not very successful.

Dr. Mason of Armstrong Labs had been working with a Trinity student, Laura Weigel. They had been trying to quantitate the appearance of c-fos protein in various regions of the hypothalamus of rat brain which had been treated with microwave. The decision was made to examine the brain tissue digitally using our equipment at Trinity. Brain slices were optically scanned using a Microtek 600ZS optical scanner. The images were manipulated with several software packages and three dimensional models were built. Also textbook images from rat brain atlases were scanned and similar three dimensional images built. Brain tissue was also examined with a digital imaging workstation and viewed at higher magnification. Based on these results, Dr. Mason decided to expend resources to equip his lab at Brooks with a digital imaging system.

And finally with undergraduate student Jim Jordan, the decision was made to continue to follow how Labtek well slides influenced the growth of RAW cells. Mr. Jordan built numerous modifications of the chamber slide system and examined its ramifications on growth patterns. Explored was the effect of the meniscus and the angle of the cell wall of the chamber on RAW cell growth. RAW cells were cultured in the chambers and prepared for microscopy by the following procedure. The cells were fixed with 3% glutaraldehyde buffered with 0.1M phosphate buffer. These cells were then stained with 1% OsO_4 for exactly one hour. After washing, the cells were dehydrated while still attached to the surface of the chamber slide. The upper chamber and its gasket had been removed. The stained slide was digitized directly with a Microtek 600ZS optical scanner. The digital image of the slide was rendered in several gray scale expressions using either Adobe Photoshop or NIH Image software.

With suggestions of technique provided by students Daniel Blystone and Tod Romo, computer simulations of cell growth in the chamber slides were created.

Positioned culture wells with different quantities of initial seed of cells or cells allowed to grow for differing intervals were prepared for digital analysis. Well chambers from multiple experiments were computer averaged into a single image. The images under differing conditions were "glued" together using Gryphon Morph software. This imaging processing resulted in a movie that modeled how the cells grew within individual chambers.

SUMMARY OF COLLABORATIONS AND RESULTS OF THE SPONSORED PERIOD

The RIP resources have been used in a number of ways, as outlined in the methodology section above, during the 1992 budget year.

Collaborations:

- Dr. Johnathan Kiel, Armstrong Laboratory scientist, Brooks AFB. Topic: The microscopy of Anthrax bacillus.
- Dr. Mitchell Garber, U.S. Air Force physician at Brooks AFB. Topic: Low atmosphere effects on the lung as measured by microscopy.
- Dr. Patrick Mason, Armstrong Laboratory scientist, Brooks AFB. Topic: Image analysis and localization of c-fos receptor sites in rat brain.
- Dr. John Bruno, Armstrong Laboratory scientist, Brooks AFB. Topic: Electron microscopy of cell cultures.

Student reserach:

- Allison Stock, senior, graduated May 1992: Development of co-cultivation methods for visualizing RAW macrophage cells stressed in two different ways.
- Jim Jordan, senior, graduated December 1992: The effect of the growth chamber geometry on the growth of RAW macrophage cells cultured in NUNC Lab-Tek culture slides.

- Laura Weigel, senior: Image analysis and localization of c-fos protein sites in rat brain.
- Daniel V. Blystone, junior (San Antonio College): Computer graphical representation of microscopy images.
- Tod D. Romo, graduate student (Rice University): Rendering of digital images of cultured cells.

Student presentations:

- Jim Jordan March 20, 1992, "Chamber geometry and cell growth." McGavick Awards presentation at Trinity University.
- Daniel V. Blystone March 27, 1992, "Computer aided three dimensional visualization of light and electron microscope images." Texas Society for Electron Microscopy Spring Meeting, San Marcos, Texas.
- Jim Jordan October 24, 1992, "Animating cell growth in a culture chamber." PEW midstates consortium for undergraduate research in biology, University of Chicago.
- Laura Weigel, March 25, 1993, "Visualization of c-fos protein in the rat brain after thermal challenges." 7th annual undergraduate research conference for the National Council for Undergraduate Research, University of Utah.

Paper presentations:

- May 14, 1992. "The role of pulmonary surfactant in extreme altitude exposures." Aerospace Medical Association annual meeting, Miami Beach, Florida (M.A. Garber, B.J. Stegmann, A.A. Pilmanis, and R.V. Blystone). Presented by Dr. Garber.
- Nov. 16, 1992. "Effects of chamber geometry on growth of cultured cells." American Society for Cell Biology annual meeting, Denver, Co. (R.V. Blystone, J.E. Jordan, T.D. Romo, and J. Kiel). Presented by Dr. Blystone.

Published abstract:

- Effects of chamber geometry on growth of cultured cells. Molecular Biology of the Cell 3(S): 91a. (R.V. Blystone, J.E. Jordan, T.D. Romo, and J. Kiel).

DISCUSSION

In experiments using Nunc, Inc. Lab-Tek® Chamber Slides, we observed that RAW 264.7 mouse macrophage cells were consistently in greatest density along the outer perimeter of each chamber. Images representing as many as 16 stained slides were aligned and computer averaged into a single representative composite which was pseudocolored or grayscale mapped. The study included Lab-Tek® 4 chamber (product code # 177399 & 177437) and 8 chamber (code # 177402 & 177445) slides with glass and Permanox® bases. Chamber walls of the upper structure are not symmetrical. Outer perimeter walls are at a greater angle which in turn influences the position of the culture media meniscus in the chamber. A silicone-based material holds the upper structure to the base. Cells can grow up to 30μm under the gasket suggesting that the gasket assembly does not retard cell growth. Occasionally cells were observed to grow in a streaked pattern on Permanox slides which have an added surface coating. 3T3 cells exhibited the same general growth tendencies although not identical to the RAW 264.7 cells. Following cell distribution after seeding proved inconclusive. Studies using inverted upper structures were inconclusive and insufficient time did not allow their repeat. In conclusion, we found the upper structure geometry and surface coating do influence how cells distribute themselves in Lab-Tek slide growth chambers.

Digital images of the cell growth were converted to 20 unit gray scale representations. Reducing the 256 gray scale to 20 units sharpened the distinction of cell density boundaries considerably. Computer averaged cell cultures were collected for several days of growth. Key transition points on each day's averaged slide were identified in the Morph software package. The computer then plotted the best transition between the key points for each day. In this way we were able to create a movie that modeled how cells grew preferentially within their growth chamber.

These two results were the most positive outcomes in handling the RAW cell portion of the research protocol. Other than determining the range in which the vital stain methylene blue works on the RAW cells, the original research protocol was not sufficiently explored. We did, however, gain a great deal of experience in computer imaging and the collaborative work with Dr. Mason is most illustrative of what was done.

In the project for which we provided assistance to Dr. Mason, the following description from Laura Weigel's work indicates what was accomplished. The mRNA for the intermediate-early gene c-fos (a proto-oncogene) expresses transcription within 15 minutes of thermal stress and is transient (1 hour) in duration. The translated c-fos protein concentrates in the nucleus in which it may remain up to 4 hours. C-fos protein can be histochemically revealed by a double antibody coupled DAB (3,3'-Diaminobenzidine tetrahydrochloride) reaction. A low-cost computer visualization procedure was developed to automate the quantitation of c-fos protein produced after thermal stress in circumventricular nuclei of the rat brain. However, the automated computer analysis offered no time advantage in counting c-fos reaction sites compared to counting manually. The imaging process did allow the construction of three-dimensional distribution maps which provided a new perspective on c-fos localization.

ACKNOWLEDGMENTS

The Brooks AFB laboratory of Dr. Johnathan Kiel provided my lab with numerous materials and suggestions. Dr. Parker and Sgt. Alls provided a great deal of assistance. Dr. Kiel patiently supplied advice and encouragement at critical points in the investigation.

INTEGRATED TASK ANALYSIS METHODOLOGY FOR SYNTHETIC TASK DERIVATION

Gerald P. Chubb
Assistant Professor
Department of Aviation
Ohio State University
2160 West Case Road
Columbus, OH 43235

Final Report for: Research Initiation Program

Armstrong Laboratory
Sustained Performance Branch
Brooks AFB, TX

Sponsored by the
Air Force Office of Scientific Research
and
The Ohio State University

December 1992

INTEGRATED TASK ANALYSIS METHODOLOGY FOR SYNTHETIC TASK DERIVATION

Gerald P. Chubb
Assistant Professor
Department of Aviation
Ohio State University

Abstract

Synthetic tasks are a surrogate for whole task or high fidelity simulation and are meant to better mimic real-world task dynamics than simple laboratory task paradigms typically used in academic research. To construct a synthetic task that represents some real-world counterpart, some form of task analysis is required, and several approaches are recommended in the literature.

The present effort examined the reliability and validity of Miller's proposed task analysis strategy. The TRACON II simulation of Terminal Radar Approach Control was used as the task context. Subjects first learned the task, then prepared an analysis. Subjects were also asked to provide paired comparison ratings of similarity for twenty-five task terms. PATHFINDER network analysis showed how subject's interpretation of the task analysis terminology differed among subjects, and how relationships among terms varied on repeated measurement.

The present study also outlines ITAM, a recommended, methodology or approach for conducting an integrated task analysis: looking at requirements, task characteristics, performance criteria, abilities, and descriptions of real-time behavior from a common framework. Originally, the study proposed examining the reliability and validity of ITAM, but this is left for further study.

Table of Contents

List of Figures

List of Tables

INTEGRATED TASK ANALYSIS METHODOLOGY FOR SYNTHETIC TASK DERIVATION

Gerald P. Chubb

SECTION 1.0

INTRODUCTION

The goals of the present effort were to: a) study the reliability of Miller's Task Strategies approach to task analysis, using his terminology as descriptors of human activity in a simulated task environment (Terminal Radar Approach Control), b) determine the validity of the derived description as compared with existing task analysis data for that same operation, c) recommend an improved Integrated Task Analysis Methodology (ITAM) that includes a strong cognitive component and test its reliability and validity, and d) define a method for deriving reductive and composite synthetic tasks using ITAM results. The reliability and validity of the ITAM methodology have not been tested, in part because specification for an ITAM was more difficult than anticipated.

Reductive synthetic tasks are an abstraction of some real-world task that serves as its surrogate for laboratory studies. Composite synthetic tasks are collections of simpler tasks that require time sharing in a manner that mimics real-world tasks. Presently, reductive and composite synthetic tasks are developed ad hoc, not directly linked back to real-world activities. A systematic procedure for deriving synthetic task characteristics and formulating performance predictions is sought. The methodology should rest on some form of task analysis which can characterize both the real and synthetic tasks.

The ITAM methodology should also be useful for training device design. So, the ITAM methodology could serve the needs of multiple users. How well it meets those needs is not yet known.

Task analysis is a prerequisite to at least three major activities during weapons system development: 1) design evaluations, 2) instructional system design, and 3) production of maintenance performance aids. Two problems are often encountered: 1) different analyses are not comparable, and 2) details of cognitive activities are not included to the degree desired.

The first problem might be solved if there was a standardized, teachable method for doing task analyses. But little is known about the reliability and validity of a task analysis. Any proposed standardization should begin by trying to overcome the root causes for non-reliability and invalidity, if those causes can be identified. The second problem is solved only by developing suitable techniques, and then submitting them to appropriate reliability and validity testing. The present work addressed these problems in the following fashion.

First, a study of task analysis reliability and validity was conducted to determine how well beginning task analysts could produce the desired product. The task studied was terminal radar approach control to an airport. Students in the Ohio State University's Department of Aviation served as test subjects, so this task was intrinsically interesting to the students who chose to participate. A computer game of that operation was available, and it was known that a professional task analysis was available for validation study comparisons.

Details of this empirical study effort are documented in Section 2.0 of this final report.

Second, a literature search was conducted to determine what work has been done that relates to the development or improvement of cognitive task analysis methodologies. Verbal protocol analyses of individual subjects are often used to obtain information about cognitive activities, and there is some controversy about how suitable such materials really are. Any ultimate methodology must in some way integrate both objective and subjective descriptions of activity if the composite definition of human operator activities are to be identified. Details of the literature review and our recommended approach to this problem are treated in Section 3.0 of this Final Report.

Section 4.0 briefly describes an ongoing effort to instrument an Air Force T-40 simulator as a test bed for future research activities, since some of the RIP funding was spent for this purpose with the expressed approval of the sponsor, who operates a similar simulator (which was the basis for the work discussed here).

Section 5.0 summarizes the conclusions of the empirical study of Miller's task analysis terminology and recommends further ITAM research and development activities.

SECTION 2.0

EMPIRICAL TASK ANALYSIS STUDY

The empirical studies of task analysis took place over two quarters. Initially, the baseline reliability and validity study was to be done in a single quarter, followed by the ITAM study during the next quarter. Two problems prevented that plan from being realized: 1) the sample size for the baseline study was smaller than desired, and 2) the literature on cognitive task analysis was small, but the literature on cognitive processes is immense. Developing the ITAM therefore required more time than had initially been allocated. Also, preliminary baseline study results indicated that some of the causes for non-reliability of task analyses would not be easily overcome, and development of ITAM should necessarily be done more slowly and carefully.

The baseline study was therefore extended to increase sample size for reliability analyses, but the validity evaluation effort had to be constrained to the smaller sample size because of timing considerations (overlap in study efforts). In a sense, it can be regarded as a pilot study for the validation methodology, now that the larger pool of task analysis results is in-hand.

All studies were conducted during Winter and Spring quarters of 1992. Data reduction and some minor analyses were completed during the Summer. Final data analysis and report preparation were accomplished in the Autumn quarter. The following sections document respectively: 1) the method, 2) data analysis, 3) results, 4) discussion, and 5) conclusions.

2.1 Method

Students from three separate sections of Aviation 540 (Human Factors in Aviation) served as subjects in the task analysis study itself. Their participation was voluntary. Student motivation for participation was 10 extra points toward their final course grade. This is sufficient to change the assigned grade by one letter grade (eg., B to A, or C to B, etc.). Students in Aviation 640 (Simulators for Training and Research) performed the reliability and validity assessments as a required class assignment.

During the Winter quarter, a pool of 15 subjects from two sections of Aviation 540 completed the TRACON II training, but three did not complete the task analysis materials. Four of 27 students (12.5%) in section 1 participated, all finishing the task analysis. Eleven of 50 (22.2%) in section two participated, but only eight (16% of the section) finished the assignment. Only 12 (total) turned in all of the task analysis materials for the baseline reliability / validity study, but thirteen did complete the task analysis. In the Spring quarter, another 19 of 45 students (42.2%) volunteered, nine (20% of the section) completed some of their assignments after TRACON II training, but only 2 (4.4%) actually completed their task analyses, and one of those failed to complete the comparison report. The two section 3 task analyses were therefore not included in the reliability and validity assessments: 1) they were not available in time to do so, and 2) there was little point in executing a separate evaluation for only two additional task analyses.

Each student was trained in the operation of the TRACON II simulation of airport terminal radar approach and control (marketed by Wesson International, Austin, Texas) for the Los Angeles region, the default area for this software package. Students scheduled blocks of time that were of the same duration as a typical class at OSU (nominally, 48 minutes per session, allowing 12 minutes each hour for class changes to take place). The traffic density (a changeable parameter for the TRACON II simulation) was kept at modest levels, and students continued practicing until they believed they had sufficient familiarity with the operation to be able to perform a task analysis of their activities. Performance score data was recorded for these sessions by copying the built-in TRACON II scoring algorithm to a data sheet and then entering the data into an ASCII file for subsequent analysis. Performance data are missing for two students who had access to TRACON II outside of class and therefore worked independently.

During initial instruction, arrival rates were set at 3 aircraft per five minutes. Subsequent sessions increased this to 8 aircraft in five minutes. All students were assisted by a trained and experienced tutor to minimize the time it took to learn the operation of the TRACON II simulation game. Students were encouraged to ask questions to clarify their understanding as they developed some experience with the operation. All were aware from the outset that they were expected to prepare their own task analysis of approach (not departure) control and to compare their work with an abridged version of the FAA's formal task analysis for this same operation.

In addition, students were asked to respond to a questionnaire at three separate times: 1) before doing their task analysis, 2) after the task analysis had been completed, and 3) after they had documented the comparison of their own effort with the formal FAA task analysis results. The instructions for the three administrations, as well as a copy of the questionnaire itself appear in Appendix A. The questionnaire data provided a set of 300 paired comparison similarity ratings that were analyzed using PATHFINDER (Schvaneveldt, 1990), prepared as follows.

Miller's twenty-five terms, as documented in Appendix A of Fleishman and Quaintance (1984) was provided to each subject. Instructions for performing a task analysis were provided, along with examples of three different formats: 1) indentured outline, 2) tabular format, and 3) graphic. The examples were developed from a concept map of how one would compute a z-score. Students were encouraged to use Miller's terminology in their own task analysis. Further, they were instructed to define any terms they used that were not part of Miller's terminology.

Students were also asked to prepare a short, written report based on their comparing their own task analysis with portions of the formal task analysis developed by CTA for the FAA (Alexander, et al., 1989). Written instructions for preparing this comparison were provided as well as an abridged set of graphic task analysis pages from CTA's report. The CTA report itself was placed on closed reserve in the Engineering Library, and students were encouraged to examine that document, but they were not required to read it in its entirety.

Not every student completing the task analysis submitted the requested comparison report.

Finally, a questionnaire was also administered that solicited subject's opinions regarding what was most and least difficult about the task analysis assignment, as well as general questions about the adequacy of instructional materials they were given. These data are regarded as anecdotal for the purpose of reporting study results. Subjects were also asked to provide selected demographic data. Not every subject complied with these requests either.

Eight students taking Aviation 640 (Simulators for Training and Research) were given the set of thirteen task analyses and the Fleishman and Quaintance Appendix A that describes Miller's terminology. They were asked to review the task analyses, and to do so carefully, they were instructed to answer several questions about those analyses. Those answers are of little importance here, other than having assured that the evaluators did familiarize themselves with the content of the task analyses before evaluating them.

The first evaluation these eight AV 640 students performed was to compare, on a pairwise basis, how similar one task analysis was to another. These similarity measures were taken as the raw data for assessing the reliability of the set of task analyses. The second evaluation performed by the AV 640 students was to compare each AV 540 student's task analysis with the abridged portions of the CTA analysis that treated the same terminal approach operation, albeit for the actual system rather than for TRACON II. The scoring forms for these assessments are presented in Appendix D.

2.2 Data Analysis

Study results are broken out in the following subsections: 1) performance data on TRACON II, 2) questionnaire data on the similarity of Miller's terminology, 3) reliability and validity assessments, and 4) anecdotal data from subjects regarding other aspects of their task analysis efforts. All of the statistical analyses of these data were performed using SYSTAT, a package developed specifically for Personal Computers (PCs), hostable on both IBM (and compatible) computers as well as the Macintosh.

One of the more interesting results is that all but one of the students who completed this assignment had already earned an A in the course. The one exception had earned a B. None were therefore in need of the extra points that project completion provided. This suggests that only the better students are represented in the study results. It may also suggest that task analysis is sufficiently difficult that less capable students choose not to attempt doing it, even when it could change their course grade by an entire letter grade.

It is also apparent that by Spring quarter, many of the AV 540 students had been told of the TRACON II experience, since more volunteers signed up for tutoring sessions. But the poor completion rate (1 in 19) also indicates that few of those students were serious about completing the assignment, or Spring quarter is just a bad time to expect high completion rates because of the many competing activities and interests students have at that time of year. Section 1 completed 14 TRACON sessions (total), section 2 completed 61 sessions, but section 3 completed 116 sessions. It appears students were motivated to learn TRACON for the experience alone, but not to do task analysis, even for a letter grade change.

Also, not every student cooperated with completing the assignments. Twenty-four completed the first of the three questionnaires that asked for similarity ratings among pairs of the Miller terms. Only 79% (19) completed the second administration of the questionnaire, and only 42% of those completing two versions (8 of the 24 starters) bothered to complete the third. The three hundred item questionnaire clearly was too bothersome to do three times in succession for most of the participants (16 of the 24 refused to comply with directions to complete the questionnaire all 3 times).

Three sets of data had to be encoded: 1) performance data scores on TRACON II, 2) questionnaire data for PATHFINDER and SYSTAT analyses, and 3) reliability and validity questionnaire data for PATHFINDER and SYSTAT analyses. Three of the items in the task terminology questionnaire (1%) had typographical errors that were not caught before administration to the subjects. For these items, a score of four was assigned, midway along the scale. Preliminary work with the PATHFINDER (PF) software had demonstrated that this approach to missing data did not distort the other relationships among the data, as evidenced from visual inspection of the graphic layout of the PF network.

The initially encoded data were independently checked by a second person. Errors in transcription and encoding were corrected. Subsequent analyses identified only two minor typographical errors, both of which were corrected in subsequent editing by a third reviewer, prior to the SYSTAT analyses. Several programs had to be written for reformatting data in a form suitable for the SYSTAT analyses, and these are described later.

The PATHFINDER input for analyzing the paired comparison similarity ratings of the twenty-five task terms were prepared from the raw data by writing two DOS 5.0 QBASIC programs (PROG1A.PJC and PROG2A.PJC) to do the reformatting. The first program takes the compacted form of the encoded raw data and creates an intermediate file consisting of a simple string of single data elements for each subject's raw scores. The second program sets up PATHFINDER parameters and header information and then generates the data matrix in the proper format for PATHFINDER analysis. Programs 3 and 4 are modifications to handle the reliability questionnaire data, and programs 5 and 6 are modifications to handle the validity rating data.

One subject completed the task terminology paired comparison questionnaire for the third administration without completing one for the second administration. In the SYSTAT analyses, this was treated as a set of two completions, but in the PATHFINDER analyses, the layout diagram for this subject is included with other questionnaire layouts for the third administration (last subsection of Appendix E).

2.2.1 TRACON II Performance Data

Since each student decided for themselves when they thought they were ready to do the task analysis, even the students who finished the assignment spent different numbers of sessions exercising the TRACON II simulation. The data for a session were first transcribed onto a score sheet and later entered into the computer in an ASCII file. Using the DOS line editor (EDLIN), the original files were then stripped of extraneous characters (file description information), leaving the raw data.

The performance data record had nine elements: 1) the Julian date, 2) the number of planes generated, 3) the interval of generation (base rate: per x minutes), 4) session duration, 5) separation, 6) handoff errors, 7) missed approaches, 8) pilot requests, and 9) emergencies. Emergencies were not exercised, so this data element was zero for all subjects. The dates of participation are of little interest. The numbers of planes and interval data simply confirm that what happened corresponded closely to the selected parameter settings. The primary criterial data are conflicts and handoff errors. Missed approaches are not solely the result of controller error (poor piloting contributes), so this data element is confounded. The pilot requests is essentially a distractor. They were infrequently encountered in these runs.

The raw data were then accessed by the SYSTAT Data Module (which converts ASCII files to a computationally more efficient but unprintable binary format) and converted to a form suitable for analysis using other SYSTAT modules. The data files were first sorted by subject name. Descriptive statistics were then computed both for the three individual sections and then by subject name within those sections.

Since there is reason to be concerned about the comparability of the various sections, two inferential tests were performed. The first t-test examined whether those who finished the task analysis had significantly more or less experience with TRACON II than those who chose not to finish the task analysis, since the amount of practice might influence relative "expertise" in understanding the operation being analyzed. The second t-test examined whether the performance data (specifically conflicts and hand-off errors) for sections 1 and 2 differed significantly (since these were the sections generating the task analysis data submitted to further analysis).

2.2.2 Similarity Data on Miller's Terminology

The similarity data were subjected to two kinds of analysis: 1) SYSTAT analyses of the statistical significance of differences between the three administrations of the questionnaire, and 2) analyses using PATHFINDER (Schvaneveldt, 1990), which forms a network based on scaling the similarity data as distances between terms: a kind of semantic network or concept map.

2.2.2.1 SYSTAT Analyses

The similarity data were tested for significant differences between the various administrations of the questionnaire. If the data are not significantly different between questionnaires, then averaging across the three administrations would essentially filter some of the measurement noise. If significant differences exist, then each administration needs to be interpreted separately: subjects' understanding of relationships among the terms may not be stable, possibly reflecting perceptual or conceptual shifts in the subjects' interpretation or understanding of those relationships.

The SYSTAT data input presented the largest problem in the analysis of similarity scores. Several programs had to be written in order to format the data for these analyses. The data set was larger than SYSTAT could handle as an unbalanced design. The first approach to handling this problem segmented and grouped the 300 questions. The upper limit of SYSTAT variables is 256, and the upper limit on levels within a dimension is 99.

The three hundred questionnaire items were therefore first reduced to four segments: 1) items 1-75, 2) 76-150, 3) 151-225, and 4) 226-300. Within each segment, the dependent scores were grouped into 15 sets of five scores. This then permitted Analysis of Variance and contrast tests to be performed using the Multiple General Linear Hypothesis (MGLH) routine. Unfortunately, that confounds variables, and the results were not deemed trustworthy. Therefore, another set of BASIC programs was written that treated the questions as individual cases, since SYSTAT is not limited in the number of cases it will accept.

Since not every subject completed every administration of the questionnaire, two sets of analysis of variance were performed. The one compared questionnaires 1 versus 2 for those 19 participants who completed the questionnaire twice. The other examined the contrasts between questionnaires 1 versus 2 and between 2 versus 3 for those 8 participants who completed all three administrations of the questionnaire.

2.2.2.2 PATHFINDER Analyses

The spatial configuration of the resulting network conveys which terms are closer versus distant from each other. This permits some interesting insight into the nature of an individual's interpretation of the concepts: how they relate to one another. Unfortunately, the software provided for these analyses still has some problems generating good graphics. Not all of the plots were generated properly. The printer manuals warn that more than one command exists in BASIC, and a particular form of the print command is needed for IBM compatible computers. Since we do not have access to the source code, we cannot change the program to correct this problem. Since the graphs do appear correctly on the screen, some are in a different scale than the others. Only one was drawn by hand, in part or whole. Unfortunately, all of the plots contain the same typographical error: the term "interpret" was entered as "interept" in the input file and was not caught until the analyses were being printed.

PATHFINDER datasets were prepared for each questionnaire a subject completed. There were 24 analyses for the first administration, 18 sets for the second administration, and 9 sets for the third administration. The program generates a graphic which visually portrays the spatial relationships among a set of terms (twenty-five terms in this case: 300 paired comparisons). The character of that graphic is influenced by parameter values supplied by the analyst. The parameter values used in these analyses corresponded to the recommended defaults. The parameter r was set to infinity and the q parameter was set to (n-1) = 24. The coherency cutoff criterion was set to a value of three. These parameter selections restrict the number of paths generated between terms, producing a more meaningful set of connections than other choices, which were explored with one subject's data to see the effects.

The KNOT software is basically self-installing. A subdirectory called \KNOT is created, and the compressed files are unpacked. All data used for an analysis needs to be copied into the KNOT subdirectory before trying to perform the analysis. The PATHFINDER program set was exercised in six major steps, by issuing the following commands at the prompt in subdirectory KNOT. In the following, filename refers to the character string used to name a particular set of data, and that will change for each analysis. However, the filename with the extension: .trm refers to a set of labels that you want assigned to the nodes in the network. Otherwise, the entries required are exactly as shown before the --- which initiates our explanation of what the commands does.

pf -d filename.prx --- computes the pathfinder network, creating an intermediate file with the name: filename.pf

dist -pf filename.pf --- computes distances between the nodes in the pathfinder network, using the intermediate file generated by the pathfinder (pf) routine, generating another intermediate file with the .gds extension

spring -d filename.gds -t filename.trm --- scales the locations for the nodes that will appear on diagrams and associates assigned labels (terms: filename.trm) to the nodes, generating yet another intermediate file, this time with the extension: .spr

copy filename.spr+filename.pf filename.lo --- combines the intermediate files needed as input for the layout routine, which actually draws the diagrams from the new intermediate file with the extension: .lo

layout -i filename.lo --- generates a graph on your computer terminal screen; this can only be printed on a dot matrix printer, not a laser printer, which is why the diagrams appear to be a bit crude

coh -c 3 -d filename.prx --- this routines computes the cohesiveness of the data, and here we have specified a cutoff score of three; unless the similarity score is 3.0 or larger, the two task terminology labels will not be associated

The output displayed on the computer terminal can be shrunk and then enlarged. All of the relationships are proportionately scaled. The largest size best illustrates the relationships, but often at this level of magnification, one or more terms may be off the viewing screen (and too large to print as well). Reducing the image size allows these terms to be seen, but only at the risk of possibly covering one term by another, in part or in whole, depending on the number of characters in each of the overlapping terms and which falls in front versus behind the other.

Also, printouts tend to span multiple pages on a dot matrix printer. By trial and error, it was discovered that the first reduction of the image was sufficient to print the graphic on paper sized 8 1/2 by 14 inches. Also, by rolling the printer's platen, it was possible to adjust the printer manually so the image did not print across a page perforation. The 8 1/2 x 14 inch format was used to interpret run results, but often the diagram printed was incomplete, having to be completed by drawing missing parts by hand.

To put the image on the 8 1/2 by 11 inch paper of Appendix E in this document, another two reductions in image size were necessary. In a few cases, the 8 1/2 by 14 inch presentations were reduced using a copier, since they are a substitute for an 8 1/2 x 11 inch diagram that did not print properly. All of the diagrams would have been done this way, except these results printed more poorly than the smaller diagrams, for some unknown reason, presumably the print command used in the program source code.

2.2.3 Reliability and Validity Assessments

The reliability and validity data were also analyzed using SYSTAT as well as being subjected to PATHFINDER analysis. The two QBASIC programs used for questionnaire data reformatting (PROG1A.PJC and PROG2A.PJC) were modified for this purpose. The PATHFINDER results show which subjects task analyses are most like others' and how each is positioned with respect to the CTA analysis. QBASIC programs PROG3A.BAS and PROG4A.BAS were used to prepare reliability data for PATHFINDER. QBASIC programs PROG5B.BAS and PROG6B.BAS were used to prepare the validity data for PATHFINDER. QBASIC programs PROG7A.BAS and PROG8A.BAS were used to prepare reliability and validity data respectively for SYSTAT analyses.

The datasets generated by the QBASIC programs had two problems easily corrected by editing the files with EDLIN. First, leading quote marks were inserted which need to be removed before PATHFINDER processing. Second, line 8 of the dataset has a spurious zero inserted, which needed to be removed. Future use of these programs should correct these problems by appropriately changing the source code. The cause of the first problem was not known, so it was left uncorrected. The cause of the second problem was later found to be due to reading subject names from the raw datafile, putting them into the intermediate datafile, and then reading them in as data (instead of characters), which creates a zero data entry. Not writing the names to the intermediate datafile eliminates the problem (with some additional changes in loop logic in the companion program -- the even numbered one). These changes were incorporated into PROG5B.BAS and PROG6B.BAS.

The reliability data consist of ratings of similarity among the 13 task analyses. This resulted in 78 paired comparisons for each of eight raters. Layout diagrams for the reliability data are presented in Appendix F. The validity data consisted of comparisons of each subject's task analysis with that done by CTA: one similarity rating score for each of the thirteen task analyses. The same eight evaluators judged both reliability and validity.

To use PATHFINDER for analysis of the validity ratings, the data matrix generated consists largely of zero entries. Each of thirteen entries is not compared with the other twelve, but only with the fourteenth item: CTA's analysis. The upper triangular matrix has all zero's except for the last entry in each row. This leads to a very simple and consistent pattern in the layout diagrams, as shown in Appendix G.

SYSTAT was used to get descriptive statistics for these data, and to test two null hypotheses: 1) that the reliabilities ratings were effectively of zero similarity, and 2) that the validity ratings for student task analyses reflected no similarity to the CTA task analysis. In each case, a simple t-test was used to test the null hypothesis that the evaluator's ratings were essentially zero. In the case of the reliability data, this null hypothesis assumes that unreliable task analyses would exhibit no similarity from one task analysis to another (and some evaluators did use the extreme scores of 0 and 9). In the case of the validity data, the null hypothesis assumes that there is zero similarity between the student's task analysis and the CTA task analysis.

Between paired-comparison differences for the reliability data and between analyst differences for the validity data were assessed using a simple one-way ANOVA. Similarly, one-way ANOVA was also used to test for statistically significant between evaluator differences for both the reliability and validity assessments. Finally, a post-hoc contrast test was administered to determine whether there was a significant difference between the ratings of relatively more or less valid task analyses, using the middle score of the thirteen as a neutral point (weighted zero in the contrast).

2.2.4 Anecdotal Data and Other Study Results

Three sets of anecdotal data exist: 1) the task analyses themselves (since subjects sometimes included annotations), 2) comments appearing in the comparison reports (most, but not all) subjects prepared, and 3) comments or remarks made on the final task analysis questionnaires (which some subjects failed to fill out). The results presented are abridged to reflect only those of some lasting interest (meaningful insights), not complaints or exercise specific caveats.

2.3 Results

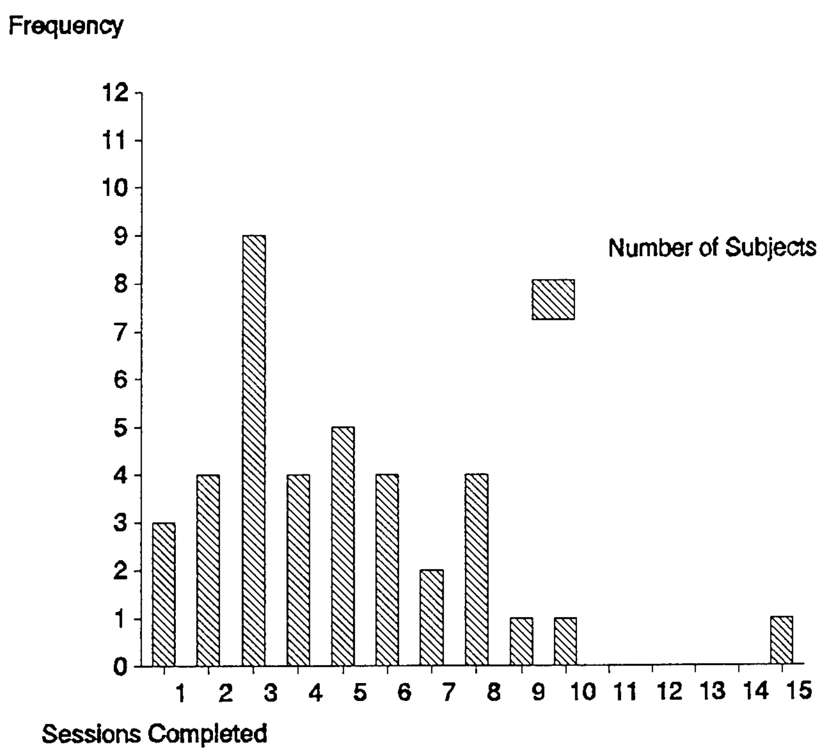
The results of the data analysis follow the same format as section 2.2. The between evaluator differences were not significant for either the reliability or the validity data. While reliability and validity were both shown to be significantly different from zero, the contrast effects demonstrated that some of the student's task analyses were more similar to CTA's than others' were.

2.3.1 TRACON II Performance Data

Figure 2-1 is a histogram of the frequency of session completions for the subject pool as a whole (task analysis finishers and non-finishers). The mode was three sessions. The distribution is positively skewed by the one section 3 subject who performed 15 sessions. This outlier drove the average number of sessions to nearly 5 (4.82), with a standard deviation of 2.9. These data are strongly influenced by section 3 participants. Four students from section 1 account for 14 sessions. Fourteen students from section 2 account for 61 sessions. Twenty students from section 3 account for 116 sessions.

The eighteen (18) students from sections 1 and 2 completed 75 sessions total, while twenty (20) section 3 students completed 116 sessions. Thirteen task analyses were completed by sections 1 and 2, but only two by section 3. Not only did section 3 tend to spend more sessions exercising TRACON II, there was also a greater variability in how many sessions were completed and how many completed sessions were considered "adequate" before doing the task analysis. One section 3 student served in 8 sessions, the other in 15, before doing the two submitted task analyses for section 3. Because of these apparent differences in exposure, we need to examine the performance scores to determine whether those with greater exposure did better.

Figure 2-1. Histograms of TRACON II Session Completions



For the task analysis finishers (ten people from sections 1 and 2), performance data are available for only 9 of the 13 subjects. Three of the nine subjects spent 1, 2, and 5 sessions, and the other six paired off, with two subjects spending 3, 4, and 6 sessions respectively. The mode for this group was 4 sessions (3 subjects); two subjects spent 3, or 6 sessions respectively; and three spent 1, 2, or 5 sessions. The average for these finishers was 3.8 sessions, with a standard deviation of 1.6. For the non-finishers in sections 1 and 2 (eight people), there was a much broader spread: ranging from 1 to 9 sessions, with an average of 4.6 sessions and a standard deviation of 2.9. The difference between these groups is 0.8 sessions, which is not significant ($t = 1.56$), given the substantial variance in the two samples, especially the non-finishers.

Table 2-1 presents the descriptive statistics for performance scores for each of the three sections. Section 1 committed an average of 2.57 conflicts: separation minimums were not adequately maintained. Section 2 averaged 0.49 conflicts per session by contrast. Section three had 0.53, comparable to section 2. As for handoff errors, section 1 committed .43 while section 2 committed .30. Section 3 committed .27, closer to section 2 than to section 1 performance. Simple t tests of differences between sections 1 and 2 on both the number of conflicts and the number of errors were not statistically significant (1.34 and 0.59 respectively). Therefore, we conclude that there are no significant performance differences between the two groups that provided the task analyses subjected to reliability and validity assessments.

Table 2-1. Performance Score Statistics by Section.

	Section 1	Section 2	Section 3
Number	4	14	20
Sessions	14	61	116
Average Conflicts	2.57	0.49	0.53
Conflict Variance	25.34	3.55	2.13
Average Handoff Errors	0.43	0.30	0.27
Error Variance	0.57	0.31	0.30

2.3.2 Similarity Data on Miller's Terminology

Subjects were asked to keep time of how long it took them to complete the questionnaire, consisting of 300 paired comparisons of the twenty-five terms. Eleven subjects actually provided data. The minimum time recorded was 24 minutes, and the maximum was 90. The average was 52 (median of 50) minutes, with a standard deviation of 25 minutes. Based on the standard error of the mean being 7.25 and assuming the data are normally distributed (which is not the case) then it is reasonable to expect that 52 + 3(7.25) = 71.75 minutes (more than an hour, but less than an hour and a quarter) should be sufficient for 99% of the subjects given this questionnaire. However, that criterion was not met by three of the eleven subjects, due to the positive skewness of the time data. One and a half hours would suffice for all of the subjects reporting.

Two kinds of analyses were performed: 1) statistical analyses using SYSTAT, and 2) PATHFINDER network analyses. The PATHFINDER results are graphical in nature, presented in Appendix E and described in Section 2.3.2.2. The SYSTAT results confirm that: 1) significant between subject differences do exist in the questionnaire data submitted for PATHFINDER analyses, and 2) significant differences also exist between: a) first and second administrations of the questionnaires for those completing only two sets of paired comparisons, and b) significant differences also exist between administrations for those completing all three sets. However, the between questionnaire contrast tests were not statistically significant, even at the 0.05 level, a result not uncommonly encountered because of the reduced degrees of freedom for such tests.

2.3.2.1 Terminology SYSTAT Results

The first two analyses of variance, using different models to estimate the least squares best fit indicated significant differences between the separate administrations of the questionnaires, so the data cannot be pooled across questionnaire administrations. Each administration leads to unique patterns of comparisons among the terms being used. Either subjects cannot provide stable interpretations, or those interpretations are drifting over time.

Table 2-2 presents ANOVA results for the comparisons between administrations for the 19 subjects who completed the questionnaire twice, while Table 2-3 presents corresponding ANOVA results for the subjects completing all three administrations of the questionnaire. In both cases, the main effects for differences between questionnaire administration were statistically significant as were the subject effects and the interaction of subject and questionnaire. This interaction effect suggests that some subjects were less stable in their interpretations of the Miller terminology than others. An alternate interpretation is that some subjects were less cooperative about taking the questionnaire, so their variability may reflect unwillingness rather than inability to provide stable similarity ratings.

While the specific contrast tests for questionnaire effects across the three administrations was not statistically significant for comparisons among the more diligent subjects (who completed all three administrations), the differences among the 19 subjects completing two administrations were statistically significant. This seems to support the suspicion that not all of the subjects were cooperating, and those who did comply with the instructions exhibit greater stability in their similarity scores.

Table 2-2. ANOVA Results for Q1 Versus Q2.

Source	Sum of Squares	df	Mean Squares	F	p
Questionnaire (Q)	50.086	1	50.086	10.143	0.001
Subjects (S)	13230.509	18	735.028	148.847	0.000
Q x S	1067.420	18	59.301	12.089	0.000
Error	56107.251	11362	4.938		
Q1-Q2 Contrast	50.086	1	50.086	10.431	0.001

The between subjects effects were substantial and highly significant, indicating little is gained from a standard set of terms, which are still subject to personal interpretation, definitions not withstanding. This is a somewhat discouraging result, in that it suggests the magnitude of the difficulty one encounters in getting people to agree on the interpretation of any set of labels used for task analyses. Even with defined terms, subject's interpretations of relationships among those terms is significantly different. While the ANOVA results confirm this result, the PATHFINDER layouts in Appendix E give very graphic evidence for this result. (Alternately, the SYSTAT ANOVA results simply confirm the statistical significance of what you can see so graphically as between subject differences in PATHFINDER layouts of Appendix E).

Table 2-3. ANOVA Results for Q1-Q2 & Q2-Q3 Comparisons.

Source	Sum of Squares	df	Mean Squares	F	p
Questionnaire (Q)	50.888	2	25.444	6.021	0.002
Subjects (S)	3881.864	7	554.552	131.222	0.000
Q x S	970.181	14	69.299	16.398	0.000
Error	30326.173	7176	4.226		
Q1-Q2 Contrast	11.408	1	11.408	2.699	0.100
Q2-Q3 Contrast	14.083	1	14.083	3.333	0.068

2.3.2.2 Terminology PATHFINDER Results

Describing a graphic is a difficult task in its own right, and the PATHFINDER output seems particularly difficult to describe. It may be best to examine the charts individually and collectively to discern the differences that exist. Several questions might help focus attention on some of the more interesting aspects of these diagrams.

First, is there one concept (or even a cluster of terms) that appear to be a central focal point for the diagram? Second, what is the overall orientation of the diagram (vertical, horizontal, or diagonal)? Third, what kinds of connections or patterns appear to be a dominant characteristic (chains of terms, a star term with spokes, diamonds or rings of terms, combinations of patterns interlocked)? Fourth, how many connections emanate from a particular term, especially "stars" that radiate spokes versus rings, diamonds, or chains that link term-to-term.

The other consideration is change: 1) between subjects, and 2) within a subject for different administrations. What term relationships seem to be relatively resistent to change, and which appear to be less stable: drastically changing position or connectivity. There are at least two possible causes for change: 1) systematic shifts, and 2) random shifts. The random shifts may be due to the subject's inability to discern any stable relationship among some set of terms, an inability to find the paired comparisons to be meaningful, or a basic instability in interpreting complex relationships of any kind, these or others.

The systematic shifts may also be due to multiple causes, like growing appreciation for certain relationships, reconsideration / reinterpretation of relationships, and redefinition of the semantic relationships for the terms themselves. Any one or all of these effects could be present.

Overall, the most common central term was decide / select. Patterns were extremely varied, even for a given subject. Two subjects (Cepec and Minor) produced patterns that appear as double, nearly concentric rings. Another common pattern was a cluster with one or more chains of terms running off in a vertical, horizontal, or diagonal direction from the central cluster. A number of subjects generated ratings that resulted in more complicated internal structures where diamond or ring patterns could be discerned internal to the diagram, rather than being an overall shape to the pattern. None of the patterns observed seemed to retain its character over different administrations, suggesting that the interpretation of what these patterns mean may be difficult to discern (since they do not appear to be stable conceptual structures).

2.3.3 Reliability and Validity Assessments

The reliability and validity ratings were also subjected to several kinds of analysis. SYSTAT was used to perform descriptive and inferential statistical analyses, and PATHFINDER analyses were conducted to discern the pattern of relationships in these data.

2.3.3.1 Reliability and Validity SYSTAT Results

The reliability data represent paired comparisons of the similarity among the 13 task analyses, a set of 78 paired comparisons. The means, standard deviations, and coefficients of variation for these are presented as upper triangular matrixes in Tables 2-4, 2-5, and 2-6. Recall that similarity scores could range from 0 to 9 in value, with a mid-range score being 4.5 on this scale. The largest and smallest scores for reliability assessments included the extremes of 0 and 9. However, the validity ratings only varied from a low score of 2 to a high score of 9. In general, the reliability comparisons also showed a larger variability (standard deviation (S. D.) = 2.085) than the validity ratings (S. D. = 1.907).

The average of the 624 similarity scores for the reliability data was 5.245 with a standard error of 0.083, resulting in a t-value of 63.193, which is clearly significant well beyond the 0.01 level. The null hypothesis may be rejected, leaving us to conclude the task analyses were reliable.

The average of the 104 similarity ratings for the validity data was 5.067, slightly smaller than for reliability, with a standard error of 0.187, resulting in a t-value of 27.096, which again is significant beyond the 0.01 level. The null hypothesis may be rejected, leaving us to conclude the task analyses were also valid.

Examining the descriptive statistics in greater detail, Table 2-4 shows variations in mean value range from a minimum average similarity score of 3.00 and a maximum average score of 7.25. Variations in standard deviations around these means (Table 2-5) range from a minimum of 0.77 to a maximum of 3.32. The lower confidence bound (-3 sigma) on the lowest value and the upper bound on the highest value (+ 3 sigma) will therefore span the range of scores, suggesting the scale is sensitive to variations in similarity among the different analyses. The coefficients of variation (table 2-6) ranged from a minimum of .116 and a maximum of .663. These variations suggest the degree of sensitivity in the scale values. Some subjects had little reluctance to use the extreme score values in evaluating the similarity (or differences) between terms in the set of 300 paired comparisons of the twenty-five terms.

The one-way analysis of variance results for differences among raters and for differences among pairs of comparisons are presented in Tables 2-7 and 2-8 respectively. The evaluators do not show statistically significant differences ($p>.05$), indicating that they provided a relatively homogeneous set of similarity scores. However, certain paired comparisons of the task analyses were significantly different ($p<0.01$), indicating some were substantially different from the others. No contrast tests were run due to the large number of paired comparisons involved (78).

The descriptive statistics for the validity ratings are presented in Table 2-9. These represent similarity ratings between student's task analyses and that done by CTA under contract with the FAA. The average ratings are all well above zero, with fairly small standard deviations, indicating substantial agreement among evaluators and reasonable comparability with the professional analyses. In particular, subjects numbered 2, 7, 9, 12 and 13 demonstrate rather strong similarities to the CTA analysis, as rated by the evaluators. In contrast, subjects 3, 4, 5, 6, 8, 10, and 11 were given relatively lower similarity ratings.

Table 2-4. Task Analysis Similarity Means.

Subj.	2	3	4	5	6	7	8	9	10	11	12	13
No. 1	5.00	4.38	4.25	5.13	4.75	6.88	6.50	4.75	6.13	4.13	6.25	3.50
No. 2	----	5.38	5.50	5.75	6.88	7.25	6.13	5.38	5.88	4.50	6.00	3.88
No. 3	----	----	6.00	6.25	4.25	5.00	4.25	4.25	5.75	4.38	5.88	5.38
No. 4	----	----	----	3.50	6.50	6.13	5.88	5.88	4.50	3.88	5.63	4.88
No. 5	----	----	----	----	5.50	6.75	5.50	5.50	4.13	4.63	5.25	4.63
No. 6	----	----	----	----	----	5.13	4.50	3.00	6.88	5.13	4.38	5.25
No. 7	----	----	----	----	----	----	5.13	5.13	6.75	5.75	6.13	5.88
No. 8	----	----	----	----	----	----	----	3.75	7.13	3.88	4.50	5.38
No. 9	----	----	----	----	----	----	----	----	6.88	4.00	4.38	6.25
No. 10	----	----	----	----	----	----	----	----	----	4.88	6.38	5.00
No. 11	----	----	----	----	----	----	----	----	----	----	4.38	4.00
No. 12	----	----	----	----	----	----	----	----	----	----	----	5.25

Table 2-10 presents the one-way ANOVA results for differences among evaluators. Again, we cannot reject the null hypothesis at the 0.05 level, indicating a relatively homogeneous set of similarity ratings for the eight evaluators who compared student's analyses to those done by CTA. Table 2-11 on the other hand looks at differences among the analysts themselves, and here we see a statistically significant difference (p= .005), indicating some analyses

Table 2-5. Task Analysis Similarity Standard Deviations

Subj.	2	3	4	5	6	7	8	9	10	11	12	13
No. 1	1.93	1.69	2.12	2.59	1.83	2.10	0.76	2.05	2.36	1.64	1.83	1.20
No. 2	----	1.41	1.31	1.58	1.36	1.58	1.36	1.51	2.17	1.20	1.60	1.89
No. 3	----	----	1.31	3.32	2.55	3.25	1.83	1.83	2.19	2.20	2.03	1.60
No. 4	----	----	----	2.00	1.41	1.55	1.55	1.89	2.00	1.46	1.60	2.17
No. 5	----	----	----	----	2.14	1.91	2.14	1.93	2.30	1.85	1.49	1.41
No. 6	----	----	----	----	----	2.95	1.69	1.85	1.55	1.46	2.00	1.98
No. 7	----	----	----	----	----	----	3.18	3.40	1.58	0.92	1.01	0.77
No. 8	----	----	----	----	----	----	----	1.83	1.13	2.10	2.27	1.30
No. 9	----	----	----	----	----	----	----	----	1.55	1.31	2.20	1.58
No. 10	----	----	----	----	----	----	----	----	----	2.23	1.85	1.41
No. 11	----	----	----	----	----	----	----	----	----	----	2.67	2.51
No. 12	----	----	----	----	----	----	----	----	----	----	----	2.32

were more like CTA's than others were. In contrasting the six highest ratings and with the six lowest ratings, we find the difference is statistically significant well beyond the 0.01 level.

2.3.3.2 Reliability and Validity PATHFINDER Results

The reliability layouts generated by PATHFINDER are presented in Appendix F. A separate diagram was generated for each of the eight task analysis evaluators (identified as subjects 1 through 8). Each evaluator rated each of

Table 2-6. Task Analysis Similarity Coefficients of Variation.

Subj.	2	3	4	5	6	7	8	9	10	11	12	13
No. 1	.385	.385	.499	.505	.386	.305	.116	.432	.385	.398	.293	.341
No. 2	----	.262	.238	.275	.197	.218	.221	.280	.369	.266	.267	.486
No. 3	----	----	.218	.370	.600	.650	.431	.431	.380	.503	.346	.297
No. 4	----	----	----	.571	.218	.253	.264	.321	.444	.376	.284	.445
No. 5	----	----	----	----	.389	.283	.389	.350	.556	.399	.283	.304
No. 6	----	----	----	----	----	.575	.376	.617	.226	.284	.456	.378
No. 7	----	----	----	----	----	----	.621	.663	.234	.453	.465	.369
No. 8	----	----	----	----	----	----	----	.489	.158	.542	.504	.242
No. 9	----	----	----	----	----	----	----	----	.226	.327	.503	.253
No. 10	----	----	----	----	----	----	----	----	----	.458	.290	.283
No. 11	----	----	----	----	----	----	----	----	----	----	.610	.627
No. 12	----	----	----	----	----	----	----	----	----	----	----	.441

Table 2-7. ANOVA Results for Evaluator Differences in Similarity Ratings.

Source	Sum of Squares	df	Mean Squares	F	p
Evaluators	60.268	7	8.610	2.003	0.053
Error	2647.218	616	3.875		

Table 2-8. ANOVA Results for Task Analysis Differences in Similarity Ratings.

Source	Sum of Squares	df	Mean Squares	F	p
Comparison Pairs	591.861	77	7.687	1.984	0.000
Error	2115.625	546	3.875		

Table 2-9. Descriptive Statistics for the Validity Ratings Across Evaluators.

Analyst No. or Task Analysis:	Average Similarity	Standard Deviation	Coefficient of Variation
1. Anderson	5.00	0.926	0.185
2. Cepec	5.75	2.053	0.357
3. Deatheridge	4.25	1.389	0.327
4. Eppley	4.88	1.642	0.337
5. Jergens	4.13	0.991	0.240
6. Lerdon	4.38	1.923	0.439
7. Lubinsky	5.88	2.949	0.502
8. Meyer	4.50	1.604	0.356
9. Schneider	6.00	1.604	0.267
10. Stedke	3.75	1.982	0.529
11. Torok	4.50	1.852	0.411
12. Vrabel	7.38	1.188	0.161
13. Wykoff	5.50	1.690	0.307

Table 2-10. ANOVA Results for Differences Among Reliability Evaluators.

Source	Sum of Squares	df	Mean Squares	F	p
Evaluators	60.268	7	8.610	2.003	0.053
Error	2647.218	616	4.297		

Table 2-11. ANOVA Results for Differences Among the Paired Comparisons.

Source	Sum of Squares	df	Mean Squares	F	p
Comparisons	591.861	77	7.687	1.984	0.000
Error	2115.625	546	3.875		

Table 2-12. ANOVA Results for Differences Among Validity Evaluators.

Source	Sum of Squares	df	Mean Squares	F	p
Evaluators	41.606	7	5.944	1.714	0.115
Error	332.923	96	3.468		

Table 2-13. ANOVA Results for Differences Among Task Analysts / Analyses.

Source	Sum of Squares	df	Mean Squares	F	p
Analysts	95.654	12	7.971	2.601	0.005
Error	278.875	91	3.065		
Contrast	72.020	1	72.020	23.5	0.000

the thirteen completed task analyses for similarity to the others (reliability) and similarity to the CTA task analysis (validity).

Again, several interesting patterns emerged. The simplest is the simple star, where the data may be interpreted relative to one of the task analyses as a central element. The best example of this is subject 3. Several variations of that pattern also exist, such as subjects 1, 4, 5,and 7. More circular patterns were also evident, such as subject 6. The more interconnected patterns (subjects 2 and 8) were also the ones that did not print well. Some cases show constellations around two or more analyses, such as subject 1, 4, 5, 6, and 7.

All of the validity layouts are the same. PATHFINDER provides nothing of particular interest in these analyses. However, that alone is encouraging, since one would not expect complex patterns from such a simple set of ratings. The output at least confirms the reasonableness of the PATHFINDER results, lending credibility to the more complicated patterns generated from the reliability data, and the even more complex relationships generated from the task analysis questionnaire data.

2.3.4 Anecdotal Data and Other Study Results

The anecdotal data were obtained from three sources: 1) the content of the task analyses themselves (notes, comments, suggestions, etc.), 2) the comparison reports (for those who did them), and 3) the Task Analysis questionnaires, which again, not every participant completed.

Thirteen subjects completed the task analysis in Winter quarter, but only one completed both the analysis and a comparison report in the Spring quarter. While the Spring quarter analysis was not included in the reliability and validity assessments, it is included in the following analyses.

2.3.4.1 Task Analysis Anecdotes

Anderson provided both outline and graphic forms of his task analysis. Anderson defined three terms: Determine (similar to Miller's "categorize"), Route (to specify flight parameters, implying control), and Ensure (similar to Miller's "test"), noting that "I do not think in Miller's Terms. ...it was easier to come up with something using some of the terms above." Anderson notes the following as hazards: 1) giving instructions to the wrong aircraft, 2) confusing left and right hand turns, 3) not following all aircraft to their destinations, assuring appropriate parameters for each, 4) directing an aircraft to an ILS approach before it is in range, and 5) forgetting approach speed and altitude constraints.

Brown (the student completing her analysis in the Spring quarter) prepared a graphic task analysis. She indicated she thought adapt / learn is important to each of the steps identified in the task analysis, especially items such as keystrokes, airport names and altitudes, and correct terminology for turning (eg. relative heading changes versus absolute headings). She also expressed the opinion that it is easier to control many aircraft if few of them are landing.

The problem of overlapping track blocks was noted. She thought TRACON was a "very stressful simulation."

Cepec developed a conceptual dependency chart first, noting that it was not followed closely in the task analysis but did serve as a starting point. His actual task analysis was done in outline form. Cepec states: "aircraft separation ... is the goal image." He also notes that, "Time is the centerpiece: 1) as time increases the overall knowledge of the game increases; 2) as time decreases different levels of the approach control problem increase." Please note that the concept of time appears to have been used two different ways in Cepec's observations: 1) total duration or experience (number of trials or sessions of practice) and 2) time available to get things done, conjectures confirmed by other comments he made later in his analysis documentation. He also noted that there were a "massive amount of commands to learn; if you really want to be proficient, memorizing them is a must... voice response can make this phase easier." But he earlier noted "... the audio portion was unintelligible and may as well not have existed."

Deatheridge documented her task analysis in a hierarchical, graphical form after first producing a conceptual dependency chart. Deatheridge said "I knew what the terms meant until I read the definitions." She defined four terms: 1) receive -- to acquire something, 2) monitor -- to watch over something, 3) review -- to look over again, and 4) locate -- to find something.

Eppley provided a conceptual dependency chart and both graphical and outline forms of his analysis. No terms were defined, nor were any comments made.

Jergens provided a conceptual dependency chart and a graphic documentation of his analysis. He observed that some things are done for all aircraft (eg. accessing their flight strip data and determining their state vector), while some actions are specific only to particular aircraft (eg. altering clearances, issuing directives). His conceptual dependency chart did not draw heavily on Miller's Terms, but he did not define his own: adjust, handoff, change, look at, accept/release, observe turn, hold, and "don't let..." In his task analysis, handoff is the only non-Miller term used.

Lerdon documented his task analysis in graphic, flow-chart form with little adherence to Miller's Terminology, but provided no verbal commentary.

Lubinsky provides three logic flow charts: 1) smooth operation, 2) very high priority situation (VHPS), and 3) emergency break (EB). Little attention was given to Miller's Terms, however, the approach taken revolves around defining a task box as a mental image of how many subtasks need to be accomplished to achieve a goal. The flow logic addresses operations on the contents of the task box. This concept is sufficiently compelling that it is presented in only a slightly abridged and amended form as Table 2-14.

Table 2-14. Lubinsky's Task Box Concept.

TASK BOX: a mental image that relates the size of a box associated with an aircraft with the number of subtasks involved in reaching a goal, determining priority when not in VHPS or EB modes. For example:

Task Box

N4353D

needs to steer clear of N6272C

needs to get near the airport

needs to be brought to approach altitude

needs to get near the ILS

needs to be on approach heading

needs to be handed off to tower

In smooth operations, the subtask list may not be followed exactly in the order subtasks are listed; some (re)-organization of the list may occur.

VHPS --- a situation where attention is given to an aircraft for reasons other than the size of the task box or the existence of an emergency, like reaching the edge of a sector where handoff is imminent even though the task box is small.

EB --- the task box shrinks because "Can I?" and "Should I?" steps have been eliminated and there is a clear need for immediate action.

NOTE: The biggest problem I had is that I couldn't incorporate planning into my flow charts. If taken by surprise, the EB task is always the highest priority.

Meyer provides a conceptual dependency chart and both an outlined task analysis and a graphic form. No verbal comments are provided. Miller's terms are not evident, and no definitions were provided for the terms used, like: determine, check, clear, change, and hand off. In some places, actions were implied, not stated : "3.2 Pilot requests; 3.2.1 Do best to answer them."

Schneider provided a two page graphic with no use of Miller's terms, nor any definitions for his own terms: divert, allow, determine, instruct, pull-up, vector, monitor, line-up, and descend (ie. direct aircraft to descend).

Stedke provides detailed logic flow charts, referenced to tabular appendices. All flow chart symbols are defined, but terms are not: highlighting, pending, strike, input (enter), communicate, and recognize.

Torok did a conceptual dependency chart first and then provides both outline and graphic task analyses. No comments are provided and terms are left undefined: prioritize, release, admit, and hand-off. A number of actions are implied: emergencies, pilot requests, and safe handling of aircraft.

Vrabel provides only a brief outline, does not use Miller's terms extensively, and leaves his own terms undefined: allow, check, look at, make plan, use plan, vector, reduce/increase (speed), and hand-off.

Wykoff provides both outline and graphic task analyses, but does not define terms: type-in, reduce, descend, turn, make sure, check, clear, and hold. Wykoff also suggests that priorities are as follows: 1) emergency situation, 2) lowest altitude, 3) fastest speed, 4) aircraft weight, 5) closest to destination, 6) length of time waiting (holding), and 7) missed approach. Certain knowledge "was necessary for smooth flow of traffic;" namely: 1) keyboard control commands, 2) quick reference menus (possible control functions), 3) meaning of all elements of the fight strip, 4) sector boundaries, and 5) name and location of all airports in the sector. Three additional items were deemed "helpful to know beforehand:" 1) approach information for each airport, 2) approach speeds of aircraft, and 3) the name of at least one fix point near each airport.

Table 2-15 summarizes the products provided. Subjects were allowed to choose which forms they wanted to use. Only one attempted to use the tabular format and quickly abandoned it. Four subjects provided all three products: 1) conceptual dependency chart (concept map), 2) graphic , and 3) outline. Four subjects provided only the graphic. Two provided both the concept map and the graphic, and another two combined the concept map with only the outline. One subject provided only the outline.

Subjects were also asked to keep track of how much time they spent doing their task analysis and documenting it. Only nine of the thirteen analysts reported the duration of their activity, as indicated in Table 2-15. The longest (654 minutes) and shortest times (165 minutes) were both reported by individuals producing all three products. The average time was 368 minutes (approximately 6 hours), and the median was 360, indicating a positively skewed distribution.

Table 2-15. Summary of Task Analysis Products Provided.

Subject Name:	Concept Map	Graphic	Outline	Time
1. Anderson	---	X	X	253 min.
2. Cepec	X	X	X	645 min.
3. Deatheridge	X	X	---	360 min.
4. Eppley	X	X	X	
5. Jergens	X	X	---	405 min.
6. Lerdon	---	X	---	330 min.
7. Lubinsky	---	X	---	380 min.
8. Meyer	X	X	X	
9. Schneider	---	X	---	
10. Stedke	---	X	---	570 min.
11. Torok	X	X	X	165 min.
12. Vrabel	---	---	X	
13. Wykoff	---	X	X	210 min.
14. Brown	---	X	X	

The standard deviation was 158 minutes, and the standard error of the mean was 52.6 minutes. In theory, nearly 99% of the analysts should be able to finish within approximately 8 1/2 hours (510 minutes), but here two of the nine analysts (22%) reporting took longer than that. Many of the participants expressed surprise at seeing how detailed the CTA analysis was compared to their own. Yet there were numerous complaints about how long it took to do even the less detailed analysis.

2.3.4.2 Comparative Report Anecdotes

In comparing their own work to the CTA task analysis, all participants were given an abridged extract from the CTA documentation. While both they and the evaluators were encouraged to go look at the complete set of CTA documentation, library records indicate that no one did. The instructions for their comparison task are presented in Appendix C. Their report was to include seven sections: 1) assumptions, 2) partitioning, 3) labeling, 4) ordering, 5) branching, 6) easiest and most difficult aspects of the task analysis, and 7) other comments. Only twelve of the thirteen analysts submitted a comparative report. However, there was an additional analysis completed that did not get included in the reliability and validity assessments. That comparison report was included in these anecdotes.

As one student said, "I definitely found it easier to compare the terms I used with the terms they (CTA) used than with Miller's ..." Another commented, "I used ... the terms in the course handout. I found that they limited my clarification of certain situations." Using a standard set of terms apparently seems unnatural, complicated, and confusing to many people. The analysts studied here seem insensitive to the lack of scientific comparability induced by using non-standard terminology. In part, this may be due to inexperience in terms used in behavioral science. Few American (Aviation) students are fluent in more than English. They have relatively little experience translating terms and concepts.

Five analysts suggested possible equivalences between operational terms and the Miller terms, as represented in Table 2-16, which also suggests some of the ambiguous relationships which can arise in the process. Sometimes the same operational term has been assigned (by different people) to more than one of Miller's terms. However, the encouraging discovery is that at least two people voluntarily suggested the same substitution of terms. All of the contents of Table 2-16 are unsolicited comments from the participants.

When discussing the level of partitioning, it was surprising to see how many ways the participants interpreted the CTA analysis. There was little consistency or agreement in the number of levels (from 3 to 8) in the CTA analysis, but most agreed that the top level had six tasks. Some of the confusion may be due to alternate interpretations of what the term "levels" meant. In some cases, the six terms used at the top level were seen as identifying the number of levels in the subsequent analysis. Others noted the numbered indenture scheme and took that as their cue for numbering "levels." And in some cases, it simply is not clear what the commenter used as the indication of "levels."

Each analyst seems to have gleaned different information from the CTA material as they did their comparisons. However, a number of participants did notice that there were elements of the CTA analysis that incorporated considerations they would not have reasonably thought to include, like: 1) supervision, 2) resource management, and 3) adverse weather, since these factors were not a part of their TRACON II experience.

Table 2-16. Possible Similarities Between Operational Terms and Miller's.

Miller's Term	CTA/FAA Term
edit	review
filter / detect	review
filtering	analyzing
filter	hand-off
transmit	hand-off
	forward
	enter
	initiate
queue to channel	initiate
	track history (sic)
transmitting	issuing
detect	perceive (observed twice)
	observe
store	record
purge	suppress / restore
control	route
interpret	determine
decide / select	determine (observed twice)
plan	formulate
categorize	resequence

Analysts that did not use logic flow diagrams used a very linear sequence of tasks. The use of parallel activity sequences was not used extensively, and CTA's use of them was seen as being there to handle situations which did not arise in TRACON II. However, the possibility of performing tasks in parallel apparently did not occur to these analysts, even though they were prompted to consider that possibility. That by itself seems curious. Once the perspective is adopted that everything flows in a particular sequence, it seems difficult to introduce the possibility that some other ordering might be possible. In fact, there seems to be a tendency to rationalize why a single, linear flow is the proper representation!

Every analyst discovered that decisions they were able to recognize were included in the CTA analysis, but CTA's analysis also included decisions they had not incorporated, in some cases because of the level of detail, but in others because of the scope of coverage (eg. weather and emergencies). Several (three) analysts also thought some aspects of the CTA analysis were inappropriate, principally because they were ignorant of what the terms actually meant, like "housekeeping." In one case, the analyst indicated what his assumptions were for terms CTA used that were not familiar, like track history (incorrectly interpreted to mean a regular traffic pattern instead of a sequence of radar hits) and flow restriction (correctly interpreted).

More than one student indicated they saw nothing "easy" about doing task analysis. Those who decided anything was easy about the assignment were the one's who said learning and exercising TRACON II was the easiest part. That certainly is supported by the fact that many prospective participants chose to stop right there and go no further, not even attempting the task analysis.

Three analysts complained that finding the right Miller term was their most difficult problem. The second most often cited difficulty was in getting started, understanding what was really expected in the task description, or determining what was important to include. Two analysts stated the most difficult part of the assignment was the comparison of the two task analyses: one's own with CTA's, principally due to the disparate levels of detail.

Under "Other Recommendations," two analysts suggested looking at the CTA analysis prior to doing their own would have helped. That approach was actually considered, but it was rejected because it would also serve to bias even more their choice of content, detail, and format, and part of our objective was to see how many people chose a particular form of task documentation. While no one looked at the complete CTA documentation, every one was surprised by the detail of the abridged section they were given, which was less than 20% of the total document. A related suggestion was to do more task analysis as an integral part of the class (AV 540).

One subject commented that filling out the questionnaires seemed mindless and time consuming. Others probably thought the same, without saying so. Another observed that this experience taught him that one must first "define" the concepts involved in getting a task done, then one had to break them down "onto a more micro-level." He stated that the problem he encountered in doing the second step was that he tended to combine main and subtasks on one level without considering sublevels.

2.3.4.3 Task Analysis Questionnaire Anecdotes

A short questionnaire with twenty eight items scaled from strongly disagree to strongly agree was given at the end of the study. Additional comments to the questions asked were invited, but no one commented. Only four participants filled out and returned this questionnaire.

The questions were related to the instructions for TRACON and the task analysis, adequacy of practice sessions, terminology issues, documentation preferences, and miscellaneous related concerns. The two questions that received the greatest degree of agreement (range of 1 scale point) were: 21) some set of standardized terms is necessary for comparisons of task analyses (average score of 7.5 out of 9.0) and 22) The terms used in a task analysis at one level of detail are different that the terms I need to use at other levels of detail (average of 5.5 of 9.0, mildly agree).

The questions evidencing the greatest disagreement were: 16) I think the graphic method of task analysis is best (range from 2 to 8, average 5.25), and 18) I think the indentured outline is the easiest (range from 2 to 9, average 5.0). Clearly, preferences in this regard are radically different, which may indicate some people are more visually oriented while others are more verbally oriented in their analysis documentation preferences. Forcing a single approach, of either kind, is going to leave others unhappy with the product.

Questions separated by a two point spread included: 2) I could have learned TRACON alone, without tutoring (avg. 7.75), 6) the task analysis instructions did not answer all of my questions (avg. 6.0), 14) based on CTA's analysis, TRACON is realistic (avg. 6.0), 17) doing a concept map first is a big help (avg. 8.0, but for 3 of 4 participants, since the fourth did not do one), 23) CTA's descriptions are too brief to be well understood (avg. 4.0, mildly disagree), and 28) I think I could handle the real thing (avg. 5.0, mildly agree).

Questions showing a three point spread include: 1) TRACON II instructions are useful by themselves (avg. 6.0; they do describe the operational environment), 4) more practice was not necessary (avg. 6.5, moderate agreement), 15) now that I've done one, I could do other task analyses without much difficulty (avg. 5.0, mildly agree), 20) I see no reason to do a task analysis more than one way (avg. 3.0, moderately disagree), and 26) I never felt comfortable I knew what I was doing with TRACON (avg. 2.25, strongly disagree).

Questions with a four point spread included: 3) TRACON tutoring was quicker than trial and error learning (avg. 7.75; NOTE: this seems inconsistent with the strong agreement that they could learn it on their own, Q #2; but they may be saying they recognize the efficiency of having help, yet its not too hard to learn alone), 7) the task analysis examples were useful (avg. 6.0, mildly agree), 8) Miller's terms were difficult to understand (avg. 4.0, mildly disagree), 10) using my own terms was easier (avg. 4.5, indifferent), 12) it was difficult to understand CTA's analysis (avg. 5.75, agree), 24) doing it over, I'd include more details (avg. 4.75, neutral), and 27) I was not comfortable my task analysis was right (avg. 5.25, mildly agree).

Questions with a five point spread included: 5) the analysis instructions were clear (avg. 2.75, moderately strong disagreement), 9) Miller's terms were difficult to apply to TRACON II (avg. 5.5, moderately agree), 11) CTA's terminology was easy to understand (avg. 5.5 also), 13) it was difficult to compare CTA's analysis to my own (avg. 5.5 again), 19) I found the tabular method of task analysis confusing (avg. 5.25, but none of the respondents indicated they had tried using it, and the one subject who did try quickly abandoned it), and 25) you ought to see a real analysis first, before doing your own (avg. 5.5, moderately agree).

2.4 Discussion

Companion and Corso (1982) indicate that Miller (1967) anticipated the problems graphically indicated by the PATHFINDER data: a mere list of labels or classes is not a sufficient taxonomic structure. A taxonomy must have 'an inner structure' (Miller, 1967, p. 67) that permits useful relations to be established. Based on what we can see in our results, that statement seems a bit weak. The need is for stable, consistent relations among the categories or terms being used. That requirement appears to be even more difficult to achieve than Miller might have anticipated.

It is typical to see new or beginning analysts try to put in too many details too soon in their analysis without considering ways in which the analysis can be segmented or re-organized to reveal complex issues in a progressive fashion. Moreover, in a complex analysis product, reviewers who have not yet realized the need for structured decomposition often struggle with the large volume of material and fail to understand or appreciate the reason for its hierarchical packaging.

Although several of the participating student analysts believe Miller's terms could be used successfully, the majority thought that the terms used should relate to the operational context or to common and familiar words. One student commented, "I do not think Miller's terms should be replaced by CTA's." Another student indicated he "had a hard time trying to integrate three terms into the conceptual dependency chart ... short term memory, code, and queue to channel ... other terms seemed straight forward to me." Goal image has a very specific connotation in the cognitive psychology of the sixties era in which Miller originally wrote, but students seemed to interpret the operational significance of this technical term with little difficulty. It is not always clear which terms will present interpretation problems for a particular individual, and there is little assurance analysts will necessarily appreciate the full significance of a term popular in a discipline but uncommon in everyday usage.

People seem to have difficulty seeing the relationship between their choice of terms (eg. everyday speech and common word usage) and others' choice of alternative descriptive terminology, especially when the words used are not familiar or are used in an unusual sense. The first shift they appear to be comfortable making is to accept the terms of their trade or specialty area (eg. operational jargon). They seem to find it difficult accepting and using more abstract or arcane terminology, even if it is alleged to be more specific or generalizable for a particular purpose (eg. Miller's, scientific jargon).

At least one student expressed the notion that Miller's Terminology did establish a base on which to build. He simply found some words better described what he wanted to say, recognizing that "A standard vocabulary will make a task analysis more understandable for those who agree with the standard set of labels." Others also agreed a standard set of terms had merit. The problem is then dealing with those who choose not to agree with the use of the standard set of terms, and certainly this study found they were in the majority, and included even some of the individuals who saw the need for standard terms. They just did not like Miller's, but there is no assurance they would feel comfortable with any other set, certainly not everyone even felt comfortable with CTA's operational terms.

People will often miss the opportunity to classify as equivalent terms which share a common conceptual referent. On the other hand, some can find such broad interpretations of terminology that they may equate, as similar, certain terms which can be interpreted other ways and actually describe two distinctly different concepts or objects. Then two concepts are seen as an equivalent when they are actually quite dissimilar; category boundaries are widened to the point of increasing false positive identifications of instances. Definitions by themselves will not solve this problem, as our data indicate. Many different relationships can still exist between a set of defined terms, as evidenced by the diagrams in Appendix E.

It was unfortunate that the last questionnaire was filled out by so few participants, but by this point in the study, many seemed to feel they had already done more than they wanted to for "the cause of science." Even with the small sample size, many of the anecdotal comments were quantitatively confirmed, indicating the questionnaire anticipated many of the participant's self-generated comments about the study. Also, while tutoring seemed welcome, it was not deemed essential: automated administration of future studies appears feasible.

While the task analysis instructions need improvement, they were better than what appears in previous published sources, and the difficulty of doing these analyses for the first time may have influenced the assessment. However, it is painfully clear that teaching this kind of analysis is neither simple nor easy, and students find it very difficult to learn. Whether any substantial improvement can be made with ITAM remains to be seen. Certainly, attention needs to be given to this issue in particular.

2.5 Conclusions

The conclusions expressed in the following statements are a combination of statistically supported assertions and empirically founded conjectures.

2.5.1 Statistically Supported Assertions

The following six assertions are based on the descriptive statistics and inferential statistical tests presented in section 2.3.

1. The similarity scale used to evaluate the reliability and validity of the task analyses is sensitive to differences among task analyses, since the confidence limits on the maximum and minimum scores in both cases (reliability and validity assessments) encompass the entire range of the scale.

2. The evaluators of reliability and validity were a homogenous group: their assessments of similarity both among the analysts's products and between those products and CTA's work were not significantly different.

3. There are significant differences among the set of task analyses produced by subjects asked to use a standardized list of terms: the similarity scores show significant between subjects differences.

4. There are significant differences in the validity of task analyses among analysts, a subset of subjects' task analysis products being significantly more or less similar to the CTA analysis.

5. There were no significant differences in performance scores between those who chose to do the task analysis assignment and those who did not. However, the participants were uniformly good students, all but one having earned A's for the course (Human Factors in Aviation) even before assignment of extra credit for completing this "Project C" assignment.

6. The statistical analyses support the graphical evidence provided by PATHFINDER output diagrams: there are significant differences in the way subjects judge the similarity among Miller's proposed task analysis terminology. Their interpretations not only differ from subject to subject, they differ significantly on repeated measurement within the same subject.

2.5.2 Empirically Founded Conjectures

The following conjectures appear to be well-founded from the empirical evidence: PATHFINDER layouts and anecdotal comments made by the participants.

1. No one found task analysis "easy," and all of the analysts were well above average achievers at the collegiate level. Not everyone is willing to

undertake such analysis, and of those who will, some are better than others, at least initially.

2. Everyone seemed to be surprised at the level of detail in a professional, complete task analysis: beginners may tend to be superficial in their preliminary attempts to learn what task analysis is and how to do it.

3. The complexities of human activity are typically under estimated by beginning analysts: many opportunities for judgment and decision making are not recognized at first.

4. Beginning analysts tend to force their description into a linear sequence of activities (a single thread of control), perhaps because they see their own behavior as having been a sequence of activities, or because this is a convenient fiction that simplifies and organizes the description. This linear sequence in the activity description suppresses some of the uncertainty that existed both in learning what to do and in identifying alternate ways to do the same things but differently (achieve similar goals by alternate procedures, methods, strategies, or operating techniques).

5. Not everyone sees or understands the value of using a standard set of terms for task analysis (even when instructed to do so), and even those who do try seem to find it difficult to stick only to the terms given, although a small set of analysts can and do stick to the specified set of terminology.

6. Similarity ratings of a set of terms is more difficult and laborious than one might imagine. It is not clear that producing ratings for 300 paired comparisons of 25 terms is a totally meaningful exercise for naive subjects for at least two reasons: a) they are not accustomed to doing paired comparisons, and b) the terms themselves are not "comfortable."

7. Definitions for terminology are necessary but not sufficient to assure consistency in activity labeling. The relationships among terms, as evidenced in the PATHFINDER diagrams, illustrates how many different ways terms may be interpreted when they are used in combination with other terms. Relationships seen by one subject are not seen by other subjects. An individual's conceptual perspective influences the choice of terms. So even when the terms are standard, their selection in a given instance depends upon how one views that term in relation to other terms from which one must (or can) choose.

8. Since making forced choices among a set of terms that are not familiar and well understood is difficult, many subjects instead chose to select "none of the above" and invented their own terms or borrowed from a larger collection of relatively more familiar or meaningful labels.

9. Having compared their analysis with CTA's, nearly all subjects commented that their own analysis was less detailed. Some thought their analysis was simpler, and therefore better, especially for understanding the general character of the TRACON task. Others recognized that CTA had solved some

description problems they had struggled with but had not solved: representing judgments, alternatives, etc.

10. Not everyone arrives at the same conclusions comparing their own work to some standard. Some defend their work by finding fault with the other product, others are searching for solutions and see particular, specific, and perhaps unique merits in the other product, and some seem indifferent to the observed differences.

11. A standardized set of terminology like Miller's is a set of jargon invented by psychologists, for psychologists, and is not likely to be well received by operationally oriented analysts unless a considerable effort is expended in training them in the appropriate use and application of the terms.

12. If operationally meaningful terms are used, they will be context specific. They also appear to be: a) more numerous, b) less general, and c) not always understandable to others outside the particular profession or operational community. To generalize from that set to the more general, abstract, and scientific set of terms, it may be useful to seek similarity ratings between the operational and the standardized terminology, both from the operational perspective and from the scientific perspective, since each group must try to deal with the other's terminology.

2.5.3 Some Conclusion Based Recommendations

Because of the labor required to make paired comparisons, a more efficient way of getting the needed data must be found. The approach taken for getting the validity ratings may provide a clue for how this problem could be solved: make only those comparisons which are needed, not all that can be made. For example, pilot studies can show which terms might or might not be reasonable candidates for selection when searching for equivalence in one set compared to another (as suggested by Table 2-16). The question then is how similar are those terms which are reasonable choices. Unreasonable choices only add to the time to make comparisons and tend to confuse the issue with operationally oriented people who have little patience with the dictates of science anyway.

The application of any standardized set of terminology seems to require more than simply making the definitions available. The training required to achieve consistent application might best be accomplished by devising a spectrum of simple task analysis examples. To illustrate the generality of the proposed standard set of terms, the task analyses might first be done using operational terms and then redone using the standard terms (Miller's or others).

The first question is choosing the set of terms to use. Only one of many options was examined here. The second question is whether the trainers could agree on the choice of example applications: establish context validity. The third question is whether the trainers could agree on the operational lexicon for each example. The fourth question is whether consistent mappings can be obtained between the operational lexicons and the standard set of terms.

The final question is whether a group trained in this fashion does "better" task analysis than an equally experienced group freely choosing their own terms and methods. Part of the problem is in defining what is meant by a "better" task analysis: a) supports a single need better, b) better supports multiple needs, c) is faster, more easily understood by others, or d) some other criterion.

The present study suggests how to compare task analyses against one another and one that exists already (as the criterion). The larger problem not addressed is the issue of criterion specification when one task analysis is prepared, and it is the first of its kind. What are the criteria for assessing validity then? If it is acceptance by operational personnel, then an operational lexicon makes the analysis get better scores, whether its content is meaningful or not. A standard set of terms could be expected to do poorly on that criterion! This study certainly suggests that would be the case anyway.

One suggestion provided in subject comments was to determine whether someone else could do the task following the analysis as a performance aid. Certainly better job-oriented guides could be prepared once a valid task analysis is available, but this operational test approach may at least be one way to check for accuracy once equipment is available (static or dynamic mockups, if not the actual system). One could at least count the number of times problems were encountered.

SECTION 3.0

INTEGRATED TASK ANALYSIS METHODOLOGY (ITAM)

Materials on cognitive task analysis and related taxonomic material were obtained and reviewed. The precise nature of the cognitive task analysis methodologies is not well-defined yet. The ITAM development effort was therefore more protracted than anticipated. A more detailed and complete specification of ITAM will still be necessary prior to reliability and validity assessment testing. This section identifies some of the background material reviewed, a discussion of some perceived problems identified in the review, and then an outline of the proposed ITAM methodology itself, some treatment of the process of applying that methodology, and suggestions for future study.

Our purpose is to devise a form of task analysis which will be useful in identifying what should (and by implication should not) be included in a synthetic task. Two stages are envisioned for doing that: 1) an analysis phase that documents what has to be incorporated for the synthetic task to be like the real one (in psychologically significant ways, not necessarily as a physical replica), and 2) a way to confirm that the behaviors produced in the synthetic task are comparable to behavior in the real system. To the extent stage 2 falters (much less fails altogether), a revision of stage 1 is clearly indicated.

Such a methodology would serve not only to enhance performance risk assessments for existing systems, but it might also serve as a useful approach to training device design: both requirements specification and empirical validation. Since these are not new problems, their solution is not going to "magically" appear as the consequence of a single re-examination of available literature.

It also seems appropriate to propose a methodology that is evolutionary, not revolutionary, for the following reasons: 1) radically new approaches (any fundamentally unique behavioral technology) would be difficult to transition, 2) while science seeks standardization, behavioral science suffers from the "not invented here" syndrome: people need to be convinced before they "buy-in," and 3) an interdisciplinary arena (of psychologists and engineers) has inherently different conceptual perspectives that challenge each other, much less something new that neither yet understands. Therefore, a bridge to the past as well as to the future is needed. In the process, a linkage must be made between the engineering and behavioral science perspectives on systems and people.

3.1 Approaches to Task Analysis

DeGreene (1970), Kidd and Van Cott (1972), and Drury et al. (1987) all provide overviews of task analysis. In their review of task taxonomies, Companion and Corso (1982) as well as Fleishman and Quaintance (1984) identified four classic approaches to classifying tasks: a) behavior description, b) requirements specification, c) abilities assessment, and d) identification of task characteristics. Based on subsequent research, Fleishman and Quaintance (1984) discussed five approaches for future consideration, as well as several other pertinent taxonomic concerns. The five approaches were: a) criterion measures, b) information-theoretic, c) task strategies, d) ability requirements, and e) task characteristics. The information-theoretic approach was never extensively developed, since it was based on an empirical approach that was not pursued beyond the proposal stage.

Kirwan and Ainsworth (1992) provide the most recent material on this subject, describing twenty-five techniques for task analysis and presenting a number of case studies. Rasmussen, Pejtersen, and Schmidt (1990) provide another recent examination of a strongly related topic, asserting traditional task analysis describing a sequence of activities "is no longer an adequate approach when responses to task requirements is discretionary, involves flexible cognitive processes, and depends on subjective preferences." Earlier material addressing cognitive task analysis includes: a) Redding (1989), who reviewed the state of the art in this area and b) Roth and Woods (1989), who discuss cognitive task analysis in the context of intelligent system design.

Psychologists should also recognize the long-standing interest industrial engineers have in the related arenas: job and task analysis (McCormick, 1982), methods engineering (Geisel, 1982), work standards (Panico, 1982), and time study (Neibel, 1982), as well as predetermined motion time systems (Brisley and Eady, 1982) and standard data systems (Cywar, 1982) for industrial manufacturing and assembly operations. Several other chapters in the Handbook of Industrial Engineering (Salvendi, 1987 and 1992) are devoted to related topics.

Maes (1989) examines related problems in artificial intelligence, examining activation / inhibition algorithms as a means of controlling autonomous agents. Indeed, the entire arena of knowledge acquisition for building expert systems strongly overlaps the systems analysis concerns that are (or should be) an integral part of any adequate task analysis. It can be argued that knowledge engineering is simply good systems analysis.

3.2 Principal Problems with Current Practices

ITAM is motivated by three concerns. First, there is a gap between psychology and engineering that prevents the behaviorist from communicating to the system designer, especially in terms of the analysis tools and techniques designers know and use. Also, both have a need to interface with the operational community. Second, the need to expand task analysis into the cognitive domain requires a reexamination of basic descriptive methodologies to support what is emerging as a cognitive engineering discipline (Woods and Roth, 1988; Rasmussen, 1986). Third, there is a need to be able to relate new methods to the older recognized approaches, to show what has been altered, what has been retained, and why both actions (alteration and retention) are rational or warranted.

Companion and Corso (1982) presented 11 criteria that a task taxonomy ought to meet. These same criteria apply to task analysis as well, and are presented here as Table 3-1. Several of those criteria seem a bit naive, based upon the results of the present study. The second and third requirements seem to

Table 3-1. Companion and Corso's Criteria for Task Taxonomies.

1. Must simplify description of system tasks: make them manageable.
2. Should be generalizable.
3. Employ terms compatible with users' terms.
4. Be complete and internally consistent, dealing with all aspects of human performance in the system without logical error.
5. Compatible with the theory or system to which applied.
6. Provide a basis for performance prediction.
7. Have some practical utility, either applied or theoretical.
8. The taxonomy must be cost-effective.
9. Provide a framework around which empirical data can be integrated.
10. Should account for the interaction of task properties and operator performance.
11. Should be applicable at all system levels.

naturally conflict. A restatement of those criteria seems in order: assure that the terms (or categories) used in an operational context are relatable to the terms (categories) in the generalizable or scientific context. Cost-effectiveness is also a rather vague criterion without any further elaboration. The intent is to have an affordable methodology.

There are three principal problems with present methods: 1) the representations inherently suggest temporal sequence and linearity of control flow that overspecify what can occur, 2) option specification and evaluation do not typically appear as an explicit part of the analysis (the results of option search and assessment are assumed known and unique), and 3) definition and validation of plausible models for non-observable (i.e., cognitive) processes are not well-defined.

While protocol analysis (e.g., Ericsson and Simon, 1984) and discourse analysis (van Dijk, 1985) form a basis for determining plausible bases for non-observable behavior, Nisbett and Wilson's (1977) article is sometimes cited as a cautionary counterargument to reliance on subjects' reports. They cite factors they believe will predict when subjects can (as well as cannot) provide accurate reports of mental processes. Table 3-2 summarizes pertinent recommendations. The Judgment Mechanics identified in Table 3-2 are explained as follows: 1) selections tend to favor options recently evaluated (serial order effects), 2) items on the right are more often chosen than those on the left (position effects), 3) judgments made on bipolar stereotyping dimensions, based on presented information, are more favorable when we anticipate having to face the person evaluated (contrast effects), and 4) anchors can include prior knowledge, instructions, established beliefs, and other such biases (anchoring effects).

A strongly related factor rarely (if ever) mentioned in task analysis circles is the cultural backdrop within which all activity is taking place (Sathe, 1985). Certainly many social interactions will be influenced by the particular culture: assumed beliefs and values of those operating within the culture. Those unstated but commonly held beliefs and values shape expectations and facilitate communication. As Nisbett and Wilson (1977) indicated, cultural factors are what often bias interpretations of situations and data input, offering plausible theories in lieu of valid introspection. The extent and strength of cultural influences are reflected in what percentage of the people in that culture accept particular beliefs and values, and to what degree.

Table 3-2. Factors Affecting Subjective Reports.

Reports Are Suspect When:	Reports May Be Valid When:
1. Influential causes are not representative nor readily available to the subject.	1. Influential stimuli are: a. available b. plausible causes of the response c. few plausible but non-influential stimuli are there.
2. The report is removed from the time the mental process occurred.	2. Allowable response possibilities are extremely constrained.
3. Judgment mechanics biases: a. serial order effects b. position effects c. contrast effects d. anchoring effects.	3. Stimulus situation is fixed and static.
4. Nonevents are used as a basis: "X" did not happen, therefore the implication is: ___.	4. Highly plausible connection between critical stimulus (reinforcement) and increased frequency of response.
5. Nonverbal behavior governs (cues are less accessible).	5. Subculture clearly specifies what responses should occur.
6. Cause and effect are of disproportionate magnitudes.	

Finally, the present government interest in Computer-Aided Acquisition and Logistics (CALS) suggests that the conduit from engineering technical information to behaviorally useful systems technical information should be sensitive to opportunities to exploit the ICAM methodologies being adopted for CALS: 1) IDEF0 for functions and activity modeling, and 2) IDEF1 for data modeling. Certainly, a cost-effective solution would not ignore opportunities to exploit adaptations of those methods or possible utilization of the products generated using those methods. The coupling of IDEF0 and SAINT has been demonstrated (Chubb and Hoyland, 1989). SAINT can be a replacement for SLAM, used during IDEF2 development. The addition and demonstration of IDEF1 integration remains to be done, perhaps as ITAM progresses.

3.3 Some Conceptual Preliminaries

Tasks can be described with little reference to the context within which they occur, but that is seldom useful. More often, tasks occur in support of some system, which itself has a purpose and an operating environment. To understand what people are doing and why they do it, one must first understand what the system is within which the people are asked to perform and the environment that surrounds the system and its operators.

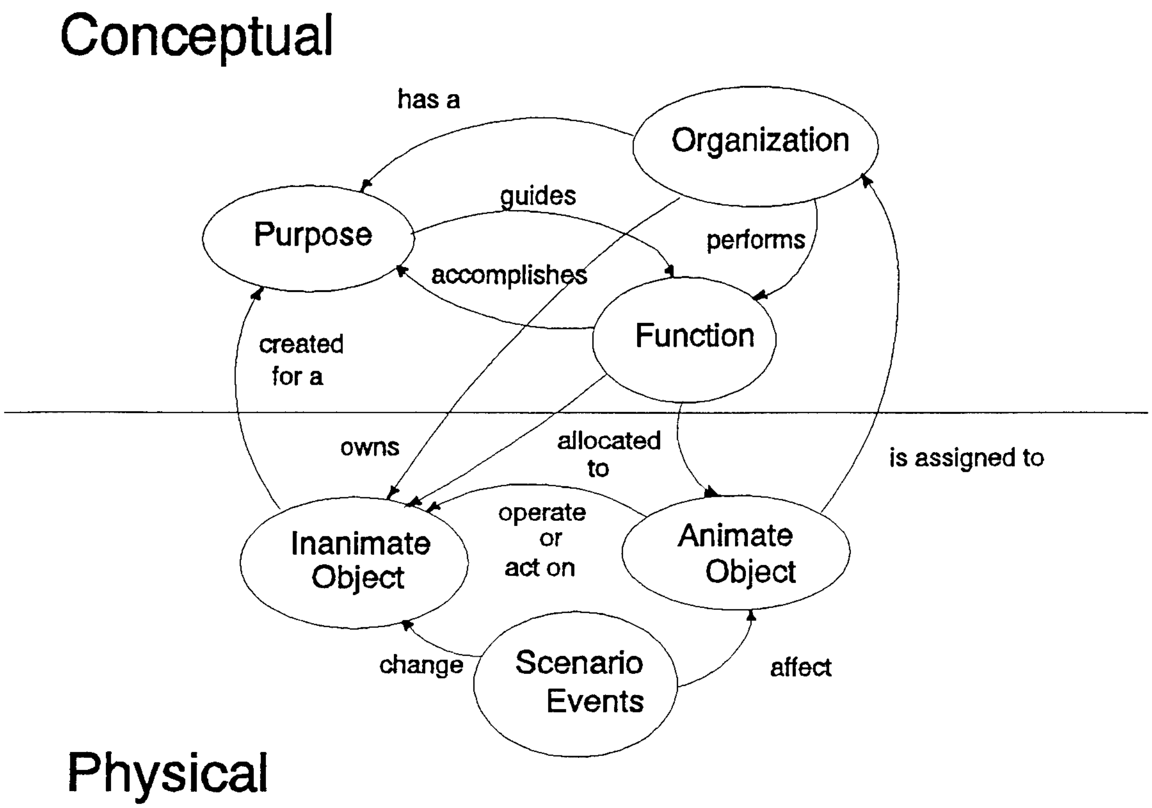
Moreover, it may not be enough to take a single system perspective. To properly understand the activities their context, one must also understand something about the systems with which one's own system interacts. It may or may not be necessary to understand something about the people in those other systems as well: what are they doing and why? The consequences of one system's actions affect what other systems may do next, as they react to the operating environment around them and pursue their own goals.

Three elements are needed: 1) a meta-system framework, 2) a system representation framework, and 3) an activity framework. ITAM will address this last issue, but not by ignoring the others. They must be implicitly incorporated in ITAM, if not by direct reference, then by assumption. People are not operating independent of the system or its environment, and to pretend they do is to miss some very important drivers of human behavior.

3.3.1 The Meta-System Framework

Most of systems theory deals with the design and/or analysis of a single system. Less is written about multiple systems. Figure 3-1 is offered as a meta-system framework for dealing with multiple systems as sets of complementary physical and logical (or conceptual) systems. For the purpose of our analyses, we separate objects in the physical world into those which are animate, inanimate, and in the background (environs). These systems tend to interact, so we need to recognize that the behavior of one may affect the others. Also, it is often our choice (as modelers) whether we choose to include each of these in our model. We are also the one who decides in which category any specific object will be placed.

Figure 3-1. The Meta-System Framework.



The conceptual system is an abstract model of the physical objects and their relationships. The meta-system framework suggests that organizations (of the physical objects) are used to accomplish functions in order to achieve goals. None of these entities is visible: we conceive (or have learned to perceive) the order of things about us.

The principal reason for making this distinction is to remind ourselves not to confuse the physical objects themselves with our representations of those objects. In modeling how people deal with objects in the real world, it is often necessary to consider that there is more information in the world than one can consciously deal with mentally. Part of any modeling problem dealing with humans (like task analysis) is to characterize these internalized and non-observable mental representations of what is happening in the world.

Consequently, task analysts should distinguish between at least two models within this framework: 1) one that reflects what is really going on, and 2) another which represents what individuals in the modeled world "think" is going on (later, this mental model will be segmented even further). The humans will be operating on what they internalized, not what "is," and those internalized mental models may be quite different for many reasons, as discussed in section 3.4.6. If the modeler fails to make that distinction, some rather strong assumptions are being implicitly made about what humans can or will do.

The other part of the problem is to then understand the structure (organization) and dynamics (functions and purpose) of the systems people operate. The designers had a purpose for building the system, but the hardware has no knowledge of that purpose: people do. The goal of the mission and system are perceptions related to what might be done with the animate objects (and may be imperfect representations of the objects and their relationships). To build a model of people operating systems, it is necessary to first understand what the systems themselves are intended to do and how they were designed to accomplish those objectives. Then we can appreciate what is expected of the operators, and why it is necessary to behave in a particular fashion.

The reverse is also true. Knowing what the operator does to or with the system, understanding system dynamics, we are then able to infer the consequences of human activities on system and mission outcomes. Our interest in human performance is of no consequence if it has no impact on the system's performance and mission outcome. We need to appreciate what this impact is: how the response of the human reflects itself in changes in system states (of our own inanimate object(s) and other's) and in changes in the environ.

3.3.2 The System Representation Framework

A canonical system description (Wilson and Wilson, 1965) consists of a description of: 1) system functions (input - output transformations), 2) control and status information, and 3) power or energy generation / distribution / dissipation. Rasmussen et al. (1990) cite a similar characterization by Alting (1978): 1) material flow, 2) information flow, and 3) energy flow. IDEF0 is well-suited to describing system functions, and IDEF1 to describing information relationships. Power (in the context of human behavior) might be viewed as the ability to sustain adequate (if not error-free) performance. Alternately, the study of stress, circadian rhythms, and fatigue effects on performance might be the appropriate analogue to studies of power generation and distribution in hardware systems.

IDEF0 was designed to address the systematic description of system functions, and IDEF1 was correspondingly developed to model data (and by implication, information systems). No corresponding IDEF methodology was developed to specifically address the power distribution or energy consumption issues of the canonical system description. The integration sought in ITAM is two-dimensional: 1) integration of IDEF0 and IDEF1 in the context of describing what people do in (with, and for) systems, and 2) integration of the various behavioral perspectives associated with proposed task taxonomies.

Rasmussen et al. (1990) propose five mean-ends levels: 1) purposes and values (including environmentally imposed constraints), 2) priority measures and flows (of mass, energy, people, information, money, etc.), 3) general work activities and functions, 4) specific work and physical processes with associated equipment, and 5) appearance, location, and configuration of physical objects. Tasks are viewed as a means to achieving some end (goal or requirement). The system itself (ergo the associated tasks) can (and typically will) be described on several levels of abstraction.

Figure 3-2 identifies the five-level system representation framework proposed by Rasmussen et al., along with two considerations: 1) system properties to be represented, and 2) couplings with (to) the work domain and the environment. The system properties captured at each level of the representation hierarchy have different behavioral implications. The couplings with the environment will affect different kinds of hardware and human activity. This figure begins to suggest why simple, linear task analyses fail to show the complexity of human behavior in systems.

The Figure 3-2. The System Representation Framework.

Means-End Relations	Properties Represented	Coupling: Work Domain	Coupling: Environment
1. Purpose & Constraints	Necessary & Sufficient Bases for Performance	Goals & Values	Requests & Intentions
2. Priorities & and Flows	Importance; accumulation & distribution of entities	Global Concerns	Exchanges Conservation
3. General Functions	Coordination requirements, irrespective of processes	Stores, Queues, Stacks	Communication & Synchrony
4. Physical Processes	Specific limitations and operating restrictions	I/O Links w/in system	Receipt & Delivery
5. Configuration and Appearance	Attributes for identifying, recognizing, & classifying	Structural Support	Topography & Connections

Rasmussen et al (1990) view goal (purpose-based) properties as being propagated top-down, performance effects (system properties) as being propagated bottom-up. In a hierarchical method, it is important to understand such relationships to know which direction to progress in the structure (up or down) from the present level. The character of the information processing will vary from level to level.

For example, mission requirements dictate system requirements, in turn dictating human performance requirements: top-down. But, human errors impact system behavior, affecting mission outcomes: bottom-up. The description of the system must anticipate the need to trace such cause and effect relationships in both directions. IDEF0 is essentially viewed as a top-down methodology. Network simulations based on them (e.g., SAINT) may be viewed as bottom-up analysis. The combination and integrated use of these analyses has been described by Chubb and Hoyland (1989) but applied only on a limited basis so far, and without inclusion of IDEF1 data modeling considerations.

Ultimately, strategy selection, resource assignment, and task management all require some processing of assimilated information that is deemed relevant to achieving stated or presumed goals. The extent to which information is unavailable, inaccessible, in error, or otherwise deficient, behavior is correspondingly affected in some fashion. Goal definitions, criterion specifications, ability assessments, and assumed restrictions or constraints can all interact with the task characteristics imposed by design and the operating environment. Task strategies must implicitly deal with this complex interaction of factors. IDEF1 provides a data modeling tool that yet needs to be successfully integrated with IDEF0 and SAINT, an issue deferred for now.

Studies examining knowledge acquisition methods (e.g., McNeese, et al., 1990) have indicated that some individuals have difficulty using the IDEF methodologies, and that the IDEF approach imposes artificial constraints that are often perceived as getting in the way of task description. These observations are not inconsistent with our findings that Miller's standardized terminology (much less a more formal methodology like SADT, also known as IDEF0) was difficult to use. However, a totally unstructured approach to soliciting information about a system provides descriptive freedom at the expense of making interpretation more difficult: the facts need yet to be organized and related in order to see if the picture is complete and accurate.

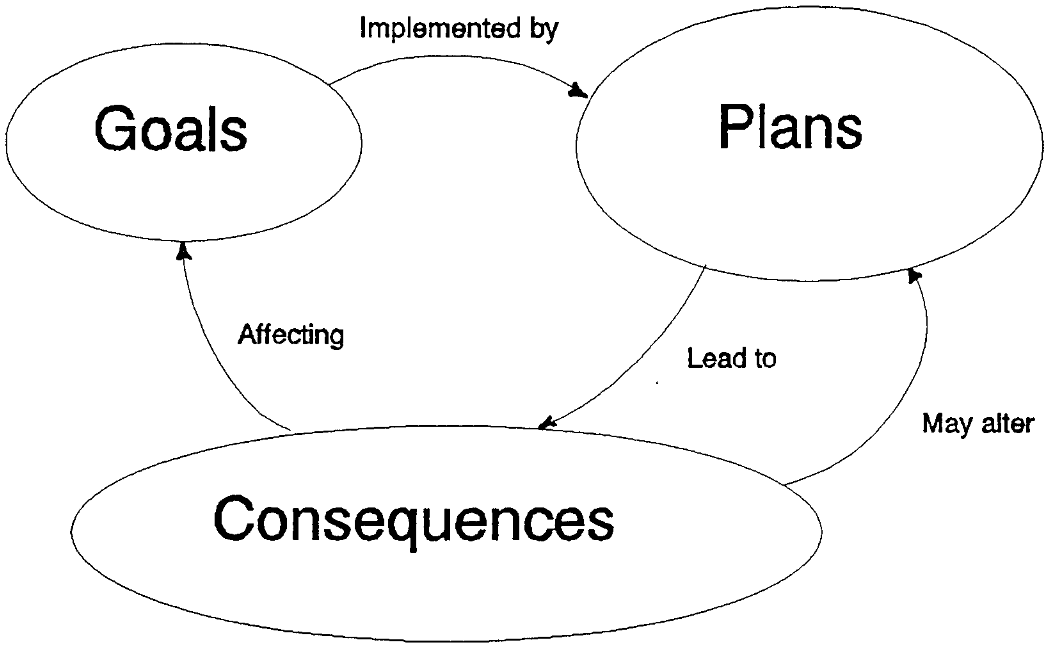
ITAM provides a framework for asking questions about the system and what people do that can guide analysis without dictating the form in which facts are documented or concepts represented. While ITAM is viewed as being applied within a system context, the subculture and operating environment associated with the mission / system context of interest must also be explicitly described and considered. As described later in the section on the ITAM process, it is also important to recognize that the task evolution that actually unfolds as a linear activity sequence is the result of a number of interactions among mental models or theories that guide activity. The ITAM framework suggests implicitly that a priori views of requirements and a posteriori descriptions of actions (governed by task strategies designed to respond to task characteristics and criteria) may not match.

3.3.3 The Activity Framework

The general characterization of activities is the simple four element diagram presented as Figure 3-3: 1) goals, 2) plans, and 3) consequences (the central elements of this concept map: key concerns). Some of the implications are also portrayed. Goals may not be well-defined until some actions take place. Plans may be non-existent as well as sketchy or incomplete. The consequences of actions (physical, cognitive, and emotional) are not always foreseen, and they often motivate (if not illuminate) goal clarification. Also, as a consequence of concomitant variation: changes occur that we may not have initially anticipated, and new goals may emerge as a result.

For example, in turning equipment on, we assume the heat generated will be dissipated. If the heat is not dissipated adequately, equipment malfunctions may occur. If we can and do detect the temperature rise, we may be able to act appropriately and avert the consequences of unanticipated equipment malfunction. If the problem with inadequate heat dissipation is not detected, then the consequences of that build-up may emerge in a wide variety of system malfunctions which will have to be dealt with as they occur. This concomitant system variation will insert new goals not anticipated initially, but arise naturally

Figure 3-3. The Activity Framework.



out of certain action (and / or inaction) consequences and failures to include latent goals in one's activity planning.

3.4 Proposed ITAM Methodology

The ITAM methodology should consist of an iterative process and should incorporate examination of: 1) mission and inferred performance goals (as well as considering the criterion measures for those goals), 2) plans and strategies (the accepted or preferred approach and / or alternatives to meeting expected and system mandated operating requirements), 3) task characteristics (stimulus and response considerations of the control and display ensemble as driven by the operating environment), and 4) abilities impacts (revisions of assumptions regarding the influence or impact of individual differences or changes in the state of all individuals involved). Task strategies are then viewed as the alternative methods (operations, procedures, techniques) by which activities are accomplished (stimuli are converted to responses using abilities, implementing and adapting any prior plan, as constrained by mission requirements, the operating environment, and system design).

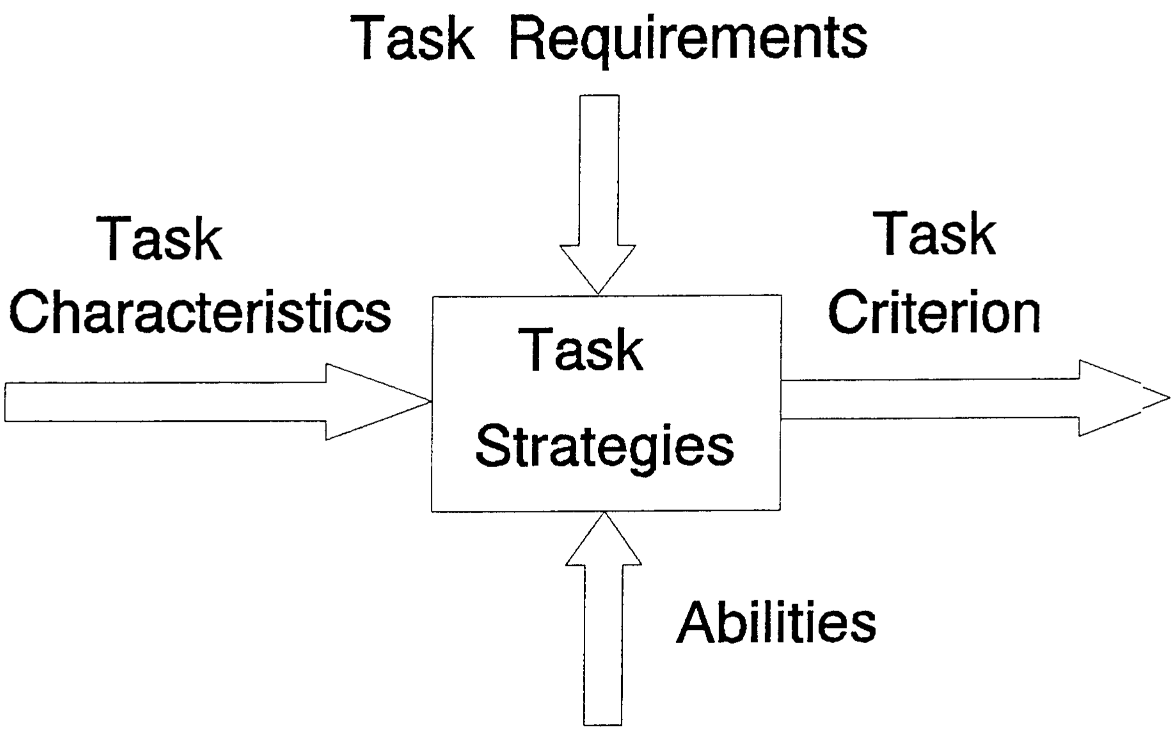
For psychomotor activities, task strategies result in body-part movements that are observable. For cognitive activities, task strategies involve information processing that is itself intrinsically non-observable: only its consequences may be observed, and then, only if psychomotor activity results. The irony is that considerable non-activity in the psychomotor realm of observable behavior may in fact imply considerable activity is taking place in the non-observable cognitive domain (as suggested by Card, Moran, and Newell (1983), among others).

Descriptive task analyses are seen as the context for validation, forcing revisions to content and product of a task analysis as necessary (when a training deficiency is not implied). It is also noted that the description of ongoing activities needs to capture considerable contextual information, including at least: 1) initial condition information (of people, equipment, and their respective environments as well as the relationships among entities, as that affects behavior or performance), 2) changes in not only the stimulus domain, but in the larger operational context portrayed by the displayed information, and 3) not simply responses, but the consequences of those responses, in terms of system, mission, and environmental impacts.

Figure 3-4 represents a graphic interpretation and integration of the various behavioral perspectives or taxonomic approaches discussed by Fleishman and Quaintance (1984). It uses the IDEF0 diagram syntax, described in Marca and McGowan (1989). In Figure 3-1, the task characteristics are seen as dominantly influencing the input side of task management, driving the strategies used to accomplish activities. Abilities are the resources or mechanisms by which tasks are accomplished, deficiencies in skill or proficiency may influence not only what is done but how. Task requirements imposed by the mission situation govern what must be done and influence priorities. The system design will enable as well as constrain or limit what strategies and actions can be applied in processing stimulus inputs. The criterion approach to task analysis is seen as focusing primarily on response or output requirements of human behavior, in order to meet the mission imposed demands, given the system dictated constraints, drawing on existing abilities, for the prevailing task characteristics.

This is in some sense an oversimplification for the sake of organizing and clarifying concepts. For example, task characteristics include response characterization. However, that concern also influences criteria for task success, so we respectfully suggest that response characterization be viewed in terms of criterion specification, not as task characterization per se. Correspondingly, other traditional concerns have not gone away, but in this depiction, they may be found in less familiar places. The reason for doing this

Figure 3-4. The Integrated Task Analysis Methodology.



is to use a paradigm familiar to at least some DoD engineers (IDEF0) and make the analysis process itself more systematic, by suggesting an orderly way in which to examine each of the task analysis concerns (requirements, abilities, characteristics, criteria and strategies).

Each of these major concerns associated with a task analysis (task characteristics, requirements, abilities, criterion, and strategies) interacts with other elements of the analysis, and each one of the major elements itself has subelements that bear on that interaction. At any one point in the analysis, one component element might be varied while the others are assumed to be constant (non varying). Subsequently, the constancy assumption can be relaxed to examine the impact multiple variations may have on the course of human activity. This may artificially constrain the preliminary analysis, but it provides a more manageable framework than trying to consider all possibilities at the same time, a clearly unworkable approach.

Which element(s) are fixed or varied will depend on the purpose for which the analysis is being done. If the analysis supports design, then the mission variability may be a principal concern, and changes in task characteristics may represent (or be associated with) design configurations to be evaluated. Other elements may be fixed when a given design alternative is examined. If the analysis supports performance assessments after the system is built, then the system design is fixed (requirements and constraints are also established), the mission variability might be restricted to a nominal and a worst case condition (for comparisons), success criteria may be of interest (e.g., how does system performance and mission success change as human reliability and accuracy change, and how will individual differences and performance variability affect mission outcomes). In a training context, the abilities element may be the variable of greatest concern. And so forth.

Subdividing each element provides insight into some of the questions that will need to be raised in the execution of the analysis. Each of the five **FICOM** elements will be addressed in turn: 1) Function (discussed last, as task strategies), 2) Inputs (identified as task characteristics), 3) Controls (identified as requirements), 4) Outputs (identified as criterion), and 5) Mechanism (identified as abilities). Hopefully, this will illustrate some of the varied concerns that might be addressed by ITAM. It also regularizes the discussion, providing a framework for discussing what needs to be addressed.

3.4.1 Requirements

Requirements can arise from a variety of sources, the most apparent being the mission situation or environment within which the task occurs. As changes occur in the operating environment, they need to be detected and evaluated to assess the need for action. Given an assumed need, a plan of action must be selected or formulated in enough detail to identify resource needs or tactical options. Invalid and incomplete plans can be updated to guide and direct activity. Requirements also form the basis for estimating anticipated or expected outcomes.

Goals instantiate and imply the task requirements: here's the objective. Mission requirements and system design often imply the task requirements. In social contexts, the goals may be considerably less clear. However, in either case, goals can be defined and achievement assessed different ways by different people. Sometimes, disagreements about performance adequacy may really be disagreements about the goal and / or its measurement. The goal at one stage of understanding may change in subtle ways as experience is gained. Goldratt (1986) suggests that consciously reformulating goals is the essence of achieving the continuous improvements needed in industrial practice to stay competitive in world-class markets.

The selection a reasonable goal may lead to successful but unacceptable performance: aiming for a landing point may get an aircraft down safely, but not as smoothly as leveling off into the ground effect and then bleeding off airspeed. Different perceptions of the task can lead to different goal specifications and performance, all for the same identical task. In team activities, introjection of a team goal may be different for each team member, as perceptions, experience, and other factors influence concept formation.

So a distinction may be usefully made between requirements as public statements and goals as psychologically significant counterparts. However, the engineering community often presumes no distinction is needed between requirements and goals. They are treated nearly as synonyms: the design goal is to meet requirements, per specifications. Satisfy the specifications, and the goal has been met. Psychologists see such statements as simplistic when they are applied to human systems or systems operated by humans. Goals as perceived may differ substantially from any stated set of requirements, however carefully worded the specifications might be.

3.4.2 Criterion

The question to be answered from this perspective is "How much is enough?" The term criterion can be used in two slightly different senses. First, it can refer to a factor or variable that can be observed and used as a standard: the criterion variable. Second, it can refer to a point value that divides this factor, dimension, or variable into two regions: 1) acceptable, and 2) unacceptable. When there is more than one dimension or criterion variable, two questions arise: 1) how will the multiple criteria be combined into a single criterion dimension (or an index for measure of merit or effectiveness), and 2) are all factors equally important or differentially weighted?

When more than one factor is used, some combination scheme (typically an equation, or alternately, some set of rules) is required to generate a new variable (an index), which is then used as the criterial dimension. If the factors are not all equally important, then some form of differential weighting is implied, whether that is formulated explicitly or not. The criterion for success can then be affected by: 1) altering the critical value on the criterial dimension, 2) changing the equation or combination rule being used, and 3) altering the assigned or assumed weightings of each criterion factor's importance.

Also, when there is more than one goal, additional questions arise: 1) can more than one goal be pursued at the same time (unification of demand), 2) can multiple goals be distributed (to others or over time), and 3) does achieving some goals affect the achievement of others (positively or negatively). Correspondingly, goals may also operate with corresponding constraints: restrictions on admissible actions, paths, states, resources, etc. Constraints may also propagate, where a constraint at one level of system description implies other constraints at other levels (above or below). The questions one might ask about goals often also apply to constraints.

Given the goal (requirements), how do we measure goal proximity and satisfaction? A potentially useful analogy is found in travel, where the goal is seen as a destination, and one's travel plans specify how one expects to get from the present state to that future state. One has either arrived or not, one can measure distance from the destination more than one way (influencing different kinds of evaluations), and if we have not arrived at the destination, then some adjustment of plans may be needed (now or later), including abandonment of the original goal (as one, often forgotten, option).

Some of the useful concerns are: 1) how can I know that the goal has been achieved (I've arrived: ambiguity or specificity of the goal definition), 2) how "close" is close enough to the goal (not only in distance, but in other measures, like rates of closure, or other higher order concerns), 3) how much flexibility exists in pursuing alternate subgoals on the path toward the ultimate goal (none, some, or unconstrained), and 4) what happens when the goal is achieved (does activity stop, shift, or simply continue: pursuing new goals)?

Other considerations include: 1) where did the requirement or goal originate (self or another, person or group), 2) was it elected (internal) or imposed (external), 3) does successful achievement depend only on one individual (self) or team (group), and are external people involved (where synchronization, coordination, and cooperation requirements may be implied), 4) what are the consequences of not achieving the goal (ever, or within some time frame), 5) what happens if pursuit of the goal is interrupted, and what happens if the desired resources are either not available to begin with or are removed during execution (goal pursuit).

Fleishman and Quaintance (1984) indicate the criterion measures can also be influenced by at least the following five related taxonomic concerns: a) personality (including temperament, mood, and emotional) factors, b) environmental (physical and social) considerations, c) organizational (superior and subordinate) demands, restrictions, or conventions, d) motivational (intrinsic and extrinsic) variables, and e) assigned or assumed team functions. Behavior tends to be influenced by the measure chosen for evaluation, whether that is made explicit or not.

Characterizing the nature of the response may also be necessary prior to specifying how its quality can be measured. Is it a one-time discrete action? Is it a series of discrete actions to a single stimulus (event): 1) precise or indefinite in number, 2) responses equally spaced physically and / or temporally, or 3) does responding simply continue until some other event or condition is satisfied? With no additional input, is the response repeated: 1) periodically or aperiodically, or 2) in synchrony with, in conjunction with, or as a result of other responses (chaining)? Are there characteristics of behavior which might have a differential impact on various observers, influencing their reactions? Do those properties of task execution need to be made explicit or can they be treated in some implicit fashion?

For example, the non-verbal communication associated with delivering an appropriate message in an inappropriate fashion may be a major determinant of success, whether the "speaker" knows the non-verbal message is being delivered or not, and whether the indications of communication failure are perceived or not. In some situations, a simple representation of success or failure might be adequate, where in other situations, the entire delivery and reaction sequence may be of interest. The purpose of the analysis will dictate what content and representation of response criterion is sufficient.

3.4.3 Characteristics

The characteristics of the task are directly related to the controls - displays interface with the person or crew operating and / or maintaining the system. At a fixed workstation location, the static and dynamic characteristics of console design and layout, display content and format, and control location and sensitivity will all play a role in determining the difficulty and complexity of what people have to do. At variable work locations, ingress / egress and transit from location to location become added concerns, including frequency and duration of relocation, and whether anything must be carried or lifted in the process, by a single person or some number of people.

The operability of equipment, tools, and fixtures depends on the complexity and difficulty of their use, which may change subjectively over time: any initially complex task seems simpler when proficiency is established, and what appears hard to do initially seems easier as experience is gained. However, for equal amounts of exposure (and equal abilities of persons doing the tasks), some tasks are harder / simpler, and more / less complex than others. Complexity deals with the structure of the task (number and diversity of activities and resources employed) while difficulty refers to the accuracy and precision of process control: sensory and conceptual discriminations and conscious psychomotor control. For example, automaticity should reduce perceived difficulty. Rasmussen et al. (1990) identify 7 factors to consider, as presented in Table 3-3.

Table 3-3. Factors Affecting Complexity.

1. Size of "The Problem Space"
2. Variety of Functional Elements.
3. Number of Goals and Objectives.
4. Compatibility of Goals and Constraints.
6. Number of Connections Between Elements.
7. Uniformity or Heterogeneity of the Workspace.

Both a description of real world (directly presented) and artificial (i.e., indirect) displays of objects, object representations or relationships, and derived or generated information about those objects, relationships, the environment, or system states is necessary to characterize the task. Three dimensions depict the characterization: 1) stimulus / response modalities exercised (visual, auditory, vestibular, haptic, kinesthetic, proprioceptive, etc.; eyes, head, hands, feet, etc.), 2) stimulus and response dynamics (intensities, frequencies, durations, etc.), and 3) functional role or purpose (alert, inform, guide; initiate, activate, deactivate, etc.).

Environmental factors may also need to be described and incorporated into the synthetic task, serving similar functions: 1) masking "noise", 2) distraction (forcing concentration), 3) interruption (diverting attention), and 4) moderation (changing difficulty or complexity). The way that noises, disruptions, interruptions, and moderating influences are implemented might differ, so long as alterations in psychological processing demands do not appear to be radically altered in the process.

For example, noise levels may not need to be as intense, as long as they complicate communication enough to evoke concentrated listening, and originate from appropriate directions. The frequency content (spectral components) of the noise might not need to be perfectly matched, so long as masking effects are roughly comparable. Special lighting may not be critical, so long a ambient lighting levels, contrast, glare effects (or shields), and colors are roughly comparable. Signal densities and rates at extreme values may be less important than matching typical or characteristic conditions, unless one's primary interest (for study or training) lies in that regime: reacting to extreme cases. Types and kinds of interruptions may not be as important as the handling they evoke (the psychological response).

If the synthetic or simulated environment is to represent the real system, several kinds of correspondence might be considered: 1) functional fidelity (operations, specifically: transformations of inputs into outputs, done in the real system, appear as required operations in the synthetic system), 2) input loads (diversity of types, number by type, rate of arrival, and mode of presentation) should be similar, 3) stimulus - response relationships ("if ___ (stimulus), then___ (response)" rules) should be the same, 4) response difficulty should be at least as difficult (but variable for proficiency development), and 5) response complexity no more demanding than the real system (appropriately coupled demands).

Representative decision making and problem solving activities need to be incorporated, subject to the purpose for which the synthetic task is being devised (performance risk assessment, training, design evaluation, etc. Decisions could occur at both the executive level (of selecting one versus other procedures) as well as at the implementation level (of getting a procedure implemented), where choices are then also necessary internal to that procedure.

Normative models (vs. naturalistic) of the decision process (Klein and Calderwood, 1991) at least suggest dimensions to consider: 1) response alternatives (comparable in number), 2) outcomes (type and number of consequences following choices), 3) uncertainties (relative frequency of outcomes), and 4) value (positive and negative aspects of those consequences). Applied problem solving dimensions are governed by the definition of the problem space (e.g., the GOMS model of Card, Moran, and Newell, 1983), which is closed for laboratory research, but may be largely open-ended in real scenarios: 1) goal specification,

2) operations (admissible actions), 3) strategies (methods), and 4) selection rules (for strategies and operations).

3.4.4 Abilities

Fleishman and Reilly (1991) provide a rather global set of assessment techniques based on years of test development and refinement. In most task analyses, the identification of requirements typically assumes abilities. Subsequent to system-task requirements specification, task analysts may review the implications for required skills and proficiency to do the indicated operation. Subsequent selection and training can then be guided by this assessment of implied abilities requirements. In theory, an individual's performance effectiveness depends on having established abilities sufficient to meet task demands: the right kinds of abilities to an adequate degree.

Performance risk assessments typically ask other related questions: What if abilities degrade; the wrong person is assigned; training is deficient; practice is inadequate; or some stressor degrades previously established abilities? The question implies that performance should also degrade: people may take longer to do the same thing, or they may make more errors (either in the number of different types of errors, in frequency of particular errors, or both). Such changes might very well affect the pattern and timing of observed activity.

Abilities may be necessary, sufficient (or neither) for task success. If an ability is necessary, the implication is that without the stated ability, the task cannot be done. An ability is sufficient if it assures task success. Many abilities may be neither necessary nor sufficient. If a desirable (but not necessary nor sufficient) ability is in some sense deficient or inadequate, the individual doing this task may alter the nature of the activity to achieve the same goal in some other way (e.g., concentration might be increased, a different hand might be used, etc.).

3.4.5 Strategies

Strategies are viewed as alternative plans of action. While requirements specify what must be done and criteria specify how well (jointly, the task demands), strategies describe mutually exclusive approaches for meeting task demands. A particular strategy may be specified or simply preferred. A plan is regarded as the apriori specification of tentative actions which collectively instantiate the strategy. Constraints encountered during execution of a plan may suggest a switch in strategies (or plans) is needed. There are two domains in which strategies apply: 1) observable psychomotor activity (work strategies, e.g., pacing), and 2) non-observable cognitive activity (problem solving strategies, e.g., trial and error).

Sternberg (1985) identifies four information-processing approaches to intelligent performance: 1) the TOTE, 2) the production, 3) the scheme, and 4) the rule (or principle). These all say nearly the same thing, but different ways. The specific representations differ, but similarities or presumed equivalences are noted by Sternberg (1985).

The TOTE (Test-Operate-Test-Exit) paradigm (Miller, Galanter, and Pribram, 1960) can characterize all behavior, observable and non-observable (but so could other models). There are a number of alternate admissible forms for this paradigm, the most common being a simple chain, where exit leads to the next first-stage test. Operate may consist of executing another TOTE of greater detail or specificity. The second-stage test (result=desired outcome "image") may be assumed successful instead of being explicitly executed (virtual open loop versus closed loop operation). If the first-stage test passes (present outcome=desired outcome "image"), operate does not typically occur, the second stage test clearly has been met already, so exit occurs, leading to continued first-stage testing of some other goal which eventually does lead to a non-degenerate TOTE module.

Some TOTE units may be seen as executives or operants. Executive TOTES are viewed as units controlling or supervising others. Operants change something (a stored concept, a retrieved memory item, a location or orientation of a body part, etc). Executive TOTES are by nature non-observable (perhaps inferable) and only some TOTE operants will be observable. The TOTE may serve as an augmented production: if condition (some testable state: first-stage test) occurs, then do (something), and be sure it happens (second-stage test).

Productions might be viewed as a variant of the TOTE, eliminating the second-stage test. The testable state may be external (which requires sensation and perception) or internal (operating on acquired, remembered, or processed information), but most implementations assume both the stimulus detection and response execution are accomplished, so productions apply to modeling the mental activity of information processing for problem solving. Also, the productions are typically strictly ordered and sequentially examined. Variations on this control structure have been examined.

Schemes are used to characterize concept structures: 1) figurative, 2) operative, and 3) executive. Figurative schemas are internal representations of information (so-called 'chunks') that govern familiarity, recognition, and identification. Operative schemas (lower order performance and knowledge acquisition components, information processes, or transformations) consist of rules applicable to figurative schema. Executive schema (plans, strategies, or metacomponents) then determine which figurative and operative schema are activated in the face of particular situations.

Rules (or principles) emphasize knowledge rather than process, being an equivalent to plan or strategy. The rules simply vary in complexity to suit the problem context and experience of the individual. Stimulus and response encoding becomes more efficient as chunking continues to evolve more advanced conceptual structures to provide automatized reactions to recognizable, repeated situations.

The subject (or team's behavior) is affected by three major classes of variables: a) the work situation (environment and task requirements), b) individual differences (abilities, strategies, knowledge, preferences, etc.), and c) the evolving scenario of events that characterize demands or imposed load. Team behavior adds communication concerns (for coordination, synchronization, and feedback). Strategy selections are now centralized, distributed, or a mixture of both, depending on predetermined rules and existing contingencies (e.g., loss of a commander, director, or other leader).

3.4.6 Cognition

Cognitive task analysis could be focused on either: 1) the processes, 2) the products, or 3) both. From evidence presented by Nisbett and Wilson (1977), emphasis on the products is safer than dealing with processes, and observers are nearly as good as participants in identifying the results of cognitive activity (based on subcultural theories about what responses should reasonably might be expected).

In searching the literature, only Sternberg (1985) and Rasmussen et al. (1990) seem willing to identify component processes. Sternberg in particular suggests which are elemental, and proposes a structural framework for linking component and meta-cognitive processes. Most other discussions are ad hoc, about specific kinds of cognitive operations (perceiving, reading, reasoning, remembering, problem solving, decision making, judging, choosing, etc.).

Techniques for componential analysis exist, but no set of components offered so far has been universally accepted as adequate in every applied context. More experience is needed in a variety of contexts.

It is often difficult to discern unambiguously where observable (psychomotor) processes cease and non-observable (cognitive) processes begin. The common example is differentiating between: 1) looking at something, 2) seeing what is potentially observable, and 3) perceiving the situation as observed. The non-observable, cognitive elements are sometimes concurrent with the observable psychomotor activity, eye position (line of regard and visual field), but may be sequential in other cases (e.g. reasoning based on the observables). Moreover, much of the processing that occurs depends on how the subject chose to encode the information, what chunking was used, what stereotypes exist, and what motivational factors that may influence processing in subtle ways. All of these individual differences have a bearing on how the internal processes might unfold.

3.4.6.1 Sternberg's Triarchic Theory

Sternberg's Triarchic Theory of Human Intelligence (Sternberg, 1985) provides a global framework, having three subtheories: 1) contextual, 2) experiential, and 3) componential. "Contextually intelligent behavior is specified to involve (a) adaptation to a present environment, (b) selection of a more nearly optimal environment ... , or (c) shaping of the present environment so as to render it better fit to one's skills, interests, or values ... for a given task or situation, contextually appropriate behavior is not equally 'intelligent' at all points along the continuum of experience ... (but) ... is best demonstrated when one is (a) confronted with a relatively (but not totally) novel task or situation or is (b) in the process of automatizing performance ... a componential subtheory, specifies the structures and mechanisms that underlie intelligent behavior... Metacomponents control one's information processing and enable one to monitor and later evaluate it... (Sternberg, 1985, pp. xi and xii)."

It is the metacognitive processes that permit flexibility. They control both performance and knowledge acquisition, in Sternberg's view (1985). The degree of flexibility that exists is also difficult to describe. Functions described in a linear appearing network (like IDEF) diagrams) will unfold in several different temporal activity patterns if there are no strong precedent relationships that force sequential dependencies. Sternberg identifies seven metacognitive components, shown in Table 3-4, seen as 'the executive' or

Table 3-4. Sternberg's Seven Metacognitive Components.

1. Deciding what the problem is that one must solve.
2. Selection of lower order components for problem solution.
3. Selection of strategy for organizing, ordering these components.
4. Selection of one or more representations that organize information.
5. Decisions about allocation of attentional resources.
6. Solution monitoring.
7. Sensitivity to external feedback.

Table 3-5. Sternberg's Six Lower Level Component Categories for Performance and Knowledge Acquisition.

<u>Performance Components</u>

1. Encoding components.
2. Combination and Comparison Components.
3. Response Components.

<u>Knowledge Acquisition Components</u>

1. Selective Encoding.
2. Selective Combination.
3. Selective Comparison.

'homunculus' control processes, used in planning, monitoring, and decision making.

3.4.6.2 Rasmussen's Prototypical Mental Strategies

Rasmussen, et al. (1990) identify three prototypical mental strategies: 1) analytical, model-based, 2) categorization-based, and 3) recognition-based, and list three major criteria for strategy selection: 1) resource requirements, 2) task specific versus wide applicability, and 3) sensitivity to disturbances. Their eight resource requirement criteria are listed in Table 3-6, and the seven elementary cognitive processes or strategies are presented in Table 3-7.

Table 3-6. Eight Resource Criteria for Strategy Selection.

1. Time
2. Information
3. Mental Capability
4. Physical Capability
5. Short Term Memory
6. Long Term Memory
7. Knowledge
8. Experience

Table 3-7. Rasmussen et al.'s Elementary Cognitive Processes.

1. Association
2. Induction
3. Deduction
4. Hypothetico-Deduction
5. Search
6. Comparison and Choice
7. Evaluate

3.4.6.3 Bolman's Theory of the Situation

Bolman (undated paper) presents an intriguing contrast among the various perspectives we need to explain why people do what we observe them doing. People may in fact base their own behavior on an incorrect mental model: their personal

"Theory of the Situation (TOS)," a short-term set of beliefs which imply what tasks are identified as appropriate and assigned to self or others. The individual's TOS is determined by the situation and by two kinds of long-term personal characteristics: 1) fundamental cognitive and behavioral parameters, and 2) the 'Theory of Practice (TOP)," which governs how to design, test, and implement one's TOS. Cognitive and behavioral capabilities and limitations must consider attention span, short-term memory, the chunking of information (patterns), and the integration of smooth, effortless response patterns (skills): reactions slow as the situation falls outside of familiar patterns or demands actions other than established skills.

Bolman suggests that problems arise when reality and theory are mismatched. The likelihood of detection and revision of faulty theory depends on: 1) "Theories of Practice," 2) an ability to combine "advocacy" and "inquiry," 3) any partner's management style and skill, and 4) the extent to which interacting partners have agreed on roles and procedures for modification (or reassignment). Bolman contends that a role defines task expectations. If the expectations are vague, role ambiguity exists. Then 1) Restrictions apply (Delegation is poor, resulting in load imbalance or thrashing.); 2) Under- and over-differentiation occur (<u>under</u>: too many crewmembers doing or focused on same thing, redundancy is good but conflicts can occur; and <u>over</u>: no coordination, so some tasks may get left undone); and 3) Boundary management fails, because the tasks in the role domain are ill-defined (What can I expect from others and what can they expect me to do?).

Both TOS and TOP are 'Theories for Action (TOA)' which help an individual select what to do in complex environments: "cognitive and behavioral frameworks that guide them in deciding what variables to attend to, what information to seek, what causal relationships to expect and what actions to take (Bolman, undated, p. 10)." TOAs contain four components: 1) core values, 2) beliefs, 3) skills, and 4) outcomes (consequences of behavior which provide knowledge of results).

Bolman distinguishes two versions of a TOA: 1) the Espoused Theory (representing the individual's account or explanation), and 2) the Theory-in-Use (which validly predicts actual behavior). These two do not always match, as Nisbett and Wilson (1977) observed. The espoused theory is: 1) incomplete because details are lost with automatizing the task, 2) sometimes irrelevant or contradictory (when individuals are unaware of major behavioral drivers), and 3) shaped more by a need to preserve self-esteem than by any need for accuracy.

Knowledge is: 1) 'intellectual' when it is in the espoused theory but not in the theory-in-use, 2) 'tacit' when it exists in the theory-in-use but not in the espoused theory, and 3) 'integrated' when someone can both do something and describe it. Reasons for inconsistency also include: 1) discrepancy induced anxiety which motivates forgetting, 2) self-fulling / self-sealing processes (I think I'm cool, so those who think I'm abrasive miss the point.), and 3) unrecognized contradictions (e.g., due to incomplete information specification).

Inverting this structure, the theory-in-use "consists of the core values, beliefs, and strategies which provide direction, meaning, and uniqueness to everything that I do ... my executive program ... incorporates a number of TOPs for different practice arenas (Bolman, undated, p. 14)." Modifying or adapting an overlearned TOP costs time and energy, it requires effort and may generate stress.

Bolman (undated, p. 15) indicates this tendency to hold onto existing TOPs is called the 'conservative impulse,' and is intrinsic to our capacity to survive. To interpret the situation, we must match it with experience, each discovery being the basis for the next, and as our understanding grows, we defend its validity. As Bolman (1975, p. 14) quotes "...we ... feel immediately threatened if our basic assumptions and emotional attachments are threatened." To let go requires accepting the risk associated with ambiguous outcomes and the uncertainty of each. There is a loss of meaning when we abandon a familiar interpretation of the world.

Table 3-8 suggests factors affecting an individual's willingness to engage in theory-revision. The basic tradeoff is whether: 1) it is more efficient to continue implementing the present theory (and risk incurring any resulting penalty costs) or to pay the price of revising the theory to correct its errors and deficiencies, in order to avoid the associated penalty costs. Answers to two basic questions determine whether theory revisions will occur on-line: 1) is there any information indicating the present theory of the situation is invalid, and 2) does the crew's present TOP lead them to use that information?

Table 3-8. Factors Affecting the Revision of Mental Theories.

1. Theories central to self-concept and self-esteem are less likely to be revised.
2. The more highly overlearned theories are less likely to be revised.
3. The more inquiry and adaptation are built into the theory, the more likely it is to be revised.
4. The more disconfirming data available, the more likely it is to be revised.
5. The greater the ambiguity, confusion, information overload, and stress, the less likely it is to be revised.

Information availability is then a necessary but not sufficient condition. It is normal for the person in charge to defend their TOS rather than asking if it is in error. A better approach is to establish a TOP that incorporates both inquiry (for testing the TOS), and advocacy (for recommending TOS alterations).

3.5 The ITAM Process

In its simplest form, the ITAM process consists of identifying: 1) goals, 2) actions, and 3) consequences. In doing so, it tries to recognize the many interacting factors which might influence how each of these constructs is perceived and might in turn affect the others. For example, an individual's anticipation of adverse consequences of a contemplated action might very well lead to an implicit goal redefinition task that is never explicitly identified in mission or system operating requirements. The ITAM framework provides a restructuring and combining of prior task analysis perspectives while suggesting that what seems like a linear sequence of observed events in a behavioral description is only the result of eliminating many other options that could have occurred and will be observed as conditions and theories of the situation change.

ITAM could be implemented more than one way, and from our present study, at least two documentation methods should be pursued: graphic portrayals and structured textual descriptions. The IDEF0 diagram has an associated node tree which supports indentured outline generation. Tabular presentations of IDEF0 information are possible from a database of FICOM labels. Similar dual manipulations of IDEF1 appear feasible (graphic and corresponding verbal representations).

It might be appropriate to list some basic assumptions, identify implications, and suggest impacts on ITAM procedures. Details of ITAM implementation will still need further description as application experience accrues. Assumptions include:

1. Any system description is essentially incomplete: all representations are potentially in error; reality generates interpretable data that permit one or more people to validate their model of a system. The consensus of some group (e.g., system design and product manufacturing) or the documented representation they sanction as "approved" becomes the Truth Model for the system. Reality may vary from the approved Truth Model.

2. In operating and / or maintaining a system, people build mental models of the system, its environment, their role in the system, and what to expect when action is taken. People begin the model building process from a pre-established theory of their own behavior and how things in the world around them actually work. Individuals also construct models about what other people are doing and why. Collectively, these central, internal mental models people bring with them today are their initial Theories-in-Use (TIU).

3. Any task analysis is itself a model of the mission / system / work / job situation. It is a formal and explicit attempt to represent one or more mental models in a publicly examinable and possibly testable form. As a representation of a non-observable entity (a mental model), any task analysis may not perfectly match what it is supposed to describe. For the operator or maintainer, their mental model of the task, including their role in team activities, constitutes a Theory of Practice (TOP): a goal image for guiding and directing their own behavior in a particular system context.

4. Jointly, the TIU and TOP form a Theory of Action (TOA), consisting of a mix of a repertoire of highly overlearned and preexisting behavior (TIU) and the context-specific TOP behaviors. Using a computer software analogy, the TOA is like an executable load module that controls behavior, consisting of application specific commands (TOPs) and more general modules (TIU) linked to those from a store of libraries. Both procedural and declarative knowledge are carried in the object or load module.

5. Any attempt to make these theories explicit leads to a Theory of Espousal (TOE) or explanation. TOEs may be distorted, intentionally, inadvertently, or unwittingly. Analysts must therefore develop a sensitivity to areas where deviations between espoused theories of operation and actual theories of practice (TOP) or use (TIU) may differ. TOEs can serve several legitimate purposes of their own, so they are not to be dismissed or "corrected," but they must be recognized for what they are.

5. All of the above are relatively static theories. They may represent unrealistic simplifications of task dynamics as they actually occur. Because of interactions that inevitably drive behavior in unanticipated directions, adjustments are required as conditions evolve. That requires the construction of a Theory of the Situation (TOS). The TOS feeds the TOA, driving it with input data and supporting it with assimilated information. To the extent the TOS is deficient, the TOA may lead to errors. However, deficiencies in the TIU or TOP components of the TOA may also lead to errors.

6. The TIU will consist of values and beliefs that shape perceptual and action patterns built into the TOS, TOP, and TOA. Those values and beliefs will also affect what gets incorporated in the TOE (and what does not appear there as well). Those values and beliefs are functions of inherited traits, personality development, and acquired cultural influences. Consequently, they tend to be robust and resistent to change. By contrast, the TOS is much easier to change. The TOA is systematically adjusted to achieve satisfactory results: first to the

TOP but potentially to the TIU as well, either as changes or in terms of additions to the library.

7. All of the internal workings of cognitive processes are based on transforming encoded information from sensed stimuli to response commands. Psychomotor tasks are the observable result of unobservable cognitive processing. The products of the processing are more easily and accurately described than the processes themselves and should therefore be the focus of the analyst's attention.

8. The data encoding operations and the chunking of assimilated information is subject to variation and its precise form and nature in actual instances is difficult to validate, yet accurate representation of the internalized constructs is central to determining (or predicting) both the speed and accuracy of the observable response to presented data (and not all data are informative).

9. Constructive models of encoding and processing can be matched to actual responses. Models which generate appropriate matches are valid; those which do not match are invalid descriptions of behavior. It is not in general possible to discern from responses what the internal processing actually is. More than one constructive model might prove valid in one instance. Models which generalize across instances are clearly preferred. Validating TOS should be easier than TOPs, which should be easier than TOAs. The TIUs will be the most difficult challenge.

10. It is best to begin the analysis with nominal operating conditions and an assumed match between reality and the approved system model. Assumptions about environmental impacts, abilities differences, and other very real considerations (like alternate TOSs and TOAs) should be incrementally added to the analysis only after a baseline analysis exists. The question" "What if ...?" is always extensible to additional considerations not yet incorporated into the task analysis.

The basic methodology suggests that there should be an iterated process of examining some central issues (like goals, plans, and expected consequences of actions) and making certain distinctions clear (goals and behavioral objectives in the mind of the individual operator may not match the designer's interpretation of system operating requirements). Synthetic tasks should support formulation of appropriate TOSs, recognizing that the TOP and TOA may vary due to task characteristic differences in the real and synthetic environments. The key element of the synthetic environment is its ability to elicit or evoke similar TIU elements in the construction of the TOA.

3.6 Recommendations for Further Study

The ITAM methodology proposed here still needs additional work. It is recommended that the ITAM methodology refinements first be applied to an analysis of an actual TRACON as well as to TRACON II in order to identify the differences that may exist in task requirements, characteristics, and criterion variables.

Second, continuing study is needed to examine the set of terminology that might best be used to standardize the concepts (especially those used to describe cognitive processes) in a more consistent and coherent fashion. This requires not only a definition of the terms themselves, but some explanation of envisioned relationships among those terms, at least first order or nearest neighbor sorts of relationships. Third, given more than one expert application of the ITAM, more novice users may be studied, comparing the consistency of their results to those of the so-called experts. Finally, the question of validity needs to be more thoroughly examined in its own right: what does it mean and how is it measured in this particular application context --- task analysis?

An integrated set of analysis tools is needed. While IDEF0 and IDEF1 tools are available, their application in this context needs to be examined in greater detail. Support aids may be needed, either as additional tools operating on IDEF outputs, or as integral components incorporated into future versions of the IDEF tools themselves. The later will be more difficult to achieve, so first consideration might be given to providing tools which augment and complement what is already available.

Further, the problem presented to the analyst is not how to use the IDEF tools. That should be presumed. The ITAM problem is how to apply the IDEF tools to the problem at hand: task analysis and activity modeling. Guidelines are needed on "How to ..." adapt these tools. That has been done for IDEF0, not IDEF1. ITAM at this point sketches a skeletal structure and suggests what alternate paths might be pursued. None have been explored, and the road map still has to be drawn so others can follow the route taken during initial navigation, making changes as experience grows.

SECTION 4.0

FLIGHT SIMULATOR INSTRUMENTATION

The present effort included the acquisition of hardware and software for performing future studies of flying tasks, using a surplused Air Force T-40 flight simulator. This training device was built by Singer-Link for the T-39 aircraft, a twin engine jet (the commercial version being the Sabre Jet), based on the original F-86 design. The simulator itself has been at Ohio State University for some time, but it has not been actively used in several years.

The T-40 was installed, and checked out. The attitude direction indicator has an erroneous pitch indication, one of the two Heading Situation Indicators (HSIs) was malfunctioning, and several other minor deficiencies were noted. Necessary parts have been requisitioned but not yet obtained. Part of the problem is in putting the device back on the Air Force list of active training devices, which will allow ordering parts from allocated provisions. It is an Air Force device, remotely located at Ohio State University for the purpose of research of mutual interest. In order to provide a formal basis that allows pursuing further research, a Memorandum of Agreement (MoA) between the University and the Armstrong Laboratory was prepared and is still being coordinated.

The instrumentation package being developed for the OSU operated T-40 was based upon the instrumentation of the T-40 operated at Brooks Air Force Base (AFB), Texas. The signal conditioning circuits, analog to digital converter, graphics display card, and personal computer are all nearly identical to the configuration at Brooks AFB. The Turbo PASCAL software developed for the Brooks T-40 will also be used initially to collect data from the OSU T-40. Castle AFB, CA supplyied several (3) sets of punched cards for the T-40 card reader. These punched cards provide input to the navigation system by identifying the geographic location and operating characteristics of various navigation aids (e.g., NDB, VOR, and TACAN stations).

Presently, the instrumentation package has been acquired and is in place, but it is still being applied to the T-40. Full checkout and use is not expected until after the termination of the present contractual effort, but this activity was initiated in anticipation of future research efforts in any case. No studies using the T-40 were to be conducted in the present research.

Two software packages that were to be procured, funds available, cannot actually be procured because other requirements of higher priority consumed the allocated monies. The instrumented version of TRACON II was one of the two to be acquired, but the price was much higher than anticipated ($2000.00), and the OSURF assesses a 47% surcharge for ordering. The IDEFine software was not acquired for the same reason ($2000.00 academic purchase cost plus an ordering surcharge). Other software acquisitions were also deferred: 1) SLAM System ($400.00), and 2) PROOF Animation ($1500.00).

Hopefully, these packages will be acquired in support of future research efforts. These software packages will support computer modeling of human operator activities, based on task analysis data. The remaining budget for the current AFOSR RIP Grant was instead used to acquire some general purpose laboratory support software to study manual control, quickening, aiding, and preview / predictor display techniques. Selected technical publications that would support our continuing research program were also procured.

SECTION 5.0

CONCLUSIONS AND RECOMMENDATIONS

Several conclusions emerge from the empirical study of Miller's terminology, and since ITAM has yet to be tested, recommendations are provided for future research.

5.1 Empirical Study Conclusions

Task analyses appear to be the analyst's best attempt to make explicit a representation of their mental model for perceived activity requirements. This implies that the lack of comparability between different task analyses may arise from several sources: 1) the ability to correctly perceive the task characteristics and criteria which drive behavior and dictate acceptable performance, 2) the ability to discern relevant behavioral processes (cognitive and psychomotor) that serve as alternative means to ends (strategies or plans for goal achievement), 3) the ability to find and use a suitable representation technique, and 4) failure (or inability) to recognize how individual differences in ability affect actual task execution.

The anecdotal data as well as the empirical evidence of our study of Miller's terminology suggest that not everyone is equally comfortable with the same representation technique. Not only do their preferences differ, they believe their's is the better choice (by implication, for others also). By itself, this presents a formidable barrier to overcome in communicating the results of task analyses from one community to another, if their preferred means of representation and analysis documentation happens to differ.

It was also clear that analysis concepts and activity labels are a major semantic problem that will not be easily solved. A language suitable for scientific purposes may conflict with terminology which best communicates with operations-oriented users. Calibrated correlation of terminology will be required if two sets of terms become necessary (one operational, the other scientific).

Also, the complexity of human behavior in an operational context is quickly lost in the struggle to describe the activity more simply: the focus on a linear sequence (the one I use, perceive is right / best, or deem is the one required) is nearly overpowering to beginning analysts. The ability to discern (much less describe) alternative courses of action is very limited, at least in the early stages of trying to learn this kind of analysis. Two problems were noticed in the anecdotal data: 1) subjects were late to recognize this multi-option problem existed, and 2) if and when recognized, they often had no tool readily available to deal with it. Those who knew how to create a logic flow diagram did so, but not everyone has that preparation and training.

The question of individual differences never seemed to surface in any analysis, nor in any analyst's comments. This is surprising only in the sense that TRACON instructions clearly indicate several different action plans can all be used to achieve the same operational objective. The TRACON II instructions recommend: create a plan, then make it work. The analyses all implicitly show the analyst's chosen plan. None identify alternate feasible plans. Whether that question would surface with additional practice or interaction with other players remains to be seen. It is recommended as a good topic for future research.

5.2 Recommendations for Further ITAM Research

The ITAM was outlined but not fully developed. The itegration of IDEF1 methods into the ITAM process is still needed. The whole ITAM application process needs to be demonstrated to illustrate the procedures for collecting, organizing, and documenting the task analysis. In particular, ITAM application should support: 1) synthetic task description, and 2) sequential network model development.

The cognitive component of ITAM has been examined but not fully specified. Of the several proposals applicable to doing cognitive analysis, all covered similar conceptual territory. Of the three macroconcepts (Sternberg, Rasmussen, and Bolman), no one scheme stands out as necessarily superior. Of the four micro-approaches reviewed by Sternberg (TOTE, productions, schema, and rules), all seem to work, but none are universally accepted. At least the domain of feasible approaches has been narrowed, and the overall architecture seems to be accepted (meta-cognitive control of micro-process elements).

To facilitate the evolution of ITAM specification and refinement, it is recommended that: 1) the Columbus TRACON facility be used a a test bed for application of the ITAM concepts, 2) the Army's SYNTAS package be used for development of a synthetic task representing some or all of the TRACON operations, and 3) the SLAM System (or C-SAINT) and PROOF Animation software packages be used for the sequential network modeling of both the real and synthetic tasks.

Using IDEFine as the IDEF0 and IDEF1 modeling package permits downloading the static functional and data models into a dBASE-IV relational database, where (in theory at least) the information is accessible to simulation and other analysis routines (e.g. Symantec's Timeline and Microsoft's Excel). Integrated utilization of existing commercial-off-the-shelf (COTS) software is an appealing approach for further development of the ITAM concept, avoiding special versions of non-supported software.

Once the ITAM application is completed, the empirical study of synthetic task construction can be examined in parallel with reliability and validity studies of ITAM itself, based on the general approach taken in the present study of the Miller terminology.

REFERENCES

Alexander, J. R., H. L Ammerman, W. S. Fairhurst, C. M. Hostetler, and G. W. Jones, FAA Air Traffic Control Operations Concepts, Volume VIII: TRACON Controllers, 8 September 1989, DOT/FAA/AP-87-01, U. S. Department of Transportation, Federal Aviation Administration, Washington, DC.

Alting, Leo (1978), "A Systematic Theory of Manufacturing," Environment and Planning B, Vol. 5, 131-156.

Bolman, Lee (undated), Aviation Accidents and the 'Theory of the Situation', Graduate School of Education, Harvard University, Cambridge, MA.

Brisley, Chester L. and Eady, Karl (1982), "Predetermined Motion Time Systems," in G. Salvendi, editor, Handbook of Industrial Engineering, Wiley, New York, NY, 4.5.1-4.5.31.

Card, Stuart K., Moran Thomas P., and Newell, Alan (1983), The Psychology of Human-Computer Interaction, Lawrence Erlbaum and Associates, Hillsdale, NJ.

Chubb, Gerald P. and Hoyland, Constance M. (1989), "Systematic Modeling of Multi-Operator Systems to Evaluate Design Concepts," in Grant R. McMillan, David Beevis, Eduardo Salas, Michael H. Strub, Robert Sutton, and Leo van Breda, editors, Application of Human Performance Models to System Design, Plenum Press, New York, 295-311.

Companion, Michael A. and Corso, Gregory M. (1982), "Task Taxonomies: A General Review and Evaluation," Int. J. Man-Machine Studies, 17, 459-472.

Cywar, Adam W. (1982), "Development and Use of Standard Data," in G. Salvendi, editor, Handbook of Industrial Engineering, Wiley, New York, NY, 4.8.1-4.8.19.

DeGreene, K. B. (1970), Systems Psychology, McGraw-Hill Book Company, New York, NY.

Drury, Colin G., Paramore, Barbara, Van Cott, Harold, Grey, Susan M., and Corlett, E. Nigel (1987), "Task Analysis," in G. Salvendi, editor, Handbook of Human Factors, Wiley, New York, NY, 370-401.

Ericsson, K. Anders and Simon, Herbert A. (1984), Protocol Analysis: Verbal Reports as Data, The MIT Press, Cambridge, MA.

Fleishman, Edwin A. and Quaintance, Marilyn K. (1984), Taxonomies of Human Performance: The Description of Human Tasks, Academic Press, New York, NY.

Fleishman, Edwin A. and Reilly, Muareen E. (1991), Handbook of Human Abilities: Definitions, Measurements, and Job Task Requirements, Consulting Psychologists Press, Inc., Palo Alto, CA.

Geisel, Charles E. (1982), "Methods Design," in G. Salvendi, editor, Handbook of Industrial Engineering, Wiley, New York, NY, 3.1.1-3.1.11.

Goldratt, Eliyahu M. and Cox, Jeff (1986), The Goal: A Process of Ongoing Improvement, Revised Edition, North River Press, Croton-on-the-Hudson, NY.

Kidd, Jerry S. and Van Cott, Harold P. (1972), "System and Human Engineering Analysis," in Harold P. Van Cott and Robert G. Kincade, editors, Human Engineering Guide to Equipment Design, Revised Edition, U. S. Government Printing Office, Washington, DC, 1-16.

Kirwan, B. and Ainsworth, L. K., editors (1992), A Guide to Task Analysis, Taylor and Francis, Washington, DC.

Klein, Gary A. and Calderwood, Roberta (1991), "Decision Models: Some Lessons from the Field," IEEE Transactions on Systems, Man, and Cybernetics, 21(5), 1018-1026.

Meas, Pattie (1989) "How to Do the Right Thing," Connection Science, Vol. 1, No. 3, 291-323.

Marca, David and McGowan, Clement (1989), SADT: Structured Analysis and Design Technique, McGraw-Hill, New York.

McCormick, Ernest J. (1982), "Job and Task Analysis," in G. Salvendi, editor, Handbook of Industrial Engineering, Wiley, New York, NY, 2.4.1-2.4.21.

McNeese, Michael D., Zaff, Brian S., Peio, Karen J., Snyder, Daniel E., Duncan, John C., and McFarren, Michael R. (1990), An Advanced Knowledge Acquisition Methodology: Application for the Pilot's Associate, AAMRL-TR-90-060, Harry G. Armstrong Aerospace Medical Research Laboratory, Wright-Patterson AFB, OH.

Miller, George A., Galanter, Eugene and Pribram, Karl H. (1960), Plans and the Structure of Behavior, Henry Holt and Company, New York.

Miller, Robert A. (1967), "Task Taxonomy: Science or Technology?," in W. T. Singleton, R. S. Easterby, and D. C. Whitfield, editors, The Human Operator in Complex Systems, Taylor & Francis, London.

Neibel, Benjamin W. (1982), "Time Study," in G. Salvendi, editor, Handbook of Industrial Engineering, Wiley, New York, NY, 4.4.1-4.4.37.

Panico, Joseph A. (1982), "Work Standards: Establishment, Documentation, Usage, and Maintenance," in G. Salvendi, editor, Handbook of Industrial Engineering, Wiley, New York, NY, 4.1.1-4.1.25.

Rasmussen, Jens (1986), A Cognitive Engineering Approach to the Modeling of Decision Making, Technical Report Riso-M-2589, Riso National Laboratory, Denmark.

Rasmussen, Jens, Pejtersen, Annelise Mark, and Schmidt, Kjeld (1990), Taxonomy for Cognitive Work Analysis, Technical Report Riso-M-2871, Riso National Laboratory, Denmark.

Redding, Richard E. (1989), "Perspectives on Cognitive Task Analysis: The State of the Art," Proceedings of the Human Factors Society 33rd Annual Meeting, 1348-1352.

Roth, Emille M. and Woods, David D. (1989), "Cognitive Task Analysis: An Approach to Knowledge Acquisition for Intelligent System Design," in G. Guida and C. Tasso, editors, Topics in Expert Systems Design, Elsevier Science Publishers, North-Holland, New York, NY, 233-264.

Salvendi, Gavriel (1987, first ed. and 1992, rev. second ed.), Handbook of Industrial Engineering, Wiley, New York, NY.

Sathe, Vijay (1985), Culture and Other Corporate Realities: Text, Cases, and Readings, Richard D. Irwin, Homewood, IL.

Schvaneveldt, Roger (1990), Pathfinder Associative Networks: Studies in Knowledge Organization, Ablex, Norwood, NJ.

Sternberg, Robert J. (1985), Beyond IQ: A Triarchic Theory of Intelligence, Cambridge University Press, New York, NY.

Van Cott, Harold P. and Kincade, Robert G. (1972), Human Engineering Guide to Equipment Design, Revised Edition, Superintendent of Documents, Government Printing Office, Washington, DC.

van Dijk, Teun A., editor (1985), Discourse Analysis, Vols. 1-4, Academic Press, New York, NY.

Wilson, Ira G. and Wilson, Marthann E. (1965), Information, Computers, and System Design , John Wiley and Sons, Inc., New York, NY.

Woods, David D. and Roth, Emille M. (1988), "Cognitive Systems Engineering," in Martin Hollander (editor), Handbook of Human-Computer Interaction, North Holland, New York.

Appendix A

Similarity Questionnaire and Instructions

The Task Analysis Questionnaire consists of three hundred items, created by pairing each of the twenty-five terms in Miller's Task Strategy list of terms (Fleishman and Quaintance, 1984, Appendix A). Only the instructions were changed for each of the three administrations, so only one copy of the questionnaire itself appears here. The cover sheet of instructions was changed only slightly (the title), as evidenced in the first three pages that follow.

AV 540 Project C

Task Analysis Questionnaire

Similarity of Miller's Functions Terminology

Instructions: We want to determine how similar (or dissimilar) Miller's 25 terms seem to you. These are the terms that were in the Course Handout. We will ask you to use these terms in your task analysis. Before you begin, we would like to get some data that help us understand whether you see certain terms as being closely related or very different. So, we will present all the terms in pairs. For each pair, you will be asked to decide the degree to which you believe they are similar or dissimilar in meaning, on a scale that has ten values: 0 through 9. We want you to think of zero as completely dissimilar: the terms are not at all alike. Then 9 will represent the case where you believe the terms are very similar, nearly identical or synonyms. For practice, consider the following two examples:

	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
1. red-green		✓	___	___	___	___	___	___	___	___	___	

In this case, you would put a check in the space below zero, since red and green are complementary colors. They are as dissimilar as you can get!

	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
2. gorgeous-beautiful		___	___	___	___	___	___	___	___	___	✓	

In this case, the pair are virtually synonymous, and if you agree, then perhaps we should put out check in the space below the number 9.

	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
3. anger-hatred		___	___	___	___	✓	___	___	___	___	___	

In this case, the answer depends a lot on how you, personally, interpret these two words. Some people assume they are hated when someone is angry with them. Others see anger and hatred as two separate emotions that may or may not be associated. If you see anger and hatred as closely related (and therefore similar), you would mark one of the spaces to the right, toward the 9 end of the scale. If you see the two words as being more dissimilar, you would mark a space to the left, maybe not close to 0, but certainly not close to 9.

If you have any questions about how to answer this questionnaire or mark the scale, now is a good time to do so.

YOUR NAME: ______________________________ DATE: ____________

AV 540 Project C: Task Analysis Questionnaire

Second Repetition

Similarity of Miller's Functions Terminology

Instructions: We want to determine how similar (or dissimilar) Miller's 25 terms seem to you. These are the terms that were in the Course Handout. We will ask you to use these terms in your task analysis. Before you begin, we would like to get some data that help us understand whether you see certain terms as being closely related or very different. So, we will present all the terms in pairs. For each pair, you will be asked to decide the degree to which you believe they are similar or dissimilar in meaning, on a scale that has ten values: 0 through 9. We want you to think of zero as completely dissimilar: the terms are not at all alike. Then 9 will represent the case where you believe the terms are very similar, nearly identical or synonyms. For practice, consider the following two examples:

	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
1. red-green		✓										

In this case, you would put a check in the space below zero, since red and green are complementary colors. They are as dissimilar as you can get!

	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
2. gorgeous-beautiful											✓	

In this case, the pair are virtually synonymous, and if you agree, then perhaps we should put out check in the space below the number 9.

	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
3. anger-hatred						✓						

In this case, the answer depends a lot on how you, personally, interpret these two words. Some people assume they are hated when someone is angry with them. Others see anger and hatred as two separate emotions that may or may not be associated. If you see anger and hatred as closely related (and therefore similar), you would mark one of the spaces to the right, toward the 9 end of the scale. If you see the two words as being more dissimilar, you would mark a space to the left, maybe not close to 0, but certainly not close to 9.

If you have any questions about how to answer this questionnaire or mark the scale, now is a good time to do so.

YOUR NAME: ______________________________ DATE: __________

AV 540 Project C: Task Analysis Questionnaire

Third Repetition

Similarity of Miller's Functions Terminology

Instructions: We want to determine how similar (or dissimilar) Miller's 25 terms seem to you. These are the terms that were in the Course Handout. We will ask you to use these terms in your task analysis. Before you begin, we would like to get some data that help us understand whether you see certain terms as being closely related or very different. So, we will present all the terms in pairs. For each pair, you will be asked to decide the degree to which you believe they are similar or dissimilar in meaning, on a scale that has ten values: 0 through 9. We want you to think of zero as completely dissimilar: the terms are not at all alike. Then 9 will represent the case where you believe the terms are very similar, nearly identical or synonyms. For practice, consider the following two examples:

	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
1. red-green		✓										

In this case, you would put a check in the space below zero, since red and green are complementary colors. They are as dissimilar as you can get!

	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
2. gorgeous-beautiful											✓	

In this case, the pair are virtually synonymous, and if you agree, then perhaps we should put out check in the space below the number 9.

	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
3. anger-hatred						✓						

In this case, the answer depends a lot on how you, personally, interpret these two words. Some people assume they are hated when someone is angry with them. Others see anger and hatred as two separate emotions that may or may not be associated. If you see anger and hatred as closely related (and therefore similar), you would mark one of the spaces to the right, toward the 9 end of the scale. If you see the two words as being more dissimilar, you would mark a space to the left, maybe not close to 0, but certainly not close to 9.

If you have any questions about how to answer this questionnaire or mark the scale, now is a good time to do so.

YOUR NAME: ______________________________ DATE: __________

		0	1	2	3	4	5	6	7	8	9	
1. message - input select	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
2. message - filter	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
3. message - queue to channel	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
4. message - detect	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
5. message - search	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
6. message - identify	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
7. message - code	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
8. message - interpret	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
9. message - categorize	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
10. message - transmit	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
11. message - store	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
12. message - short term memory	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
13. message - count	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
14. message - compute	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
15. message - plan	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
16. message - test	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
17. message - control	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar

Item		0	1	2	3	4	5	6	7	8	9	
18. message - edit	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
19. message - display	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
20. message - purge	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
21. message - reset	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
22. message - decide / select	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
23. message - adapt / learn	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
24. message - goal image	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
25. input select - filter	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
26. input select - queue to channel	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
27. input select - detect	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
28. input select - search	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
29. input select - identify	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
30. input select - code	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
31. input select - interpret	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
32. input select - categorize	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
33. input select - transmit	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar

Item	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
34. input select - store		__	__	__	__	__	__	__	__	__	__	
35. input select - short term memory		__	__	__	__	__	__	__	__	__	__	
36. input select - count		__	__	__	__	__	__	__	__	__	__	
37. input select - compute		__	__	__	__	__	__	__	__	__	__	
38. input select - plan		__	__	__	__	__	__	__	__	__	__	
39. input select - test		__	__	__	__	__	__	__	__	__	__	
40. input select - control		__	__	__	__	__	__	__	__	__	__	
41. input select - edit		__	__	__	__	__	__	__	__	__	__	
42. input select - display		__	__	__	__	__	__	__	__	__	__	
43. input select - purge		__	__	__	__	__	__	__	__	__	__	
44. input select - reset		__	__	__	__	__	__	__	__	__	__	
45. input select - decide/select		__	__	__	__	__	__	__	__	__	__	
46. input select - adapt/learn		__	__	__	__	__	__	__	__	__	__	
47. input select - goal image		__	__	__	__	__	__	__	__	__	__	
48. filter - queue to channel		__	__	__	__	__	__	__	__	__	__	
49. filter - detect		__	__	__	__	__	__	__	__	__	__	

	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
50. filter - search		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
51. filter - identify		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
52. filter - code		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
53. filter - interpret		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
54. filter - categorize		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
55. filter - transmit		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
56. filter - store		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
57. filter - short term memory		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
58. filter - count		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
59. filter - compute		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
60. filter - plan		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
61. filter - test		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
62. filter - control		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
63. filter - edit		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
64. filter - display		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
65. filter - purge		__	__	__	__	__	__	__	__	__	__	

Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
66. filter - reset _ _ _ _ _ _ _ _ _ _

Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
67. filter - decide/select _ _ _ _ _ _ _ _ _ _

Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
68. filter - adapt/learn _ _ _ _ _ _ _ _ _ _

Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
69. filter - goal image _ _ _ _ _ _ _ _ _ _

Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
70. queue to channel - detect _ _ _ _ _ _ _ _ _ _

Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
71. queue to channel - search _ _ _ _ _ _ _ _ _ _

Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
72. queue to channel - identify _ _ _ _ _ _ _ _ _ _

Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
73. queue to channel - code _ _ _ _ _ _ _ _ _ _

Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
74. queue to channel - interpret _ _ _ _ _ _ _ _ _ _

Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
75. queue to channel - categorize _ _ _ _ _ _ _ _ _ _

Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
76. queue to channel - transmit _ _ _ _ _ _ _ _ _ _

Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
77. queue to channel - store _ _ _ _ _ _ _ _ _ _

Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
78. queue to channel - short term memory _ _ _ _ _ _ _ _ _ _

Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
79. queue to channel - count _ _ _ _ _ _ _ _ _ _

Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
80. queue to channel - compute _ _ _ _ _ _ _ _ _ _

Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
81. queue to channel - plan _ _ _ _ _ _ _ _ _ _

Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
82. queue to channel - test _ _ _ _ _ _ _ _ _ _

	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
83. queue to channel - test		_	_	_	_	_	_	_	_	_	_	
84. queue to channel - control		_	_	_	_	_	_	_	_	_	_	
85. queue to channel - edit		_	_	_	_	_	_	_	_	_	_	
86. queue to channel - display		_	_	_	_	_	_	_	_	_	_	
87. queue to channel - display		_	_	_	_	_	_	_	_	_	_	
88. queue to channel - purge		_	_	_	_	_	_	_	_	_	_	
89. queue to channel - reset		_	_	_	_	_	_	_	_	_	_	
90. queue to channel - decide/select		_	_	_	_	_	_	_	_	_	_	
91. queue to channel - adapt/learn		_	_	_	_	_	_	_	_	_	_	
93. detect - search		_	_	_	_	_	_	_	_	_	_	
94. detect - identify		_	_	_	_	_	_	_	_	_	_	
95. detect - code		_	_	_	_	_	_	_	_	_	_	
96. detect - interpret		_	_	_	_	_	_	_	_	_	_	
97. detect - categorize		_	_	_	_	_	_	_	_	_	_	
98. detect - transmit		_	_	_	_	_	_	_	_	_	_	
99. detect - store		_	_	_	_	_	_	_	_	_	_	
100. detect - store		_	_	_	_	_	_	_	_	_	_	

	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
101. detect - short term memory		__	__	__	__	__	__	__	__	__	__	
102. detect - count		__	__	__	__	__	__	__	__	__	__	
103. detect - compute		__	__	__	__	__	__	__	__	__	__	
104. detect - plan		__	__	__	__	__	__	__	__	__	__	
105. detect - test		__	__	__	__	__	__	__	__	__	__	
106. detect - control		__	__	__	__	__	__	__	__	__	__	
107. detect - edit		__	__	__	__	__	__	__	__	__	__	
108. detect - display		__	__	__	__	__	__	__	__	__	__	
109. detect - purge		__	__	__	__	__	__	__	__	__	__	
110. detect - reset		__	__	__	__	__	__	__	__	__	__	
111. detect - decide/select		__	__	__	__	__	__	__	__	__	__	
112. detect - adapt/learn		__	__	__	__	__	__	__	__	__	__	
113. detect - goal image		__	__	__	__	__	__	__	__	__	__	
114. search - identify		__	__	__	__	__	__	__	__	__	__	
115. search - code		__	__	__	__	__	__	__	__	__	__	
116. search - interpret		__	__	__	__	__	__	__	__	__	__	
117. search - categorize		__	__	__	__	__	__	__	__	__	__	

	Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
118. search - transmit	_ _ _ _ _ _ _ _ _ _
	Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
119. search - store	_ _ _ _ _ _ _ _ _ _
	Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
120. search - short term memory	_ _ _ _ _ _ _ _ _ _
	Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
121. search - count	_ _ _ _ _ _ _ _ _ _
	Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
122. search - compute	_ _ _ _ _ _ _ _ _ _
	Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
123. search - plan	_ _ _ _ _ _ _ _ _ _
	Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
124. search - test	_ _ _ _ _ _ _ _ _ _
	Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
125. search - control	_ _ _ _ _ _ _ _ _ _
	Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
125. search - edit	_ _ _ _ _ _ _ _ _ _
	Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
126. search - display	_ _ _ _ _ _ _ _ _ _
	Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
127. search - purge	_ _ _ _ _ _ _ _ _ _
	Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
128. search - reset	_ _ _ _ _ _ _ _ _ _
	Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
129. search - decide/select	_ _ _ _ _ _ _ _ _ _
	Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
130. search - adapt/learn	_ _ _ _ _ _ _ _ _ _
	Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
131. search - goal image	_ _ _ _ _ _ _ _ _ _
	Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
132. identify - code	_ _ _ _ _ _ _ _ _ _
	Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
133. identify - interpret	_ _ _ _ _ _ _ _ _ _

Item		0	1	2	3	4	5	6	7	8	9	
134. identify - categorize	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
135. identify - transmit	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
136. identify - store	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
137. identify - short term memory	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
138. identify - count	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
139. identify - compute	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
140. identify - plan	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
141. identify - test	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
142. identify - control	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
143. identify - edit	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
144. identify - display	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
145. identify - purge	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
146. identify - reset	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
147. identify - decide/select	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
148. identify - adapt/learn	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
149. identify - goal image	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
150. code - interpret	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar

151. code - categorize — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar

152. code - transmit — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar

153. code - store — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar

154. code - short term memory — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar

155. code - count — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar

156. code - compute — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar

157. code - plan — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar

158. code - test — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar

159. code - control — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar

160. code - edit — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar

161. code - display — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar

162. code - purge — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar

163. code - reset — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar

164. code - decide/select — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar

165. code - adapt/learn — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar

166. code - goal image — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar

	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
167. interpret - categorize		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
168. interpret - transmit		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
169. interpret - store		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
170. interpret - short term memory		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
171. interpret - count		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
171. interpret - compute		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
172. interpret - plan		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
173. interpret - test		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
174. interpret - control		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
175. interpret - edit		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
176. interpret - display		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
177. interpret - purge		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
178. interpret - reset		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
179. interpret - decide/select		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
180. interpret - adapt/learn		__	__	__	__	__	__	__	__	__	__	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
180. interpret - adapt/learn		__	__	__	__	__	__	__	__	__	__	

Item		0	1	2	3	4	5	6	7	8	9	
181. interpret - goal image	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
182. categorize - transmit	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
183. categorize - store	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
184. categorize - short term memory	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
185. categorize - count	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
186. categorize - compute	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
187. categorize - plan	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
187. categorize - plan	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
188. categorize - test	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
189. categorize - control	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
190. categorize - edit	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
191. categorize - display	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
192. categorize - purge	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
193. categorize - reset	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
194. categorize - decide/select	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
195. categorize - adapt/learn	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar

	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
196. categorize - goal image		__	__	__	__	__	__	__	__	__	__	
197. transmit - store		__	__	__	__	__	__	__	__	__	__	
198. transmit - short term memory		__	__	__	__	__	__	__	__	__	__	
199. transmit - count		__	__	__	__	__	__	__	__	__	__	
200. transmit - compute		__	__	__	__	__	__	__	__	__	__	
201. transmit - plan		__	__	__	__	__	__	__	__	__	__	
202. transmit - test		__	__	__	__	__	__	__	__	__	__	
203. transmit - control		__	__	__	__	__	__	__	__	__	__	
204. transmit - edit		__	__	__	__	__	__	__	__	__	__	
205. transmit - display		__	__	__	__	__	__	__	__	__	__	
206. transmit - purge		__	__	__	__	__	__	__	__	__	__	
207. transmit - reset		__	__	__	__	__	__	__	__	__	__	
208. transmit - decide/select		__	__	__	__	__	__	__	__	__	__	
209. transmit - adapt/learn		__	__	__	__	__	__	__	__	__	__	
210. transmit - goal image		__	__	__	__	__	__	__	__	__	__	
211. store - short term memory		__	__	__	__	__	__	__	__	__	__	

		0	1	2	3	4	5	6	7	8	9	
212. store - count	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
213. store - compute	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
214. store - plan	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
215. store - test	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
216. store - control	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
217. store - edit	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
218. store - display	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
219. store - purge	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
220. store - reset	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
221. store - decide/select	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
222. store - adapt/learn	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
223. store - goal image	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
224. short term memory - count	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
225. short term memory - compute	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
226. short term memory - plan	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar
227. short term memory - test	Dissimilar	___	___	___	___	___	___	___	___	___	___	Similar

		0	1	2	3	4	5	6	7	8	9	
228. short term memory - control	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
229. short term memory - edit	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
230. short term memory - display	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
231. short term memory - purge	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
232. short term memory - reset	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
233. short term memory - decide/select	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
234. short term memory - adapt/learn	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
235. short term memory - goal image	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
236. count - compute	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
237. count - plan	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
238. count - test	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
239. count - control	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
240. count - edit	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
241. count - display	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
242. count - purge	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
243. count - reset	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar

	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
244. count - decide/select		___	___	___	___	___	___	___	___	___	___	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
245. count - adapt/learn		___	___	___	___	___	___	___	___	___	___	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
246. count - goal image		___	___	___	___	___	___	___	___	___	___	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
247. compute - plan		___	___	___	___	___	___	___	___	___	___	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
248. compute - test		___	___	___	___	___	___	___	___	___	___	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
249. compute - control		___	___	___	___	___	___	___	___	___	___	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
250. compute - edit		___	___	___	___	___	___	___	___	___	___	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
251. compute - display		___	___	___	___	___	___	___	___	___	___	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
252. compute - purge		___	___	___	___	___	___	___	___	___	___	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
253. compute - reset		___	___	___	___	___	___	___	___	___	___	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
254. compute - decide/select		___	___	___	___	___	___	___	___	___	___	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
255. compute - adapt/learn		___	___	___	___	___	___	___	___	___	___	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
256. compute - goal image		___	___	___	___	___	___	___	___	___	___	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
257. plan - test		___	___	___	___	___	___	___	___	___	___	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
258. plan - control		___	___	___	___	___	___	___	___	___	___	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
259. plan - edit		___	___	___	___	___	___	___	___	___	___	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
260. plan - display		___	___	___	___	___	___	___	___	___	___	

	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
261. plan - purge		_	_	_	_	_	_	_	_	_	_	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
262. plan - reset		_	_	_	_	_	_	_	_	_	_	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
263. plan - decide/select		_	_	_	_	_	_	_	_	_	_	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
264. plan - adapt/learn		_	_	_	_	_	_	_	_	_	_	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
265. plan - goal image		_	_	_	_	_	_	_	_	_	_	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
266. test - control		_	_	_	_	_	_	_	_	_	_	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
267. test - edit		_	_	_	_	_	_	_	_	_	_	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
268. test - display		_	_	_	_	_	_	_	_	_	_	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
269. test - purge		_	_	_	_	_	_	_	_	_	_	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
270. test - reset		_	_	_	_	_	_	_	_	_	_	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
271. test - decide/select		_	_	_	_	_	_	_	_	_	_	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
272. test - adapt/learn		_	_	_	_	_	_	_	_	_	_	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
273. test - goal image		_	_	_	_	_	_	_	_	_	_	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
274. control - edit		_	_	_	_	_	_	_	_	_	_	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
275. control - display		_	_	_	_	_	_	_	_	_	_	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
276. control - purge		_	_	_	_	_	_	_	_	_	_	
	Dissimilar	0	1	2	3	4	5	6	7	8	9	Similar
277. control - reset		_	_	_	_	_	_	_	_	_	_	

Item		0	1	2	3	4	5	6	7	8	9	
278. control - decide/select	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
279. control - adapt/learn	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
280. control - goal image	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
281. edit - display	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
282. edit - purge	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
283. edit - reset	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
284. edit - decide/select	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
285. edit - adapt/learn	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
286. edit - goal image	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
287. display - purge	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
288. display - reset	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
289. display - decide/select	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
290. display - adapt/learn	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
291. display - goal image	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
292. purge - reset	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
293. purge - decide/select	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar
294. purge - adapt/learn	Dissimilar	__	__	__	__	__	__	__	__	__	__	Similar

295. purge - goal image — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
_ _ _ _ _ _ _ _ _ _

296. reset - decide/select — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
_ _ _ _ _ _ _ _ _ _

297. reset - adapt/learn — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
_ _ _ _ _ _ _ _ _ _

298. reset - goal image — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
_ _ _ _ _ _ _ _ _ _

299. decide/select - adapt/learn — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
_ _ _ _ _ _ _ _ _ _

300. decide/select - goal image — Dissimilar 0 1 2 3 4 5 6 7 8 9 Similar
_ _ _ _ _ _ _ _ _ _

Appendix B
Task Analysis Instructions and Examples

The task analysis instructions provided here were given to all study participants once they said they understood the basic operation of TRACON II well enough to proceed. The course handout they received for Aviation 540 had a copy of material from DeGreene's (1970) text on the steps for doing a task analysis. The following material provided more detailed instructions on "how to ..." In the second week of class, all students are required to prepared a concept map for how they can earn a good grade for the course. That homework assignment is graded and returned. So students should be familiar with the basic notion of a concept map. They are also exposed to some elementary statistics and are asked to compute the z-score for their height and weight, based upon data collected from students attending the class. So again, the computation is not completely foreign to them by the time they get this set of examples for four formats for doing task analyses.

TASKS: WHAT ARE THEY AND HOW CAN THEY BE REPRESENTED?

Tasks are sort of like ghosts. Nobody really believes they exist, do they? Yet everybody talks about them and has their own concept of what the term means. The problem is that it is impossible to hold one in your hand or point to one so we can all publicly examine the properties of this thing called a task. In the next chapter, we will begin our own set of definitions, but for now, we examine some that are already in the literature.

Task Definitions

If a task is a set of behaviors, then the set is not well-defined. No one has produced an exhaustive list of behaviors that are accepted as defining the set of all tasks. Nor has anyone identified the attributes of tasks in a way that will allow you and I to generate the set of all tasks and arrive at the same end result. Whatever tasks may be, they do not appear to be a well-defined set. We can allow that they may be a fuzzy set, but we must still deal with "What does that mean?"

It is difficult to find a definition of "a task" that you like and that everyone else will agree applies to everything they consider a task. The following are some attempts by various authors:

R. B. Miller (1953) --- "a group of discriminations, decisions and effector activities related to each other by temporal proximity, immediate purpose and a common man-machine output." (Cited in Companion and Corso (1982), p. 461, and in Meister (1976), p. 96)

R. B. Miller (1962) --- "A Task is any set of activities, occurring at the same time, sharing some common purpose that is recognized by the task performer." (p. 11)

R. B. Miller (1973) --- "A task consists of a series of goal-directed transactions controlled by one or more 'programs' that guide the operations by a human operator of a prescribed set of tools through a set of completely or partially predicted environmental states." (p. 11)

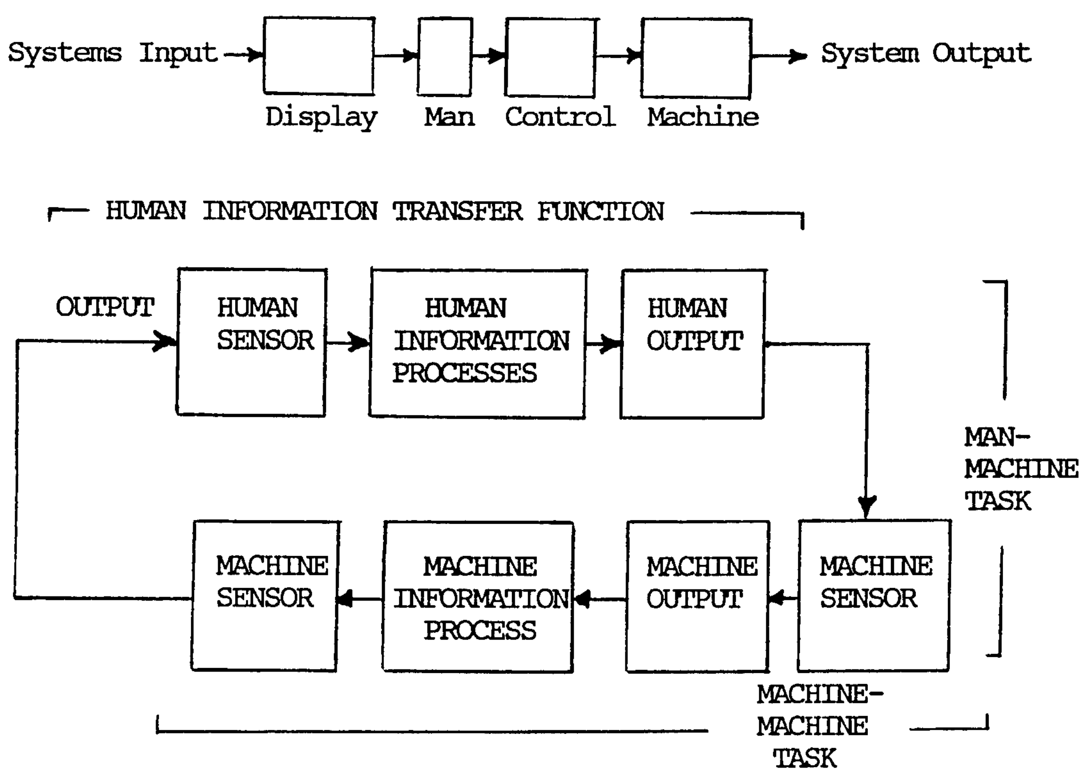
Thibaud and Kelly (1950) --- "problem, assignment, or stimulus-complex to which the individual or group responds by performing various overt and covert operations." (p. 50)

J. R. Hackman (1968) --- "A task is assigned to a person (or group) by an external agent or is self generated and consists of a stimulus complex and a set of instructions which specify what is to be done." (p. 12)

Farina and Wheaton (1973) --- what transpires between input and eventual output: "...a complex situation capable of eliciting goal-directed performance from an operator." (see Fleishman and Quaintance (1984), p. 356)

Levine and Teichner (1973) --- "...a transfer of information between components, the operation on the information within a component is a process. Processes, which appear at a more general level of systems analysis, may be subdivided into tasks." (see Fleishman and Quaintance (1984), p. 244)..."When Figure 1-A is analyzed into its subsystems, as in figure 1-B, what was a process at the more general descriptive level becomes a task. That is, there are now new transfers of information between components which did not exist in figure 1-A. Clearly, a process is carried out as a subtask." (Levine and Teichner, 1973, pp. 4-5; see our figure 1)

Figure 1. Levine and Teichner's Man-Machine System at Two Levels of Description



We wish to note in passing, that Levine and Teichner (1973)also comment that, "Although the psychologist is not concerned with the machine-machine tasks, it is important to note that such tasks exist. That is, the notion of a task is not one which necessarily involves people. This is extremely important and usually ignored. (p. 5)"

The following definitions illustrate how the concept of a task has been viewed outside of psychology: in ergonomics, computer science, and industrial engineering applications.

Davis (1983) --- "A task is composed of a number of sub-goals, each of which must be successfully completed to attain the overall or super-ordinate goal. Subgoals are a more useful technique for describing tasks than the actual methods by which they are accomplished since they are less subject to change and are more user-oriented."

Bylander and Chandrasekaran (1987) --- generic tasks---basic combinations of knowledge structures and inference strategies that are powerful for dealing for certain kinds of problems. The generic tasks provide a vocabulary for describing problems, as well as for designing knowledge-based systems that perform them.

Bradford (1990) --- "High level tasks relate to user activities that may span one or more computer sessions...Mid level tasks are subgoals of high level tasks...Low level tasks represent a decomposition of a mid level task into a sequence of simple actions that are accomplished through the commands of a command language."

A Working Definition for Our Purposes

As you do your own analysis of TRACON tasks, think in terms of what you must do to get the job done. You only have to analyze the approach control problem. You may ignore departure control operations.

The criterion measure is clearly aircraft separation: the bigger, the better! However, the goal is to get everybody down on the ground without undue delays. What you need to describe is what an operator is required to do in order to perform the job accurately, safely, and efficiently. To begin with, assume that nothing is malfunctioning, the weather is fine, and everyone is following directions perfectly. These assumptions will simply your analysis.

Using Teichner's definition as a guide, a task occurs anytime you interact with the equipment: 1) it (the pilot) speaks to you, 2) you speak through it (to a pilot), 3) you look at the display, (or something changing should catch your attention), 4) you enter keyboard commands, etc. The vocabulary we want you to use in describing these actions is the list of 25 terms found in Miller's "Appendix A", material found in the AV 540 Course Handout that you purchased (pp 46-56). So a task will be anything you think needs to be done that can be described using one of Miller's terms. If you do not find a term in that list that you think you need, you may define a new term, but be sure to provide both the name you assign to your new term (the task "title" or label) and its meaning.

Many task analyses try to describe what goes on at more than one level of detail. The first cut tries to look at the "Big Picture" and describe activities in very general terms. At the next level of detail, each of the general activities is then broken down into more specific steps that have to be taken to do the general activity. Sometimes, even these steps need to be broken down further to get to the level of actual keyboard entries being made. How far you go in breaking down the tasks into component elements depends on the detail you think is required (a matter of judgment), and the terms that exist in your task vocabulary --- they sort of imply a particular level of detail.

Some analysts try to build a conceptual model before they define their task model. The conceptual dependency charts described in chapter two of the text are one form of conceptual analysis that some people find useful. You are not required to start there, but if you are stuck getting started, trying to do a concept map may get you thinking about how you can identify specific tasks.

Other analysts seem to jump into the middle of things and just branch off in whatever direction seems appropriate. Then they step back, look at their draft, and decide how they want to edit and restructure it after they have done some preliminary thinking.

There are many different ways you can approach your task analysis. We have suggested only a few of the strategies you might use, so do not think this problem can be done one and only one way. We are interested in learning how you solve the problem: what you tried, how it worked, whether you shifted gears --- abandoning one approach to try out another, etc.

Some Problems You Can Expect

This business of doing a task analysis is not necessarily an easy job. It takes some thought. You can often describe what is going on more than one way. There are no fixed criteria to tell you what option is best, or even whether one option is better than another. As you sense these sorts of problems exist, we want to know about them! Tell us what conceptual difficulties you are running into. Try to describe what it is that is bothering you about the analysis. See if you can identify where the confusion or uncertainty lies, then tell us what you are thinking: think "out Loud" or with a pen in your hand (being applied to paper: write it down!).

While many activities may have to be done in a particular order, some sets of activities might be done in any arbitrary order. Try to identify when each situation occurs. Also, there are times you may be doing one thing (like talking) while you are simultaneously doing something else (like watching the screen). Try to capture these situations as realistically as you can.

Decisions are particularly important and are usually a bit difficult to deal with. Some times it is a question of whether to act or not to act. Other times, the decision involves making choices among a set of mutually exclusive alternatives (ie, only one of the alternatives will be done). And some decisions are a matter of timing: deciding when to do what you plan to do. Certainly other variations of decision making may exist.

There are cases where you have to specify the conditions which dictate that a task should (or should not) be done. These can often be expressed in terms of a conditional "If ... (condition X exists), then...(Do a task Y); otherwise, ...(Do: a. something, or b. nothing).

Refinements

If you think you have a fairly complete task analysis, and you believe it is time to get more realism into the description, then begin to consider what happens under less than ideal conditions: 1) people start making errors, 2) the weather turns sour, or 3) pieces of equipment (on the plane or in the ATC network) begin to malfunction. This tends to complicate the task analysis description of what operators are required to do.

This is a time when you may want to use the TRACON simulation again and exercise some of the options of the SETUP submenu on the dialog box that pops up when you select NEW from the FILE pull-down menu

Task Analysis Examples

The following pages illustrate a task analysis in three different formats to show how the same information might be portrayed in each of the three formats used to document task analyses. We begin with a conceptual dependency map, and then illustrate the three task analysis formats.

You may find it easier to understand or work with one of these three formats as your preferred method. That is perfectly alright. The choice is yours to make. They all achieve the same end. They just approach the problem different ways.

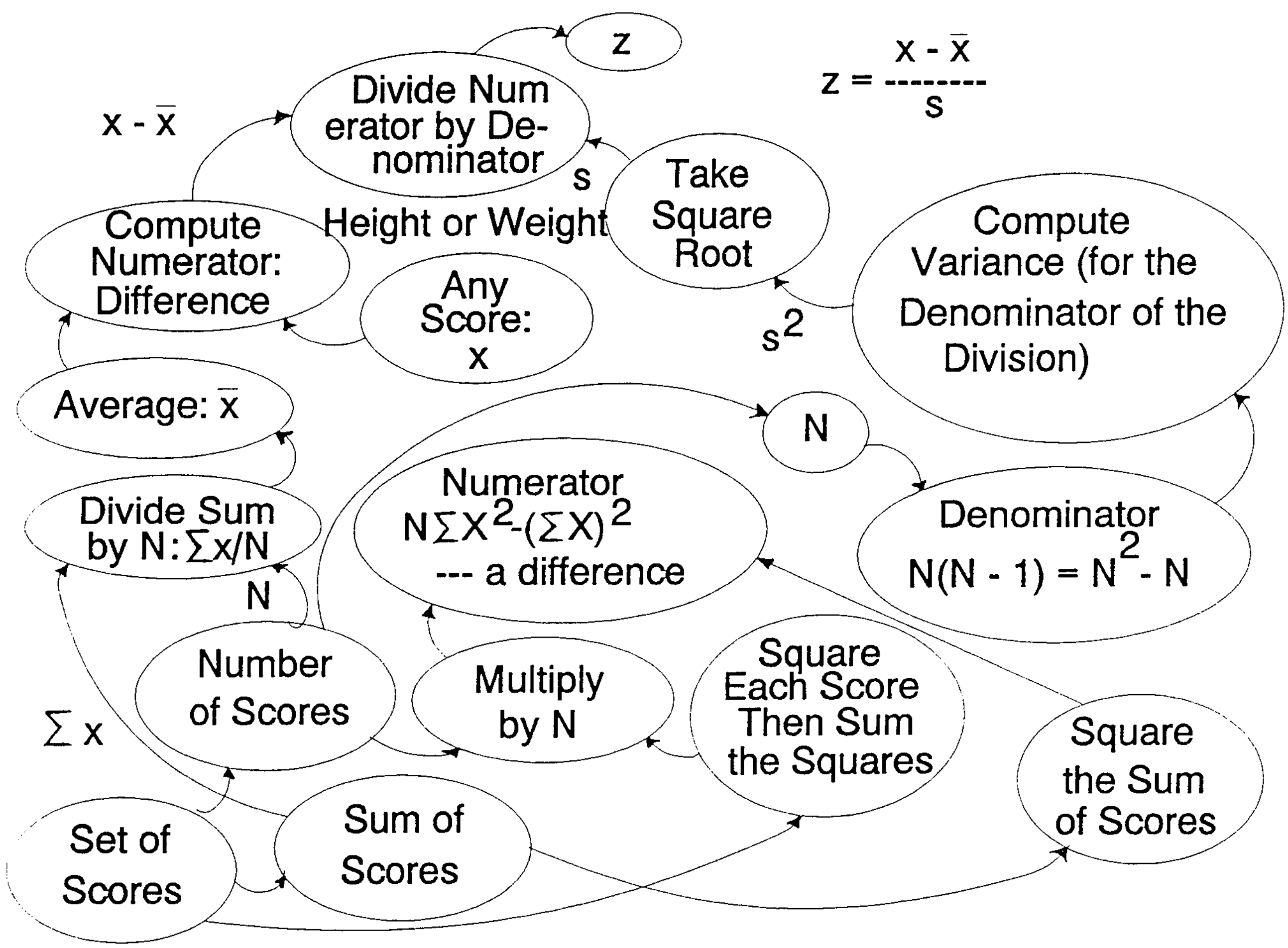
Remember though, that you will later be comparing your work with a graphic form of task analysis. The illustration we provide here is a graphic that is similar, but not identical, to the one you will see later.

The task we are analyzing is the computation of the z statistic, something you did as a homework assignment for AV 540. So you did this task yourself, and you can compare the description with what you remember doing. Please note, that you may have done the computation without thinking about the steps the same way as they are described here. Also, there are many cases where the same operation can be correctly performed more than one way. So a description of the task may be non-unique: there may be several different descriptions, all of which will work.

When that happens, it is often important to later ask, is one description better than another, and if so, why? The process of finding the best method for doing a task is the job of the industrial engineer, and they typically do time and motion studies as a part of methods engineering to define process specifications and work standards. Training also demands that some statement of performance objectives be established in advance, so the instructor knows what skills and proficiency levels are required on the job.

So what you are learning to do here has application in several areas. Your participation in this project is going to give you some special experience that may be of benefit later in your career. We hope you find this exercise interesting, not frustrating, but we do need to know where and why you are experiencing any difficulties, so please keep notes of where problems occur, and ask questions. We do not want you to bogged down in doing this, so if you get "stuck," be sure to get with Jeff Vance or Jerry Chubb and get the help you need to finish the analysis.

Conceptual Dependency Map
For Computing the "z" Statistic



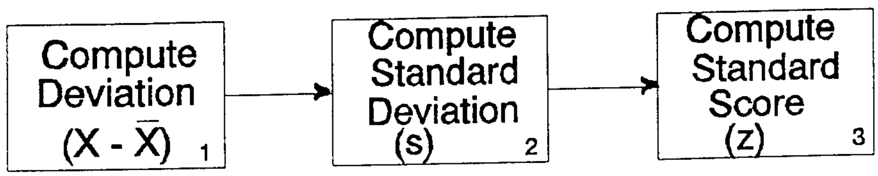
Example 1: Outline Format for a Task Analysis

1. Compute the value for the difference: subtract the mean from your score (ht. or wt.).
    1.1 Compute the mean (eg. average height or average weight).
        1.1.1 Add up all the scores.
        1.1.2 Divide by the number of scores summed (N).
    1.2 Round off to the second decimal digit.

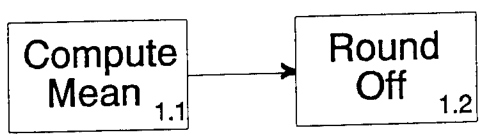
2. Compute the value for the denominator: the sample standard deviation.
    2.1 Compute the variance numerator, as follows:
        2.1.1 Compute the square of each score.
        2.1.2 Sum the squares of the scores computed in step 2.1.
        2.1.3 Multiply the sum of squares from step 2.2 by the number of scores (N).
        2.1.4 Square the sum of scores computed in step 1.1.1.
        2.1.5 Subtract the squared sum of scores from step 2.1.4 from the result of step 2.1.3.
    2.2 Compute the denominator of the variance.
        2.2.1 Subtract one (1) from the number of scores: N-1.
        2.2.2 Multiply the number of scores (N) by the answer computed in step 2.2.1 to get: N(N-1).
    2.3 Divide the numerator by the denominator to get the variance.
    2.4 Compute the square root of the variance to get the standard deviation.

3. Compute the value for z
    3.1 Find the ratio of "difference" vs. "standard deviation" by dividing the numerator (step 1.2) by the denominator (step 2.4).
    3.2 Round off the answer to the second decimal place.

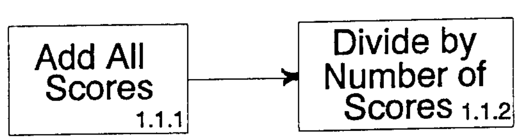
Example 2: Graphic Format for Task Analysis
Hierarchical Input-Process-Output (HIPO) Chart



NOTE: Task No. Appears in Lower Right Corner of Task Block

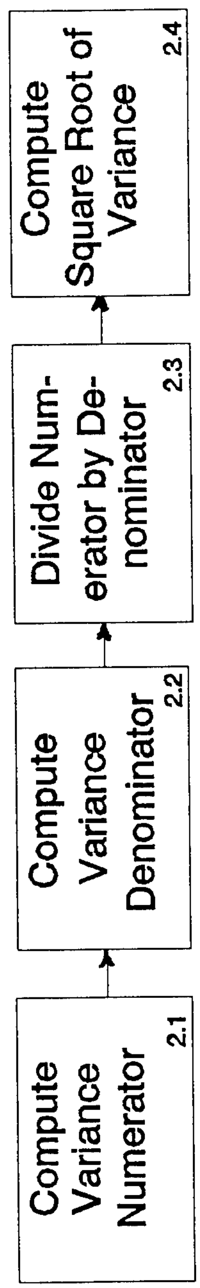


Decomposition of Task 1

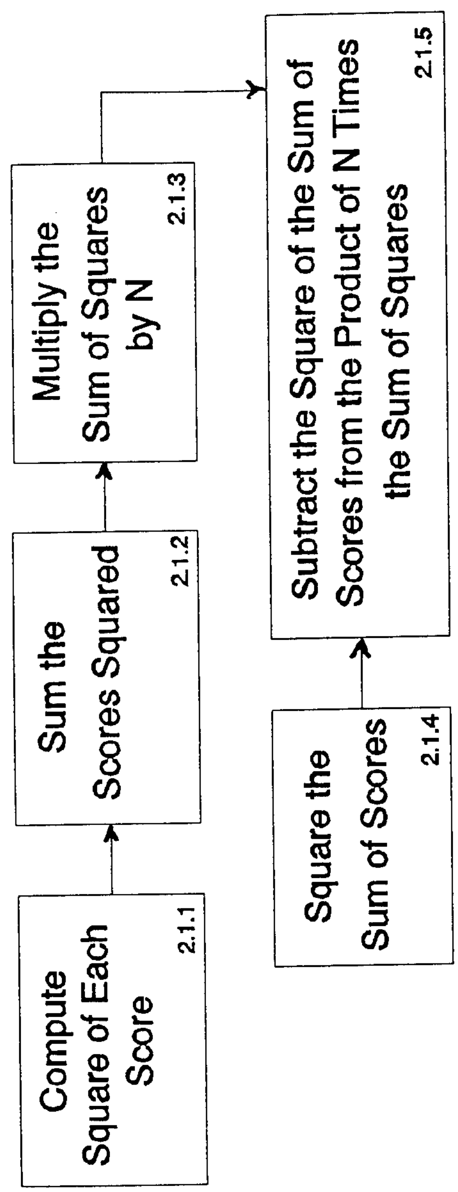


Decomposition of Task 1.1

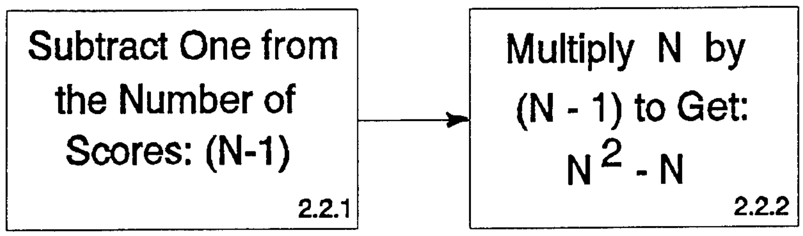
Decomposition of Task 2



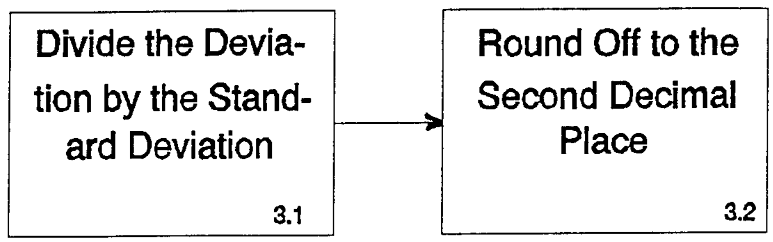
Decomposition of Task 2.1



Decomposition of Task 2.2



Decomposition of Task 3



Example 3: Tabular Format for Task Analysis
Fill In a Separate Table for Each Task

Generic Table Entries (may be organized as columns or as rows):

1. Function (parent of task) Name:

2. Task (parent of subtask) Name:

3. Subtask (this activity's) Name:

4. Action Stimulus (trigger or start event):

5. Required Action (description of what task requires):

6. Feedback (what should happen if action done correctly):

7. Task Classification (who does it normally, especially if more than one agent could be assigned the task, eg. man or machine; pilot or co-pilot; etc.):

8. Potential Errors (ways in which the required action may be done incorrectly):

9. Time (duration of activity: normally and maximum):
   a. Necessary: (otherwise: minimal, ordinary, normal, or nominal duration)
   b. Allowable: (otherwise: available, required, standard, or maximum time)

10. Workstation (when there might be more than one):

11. Skill Level (eg. unskilled/untrained, apprentice/novice, journeyman/qualified, master/experienced, expert/authority):

NOTE: for our analysis you may make the following modifications:

a. consider function as your top level task: #.
b. consider task as your next level of detail in the hierarchy: #.#
c. consider subtask as the next level of detail: #.#.#
d. if you need more levels of detail, add them by moving everything up a level: the subtask becomes a task, and the task becomes a function.
e. The required action will be the amplified explanation for your subtask name, describing the nature of the activity.
f. Feedback is typically the voice response and message you get to your keyboard entry; you may abbreviate this as: "Echoed command."
g. Task classification is irrelevant here.
h. You do need to consider what errors might occur!
i. Ignore making any estimates of time.
j. Workstation and skill level may be ignored also.

Template for Tabular Task Analysis

Function: Compute z score Task: 1. Compute deviation (difference)
Subtask: 1.1 Compute mean Action Stimulus: Provided set of scores
Required Response: computed average: quotient and dividend
Feedback: the computed number Potential Errors: wrong sum or improper division

Function: Compute z score Task: 1. Compute deviation (difference)
Subtask: 1.2 Round off to 2nd decimal digit Action Stimulus: Computed difference
Required Response: Round up/down
Feedback: Computed value Potential Errors: Too big/small

Function: Compute deviation Task: 1.1 Compute mean
Subtask: 1.1 Add up all scores Action Stimulus: Given set of scores
Required Response: Total
Feedback: Calculate sum Potential Errors: Transpose digits, miss a number, enter wrong digit, enter number twice, hit wrong operator key, or combinations of such errors.

Function: Compute z score Task: 2. Compute value for denominator
Subtask: 2.1 Compute variance's numerator Action Stimulus: Given set of scores
Required Response: Computed difference
Feedback: Calculated number must be positive Potential Errors: Difference computed wrong, first, second, or both components computed wrong.

Function: Compute z score Task: Compute value for denominator
Subtask: 2.2 Compute variance's denominator Action Stimulus: Number of given scores
Required Response: (N**2-N)
Feedback: Calculated value Potential Errors: Subtraction or multiplication errors

Function: Compute z score Task: Compute value of denominator
Subtask: 2.3 Compute Variance Action Stimulus: Num. and Denom. values
Required Response: Results of the division
Feedback: Computed value Potential Errors: Division error

Function: Compute z score Task: Compute value of denominator
Subtask: 2.4 Take square root of the variance Action Stimulus: Variance
Required Response: Calculate, look up, or derive the square root
Feedback: Calculated value when squared equals var. Potential Errors: Method dependent: derivation, look-up, or calculation errors possible

<u>Function</u>: Compute value for denominator <u>Task</u>: 2.1 Compute variance numerator
<u>Subtask</u>: 2.1.1 Compute the square of each score <u>Action Stimulus</u>: Given set of scores
<u>Required Response</u>: Scores squared
<u>Feedback</u>: Set of computed values <u>Potential Errors</u>: Entry or computation errors

<u>Function</u>: Compute value for denominator <u>Task</u>: 2.1 Compute variance numerator
<u>Subtask</u>: 2.1.2 Sum squared scores <u>Action Stimulus</u>: Squares of scores
<u>Required Response</u>: Total of squared scores
<u>Feedback</u>: Calculated sum of squares <u>Potential Errors</u>: Entry or calculation errors

<u>Function</u>: Compute value for denominator <u>Task</u>: 2.1 Compute variance numerator
<u>Subtask</u>: 2.1.3 Multiply sum of squares by N <u>Action Stimulus</u>: Sum of squares
<u>Required Response</u>: Product
<u>Feedback</u>: Calculated value <u>Potential Errors</u>: entry or multiplication error

<u>Function</u>: Compute value for denominator <u>Task</u>: 2.1 Compute variance numerator
<u>Subtask</u>: 2.1.4 Square sum of scores <u>Action Stimulus</u>: Given set of scores
<u>Required Response</u>: Product
<u>Feedback</u>: Calculated value <u>Potential Errors</u>: Entry or mult. error

<u>Function</u>: Compute value for denominator <u>Task</u>: 2.1 Compute variance numerator
<u>Subtask</u>: 2.1.5 Subtract squared sum of scores from product of N times sum of scores squared <u>Action Stimulus</u>: Intermediate components from 2.1.3 & 2.1.4
<u>Required Response</u>: Difference: 2.1.3 less 2.1.4
<u>Feedback</u>: Calculated value <u>Potential Errors</u>: Entry & subtrctn errors

<u>Function</u>: Compute value for denominator <u>Task</u>: 2.2 Compute variance denominator
<u>Subtask</u>: 2.2.1 Subtract 1 from N <u>Action Stimulus</u>: Number of given scores (N)
<u>Required Response</u>: Difference
<u>Feedback</u>: Calculated value <u>Potential Errors</u>: Subtraction & entry errors

<u>Function</u>: Compute value for denominator <u>Task</u>: 2.2 Compute variance denominator
<u>Subtask</u>: 2.2.2 Multiply: N(N-1) <u>Action Stimulus</u>: Calculated difference
<u>Required Response</u>: Product
<u>Feedback</u>: Calculated value <u>Potential Errors</u>: Entry and Mult. errors

<u>Function</u>: Compute z score <u>Task</u>: 3. Calculate standard score
<u>Subtask</u>: 3.1 Divide deviation by standard deviation <u>Action Stimulus</u>: Deviation and Std. Dev.
<u>Required Response</u>: Computed ratio
<u>Feedback</u>: Calculated value <u>Potential Errors</u>: Entry & division errors

<u>Function</u>: Compute z score <u>Task</u>: 3. Calculate standard score
<u>Subtask</u>: Round off to second <u>Action Stimulus</u>: Computed ratio

decimal digit
Required Response: Plus or minus x.xx
Feedback: Calculated value for z Potential Errors: Too big/small

Appendix C
Task Analysis Comparison Instructions

The instructions students were given for comparing their work to that done by CTA follows. The reports they prepared are not included. The library's records show that not a single student examined the actual task analysis document.

Instructions for Comparing Your Task Analysis with the Formal FAA TRACON Task Analysis

Now that you have turned in your own analysis of TRACON Tasks, you must complete the second half of the assignment. Here you will compare your analysis with one that the FAA has done. This will allow you to see how well your own work compares with that of others. Remember, you did a task analysis of a "game", while their analysis is of actual TRACON operations: the real thing. You will probably discover that your description is not as detailed as their's, and each analysis has both strong and weak points. In the first part of these instructions, we discuss where you can see the whole task analysis document. We have reproduced a portion of that document for your use. We believe everything you need is in this instruction package, but if you disagree or would feel better seeing the whole document, it is on Closed Reserve in the Caldwell Hall Engineering Library.

Comparing Your TRACON Task Analysis with the FAA's Version

The formal task analysis of TRACON activities was conducted by Computer Technology Associates (CTA) for the FAA and documented in the following report: Alexander, J. R., H. L Ammerman, W. S. Fairhurst, C. M. Hostetler, and G. W. Jones, FAA Air Traffic Control Operations Concepts, Volume VIII: TRACON Controllers, 8 September 1989, DOT/FAA/AP-87-01, U. S. Department of Transportation, Federal Aviation Administration, Washington, DC.

To do the comparison, it is essential that you take the time to study carefully the CTA task analysis material. That is not simple or easy, it takes some effort. If you have problems, see Jeff Vance or Jerry Chubb and get help. Pages you will need for your comparison are included in this present handout, along with instructions for how you are to report the results of your comparison. You are encouraged to scan the entire FAA document that has been placed on Closed Reserve; but you are not expected to read it cover to cover. The part we will be using is the material in Appendix A. This is a graphical form of a task analysis, and the symbols they use are explained on page A-2 of the document, and are included here. Be sure you understand how to read the diagrams! Ask the Jeff Vance or Jerry Chubb for assistance if needed. Other parts of the document may also be of interest to you for your comparison, but we have not reproduced that material. We believe the Appendix A materials are sufficient.

Further, we will restrict our attention to arrival control. This means that not everything in Appendix A is needed for our comparison. Refer to page A-4 of the CTA/FAA Task Analysis document (included later in this handout). This diagram is a top-level overview of the TRACON operation as they see it. Look it over and begin asking yourself, "How does that compare with my description?"

For comparison to your own work, temporarily ignore all tasks starting with the code A1.4 (we have chosen to skip pages A-5 through A-13 of the CTA/FAA document). Next, note that page A-14 is a decomposition (more detailed version) of block A1.1 on page A-4. Again, examine how it compares with your description? Now notice that pages A-15 through A-18 are yet a further decomposition of block A1.1.1 on page A-14. Look for similar decompositions in the material which follows. For example, page A-30 is the decomposition for block A1.2 on page A-4, and the following pages further decompose the task blocks presented on page A-30 (we believe that only pages A-31 through A-36 will apply to the comparison we are asking you to make here).

We also skip pages A-21 through A-29. Pages A-48 and A-49 have the decomposition of block A1.3 on page A-4, and block A1.3.4 should be of particular interest to you for comparing their work with your own. Its decomposition is presented on pages A-62 through A-69, and we also believe pages A-50 through A-52 are pertinent to your comparison effort. Compare all this material to your own.

We now return to consider portions of the decomposition of block A1.4 on page A-4. The overall decomposition of that block may be found on page A-74 and A-75. For our analysis, concentrate on blocks A1.4.2, A1.4.6, A1.4.7, A1.4.13, and A1.4.14. You may ignore the rest. The decomposition of A1.4.2 appears on pages A-81 through A-84. The decomposition of A1.4.6 and A1.4.7 are treated on pages A-95 through A-101. The decomposition of A1.4.13 and A1.4.14 appear on pages A-109 through A-117. For this analysis, we will also ignore the decompositions of blocks A1.5 and A1.6 on page A-4.

Consequently, there are only a limited number of pages you need to review and understand to compare your work to theirs. The instructions for how you are to perform this comparison follow. The format and content of the document you are expected to prepare is also specified.

<u>Making the Comparison of Your Work and Theirs</u>

There are several areas where comparisons can be made: 1) the partitioning or hierarchical decomposition of the activities (how the job was broken down into smaller pieces, and how those pieces were further broken down), 2) the labeling of the activities themselves (Miller's 25 Task functions, terms you used, and terms used by CTA/FAA), 3) the ordering of the tasks, 4) the logic of that ordering (for example, the and/or branching they used which may not have been a part of your analysis), and 5) other considerations you included that were not part of their Appendix A material (although they may appear in other sections of the overall CTA/FAA report).

A series of questionnaires follows that ask you specific questions about your analysis compared to theirs. The first is a repeat of the questionnaire you have already completed. The second asks you about the TRACON II simulation, our task analysis instructions, and problems you may have encountered. Read over the questions before you try answering any of them. If the questions appear unclear, then first get a clarification from Jeff Vance or Jerry Chubb before proceeding.

Room is provided on the questionnaires for you to make notes or comments. Please use this space to advantage. Ask yourself questions about the comparison you are trying to make. State any assumptions you think you are making to complete this assignment. If you have to guess at the meaning of some term CTA used in order to compare it to your analysis, explain what you thought they meant. Tell us other interpretations that occurred to you, if you had to decide which one meaning was better than the alternative interpretations you considered.

Report Format and Content

TITLE: AV 540 Project C --- TRACON II Task Analysis Compared to CTA's

NAME: (yours)

Length is not as important as completeness. We want you to address each of the topics in the following outline. Elaborate enough that we can understand and interpret what you discovered in your task analysis and the comparison of it with the CTA version. For each item in the outline, we first identify what we want from you, and then in the box surrounded by asterisks, we try to give you a short explanation. Read this over, and if you have questions, get them resolved by asking Jeff Vance or Jerry Chubb whatever questions you need answered for clarification.

I. Assumptions

--- Concerns you had about purpose our objective here
--- Definitions of terms you use to describe things
--- Interpretations you made of the CTA materials

****
* Things you felt you had to assume in order to make the comparison. This *
* might include your concerns about what you thought was the purpose or ob- *
* jective of this comparison. It can also include your definition of terms.*
* It might also include your interpretation of certain things CTA included *
* in their analysis that you think are ambiguous: that could be read or *
* interpreted more than one way: tell us the alternative meanings that *
* occurred to you and why you chose the one you did. Also, this is the *
* place to tell us anything else you think we should know as we interpret *
* and evaluate your comparison with comparisons made by other Project C *
* participants. *
****

II. Partitioning of Activities

--- How many levels did they use and how many did you use?
--- How many tasks did they identify at the first or top level and how many did you define at the top level?

****
* The CTA analysis had several levels of task decomposition, where a task on *
* one level is broken down and described in greater detail on the next level,*
* and then those tasks are broken down further on the next level. Compare *
* how detailed they got to how detailed you got in your respective analyses. *
****

III. Labeling of Activities

--- How many of Miller's terms did you use?
--- How many of your own terms did you define and use?
--- How many of CTA's terms could be replaced with terms from the Miller list?
--- How easy is it to compare the terms you used with the ones they used?
--- Would a standard set of terms help you make the comparison?

Activity or task labels are the names assigned: the action verbs used to characterize what is being done. The total description may include other words, but the central issue is naming the action taking place. Miller's terms are but one attempt to achieve the objective of providing a standard set of labels to choose from. No one has a list everyone else seems satisfied using, but when different people use different terms, it can make comparisons more difficult. Give us your thoughts on this issue.

IV. Ordering

--- Did the activity or task sequence they used correspond to your ideas about the order in which things ought to be done?
--- If your ordering was different from theirs, which sequence do you think better describes the nature of the work to be accomplished?
--- Are there cases where order does not matter (any sequence is arbitrary)? (CITE AN EXAMPLE OR TWO)
--- Are there cases where the sequence would be "in error" if things were done in the wrong order? (CITE AN EXAMPLE OR TWO)

There are situations where a set of tasks may occur in order, one after another. Sometimes this order is not important; it is just the sequence that was used when we observed this person on this occasion. On some other occasion, even the same person might do things in a different order. The string of tasks may be as short as two activities or as long as the entire operation being described. Sometimes there are logical reasons why things must be done in some particular order. For example, you must start the aircraft engine before you can taxi to the runway. Also, some tasks must be done in order or an error is made. If you retract the landing gear before you lift off, the consequences are not too pleasant!

V. Branching

--- Did they include decisions that you did not think about?
--- Did you include decisions they did not include?

--- Did they include put tasks in parallel that you thought were sequential?
--- Did they treat tasks as sequential that you think could be done in parallel?

****
* Sometimes, the CTA analysis had two strings of tasks that broke off from *
* one task. The break point had either an "and" symbol or it had an "or" *
* symbol. The two strings were then parallel activity sequences. In the *
* case where both strings were active ("and"), then are done concurrently, *
* that is, at the same time or nearly simultaneously, perhaps in some sort *
* of time-sharing. In the case where both strings begin with "or," they *
* the two strings are mutually exclusive (you do one or the other but not *
* both). This may be a conscious decision, or an obvious consequence of *
* the perceived conditions that occur at the time the task sequence begins. *
****

VI. The Easiest and Most Difficult Parts of Doing Your Task Analysis

--- What did you find was the most difficult part about doing the task analysis?
--- What did you find was the easiest part of doing the task analysis?
--- What would best help you do a task analysis if you had to do one again?
--- How could we improve the experiment if we repeat it next quarter?

****
* Different people struggle with different things. We want to know where *
* you had difficulties completing the task analysis assignment. It would be*
* most helpful if you could give us some insight about why you had problems,*
* what you did to resolve those problems and what you think might help *
* others get past such hurdles more easily in the future. We're also in- *
* terested in any other suggestions you might like to make on how we might *
* improve upon this exercise if we do it again with a new group of folks. *
****

VII. Other Comments and Recommendations

--- This is where you can tell us anything else you want to say that hasn't been treated elsewhere.
-- That includes asking questions you'd like us to answer for your benefit, things that occurred to you as you were writing, or things you've been thinking about but have not gotten answers to yet.

****
* **Thanks for participating in this project. Your cooperation and effort in** *
* **this project may help us improve the methods used to describe human acti-** *
* **vities in complex systems accurately and reliably. Please let us know if** *
* **you have any recommendations that might help us achieve that objective.** *
****

Appendix D
Reliability and Validity Assessment Forms

The following pages show the format and content of the two forms that AV 640 students were asked to fill out after reviewing the thirteen task analyses. These data were the primary measures used to determine the reliability and validity of the AV 540 student's task analyses.

EVALUATION OF TASK ANALYSIS RELIABILITY

Each analysis is to be compared with every other analysis. On a scale from 0 to 10, you are asked to judge how similar the two analyses are to one another. You can look at similarity in terms of distance. If two task analyses are very similar in their content, they are very close to one another and would be given a score near 0. If the two analyses are very dissimilar, then their content is very far apart, and they would be given a score close to 10.

Check the box that best represents your assessment of the similarity between the two task analyses being compared.

		0	1	2	3	4	5	6	7	8	9	
1. Task Anal. 1 vs. 2 (Anderson vs. Cepec)	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
2. Task Anal. 1 vs. 3 (Anderson vs. Deatheridge)	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
3. Task Anal. 1 vs. 4 (Anderson vs. Eppley)	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
4. Task Anal. 1 vs. 5 (Anderson vs. Jergens)	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
5. Task Anal. 1 vs. 6 (Anderson vs. Lerdon)	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
6. Task Anal. 1 vs. 7 (Anderson vs. Lubinsky)	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
7. Task Anal. 1 vs. 8 (Anderson vs. Meyer)	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
8. Task Anal. 1 vs. 9 (Anderson vs. Schneider)	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
9. Task Anal. 1 vs. 10 (Anderson vs. Stedke)	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar

	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
10. Task Anal. 1 vs. 11 (Anderson vs. Torok)		__	__	__	__	__	__	__	__	__	__	
11. Task Anal. 1 vs. 12 (Anderson vs. Vrabel)		__	__	__	__	__	__	__	__	__	__	
12. Task Anal. 1 vs. 13 (Anderson vs. Wycoff)		__	__	__	__	__	__	__	__	__	__	
13. Task Anal. 2 vs. 3 (Cepec vs. Deatheridge)		__	__	__	__	__	__	__	__	__	__	
14. Task Anal. 2 vs. 4 (Cepec vs. Eppley)		__	__	__	__	__	__	__	__	__	__	
15. Task Anal. 2 vs. 5 (Cepec vs. Jergens)		__	__	__	__	__	__	__	__	__	__	
16. Task Anal. 2 vs. 6 (Cepec vs. Lerdon)		__	__	__	__	__	__	__	__	__	__	
17. Task Anal. 2 vs. 7 (Cepec vs. Lubinsky)		__	__	__	__	__	__	__	__	__	__	
18. Task Anal. 2 vs. 8 (Cepec vs. Meyer)		__	__	__	__	__	__	__	__	__	__	
19. Task Anal. 2 vs. 9 (Cepec vs. Schneider)		__	__	__	__	__	__	__	__	__	__	
20. Task Anal. 2 vs. 10 (Cepec vs. Stedke)		__	__	__	__	__	__	__	__	__	__	
21. Task Anal. 2 vs. 11 (Cepec vs. Torok)		__	__	__	__	__	__	__	__	__	__	

	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
22. Task Anal. 2 vs. 12 (Cepec vs. Vrabel)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
23. Task Anal. 2 vs. 13 (Cepec vs. Wycoff)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
24. Task Anal. 3 vs. 4 (Deatheridge vs. Eppley)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
25. Task Anal. 3 vs. 5 (Deatheridge vs. Jergens)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
26. Task Anal. 3 vs. 6 (Deatheridge vs. Lerdon)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
27. Task Anal. 3 vs. 7 (Deatheridge vs. Lubinsky)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
28. Task Anal. 3 vs. 8 (Deatheridge vs. Meyer)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
29. Task Anal. 3 vs. 9 (Deatheridge vs. Schneider)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
30. Task Anal. 3 vs. 10 (Deatheridge vs. Stedke)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
31. Task Anal. 3 vs. 11 (Deatheridge vs. Torok)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
32. Task Anal. 3 vs. 12 (Deatheridge vs. Vrabel)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
33. Task Anal. 3 vs. 13 (Deatheridge vs. Wycoff)		___	___	___	___	___	___	___	___	___	___	

	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
34. Task Anal. 4 vs. 5 (Eppley vs. Jergens)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
35. Task Anal. 4 vs. 6 (Eppley vs. Lerdon)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
36. Task Anal. 4 vs. 7 (Eppley vs. Lubinsky)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
37. Task Anal. 4 vs. 8 (Eppley vs. Meyer)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
38. Task Anal. 4 vs. 9 (Eppley vs. Schneider)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
39. Task Anal. 4 vs. 10 (Eppley vs. Stedke)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
40. Task Anal. 4 vs. 11 (Eppley vs. Torok)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
41. Task Anal. 4 vs. 12 (Eppley vs. Vrabel)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
42. Task Anal. 4 vs. 13 (Eppley vs. Wycoff)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
43. Task Anal. 5 vs. 6 (Jergens vs. Lerdon)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
44. Task Anal. 5 vs. 7 (Jergens vs. Lubinsky)		___	___	___	___	___	___	___	___	___	___	
	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
45. Task Anal. 5 vs. 8 (Jergens vs. Meyer)		___	___	___	___	___	___	___	___	___	___	

Similar 0 1 2 3 4 5 6 7 8 9 Dissimilar
46. Task Anal. 5 vs. 9 __ __ __ __ __ __ __ __ __ __
(Jergens vs. Schneider)

Similar 0 1 2 3 4 5 6 7 8 9 Dissimilar
47. Task Anal. 5 vs. 10 __ __ __ __ __ __ __ __ __ __
(Jergens vs. Stedke)

Similar 0 1 2 3 4 5 6 7 8 9 Dissimilar
48. Task Anal. 5 vs. 11 __ __ __ __ __ __ __ __ __ __
(Jergens vs. Torok)

Similar 0 1 2 3 4 5 6 7 8 9 Dissimilar
49. Task Anal. 5 vs. 12 __ __ __ __ __ __ __ __ __ __
(Jergens vs. Vrabel)

Similar 0 1 2 3 4 5 6 7 8 9 Dissimilar
50. Task Anal. 5 vs. 13 __ __ __ __ __ __ __ __ __ __
(Jergens vs. Wycoff)

Similar 0 1 2 3 4 5 6 7 8 9 Dissimilar
51. Task Anal. 6 vs. 7 __ __ __ __ __ __ __ __ __ __
(Lerdon vs. Lubinsky)

Similar 0 1 2 3 4 5 6 7 8 9 Dissimilar
52. Task Anal. 6 vs. 8 __ __ __ __ __ __ __ __ __ __
(Lerdon vs. Meyer)

Similar 0 1 2 3 4 5 6 7 8 9 Dissimilar
53. Task Anal. 6 vs. 9 __ __ __ __ __ __ __ __ __ __
(Lerdon vs. Schneider)

Similar 0 1 2 3 4 5 6 7 8 9 Dissimilar
54. Task Anal. 6 vs. 10 __ __ __ __ __ __ __ __ __ __
(Lerdon vs. Stedke)

Similar 0 1 2 3 4 5 6 7 8 9 Dissimilar
55. Task Anal. 6 vs. 11 __ __ __ __ __ __ __ __ __ __
(Lerdon vs. Torok)

Similar 0 1 2 3 4 5 6 7 8 9 Dissimilar
56. Task Anal. 6 vs. 12 __ __ __ __ __ __ __ __ __ __
(Lerdon vs. Vrabel)

Similar 0 1 2 3 4 5 6 7 8 9 Dissimilar
57. Task Anal. 6 vs. 13 __ __ __ __ __ __ __ __ __ __
(Lerdon vs. Wycoff)

		0	1	2	3	4	5	6	7	8	9	
58. Task Anal. 7 vs. 8 (Lubinsky vs. Meyer)	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
59. Task Anal. 7 vs. 9 (Lubinsky vs. Schneider)	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
60. Task Anal. 7 vs. 10 (Lubinsky vs. Stedke)	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
61. Task Anal. 7 vs. 11 (Lubinsky vs. Torok)	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
62. Task Anal. 7 vs. 12 (Lubinsky vs. Vrabel	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
63. Task Anal. 7 vs. 13 (Lubinsky vs. Wycoff)	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
64. Task Anal. 8 vs. 9 (Meyer vs. Schneider)	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
65. Task Anal. 8 vs. 10 (Meyer vs. Stedke)	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
66. Task Anal. 8 vs. 11 (Meyer vs. Torok)	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
67. Task Anal. 8 vs. 12 (Meyer vs. Vrabel)	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
68. Task Anal. 8 vs. 13 (Meyer vs. Wycoff)	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
69. Task Anal. 9 vs. 10 (Schneider vs. Stedke)	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar

	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
70. Task Anal. 9 vs. 11 (Schneider vs. Torok)		__	__	__	__	__	__	__	__	__	__	
71. Task Anal. 9 vs. 12 (Schneider vs. Vrabel)		__	__	__	__	__	__	__	__	__	__	
72. Task Anal. 9 vs. 13 (Schneider vs. Wycoff)		__	__	__	__	__	__	__	__	__	__	
73. Task Anal. 10 vs. 11 (Stedke vs. Torok)		__	__	__	__	__	__	__	__	__	__	
74. Task Anal. 10 vs. 12 (Stedke vs. Vrabel)		__	__	__	__	__	__	__	__	__	__	
75. Task Anal. 10 vs. 13 (Stedke vs. Wycoff)		__	__	__	__	__	__	__	__	__	__	
76. Task Anal. 11 vs. 12 (Torok vs. Vrabel)		__	__	__	__	__	__	__	__	__	__	
77. Task Anal. 11 vs. 13 (Torok vs. Wycoff)		__	__	__	__	__	__	__	__	__	__	
78. Task Anal. 12 vs. 13 (Vrabel vs. Wycoff)		__	__	__	__	__	__	__	__	__	__	

EVALUATION OF TASK ANALYSIS VALIDITY

Each analysis is to be compared with the CTA, Inc. task analysis done for the Federal Aviation Administration. On a scale from 0 to 10, you are asked to judge how similar the two analyses are to one another. You can look at similarity in terms of distance, just as you did for the reliability assessments. If the two task analyses are very similar in their content, they are very close to one another and would be given a score near 0. If the two analyses are very dissimilar, then their content is very far apart, and they would be given a score close to 10.

Check the box that best represents your assessment of the similarity between the two task analyses being compared.

		0	1	2	3	4	5	6	7	8	9	
1. Anderson vs. CTA	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
2. Cepec vs. CTA	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
3. Deatheridge vs. CTA	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
4. Eppley vs. CTA	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
5. Jergens vs. CTA	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
6. Lerdon vs. CTA	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
7. Lubinsky vs. CTA	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
8. Meyer vs. CTA	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar
9. Schneider vs. CTA	Similar	__	__	__	__	__	__	__	__	__	__	Dissimilar

10. Stedke vs. CTA	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
11. Torok vs. CTA	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
12. Vrabel vs. CTA	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar
13. Wycoff vs. CTA	Similar	0	1	2	3	4	5	6	7	8	9	Dissimilar

Appendix E
PATHFINDER Layout Diagrams

This appendix contains the 24 + 18 + 9 = 51 diagrams that show how each subject's similarity scores for each administration completed are to be geometrically interpreted. The PATHFINDER program computes the relationships among the paired comparisons, and then a two dimensional projection is generated, portraying that pattern of relationships. Two kinds of diagram comparisons are instructive: 1) between subject comparisons within any one of the three administrations, and 2) within subject comparisons across two, or three administrations, depending on whether a particular subject completed any more than one administration. Questionnaire administrations are labeled Q1, Q2, or Q3 for convenience of identifying each particular administration of the same paired comparison rating scale. When a subject number is missing for Q2 or Q3, it means that particular subject did not complete that particular administration of the similarity rating questionnaire.

Figure E-1. Subject 1 for Q1.

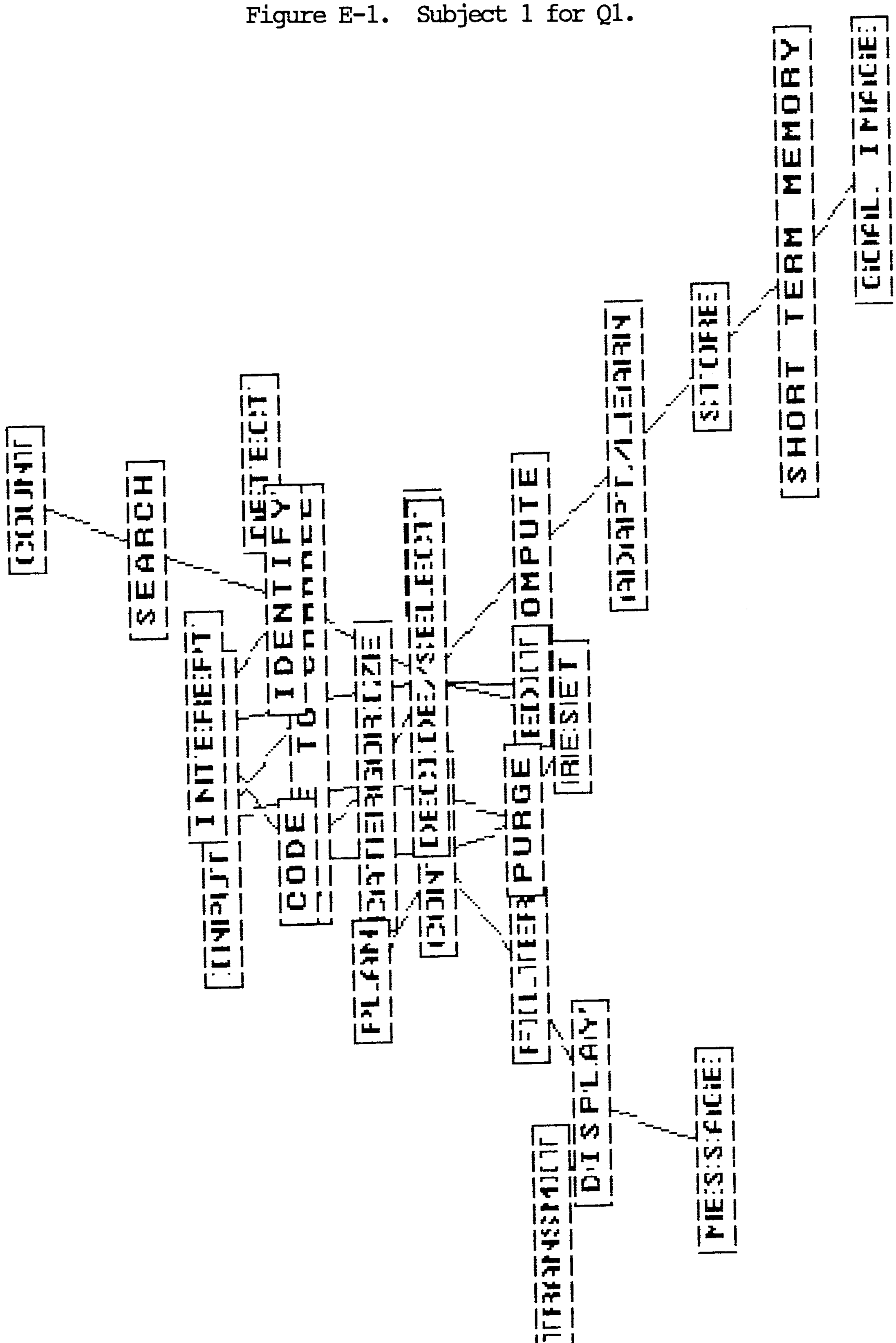


Figure E-2. Subject 2 for Q1.

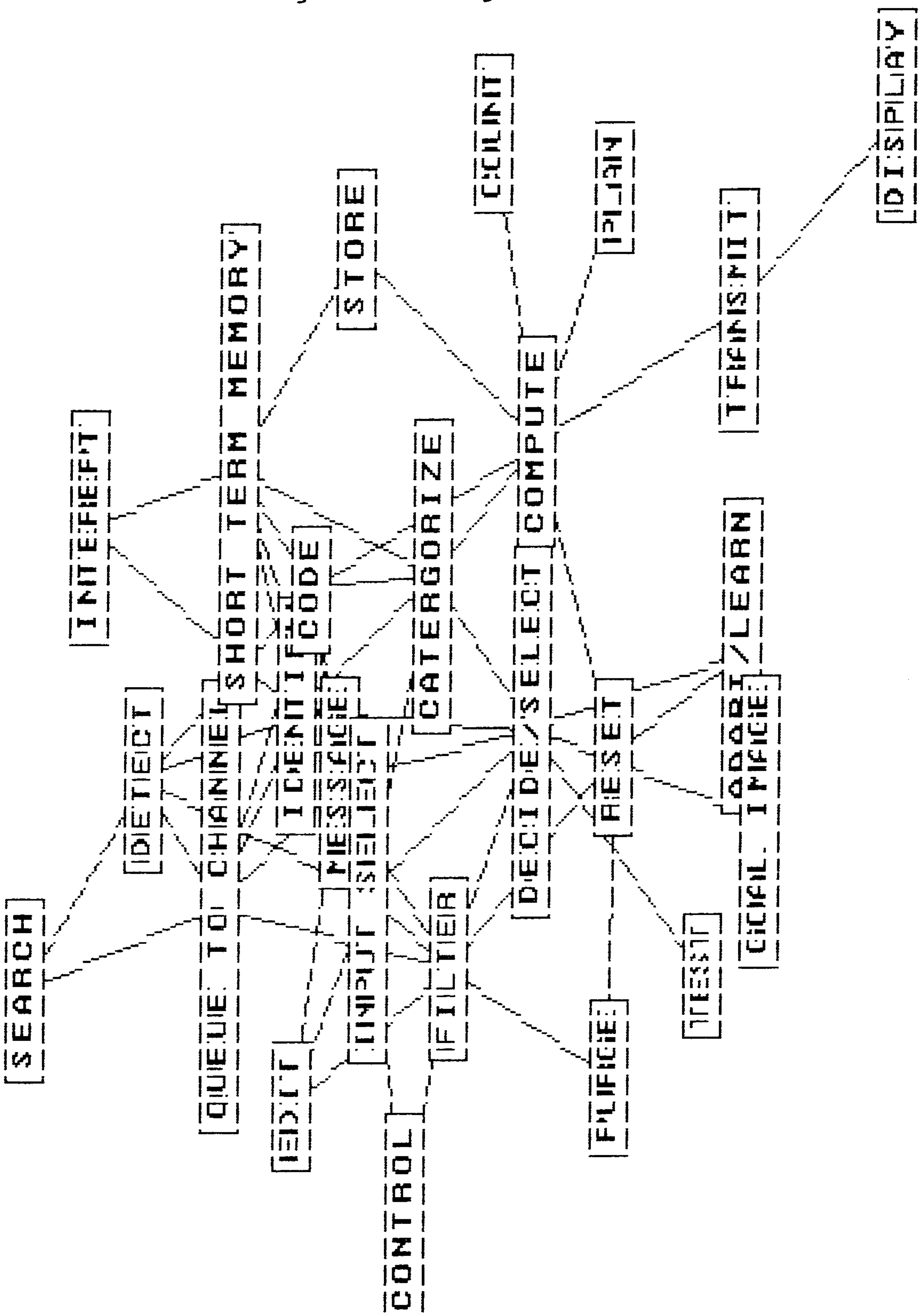


Figure E-3. Subject 3 for Q1.

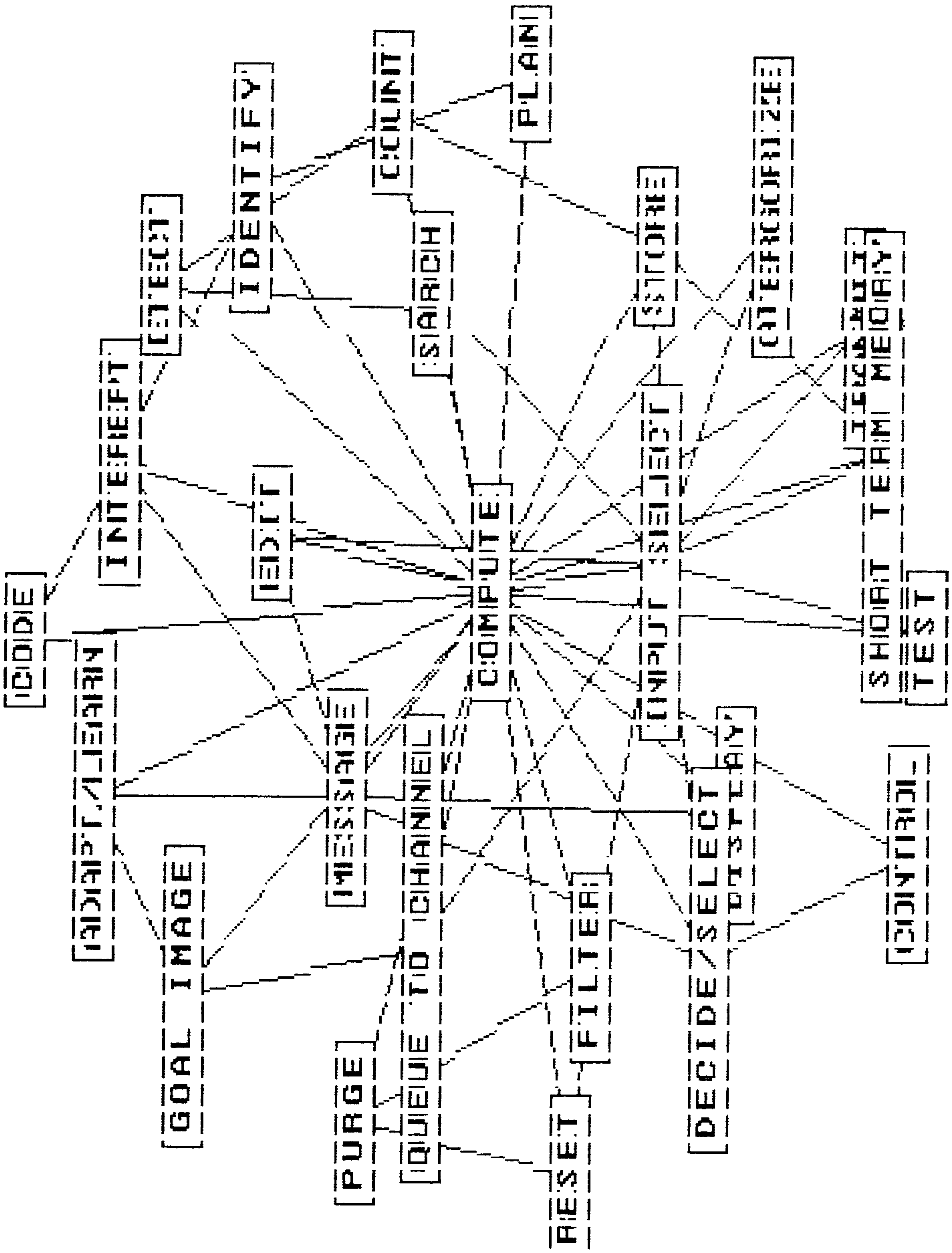


Figure E-4. Subject 4 for Q1.

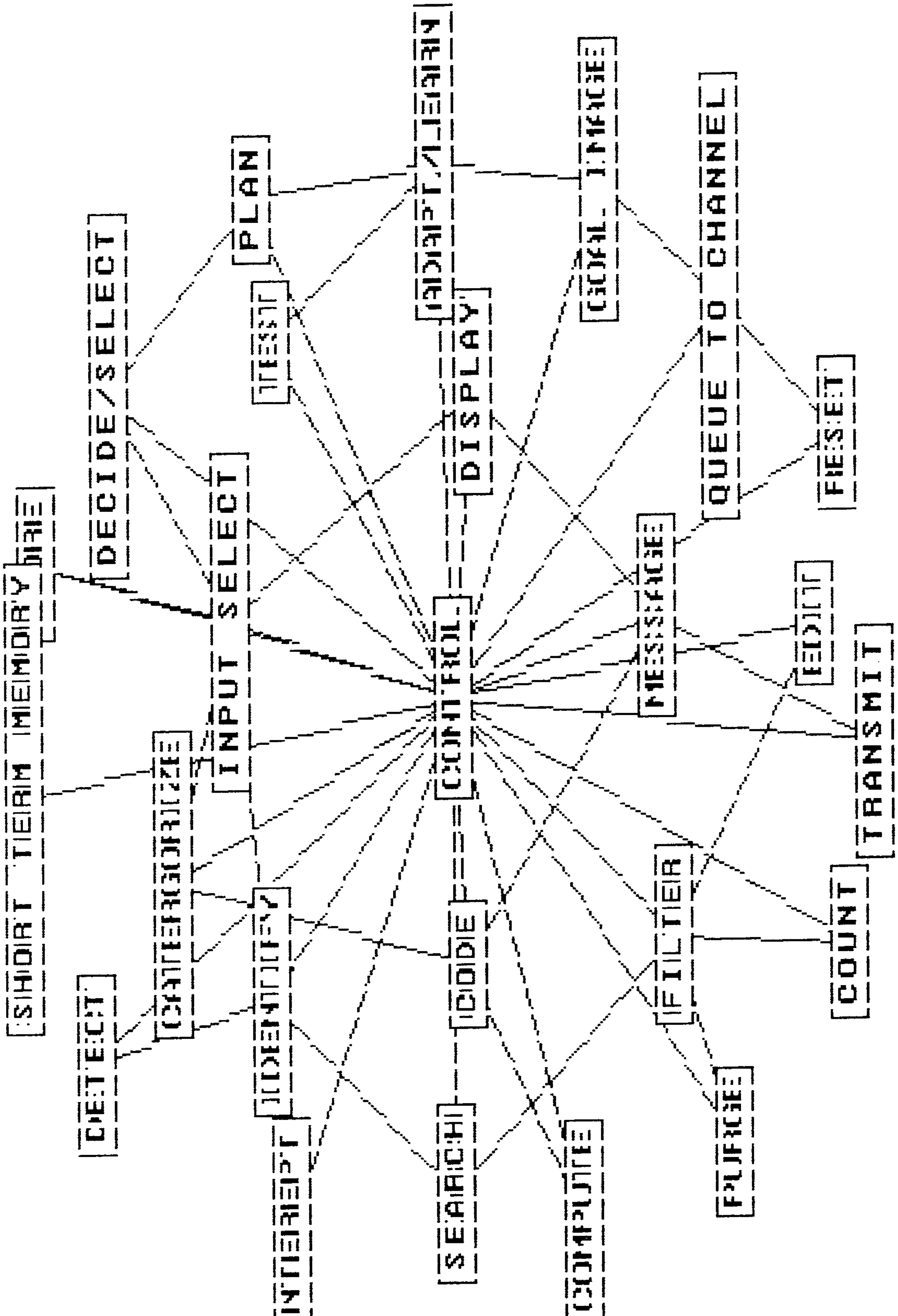


Figure E-5. Subject 5 for Q1.

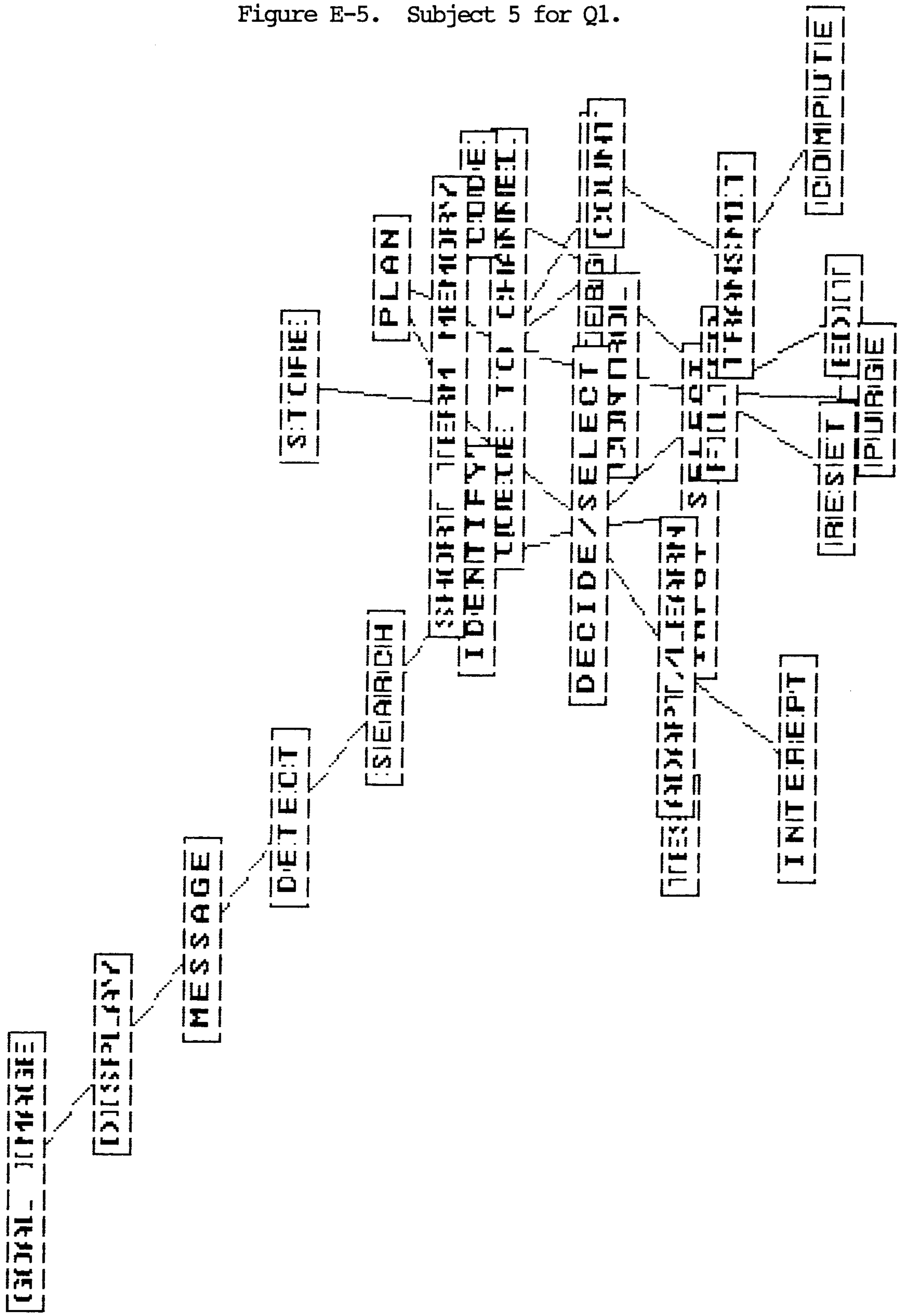


Figure E-6. Subject 6 for Q1.

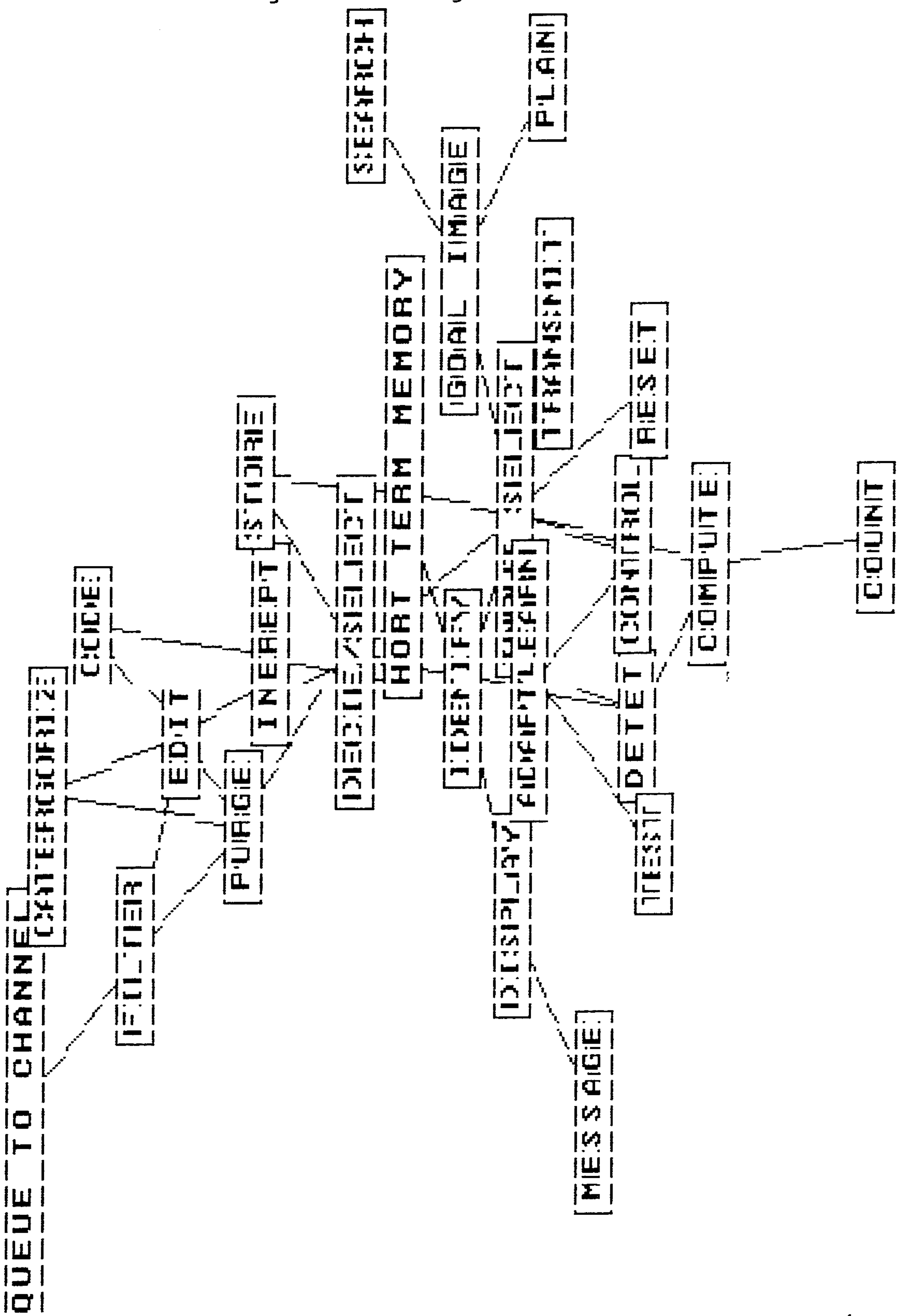


Figure E-7. Subject 7 for Q1.

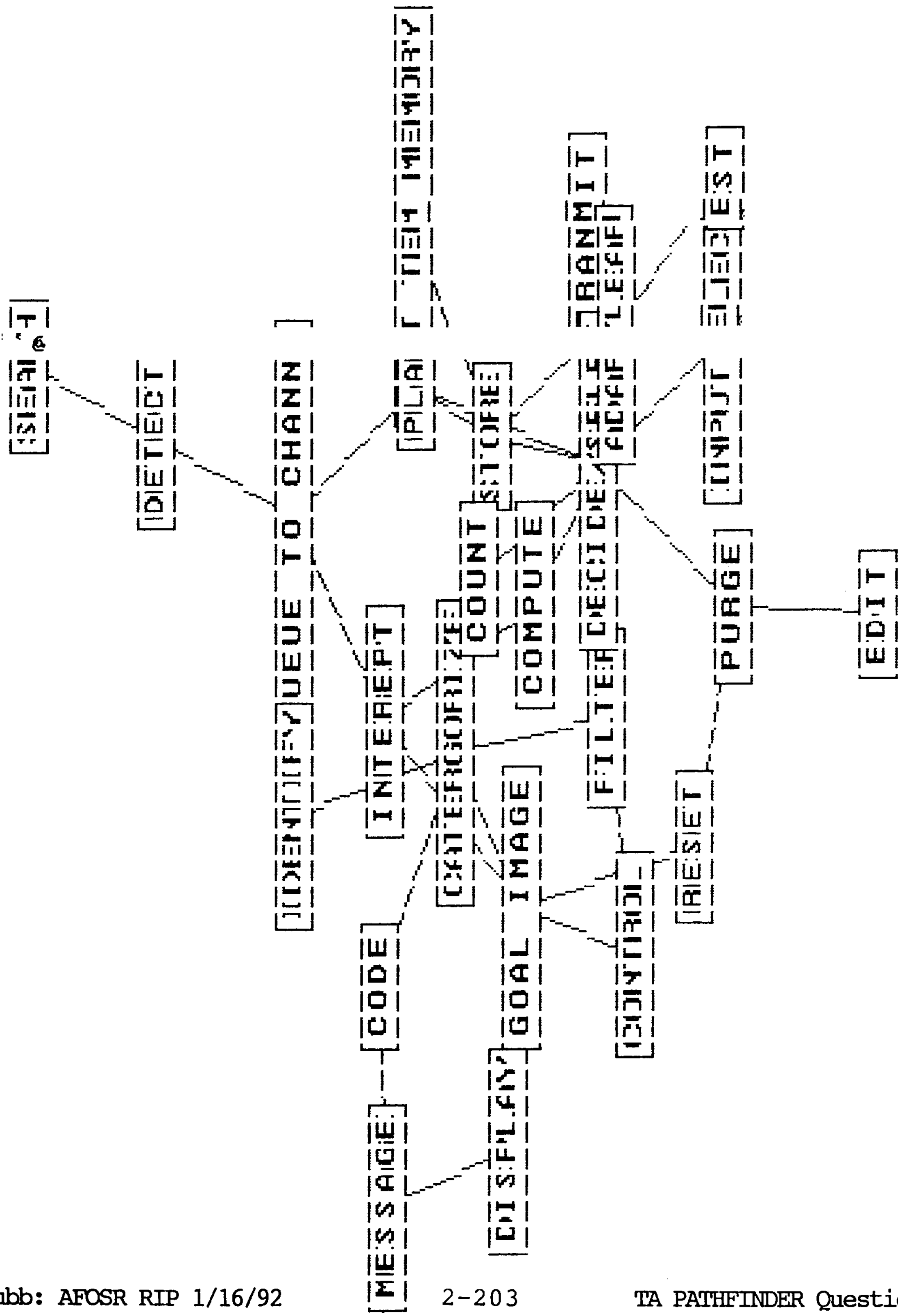


Figure E-8. Subject 8 for Q1.

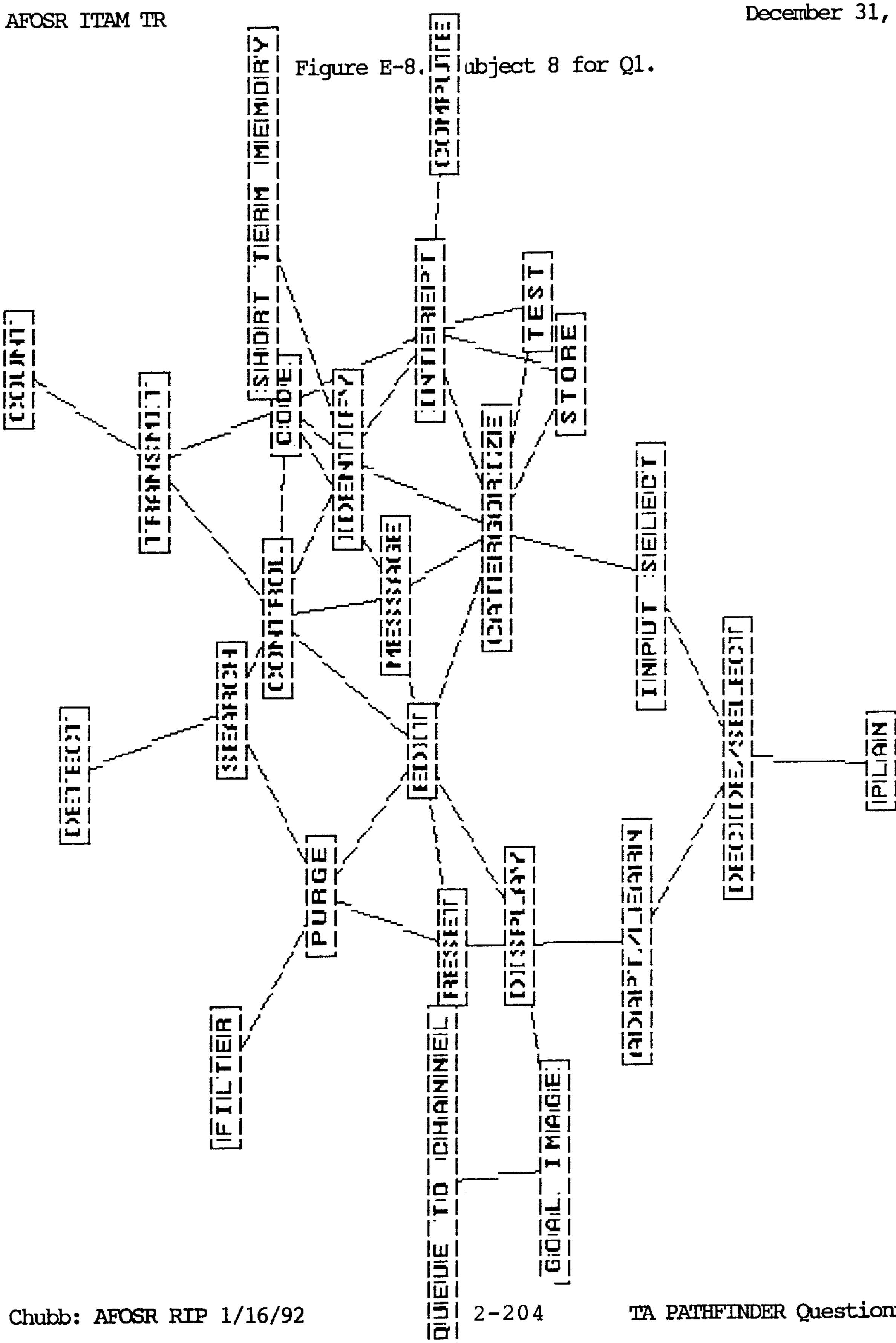


Figure E-9. Subject 9 for Q1.

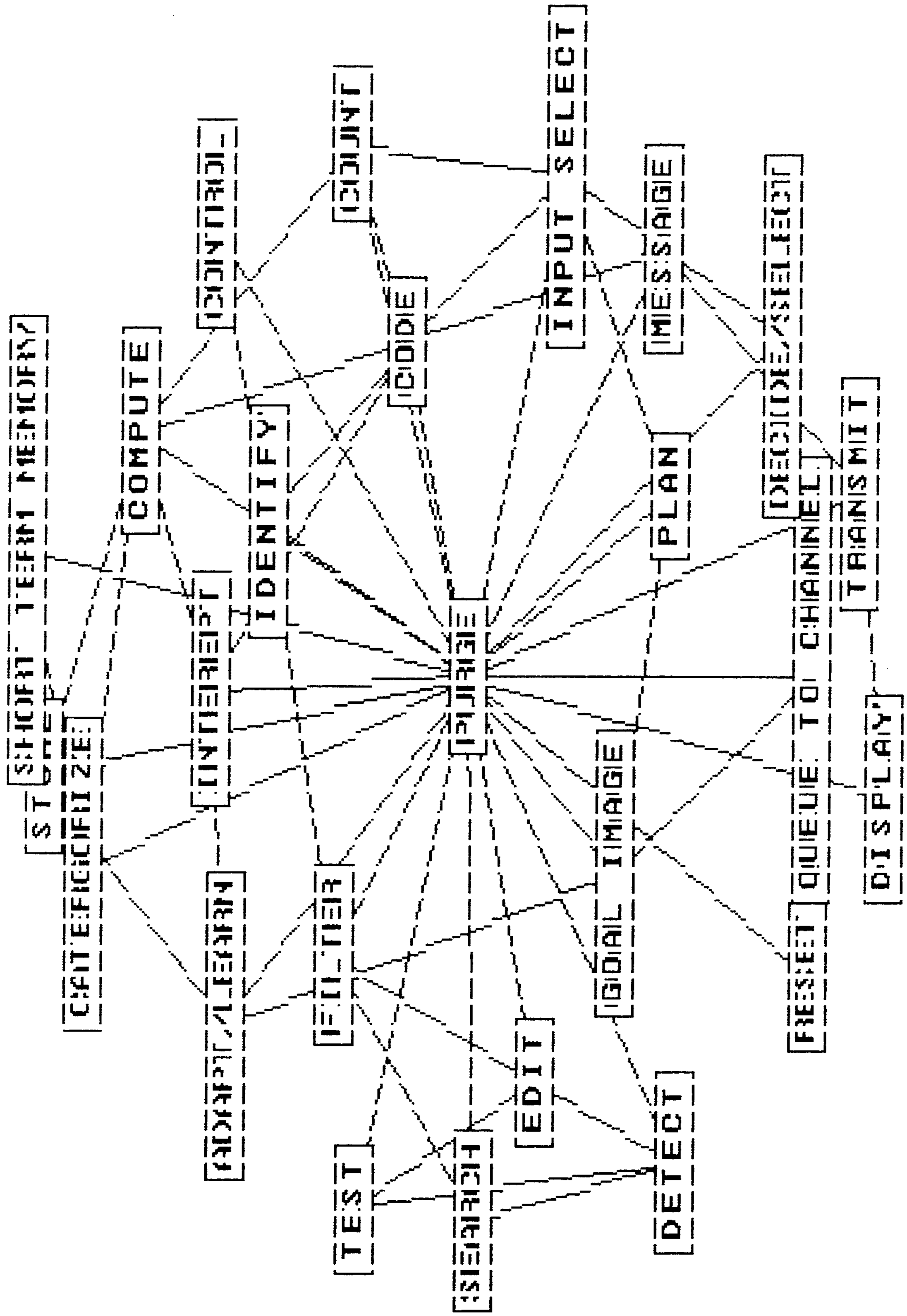


Figure E-10. Subject 10 for Q1.

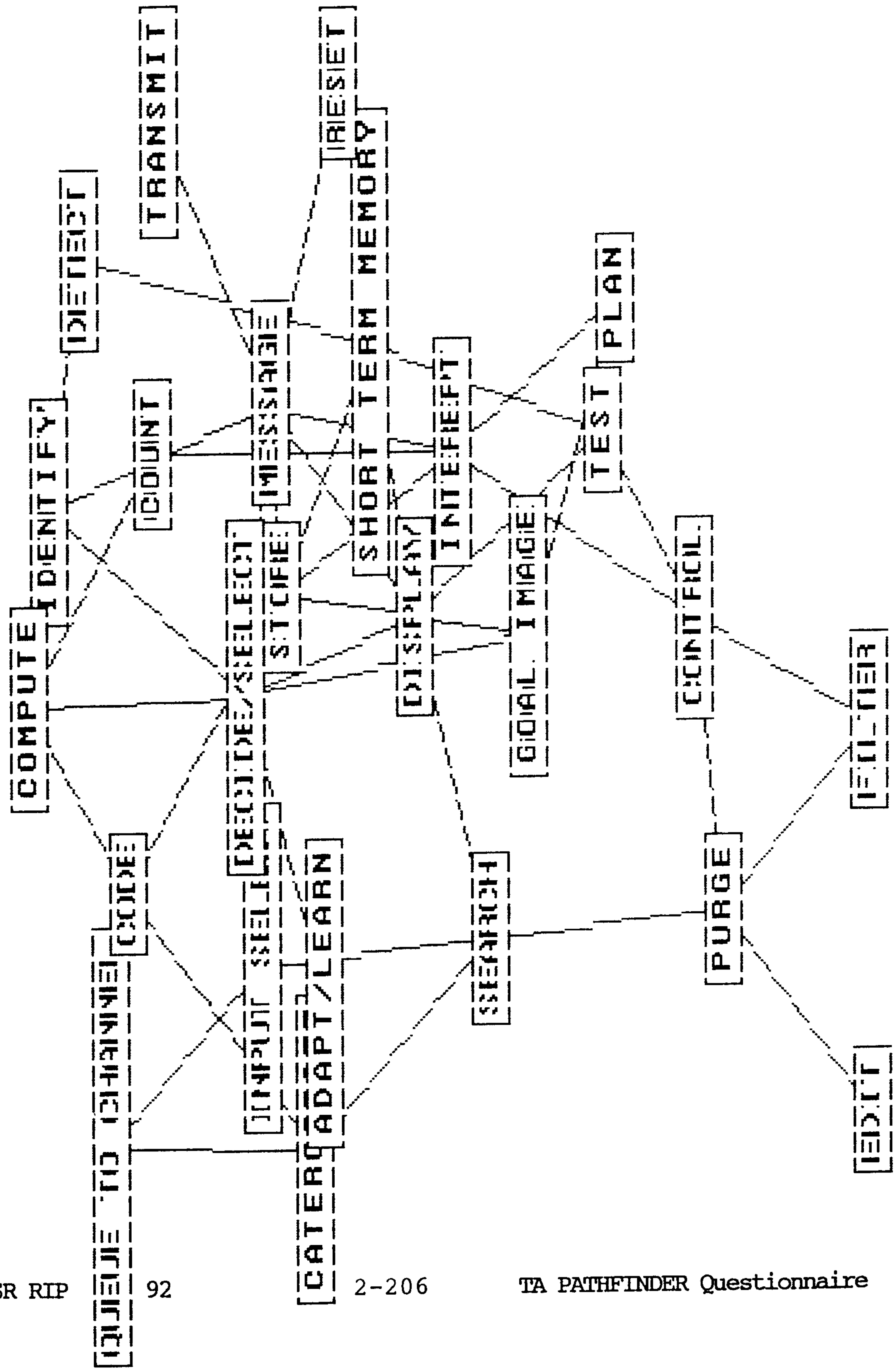


Figure E-11. Subject 11 for Q1.

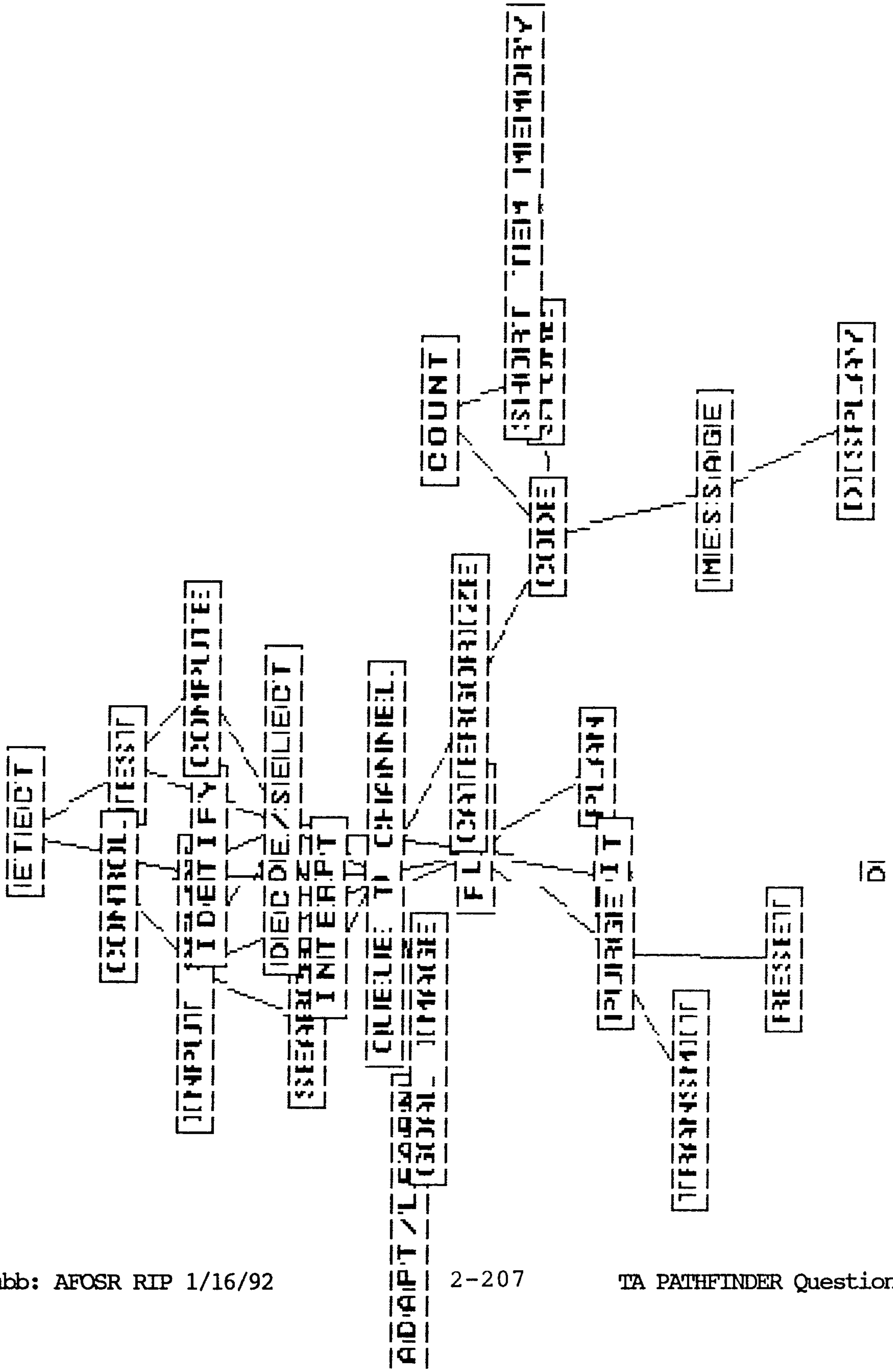


Figure E-12. Subject 12 for Q1.

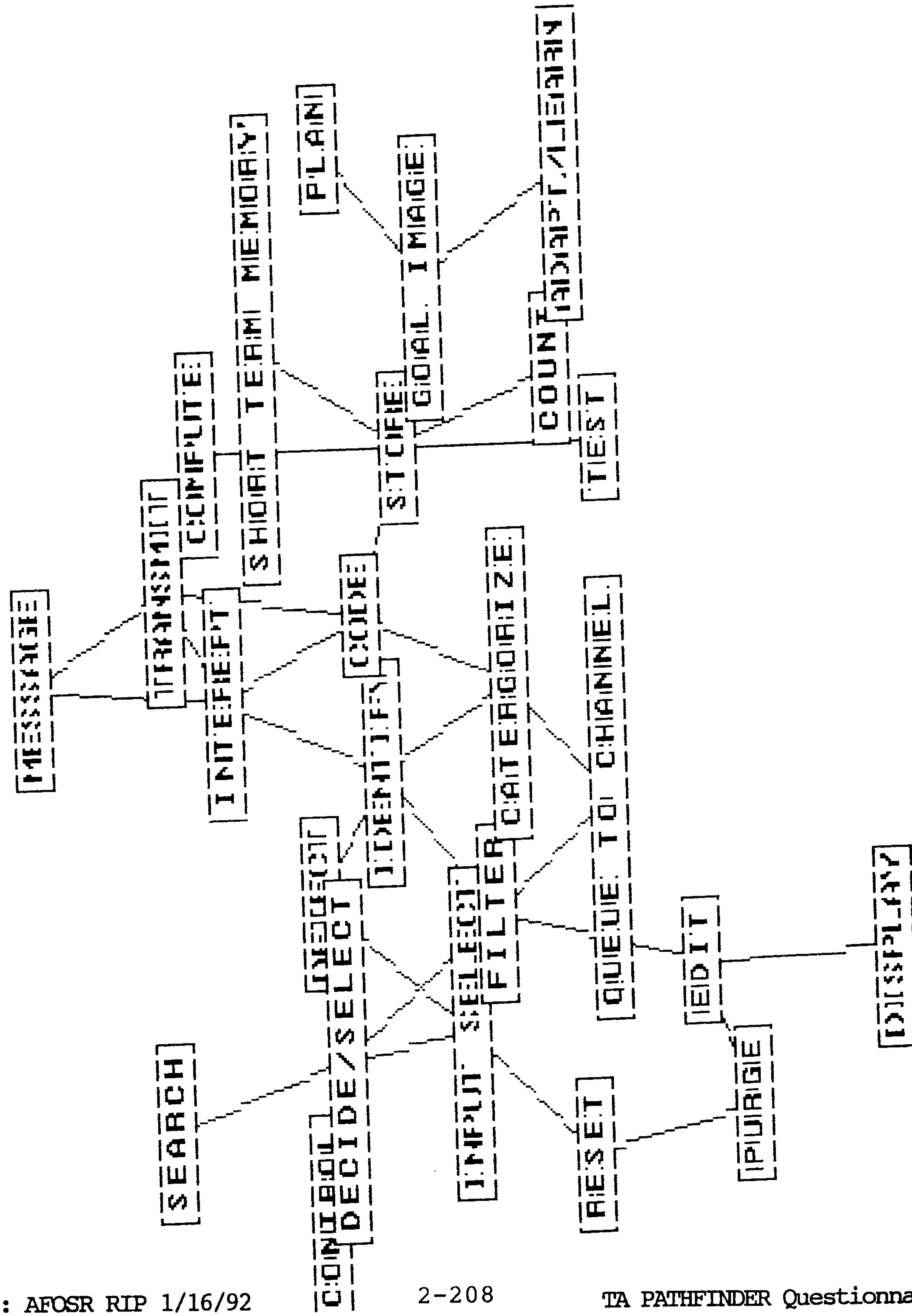


Figure E-13. Subject 13 for Q1.

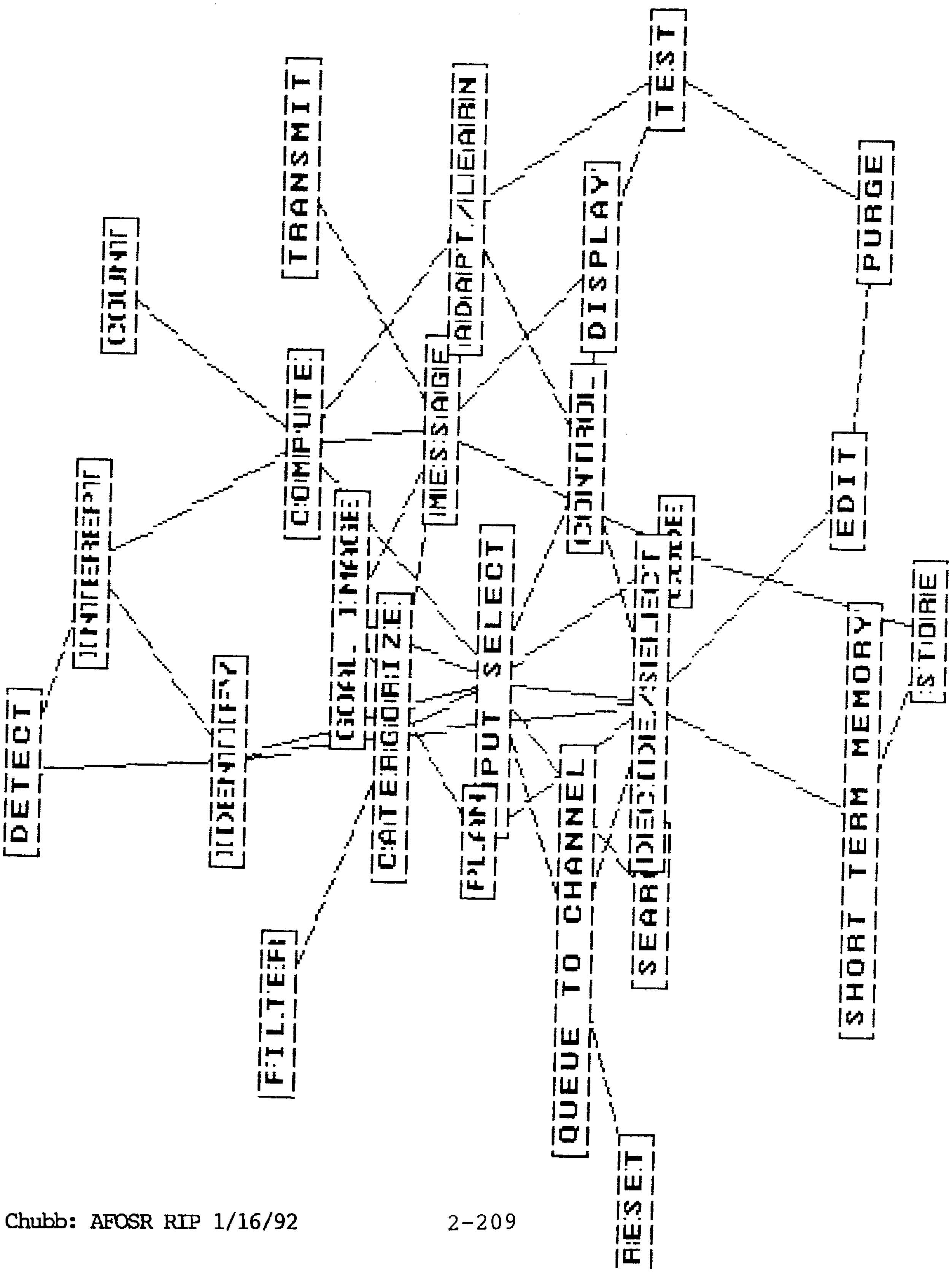


Figure E-14. Subject 14 for Q1.

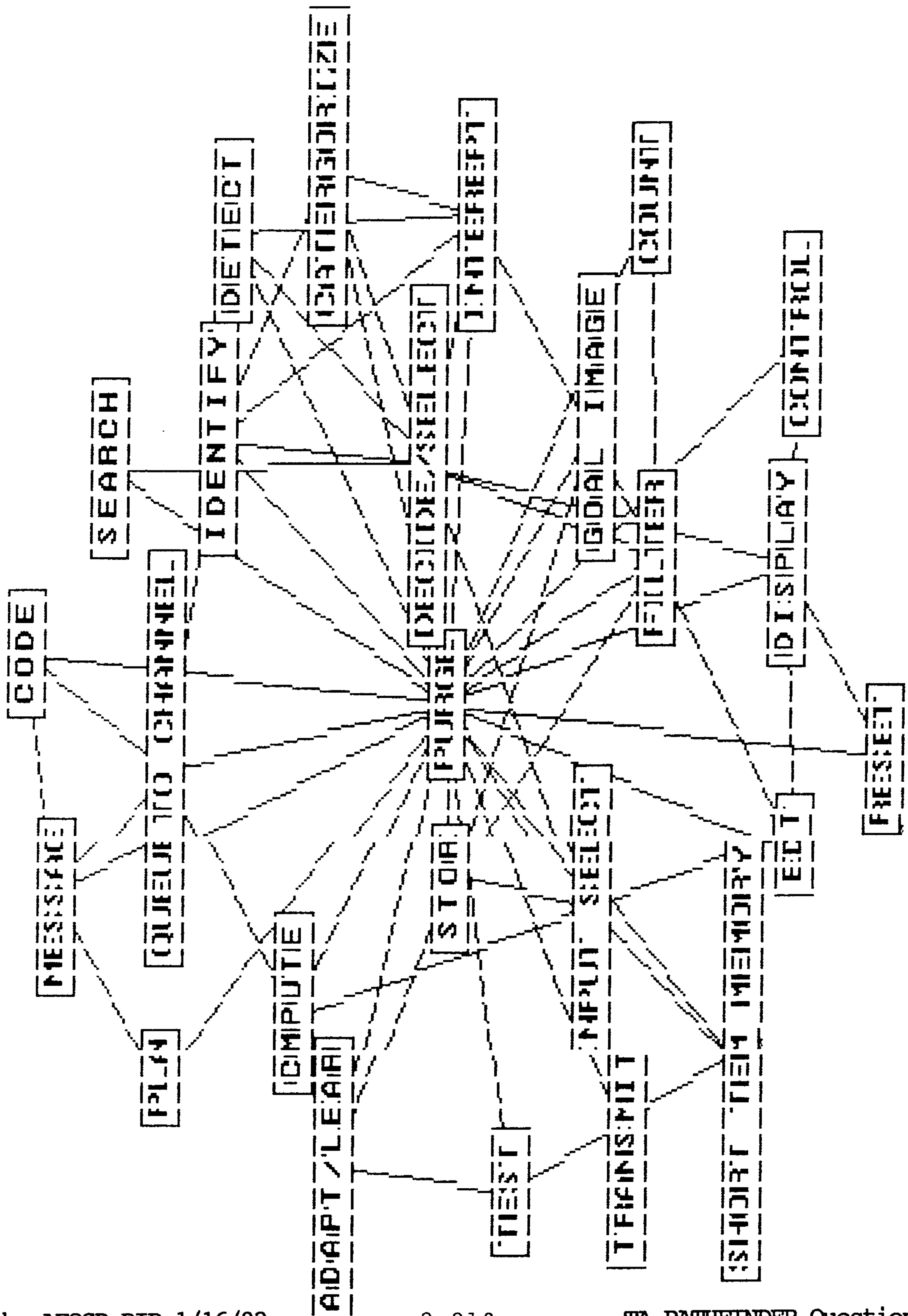


Figure E-15. Subject 15 for Q1.

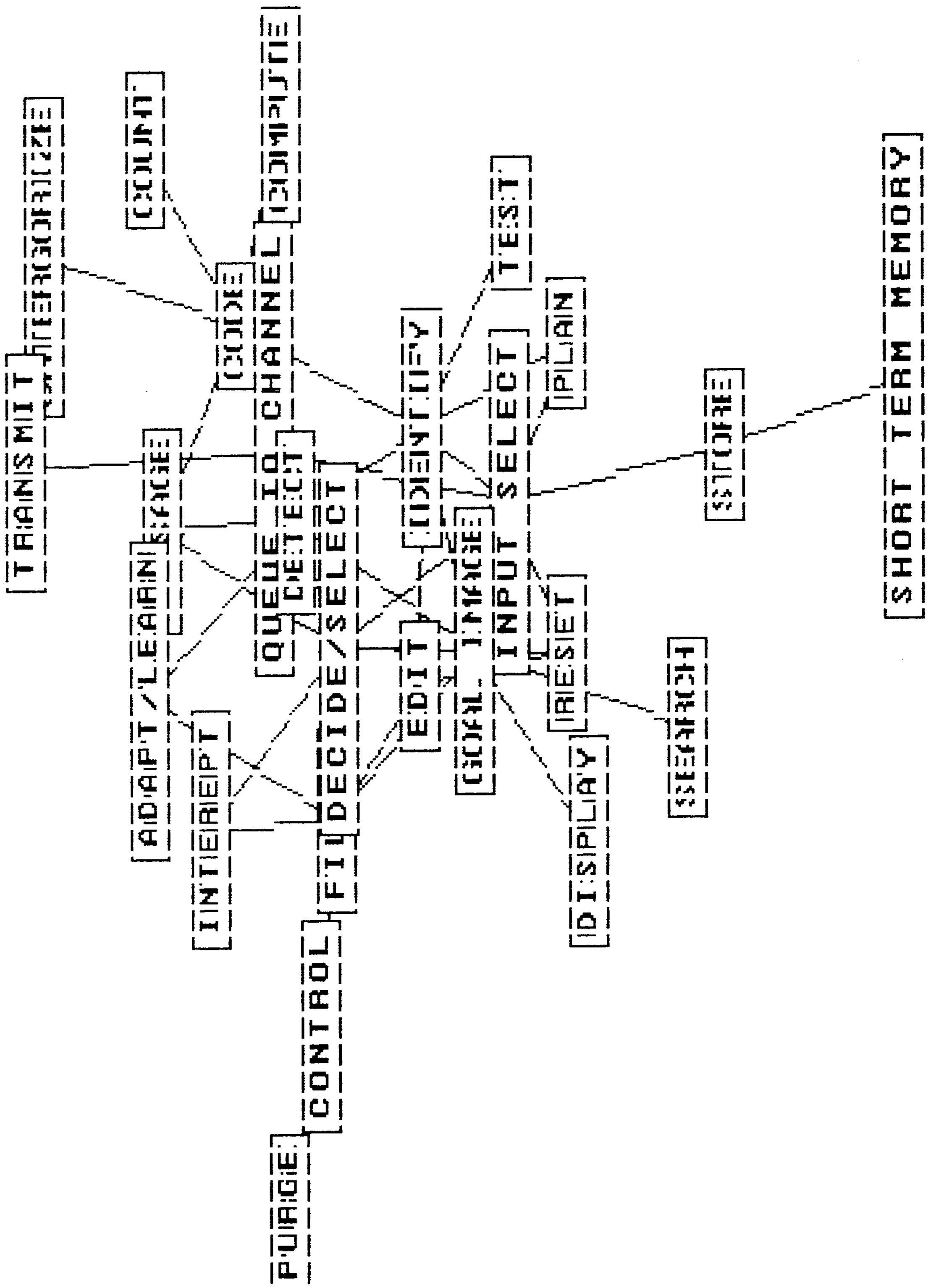


Figure E-16. Subject 16 for Q1.

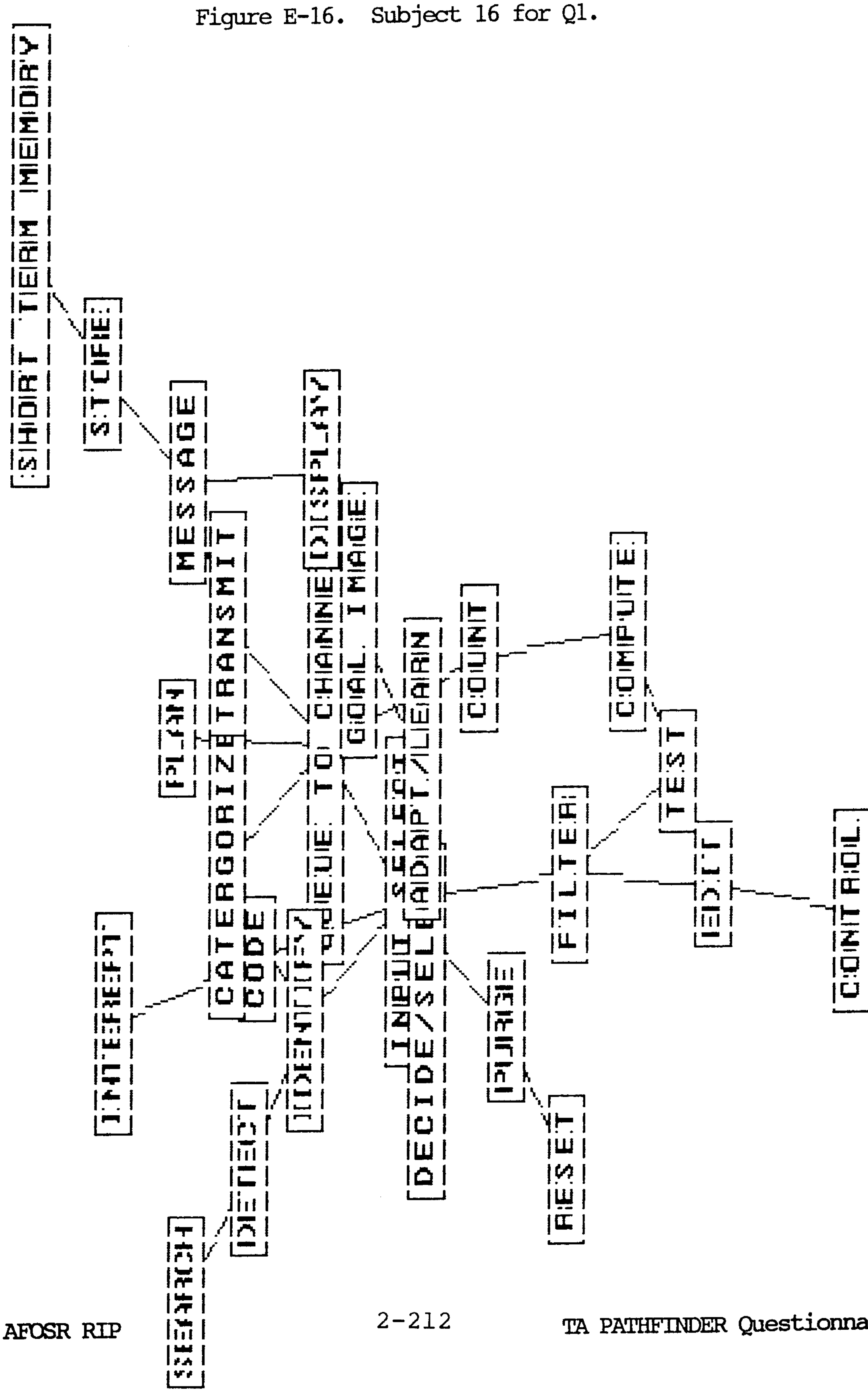


Figure E-17. Subject 17 for Q1.

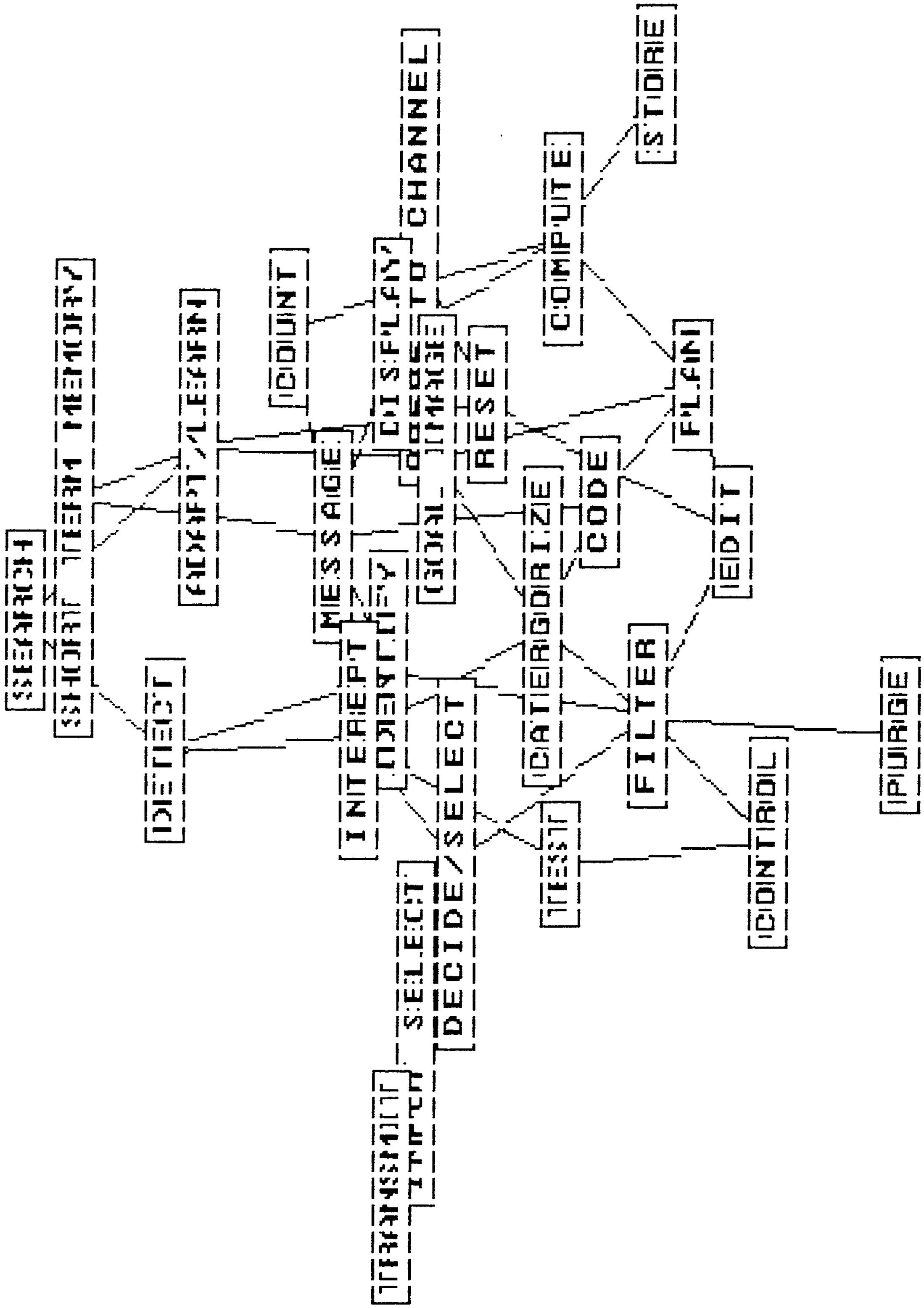


Figure E-18. Subject 18 for Q1.

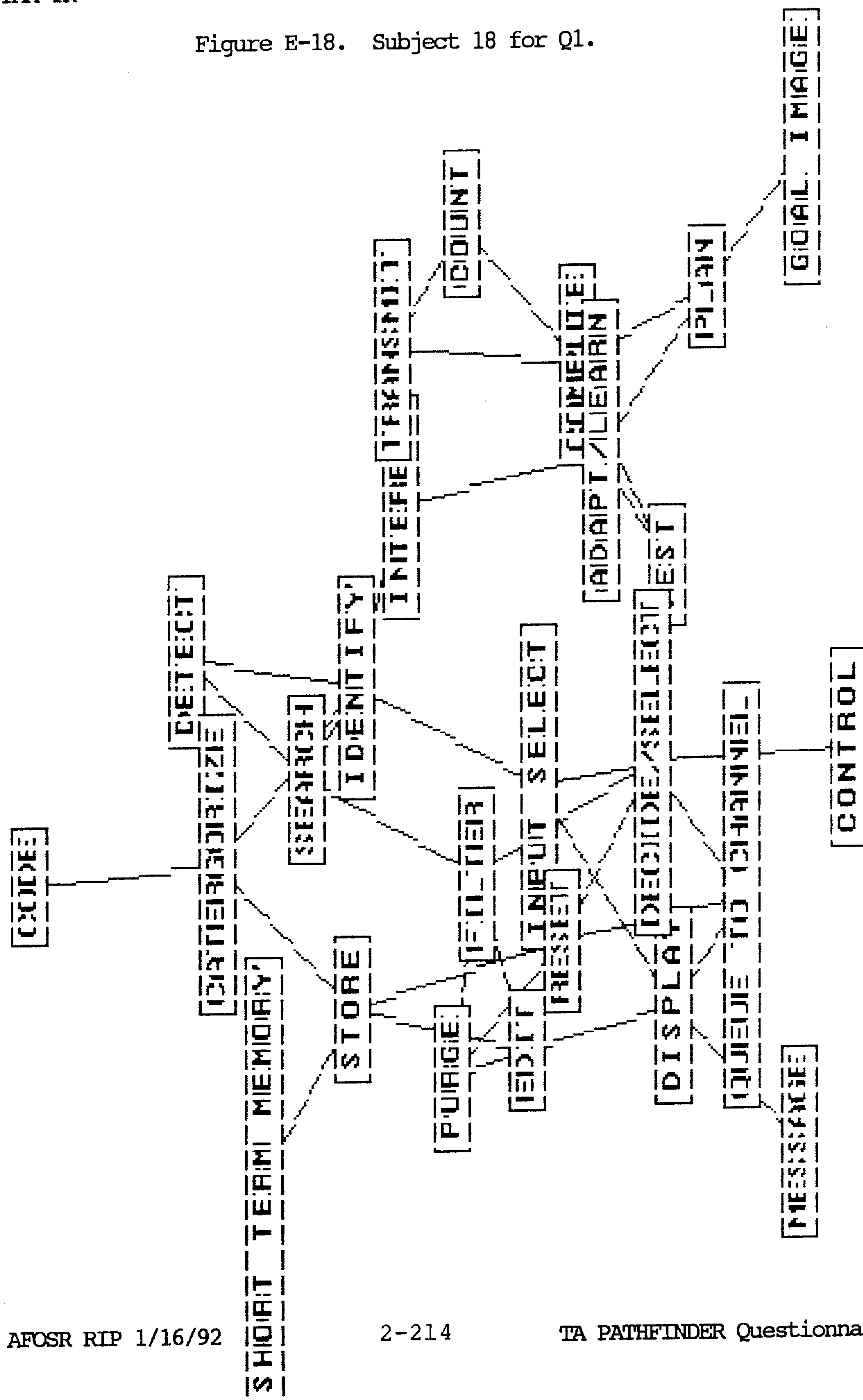


Figure E-19. Subject 19 for Q1.

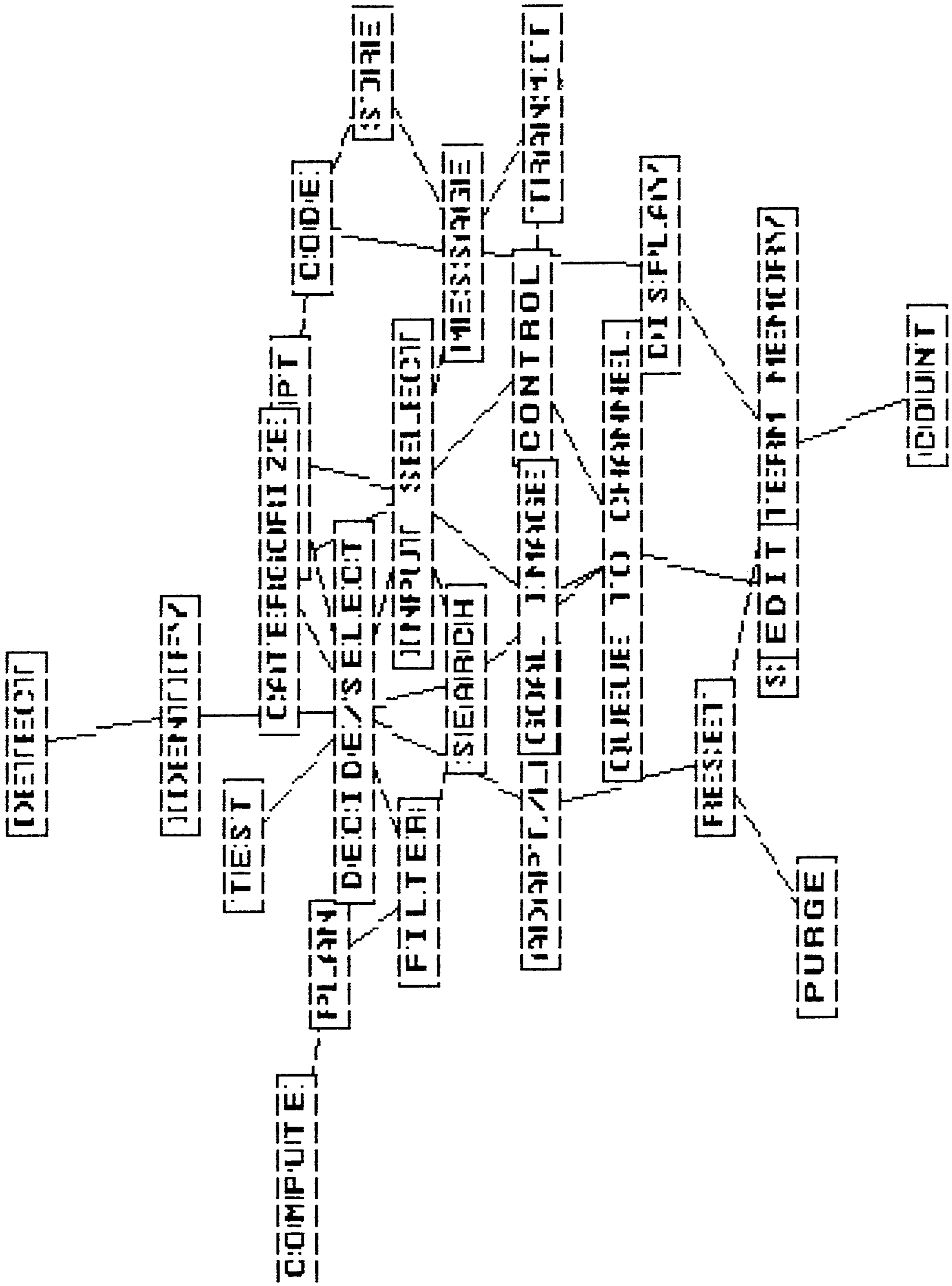


Figure E-20. Subject 20 for Q1.

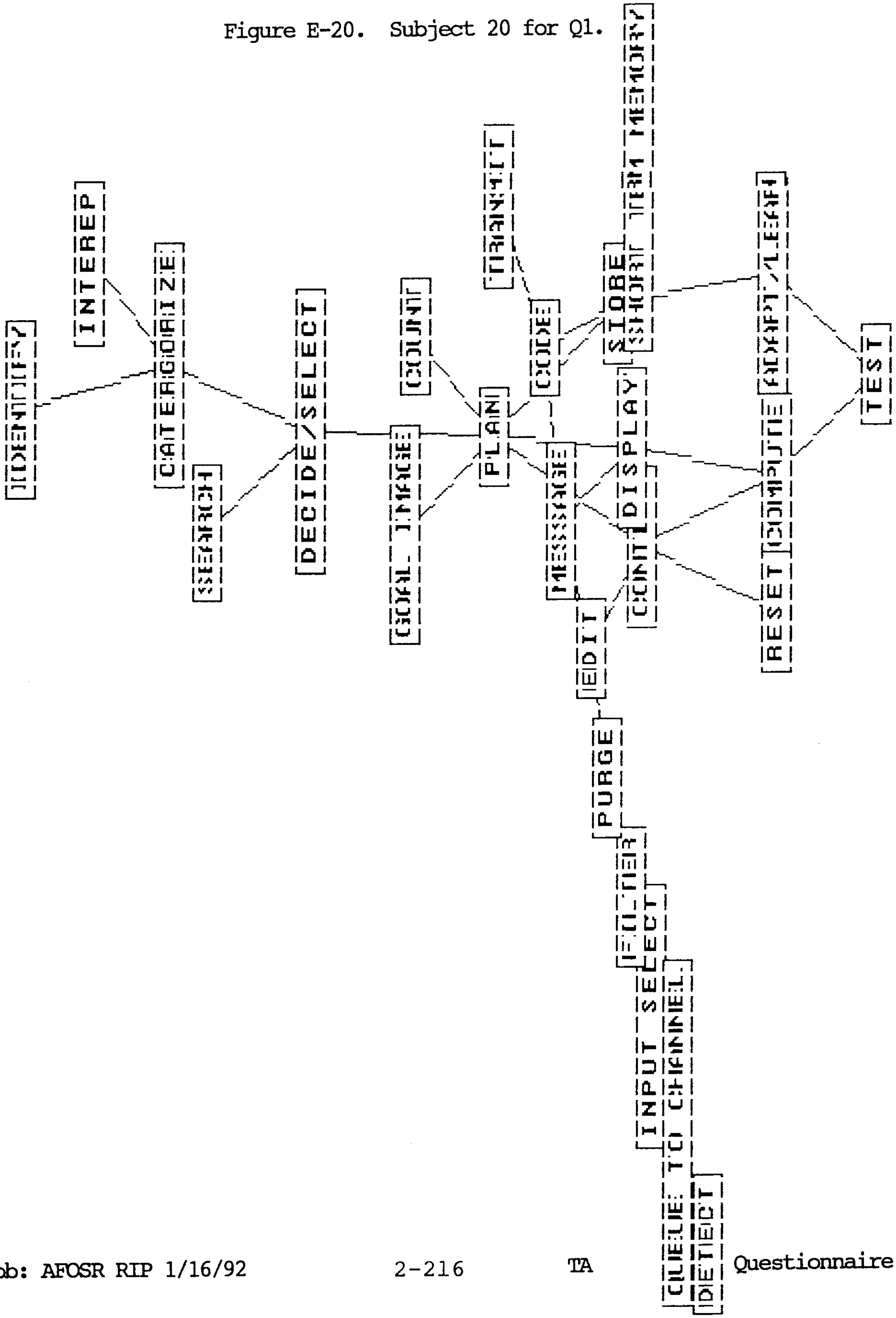


Figure E-21. Subject 21 for Q1.

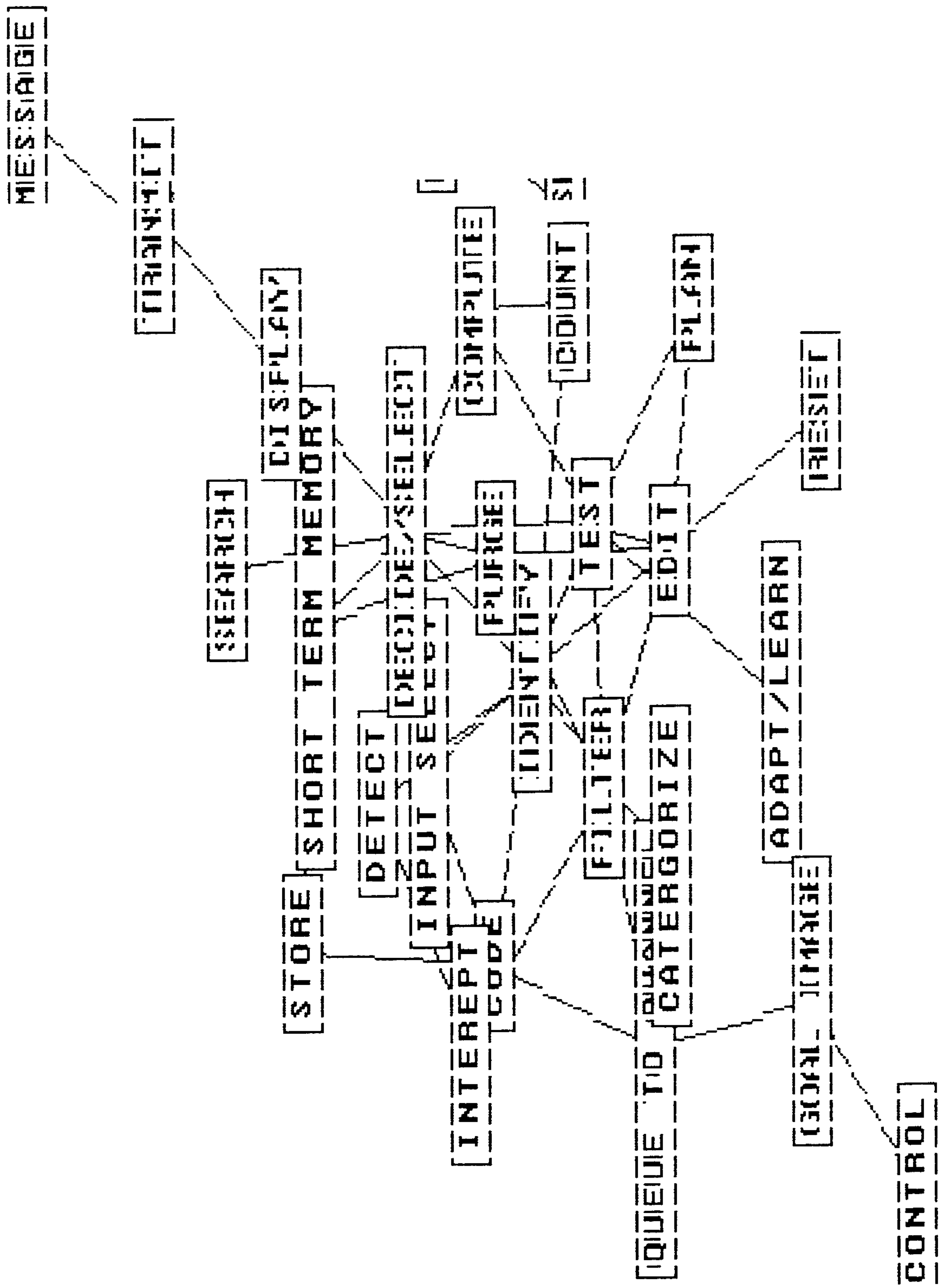


Figure E-22. Subject 22 for Q1.

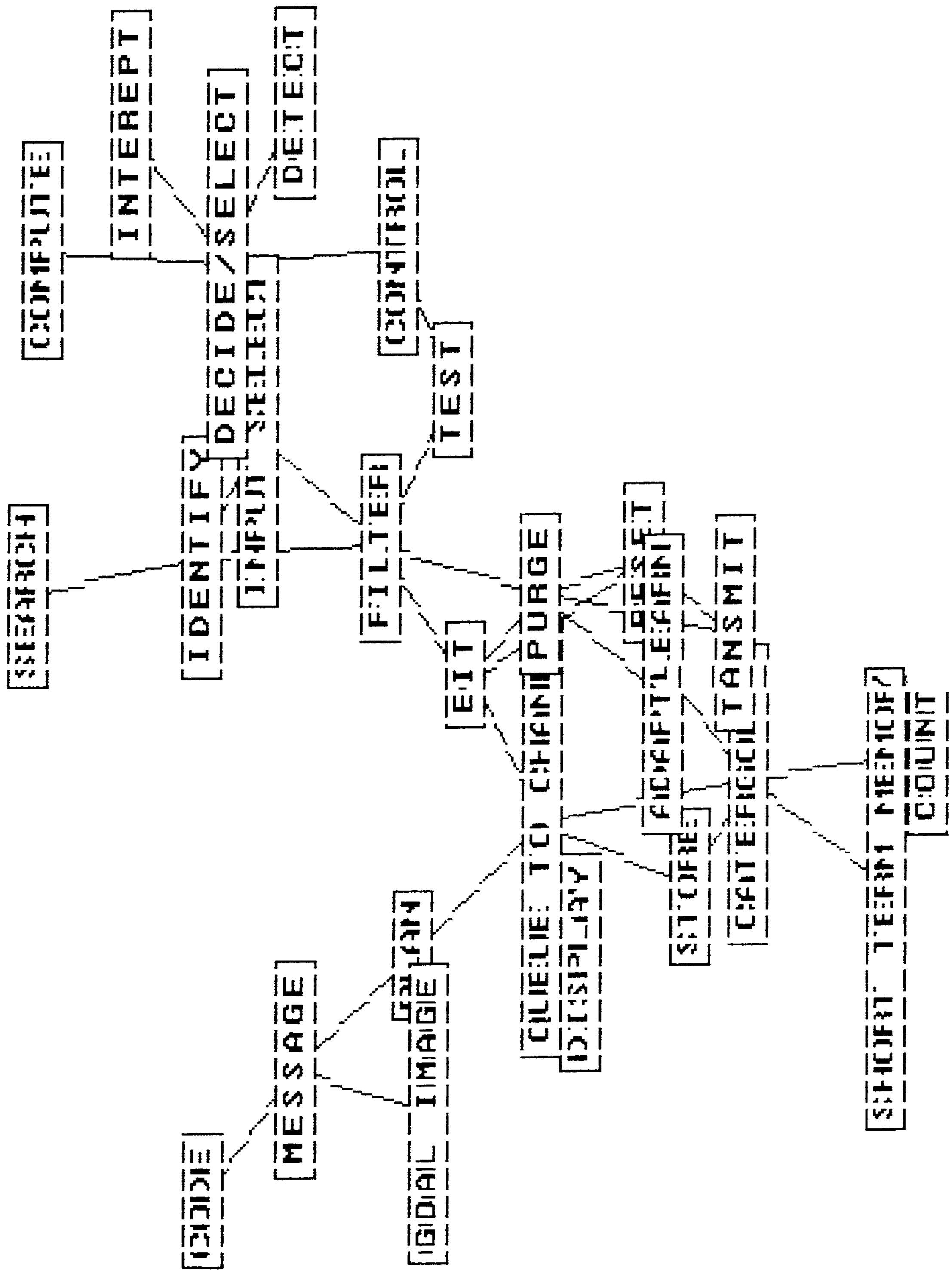


Figure E-23. Subject 23 for Q1.

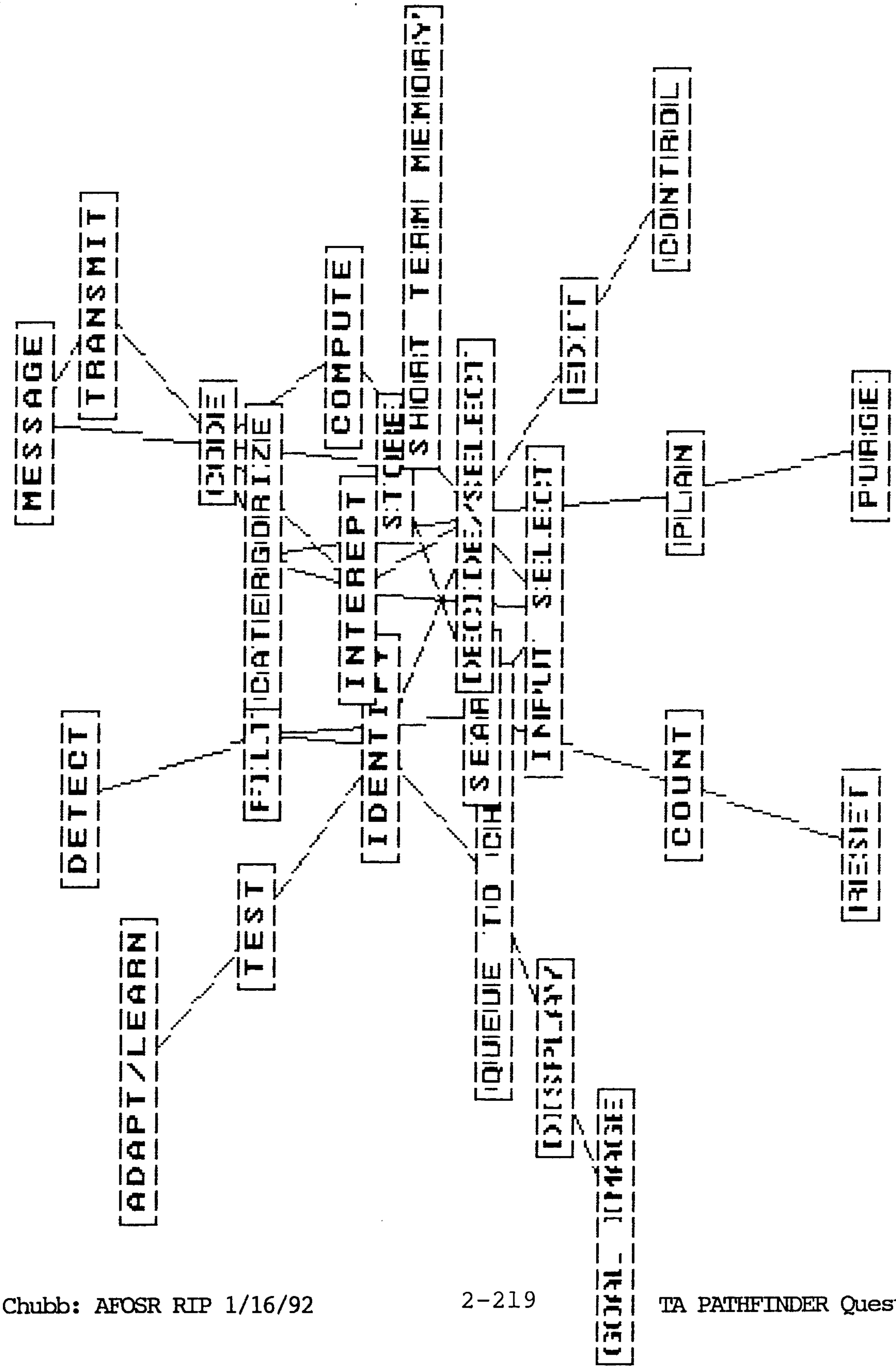


Figure E-24. Subject 24 for Q1.

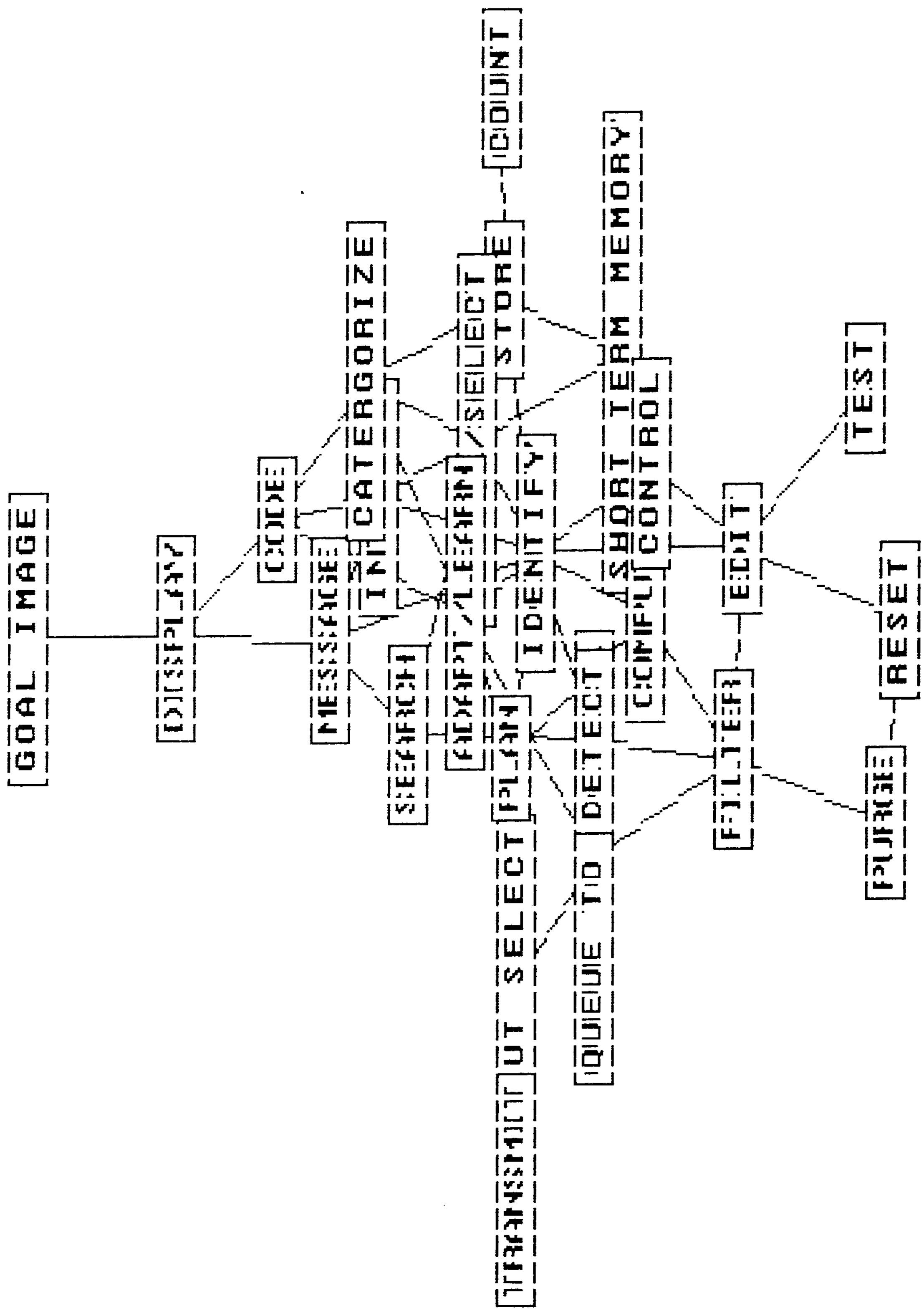


Figure E-25. Subject 1 for Q2.

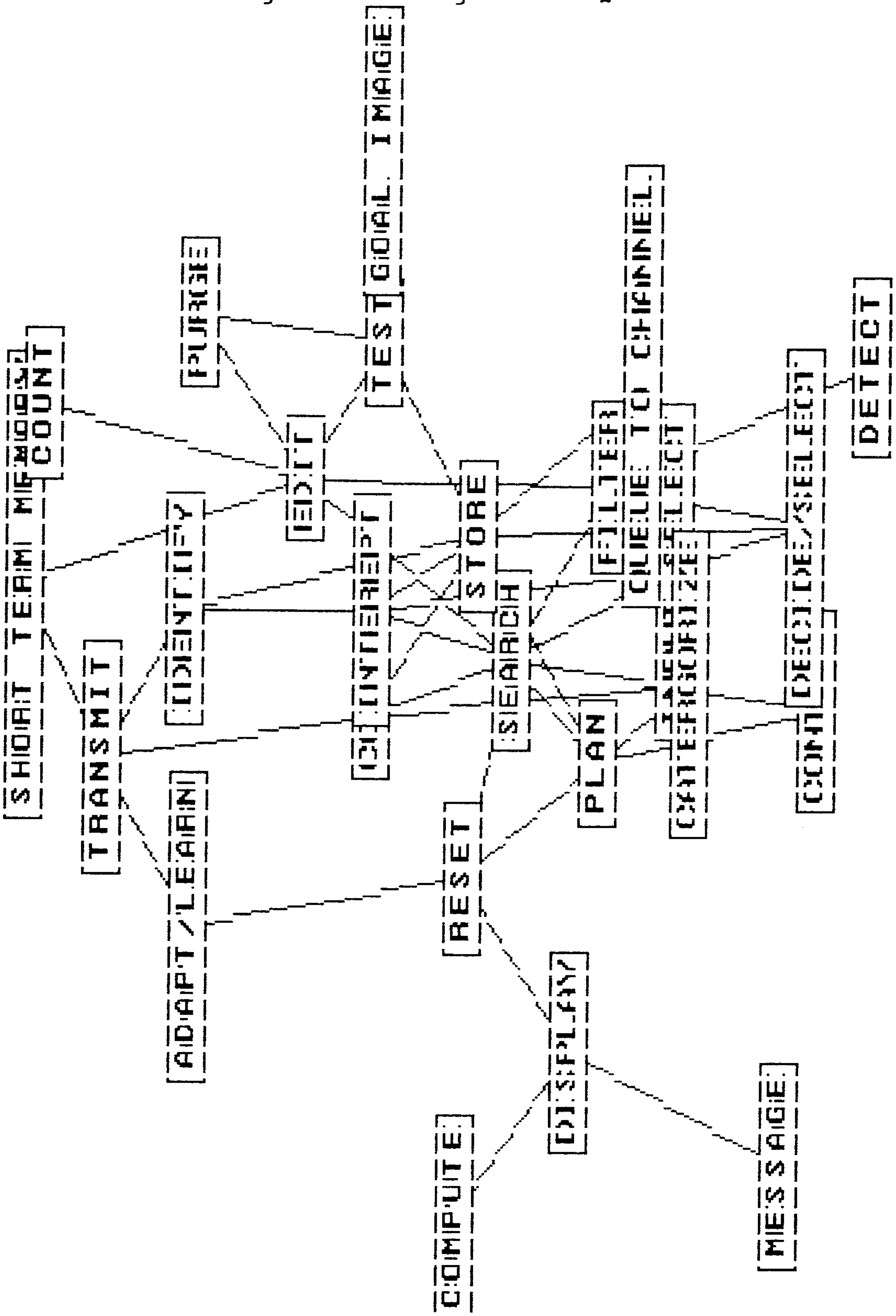


Figure E-26. Subject 2 for Q2.

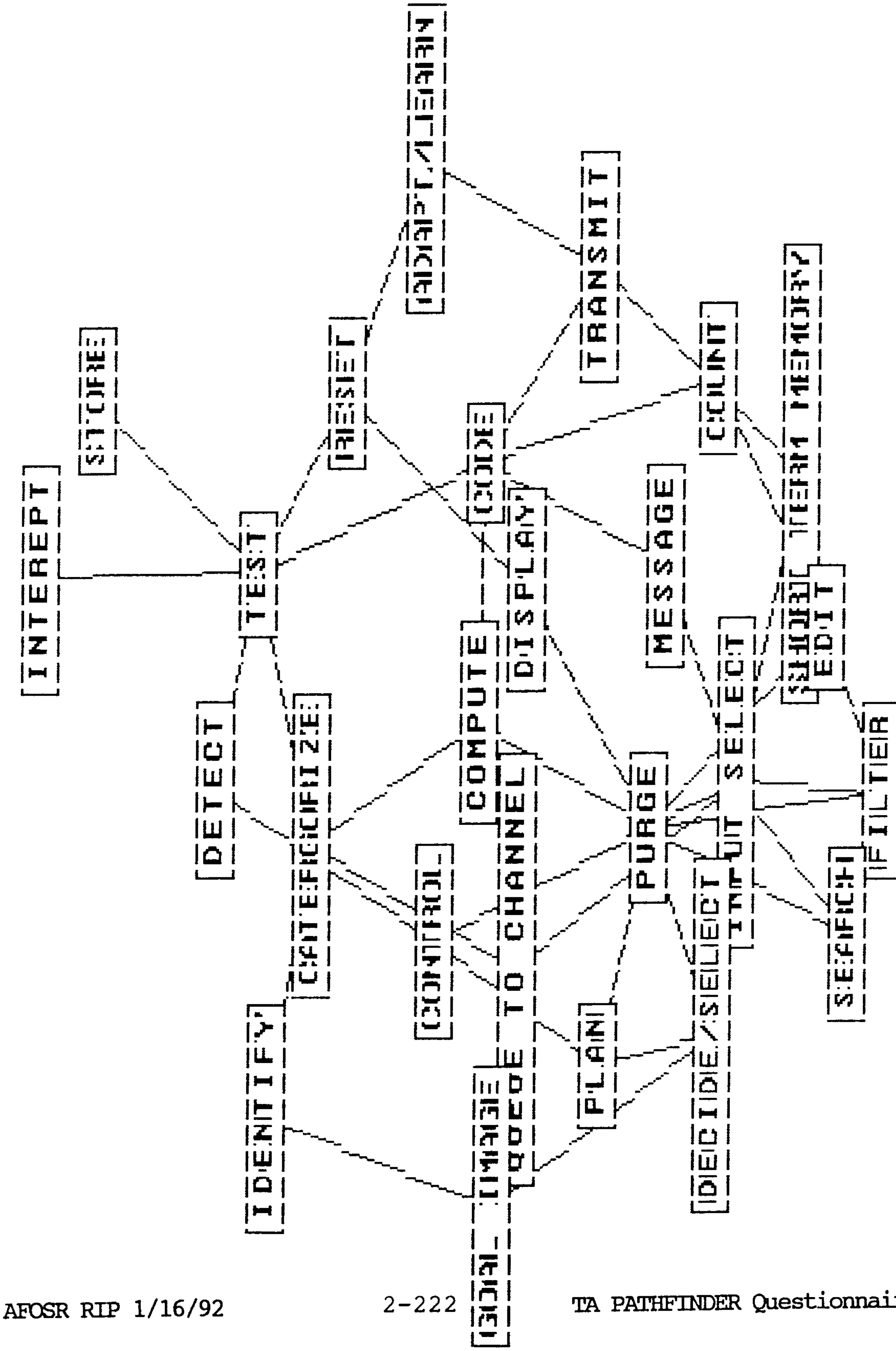


Figure E-27. Subject 4 for Q2.

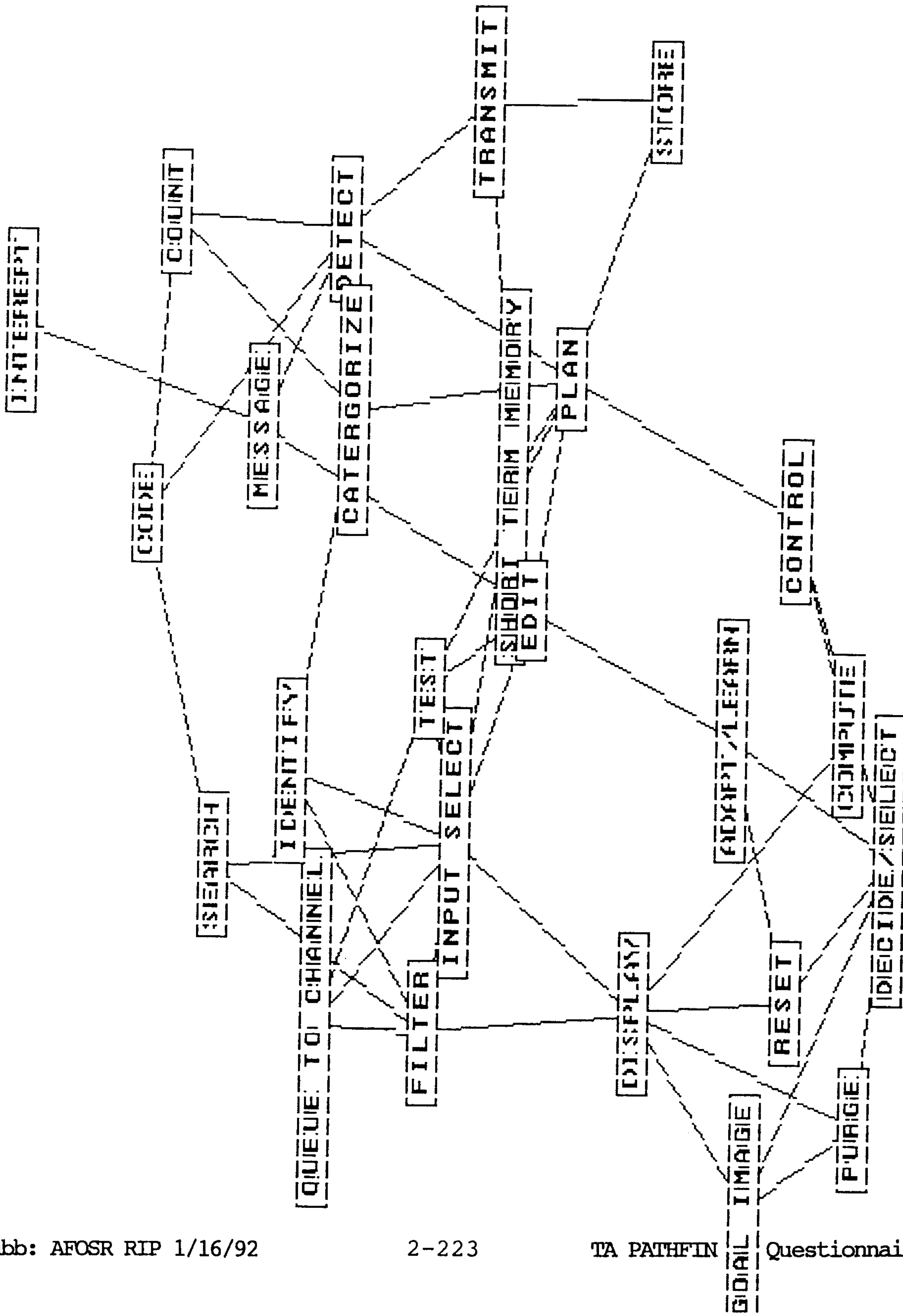


Figure E-28. Subject 5 for Q2.

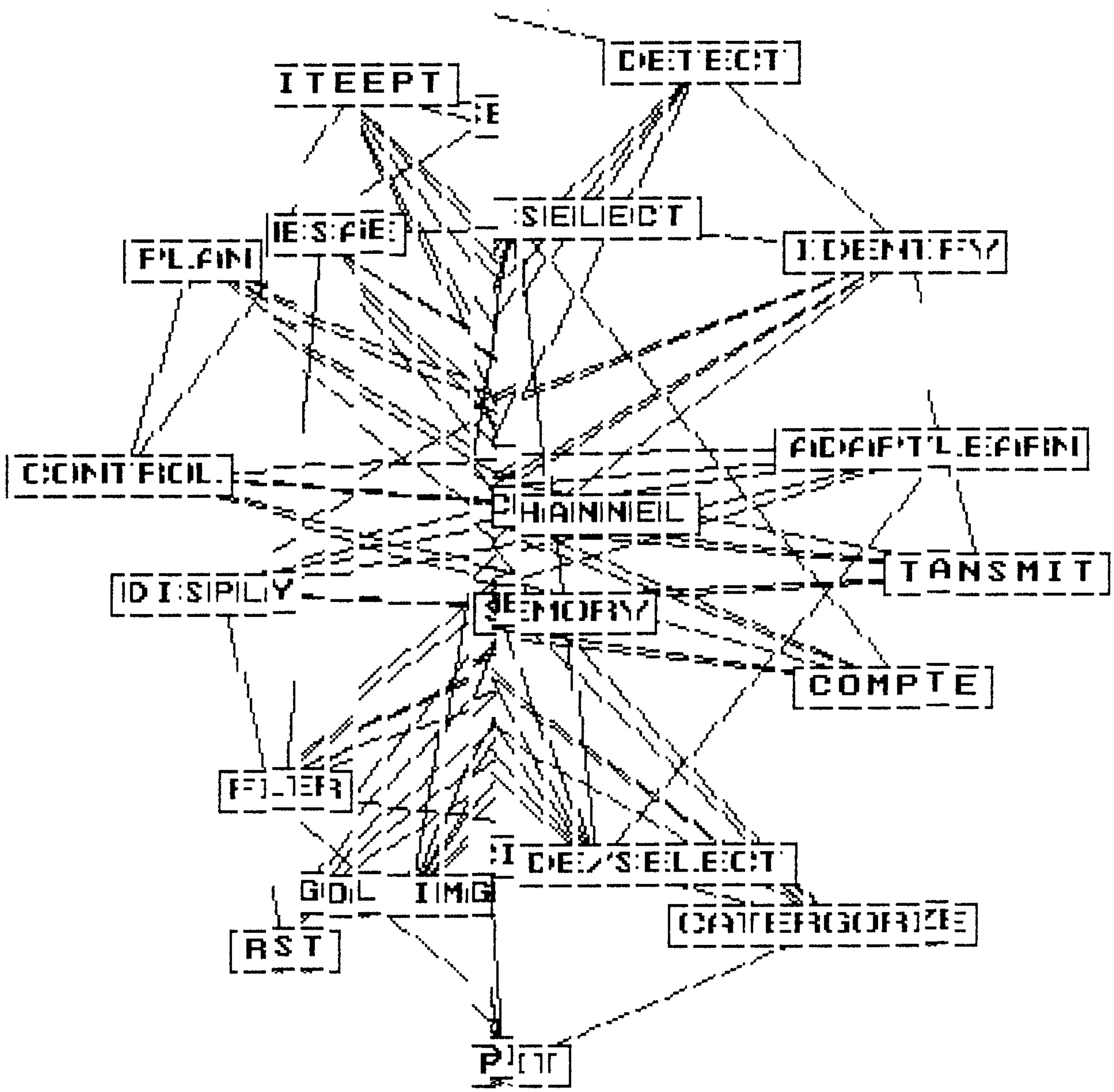


Figure E-29. Subject 6 for Q2.

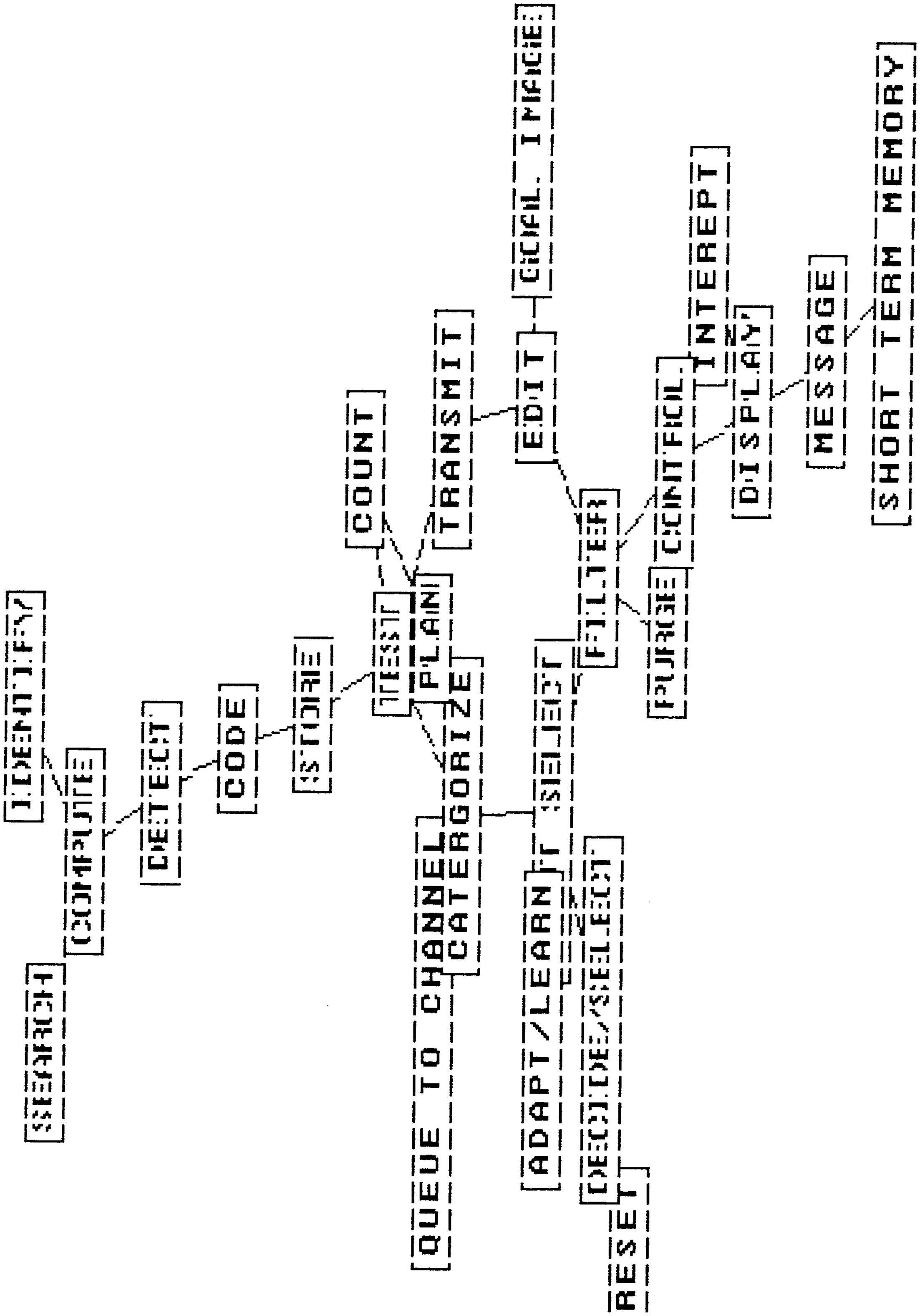


Figure E-30. Subject 7 for Q2.

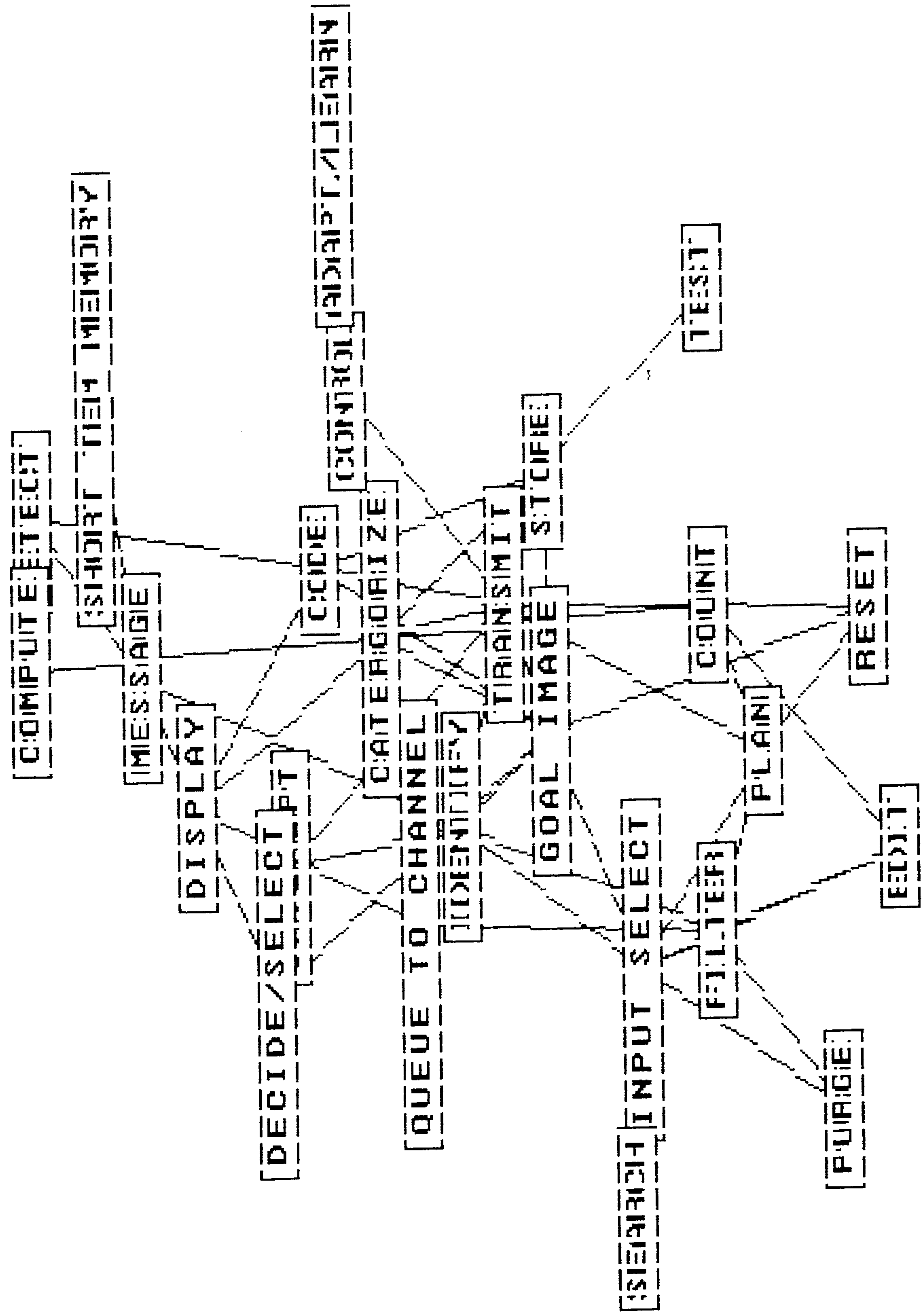


Figure E-31. Subject 8 for Q2.

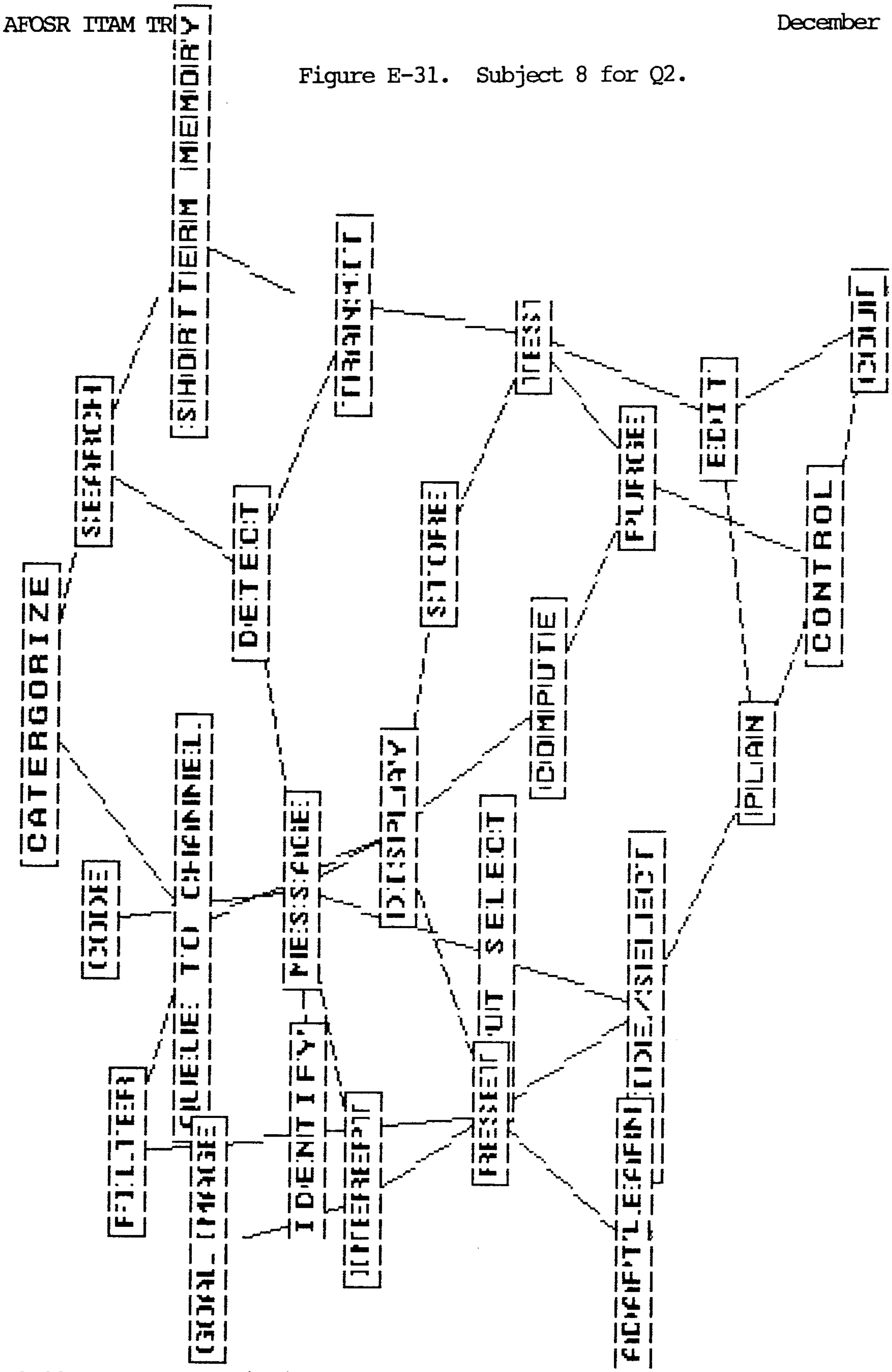
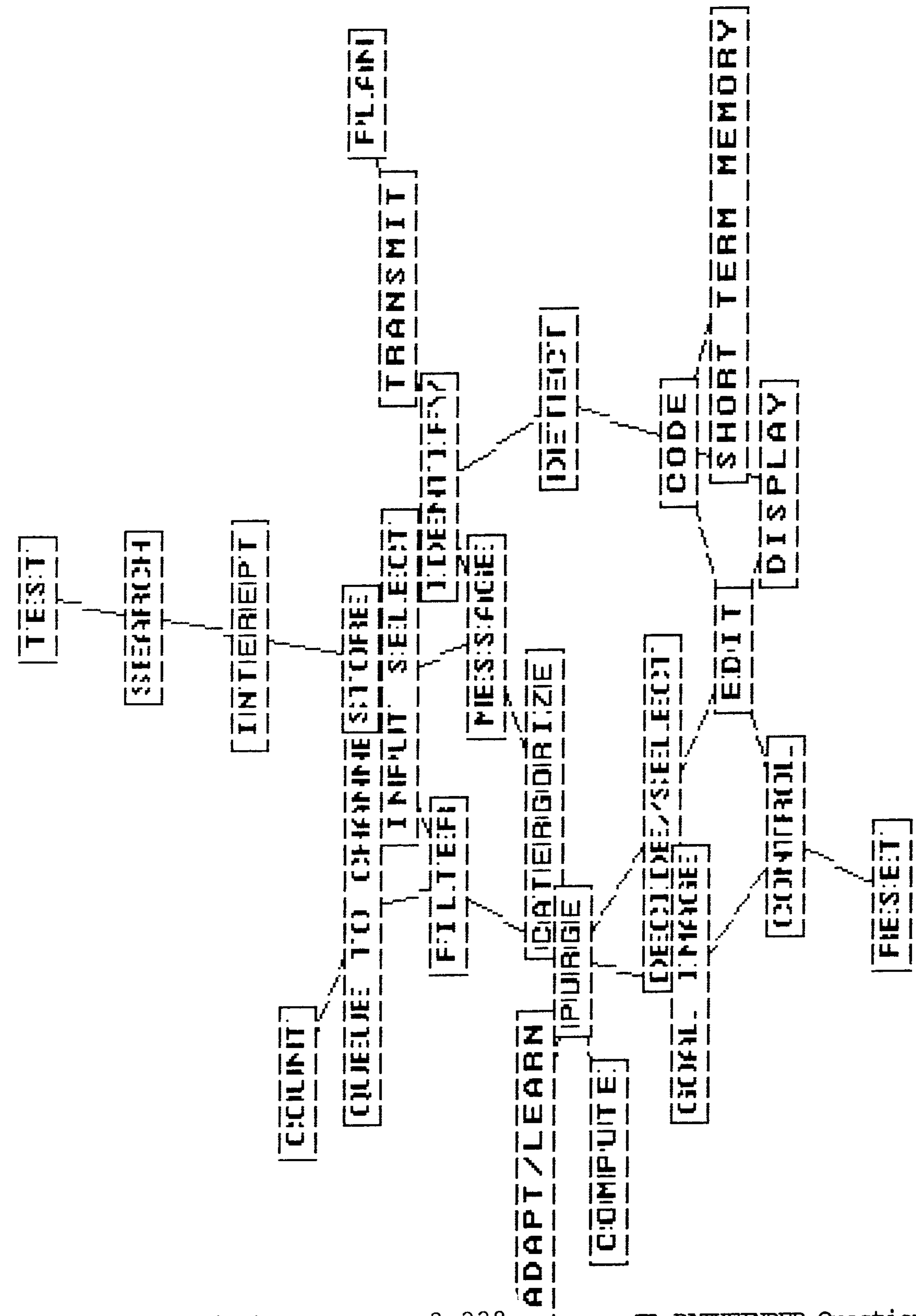


Figure E-32. Subject 10 for Q2.



E-33. Subject 11 for Q2.

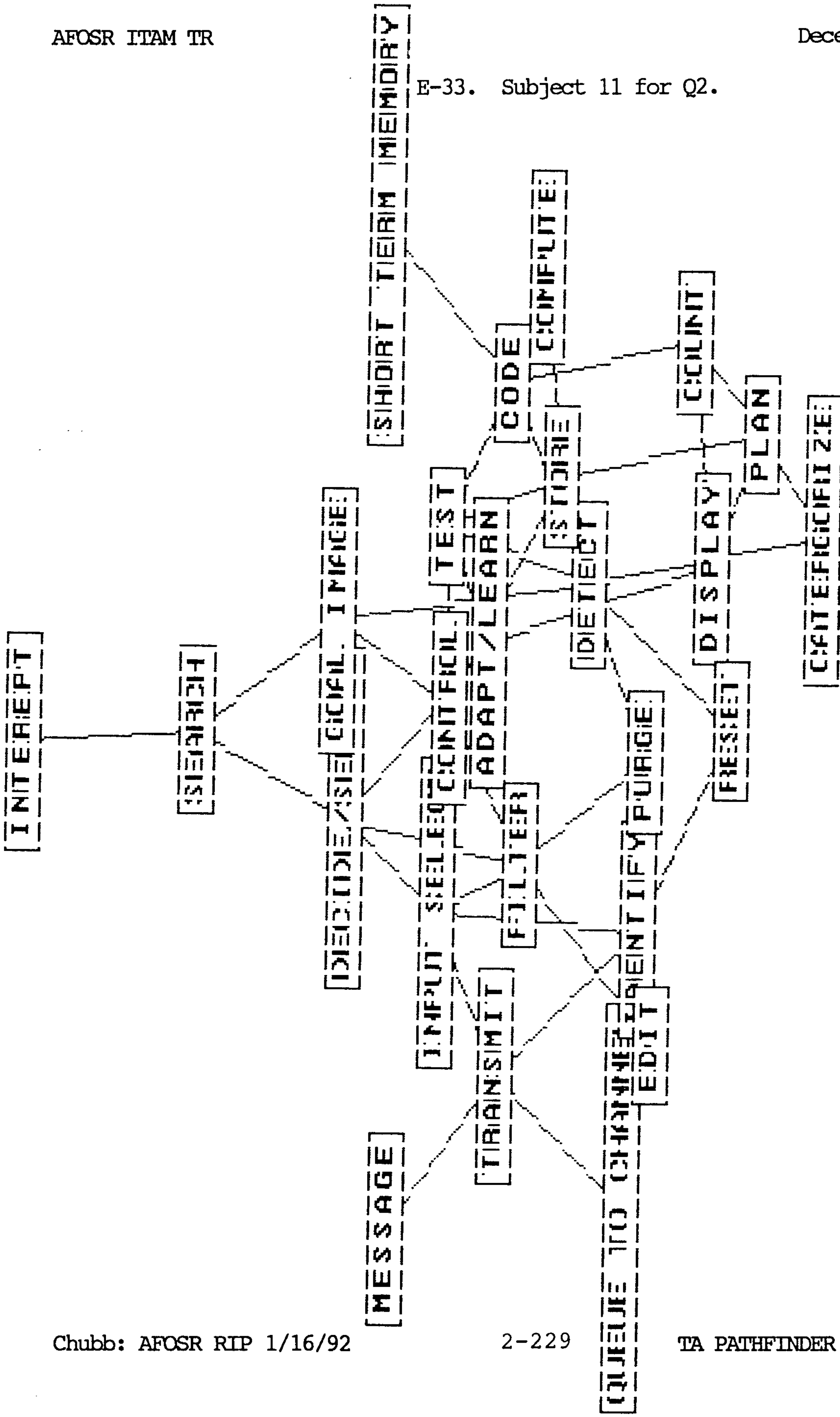


Figure E-34. Subject 12 for Q2.

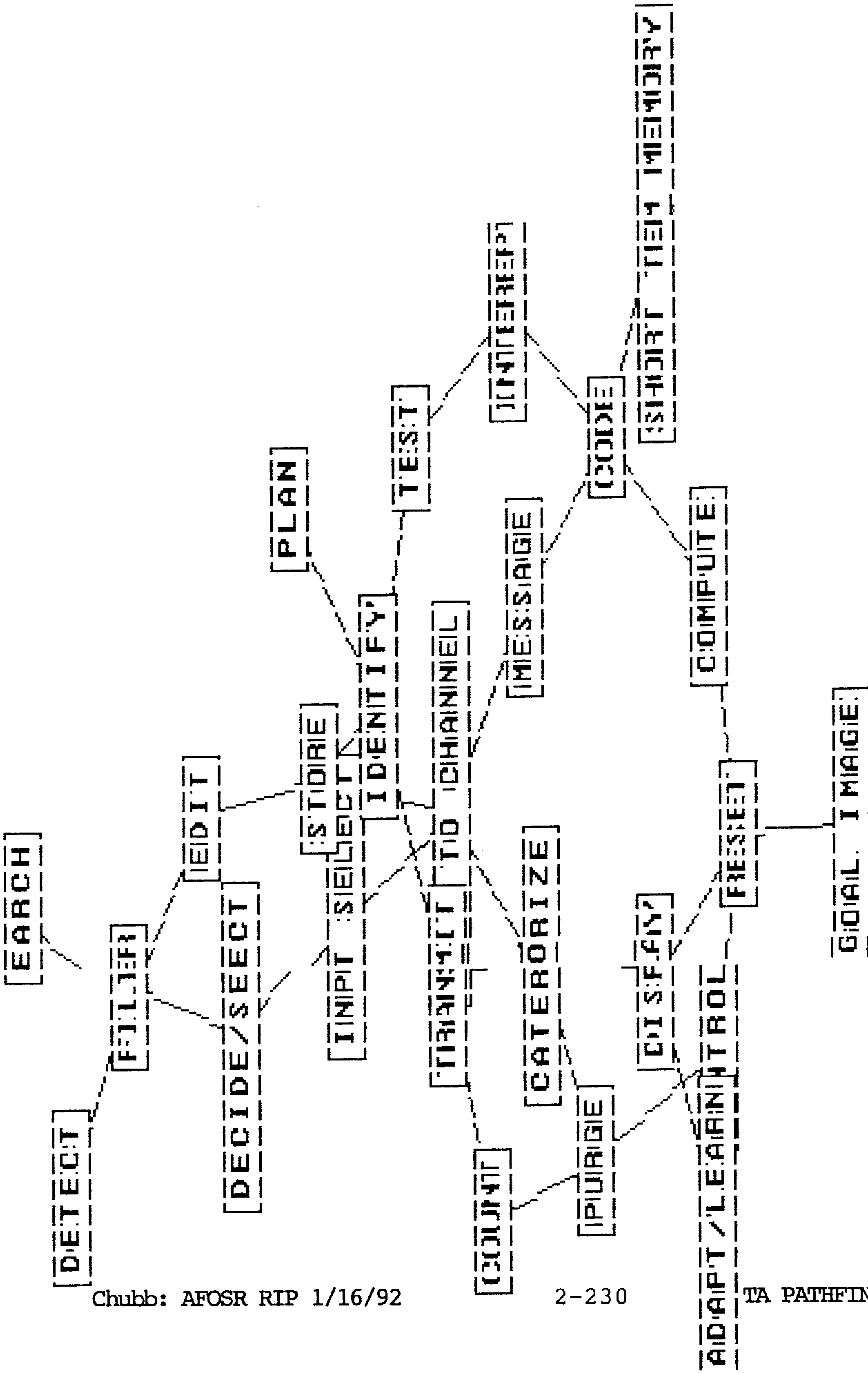


Figure E-35. Subject 13 for Q2.

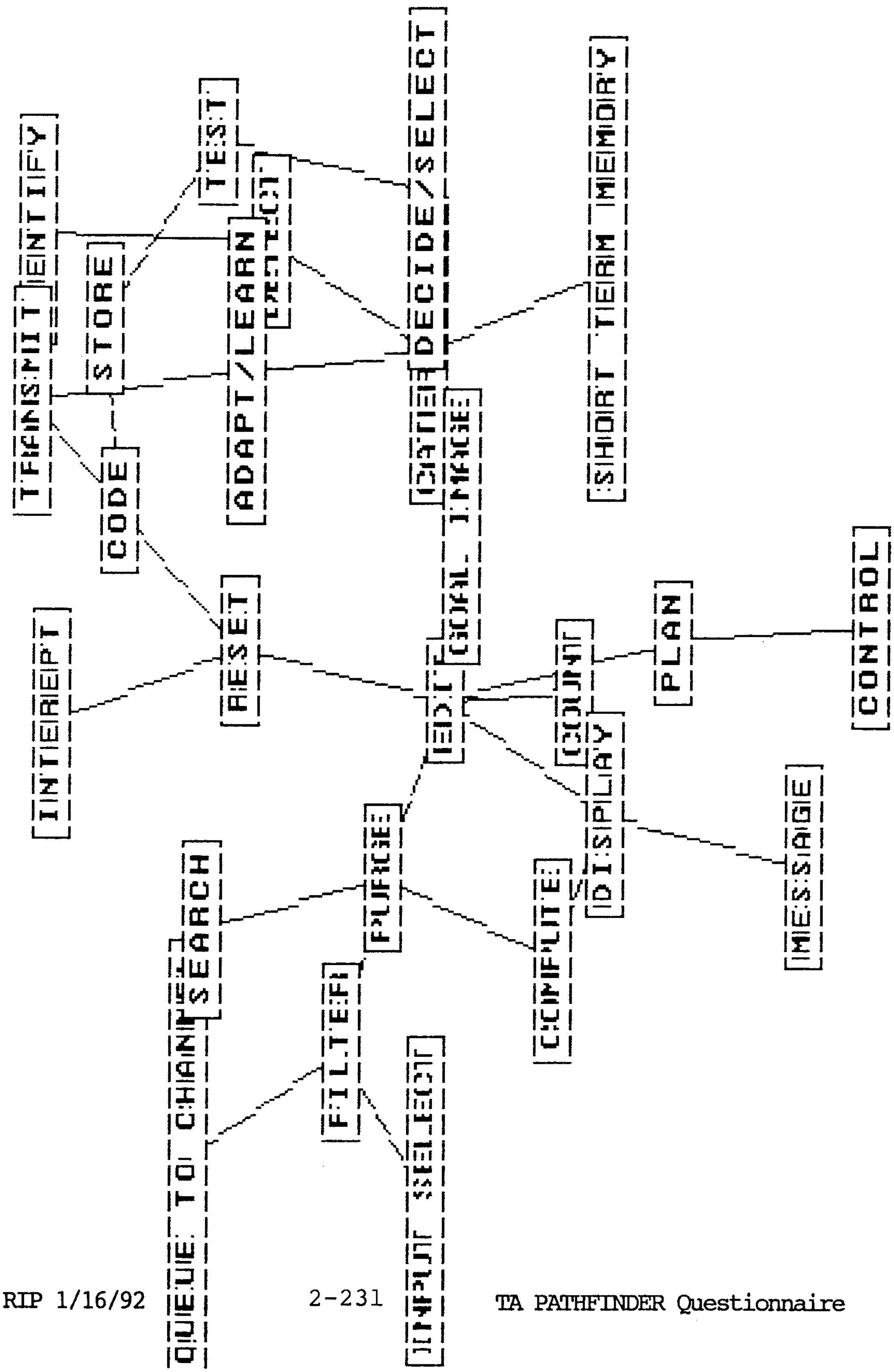


Figure E-36. Subject 15 for Q2.

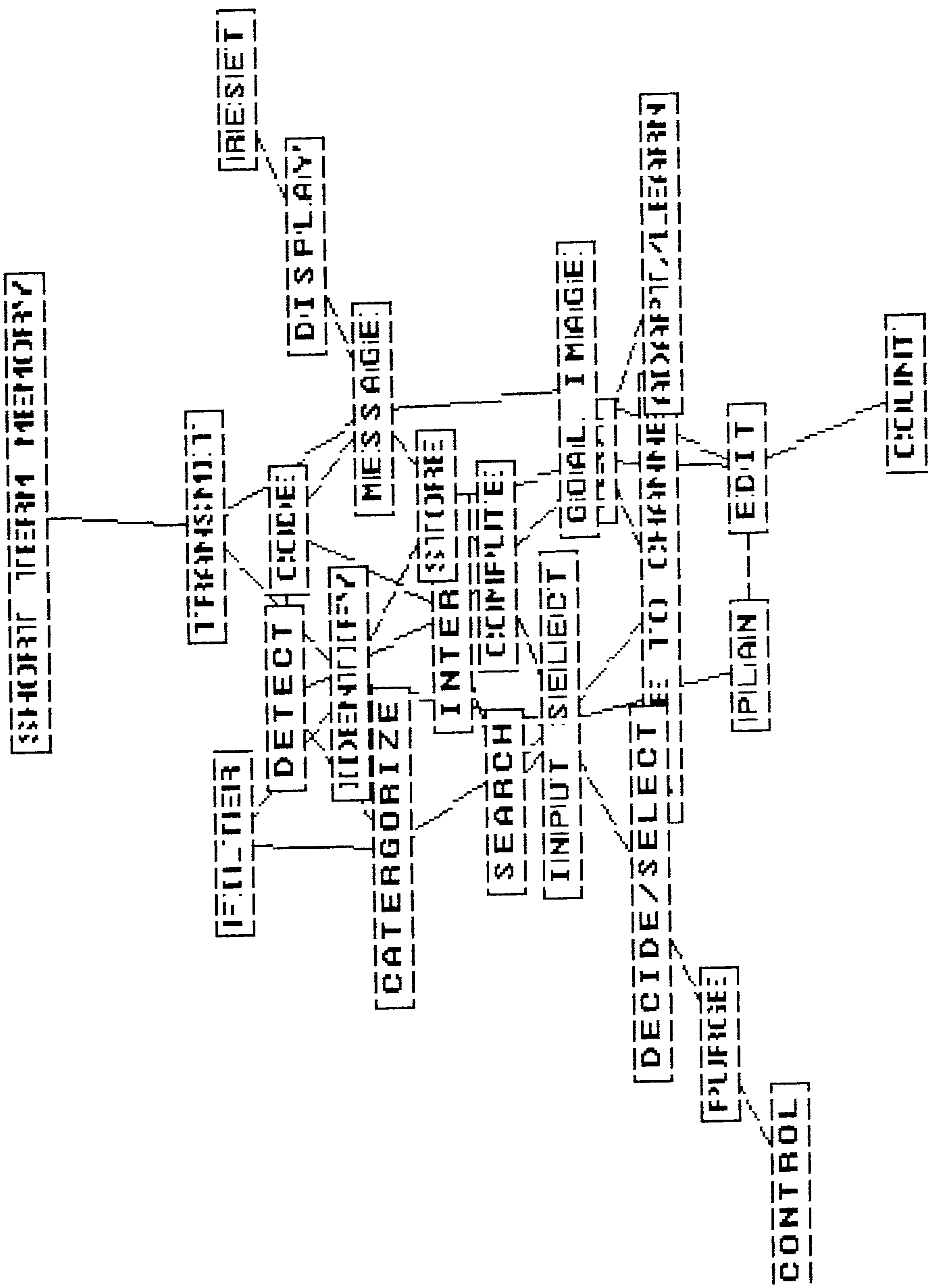


Figure E-37. Subject 19 for Q2.

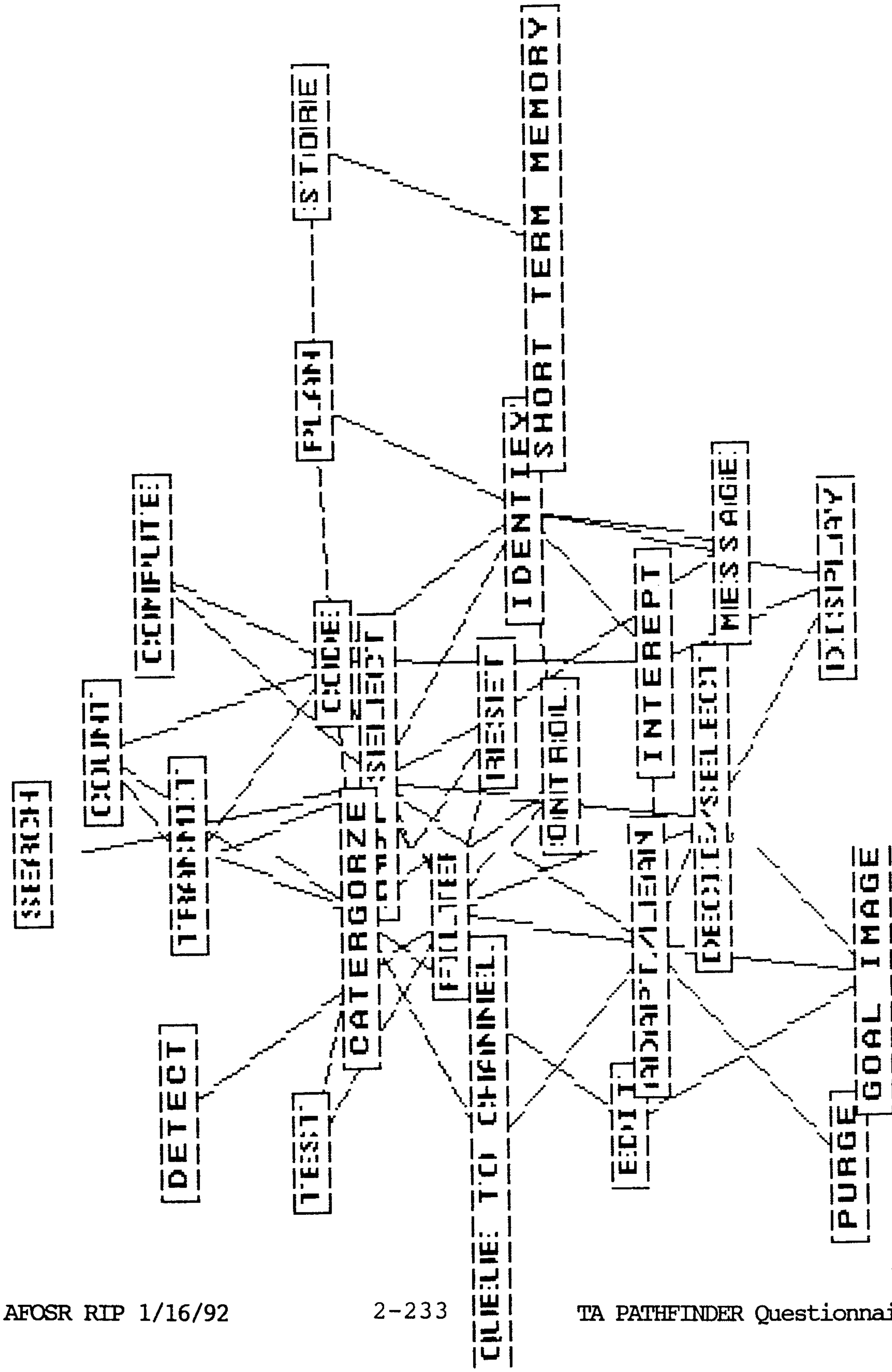


Figure E-38. Subject 20 for Q2.

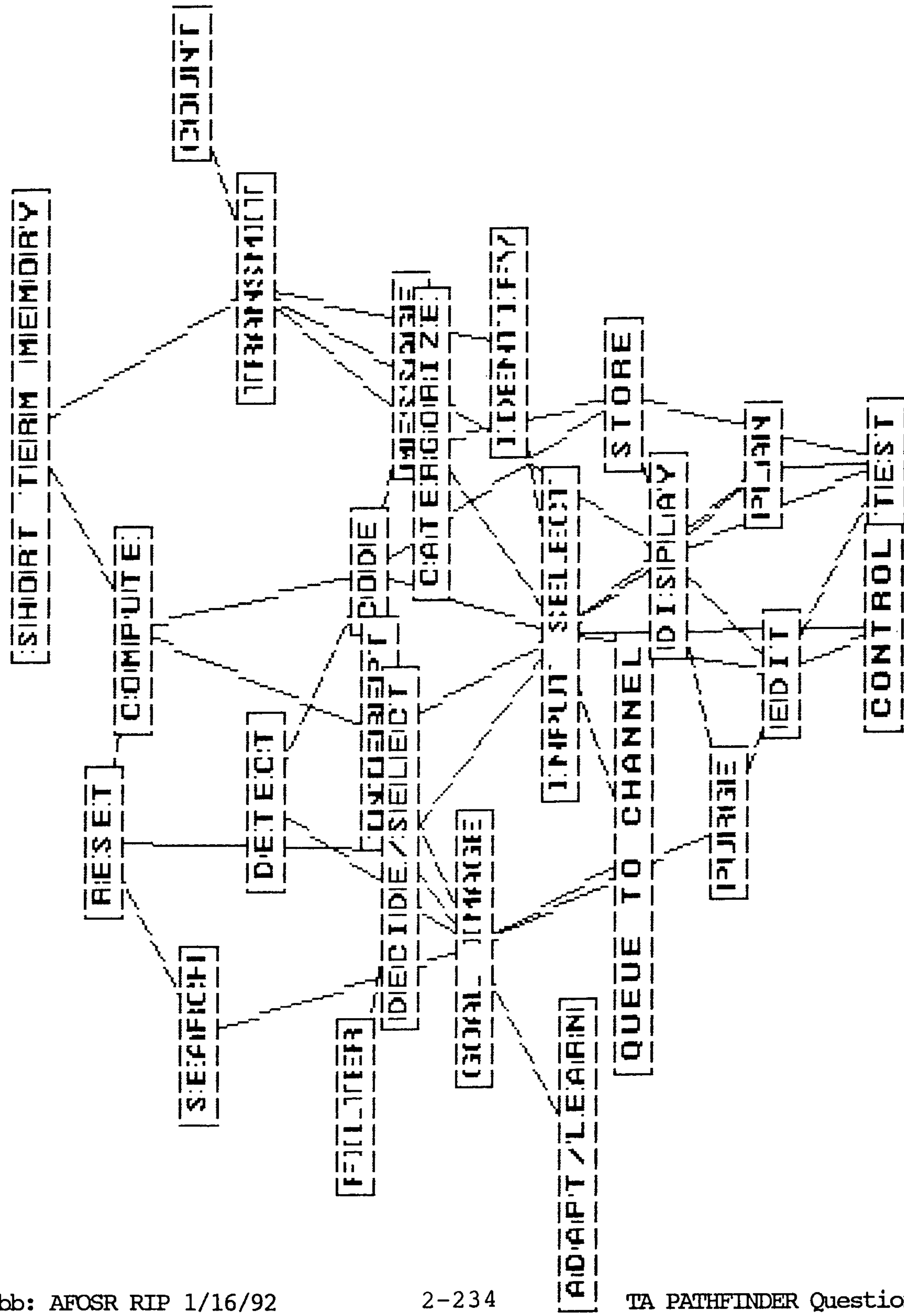


Figure E-39. Subject 21 for Q2.

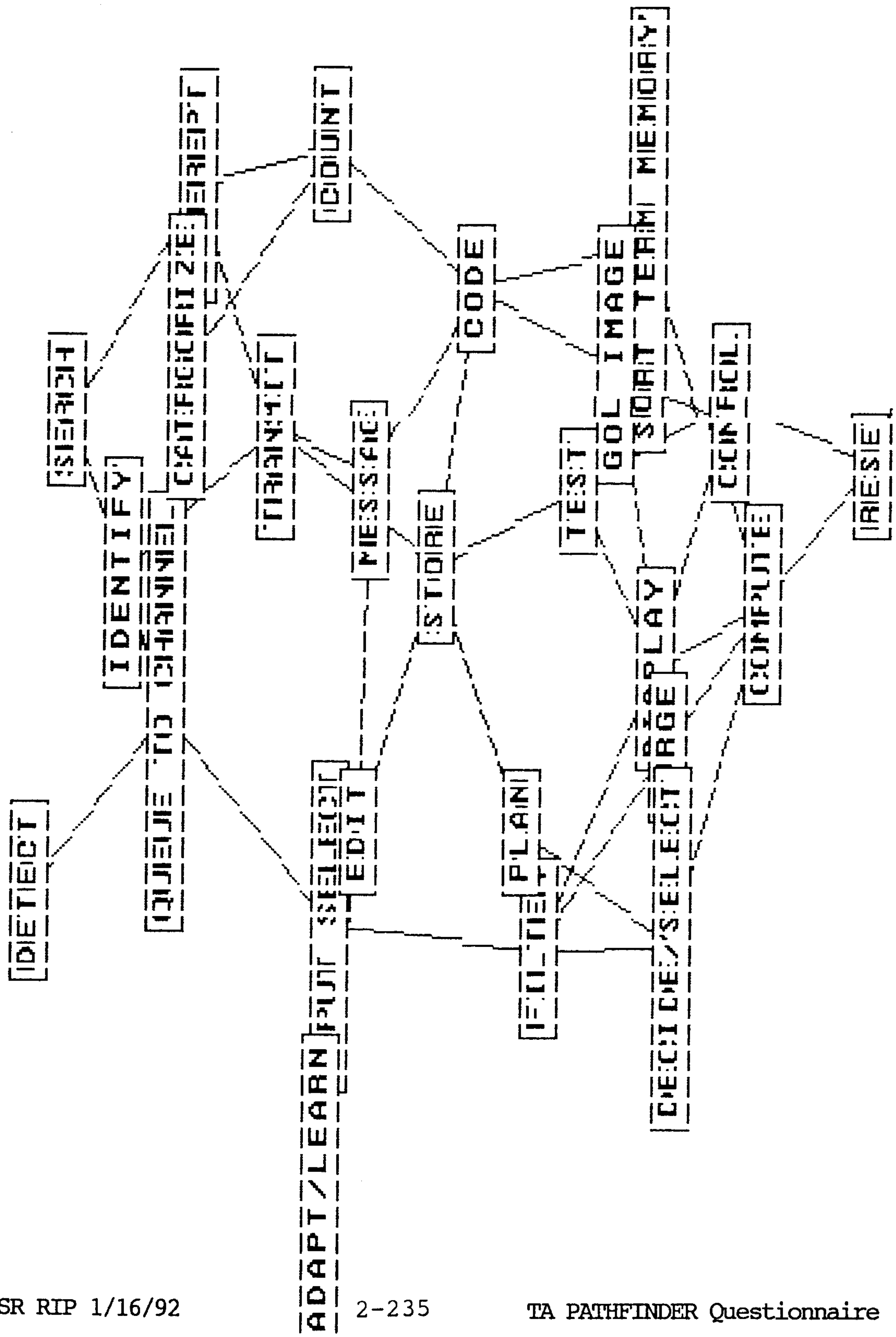


Figure E-40. Subject 22 for Q2.

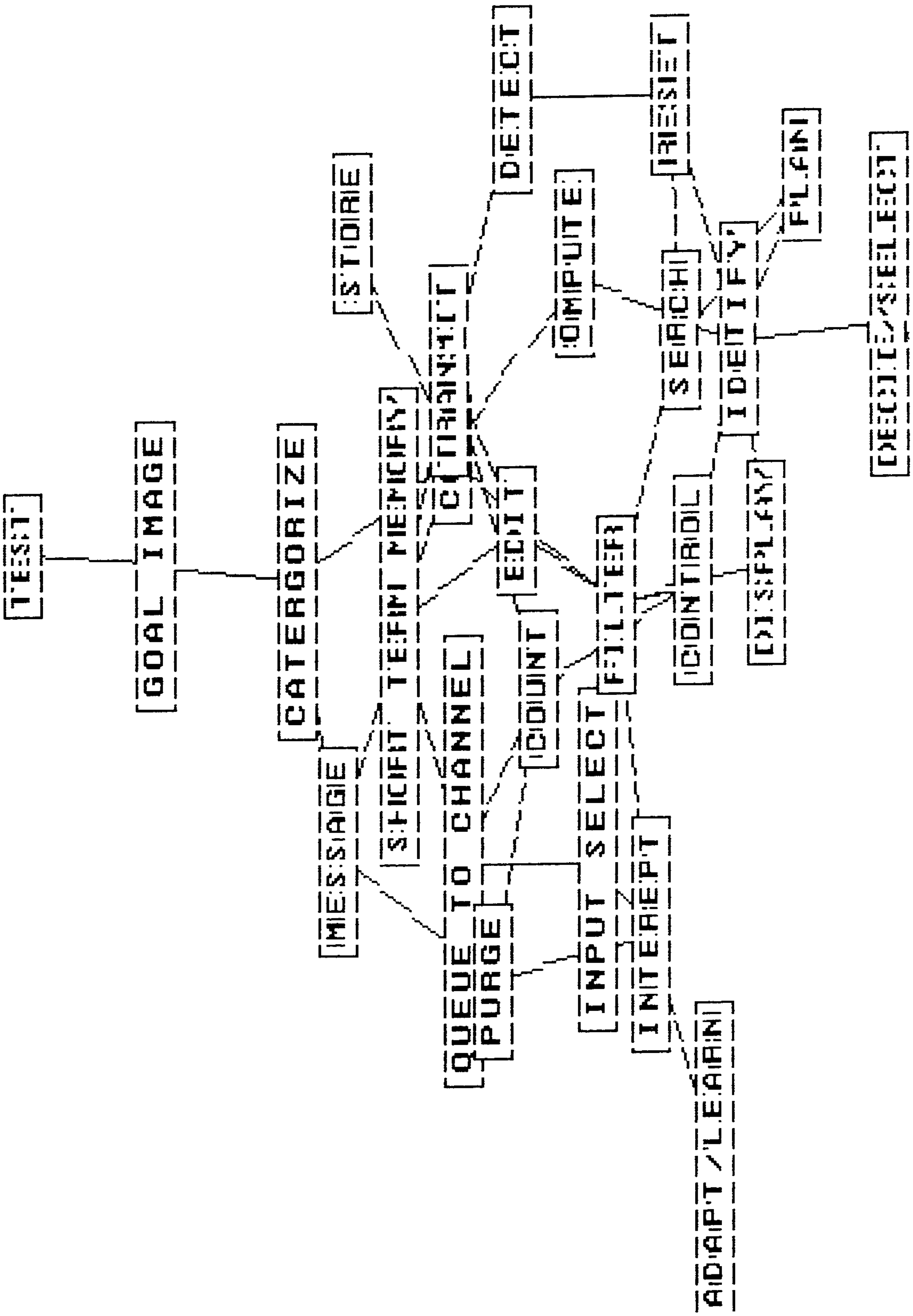


Figure E-41. Subject 23 for Q2.

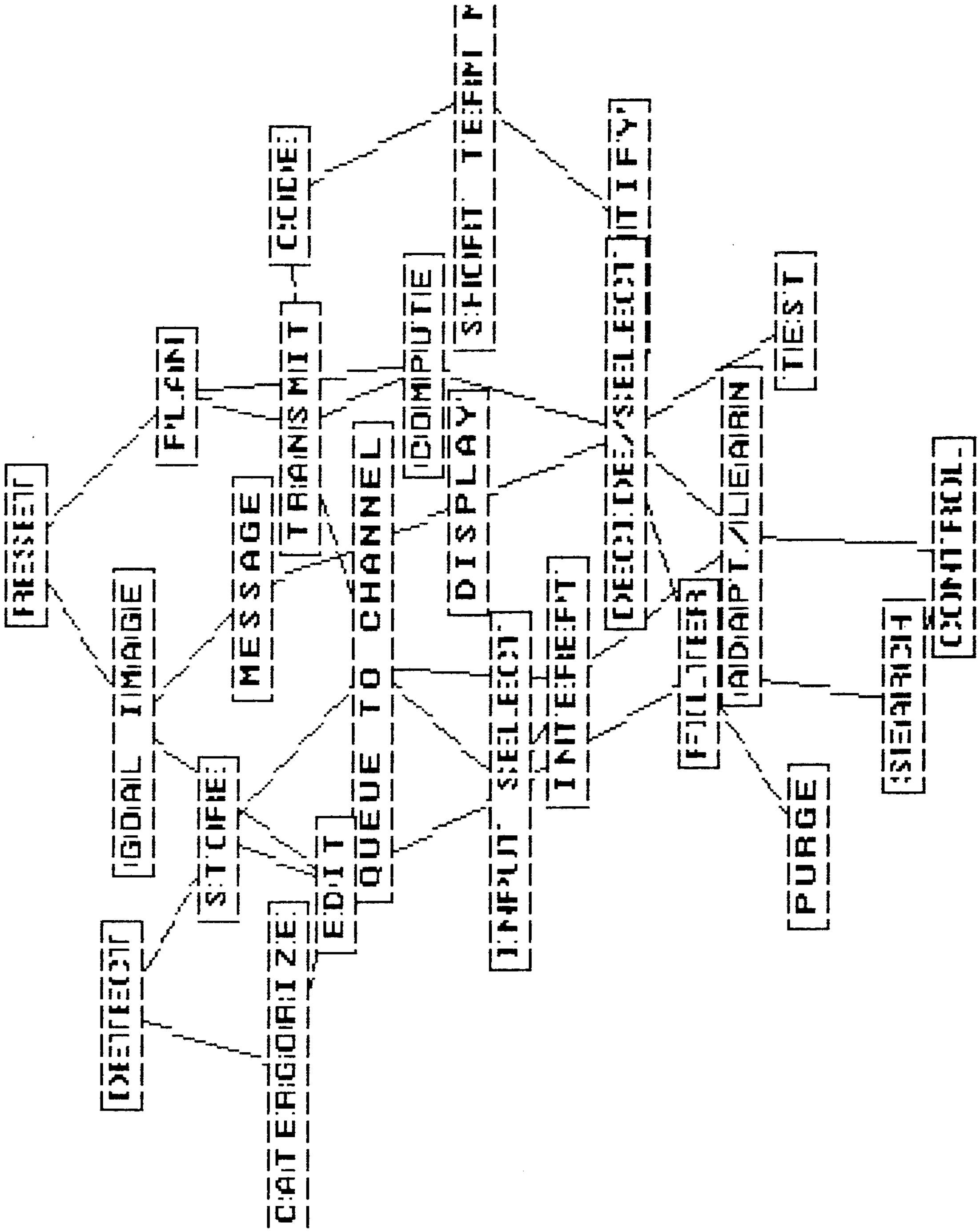


Figure E-42. Subject 24 for Q2.

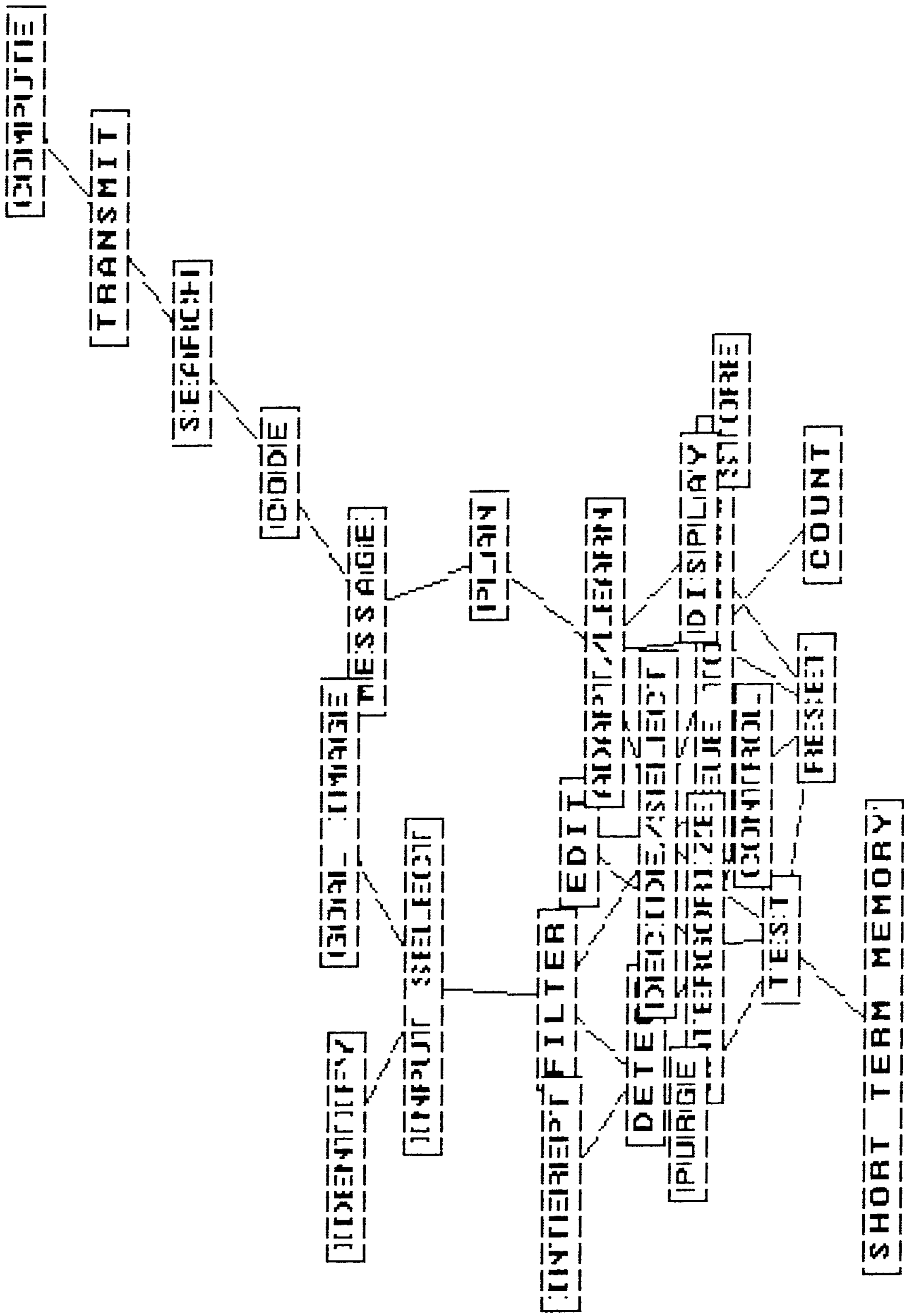


Figure E-43. Subject 2 for Q3.

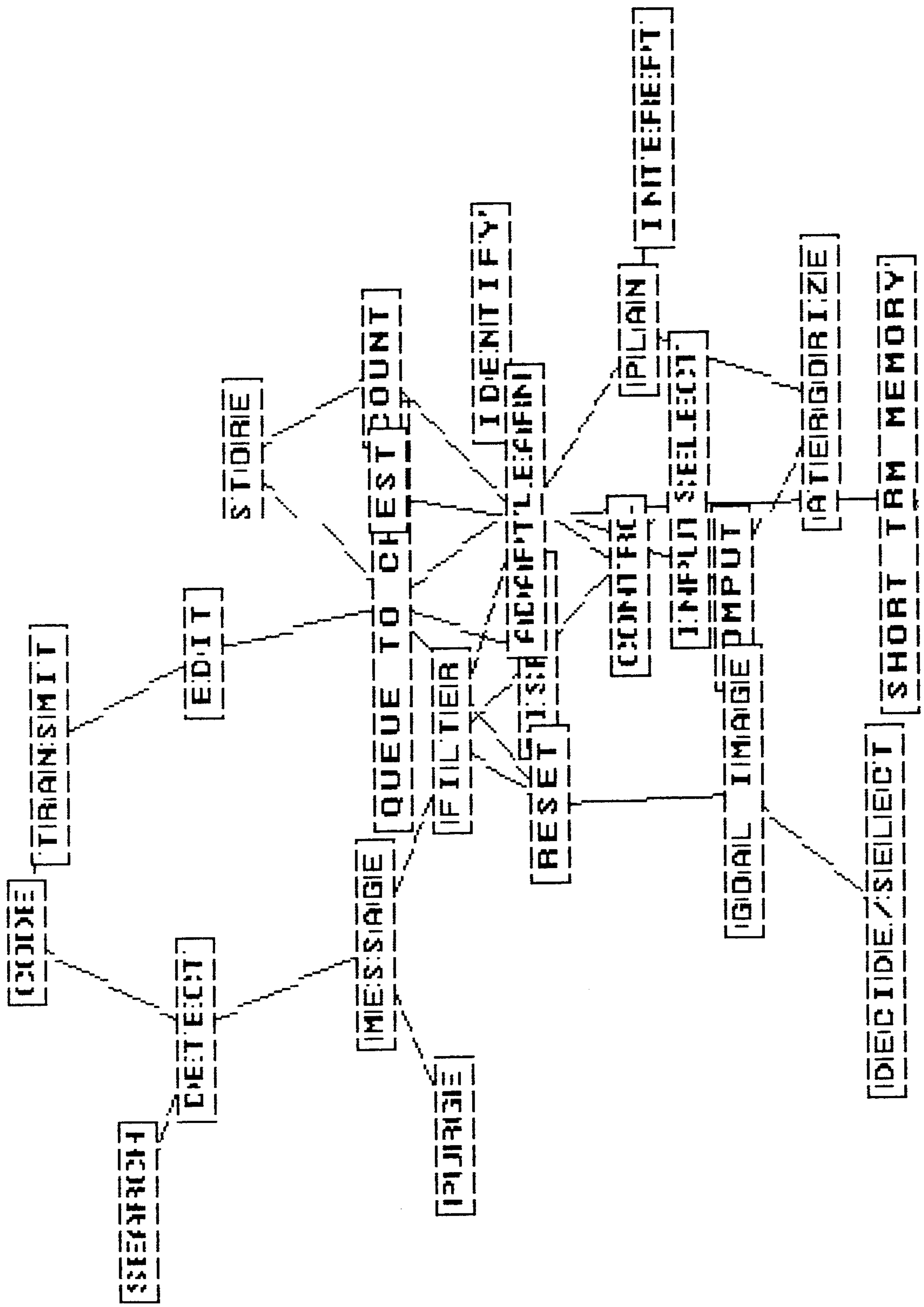


Figure E-44. Subject 3 for Q3.

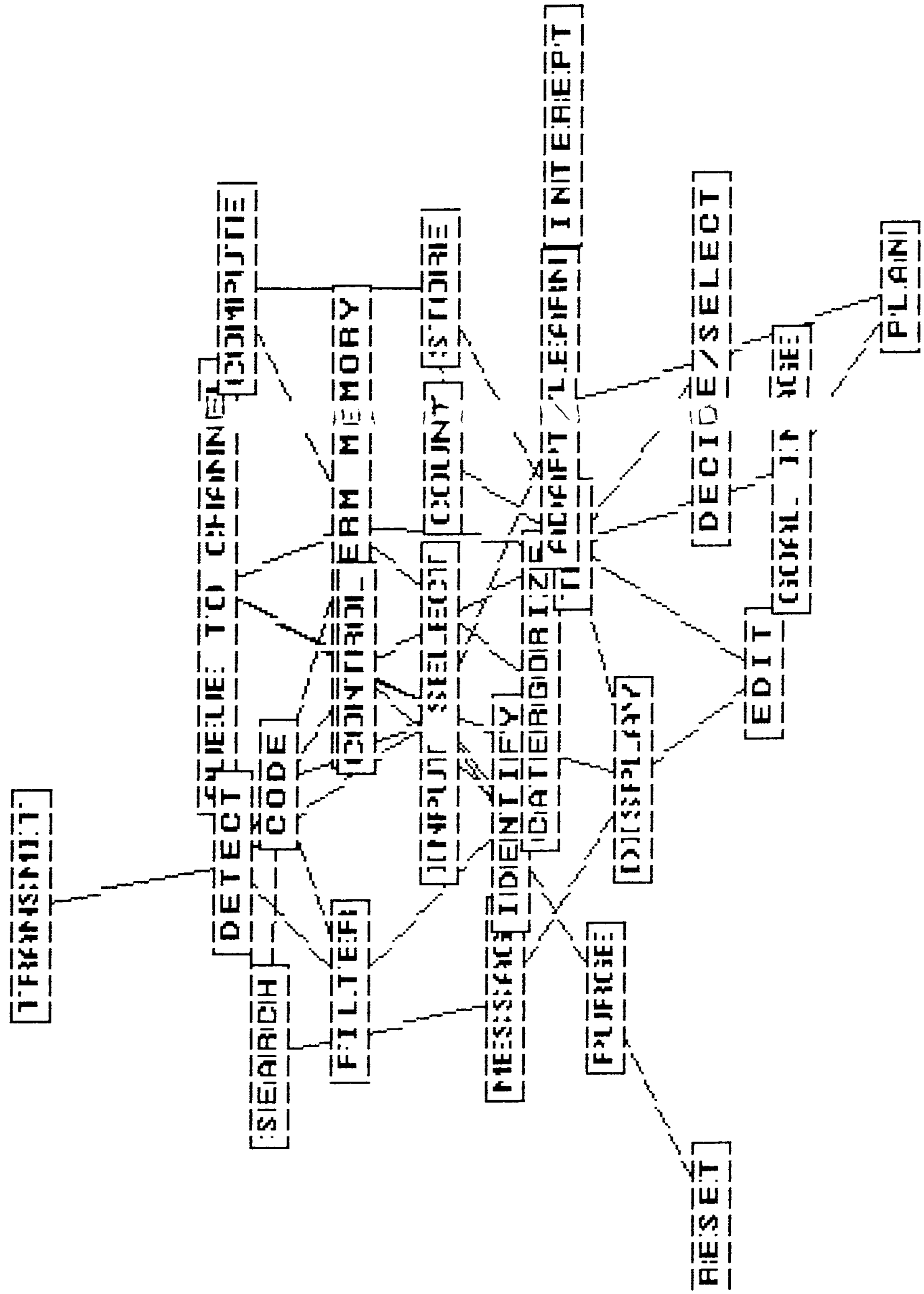


Figure E-45. Subject 8 for Q3.

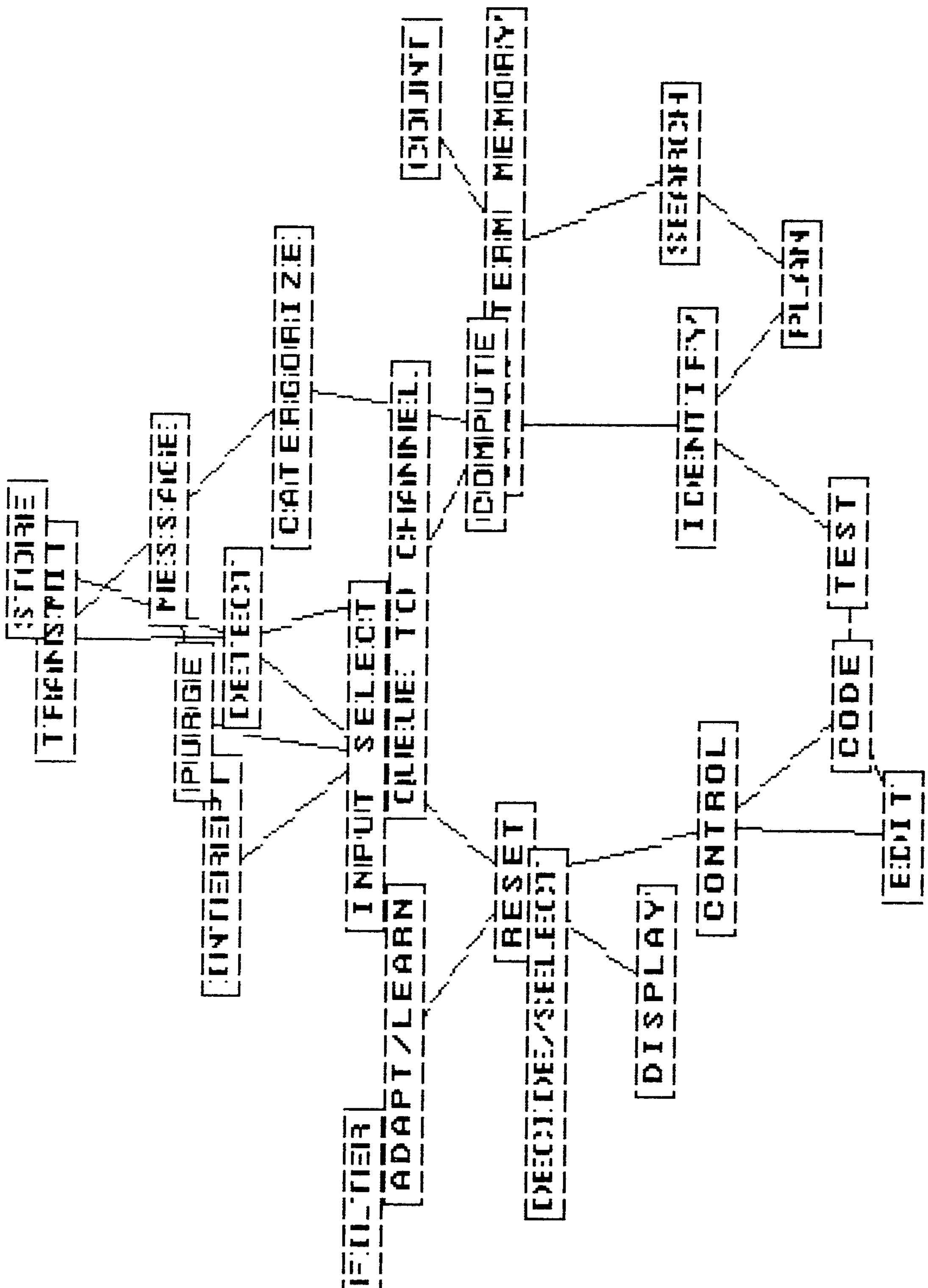


Figure E-46. Subject 12 for Q3.

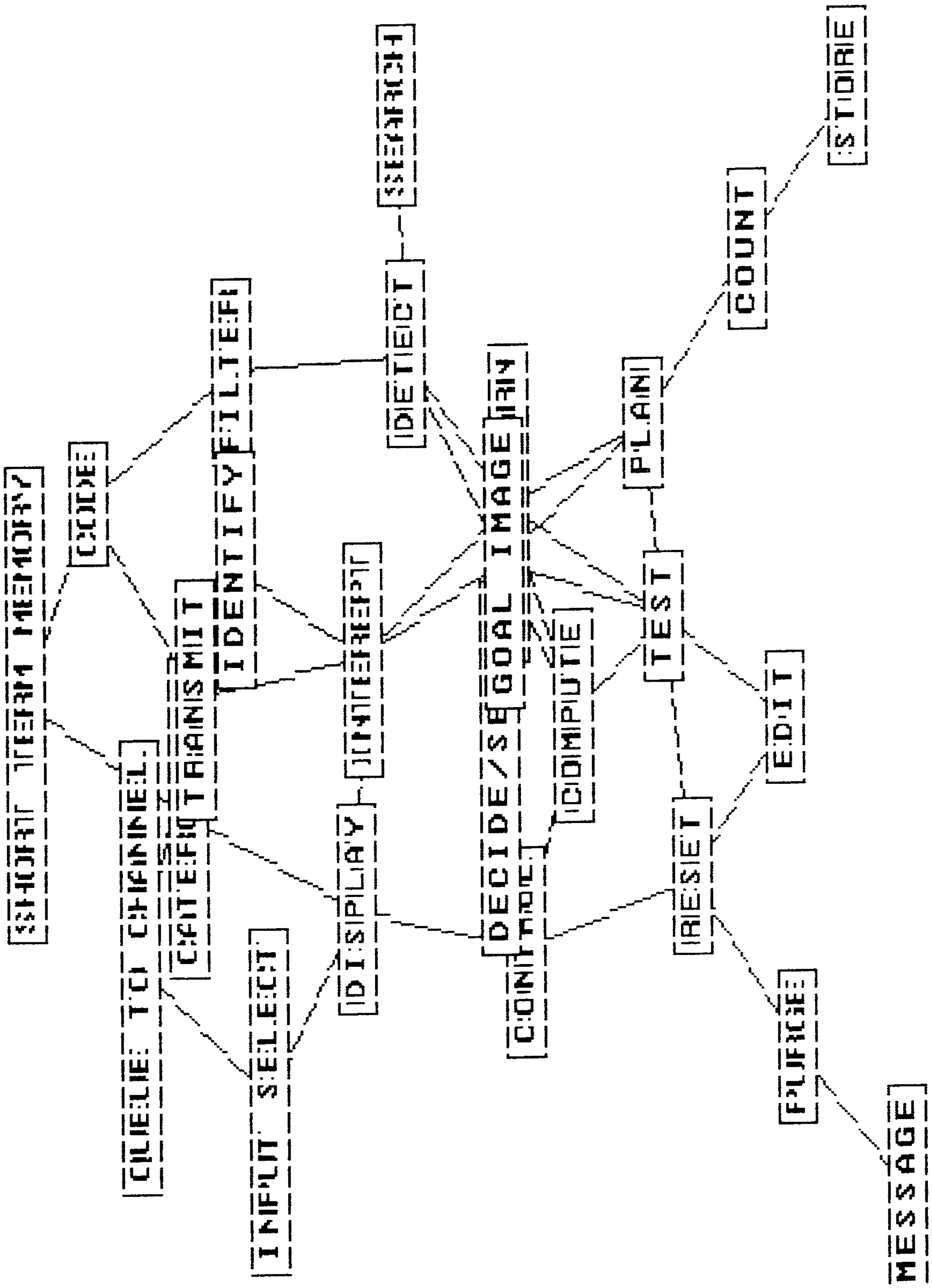


Figure E-47. Subject 13 for Q3.

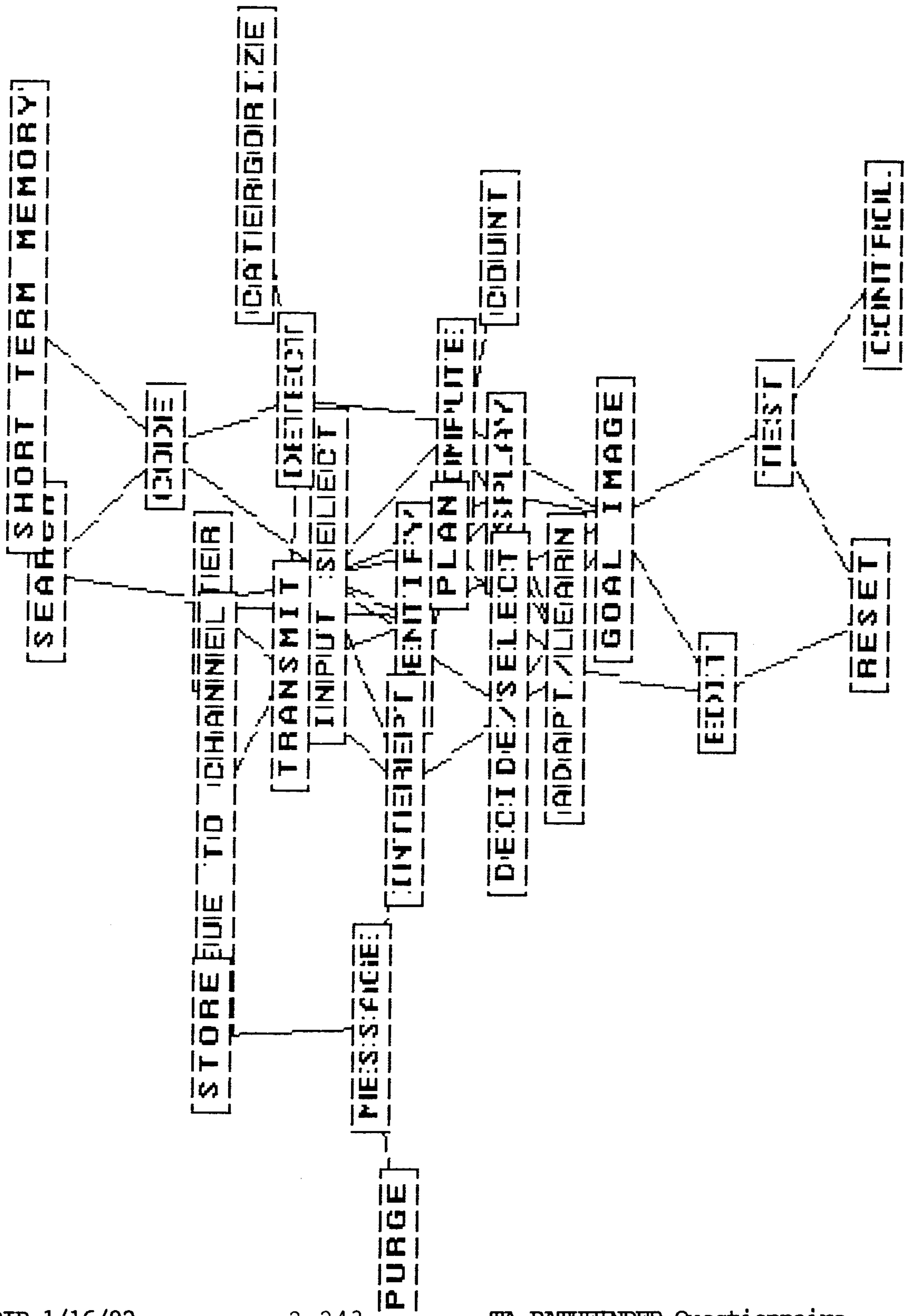


Figure E-48. Subject 19 for Q3.

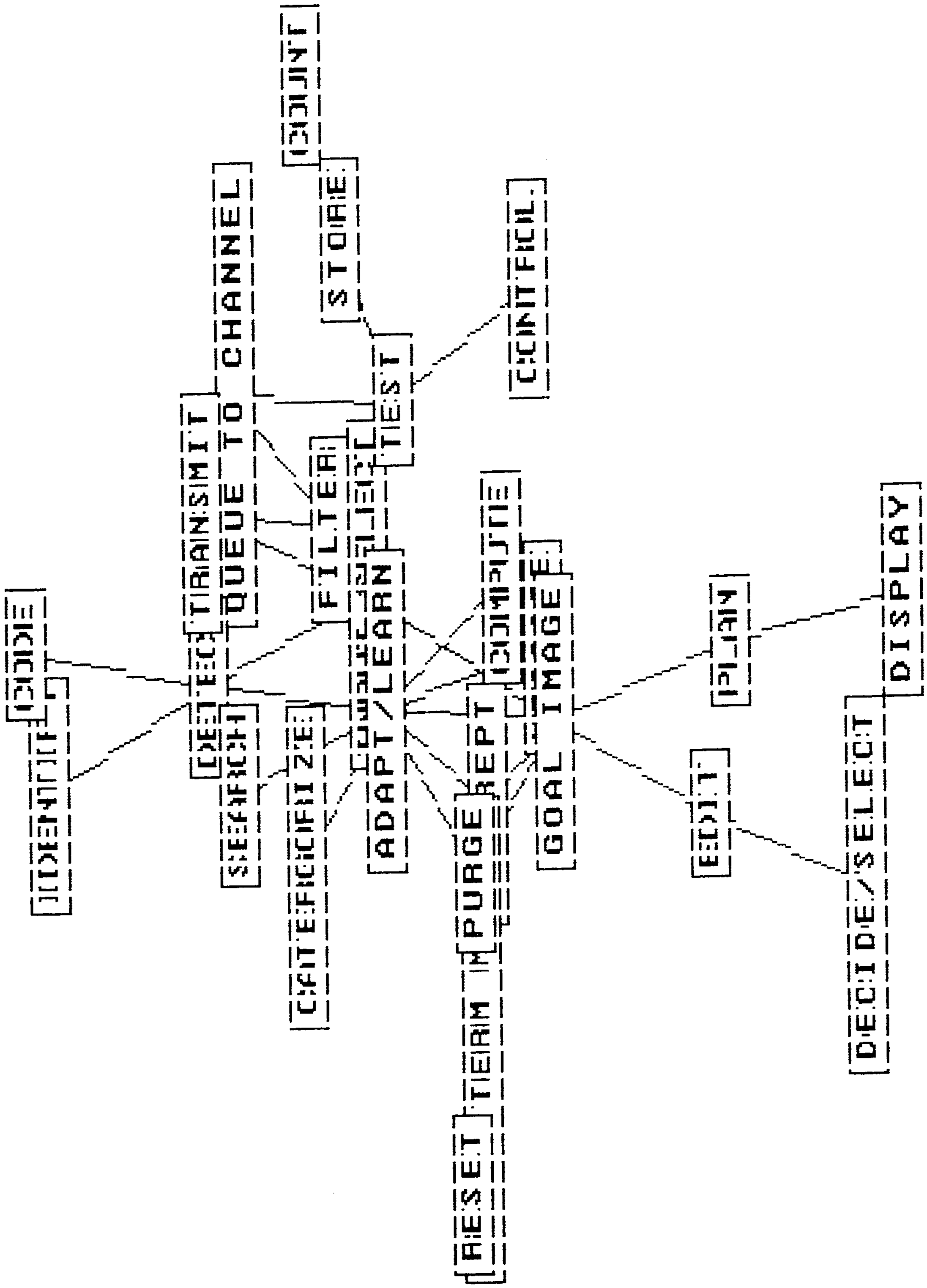


Figure E-49. Subject 20 for Q3.

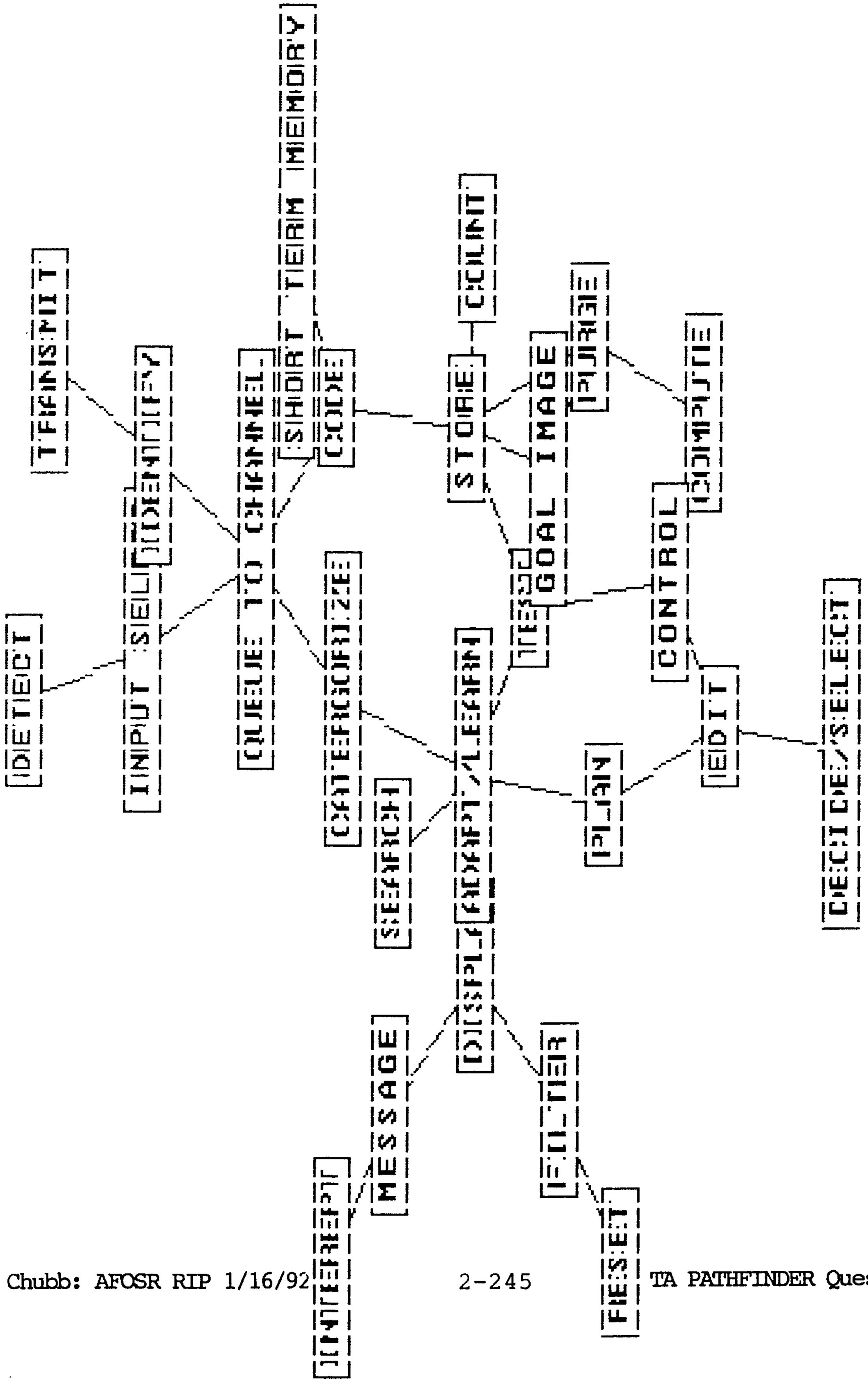


Figure E-50. Subject 23 for Q3.

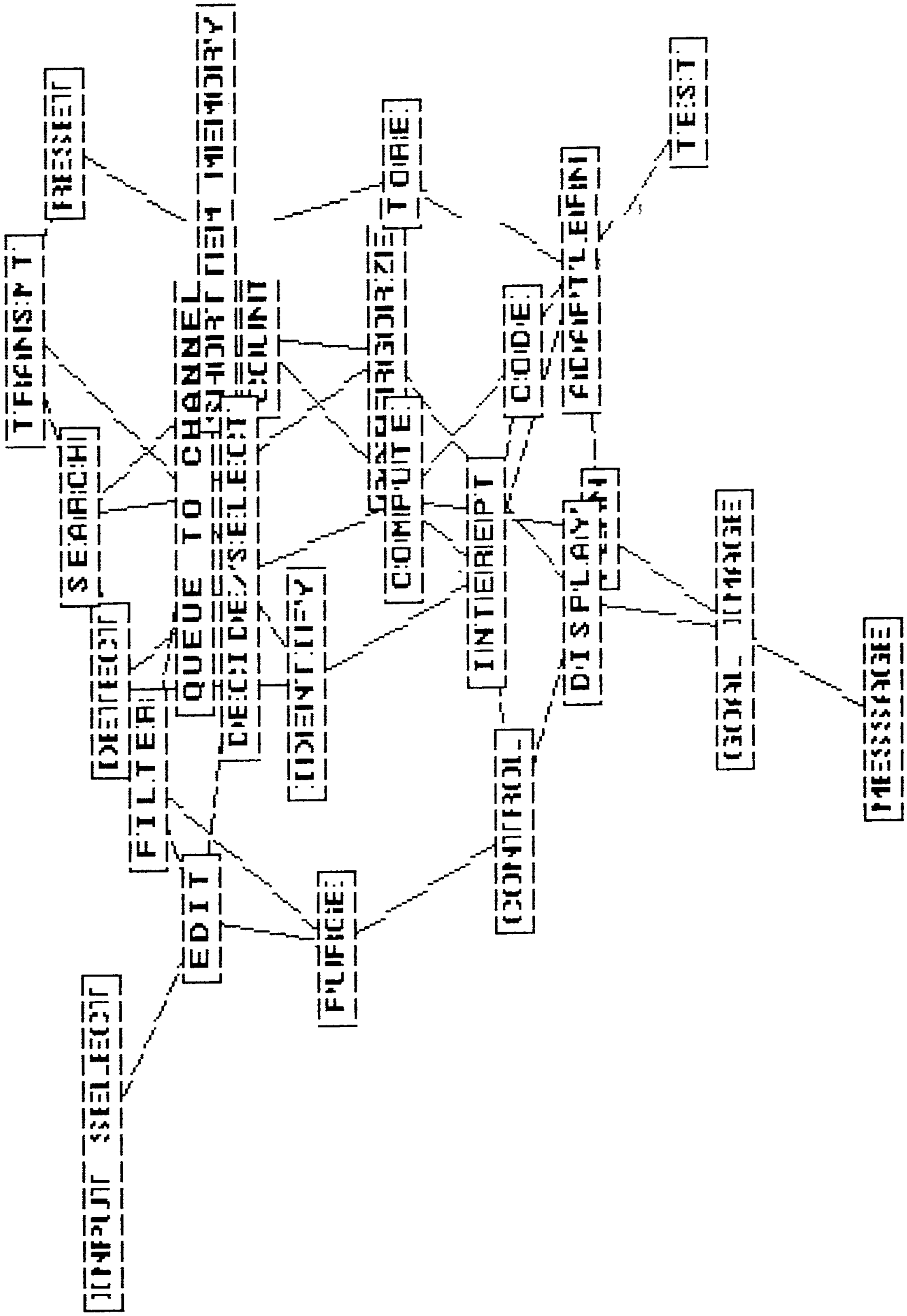
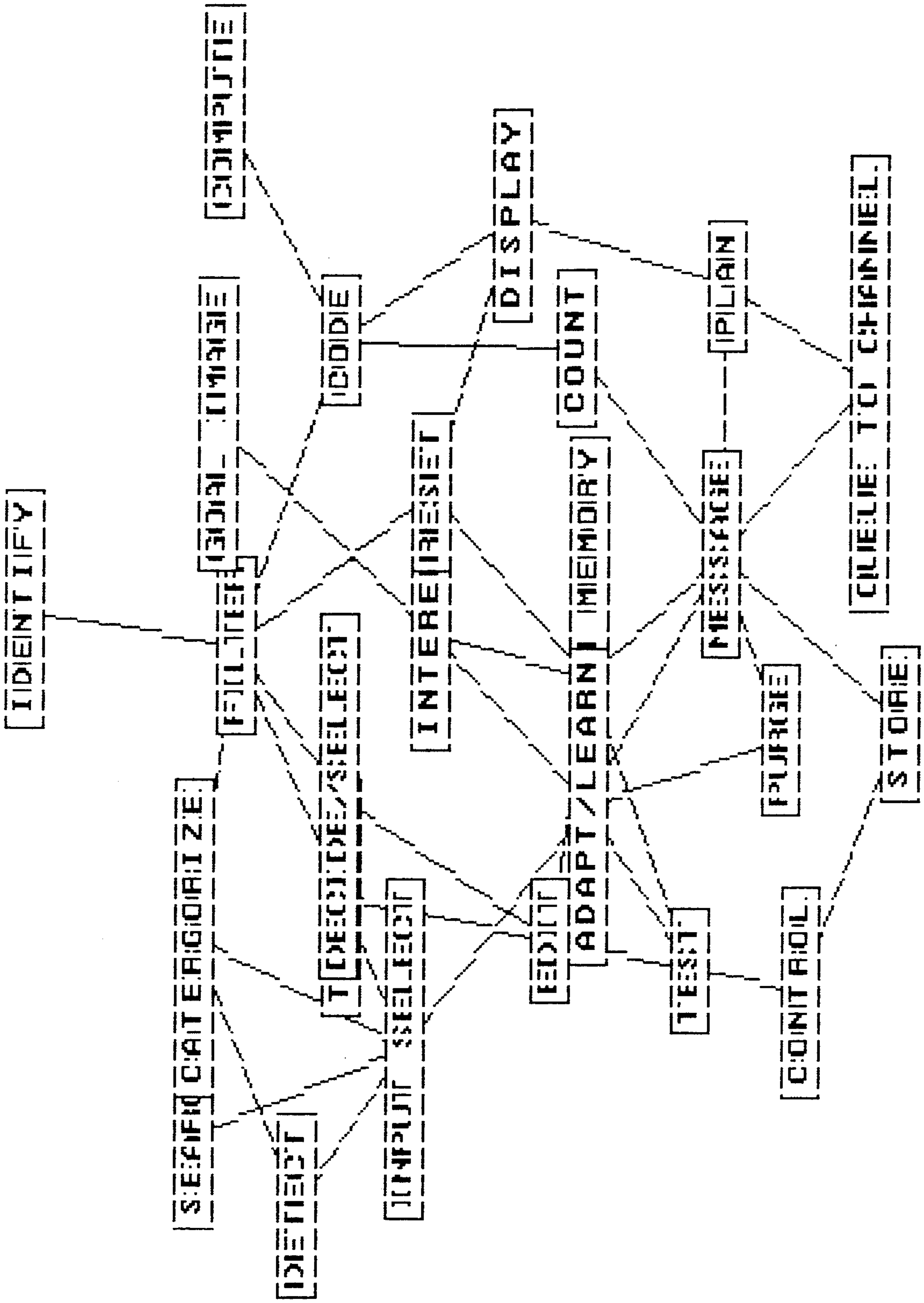


Figure E-51. Subject 24 for Q3.



Appendix F: PATHFINDER Layouts of Task Analysis Similarities: Reliability

The following PATHFINDER diagrams reflect the reliability judgments of eight evaluators. They were asked to compare each of the thirteen task analyses with each other. There were 78 paired comparisons of similarity. These similarity ratings were then input to the PATHFINDER program, which computed the distances between the analyses and plotted the following diagrams.

Figure F-1. Subject 1: Reliability Layout.

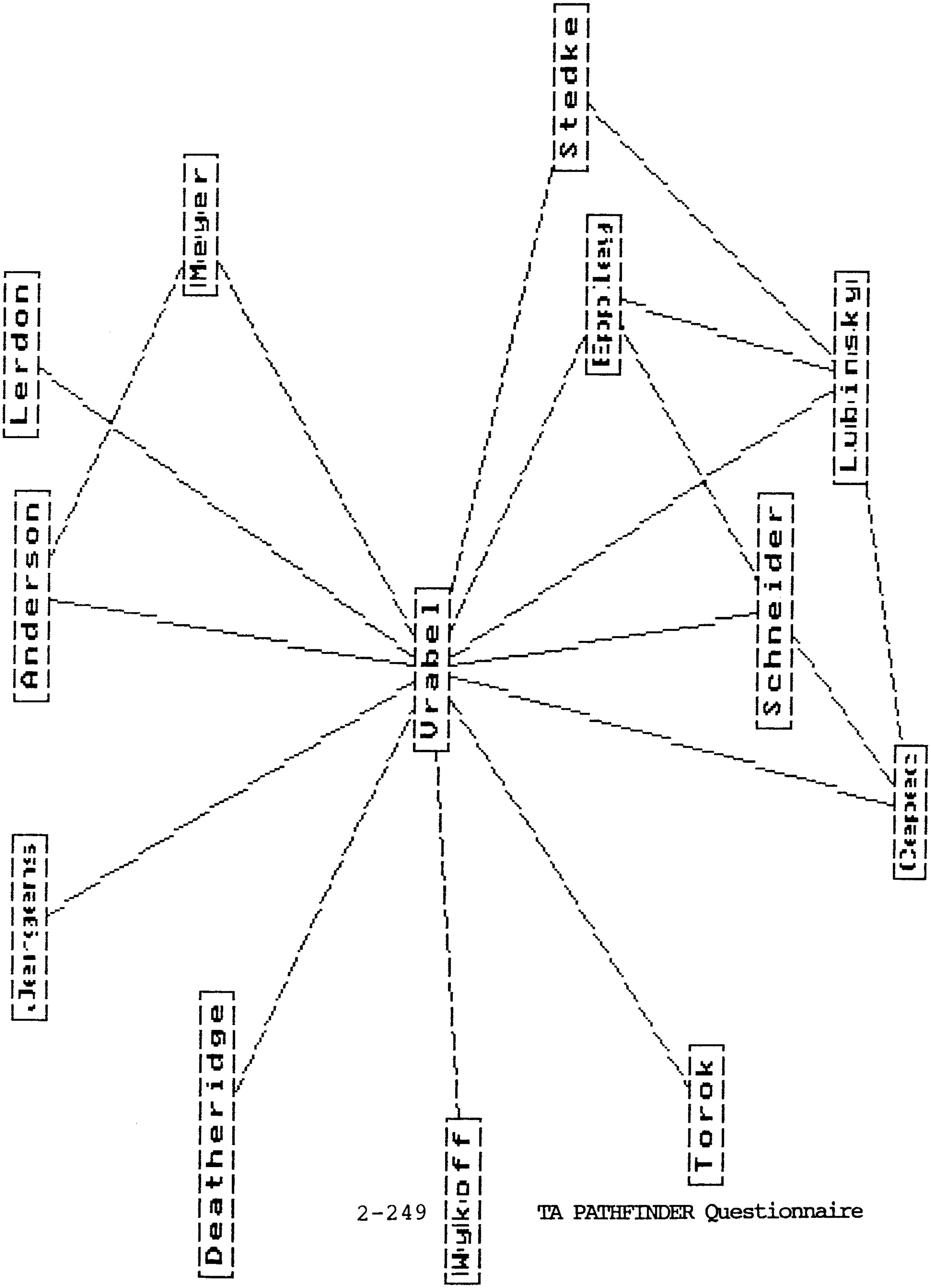


Figure F-2. Subject 2: Reliability Layout.

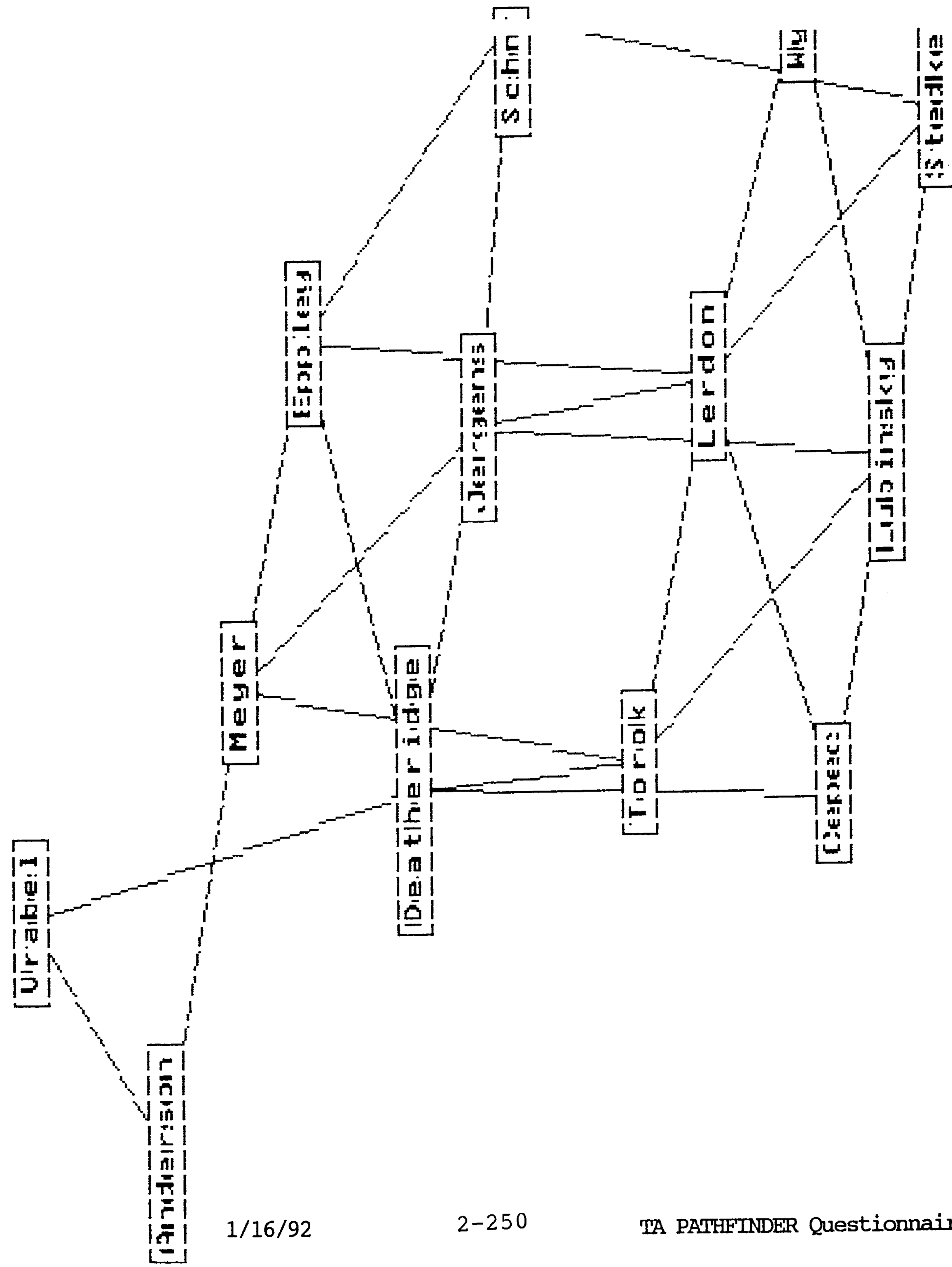


Figure F-3. Subject 3: Reliab Layout.

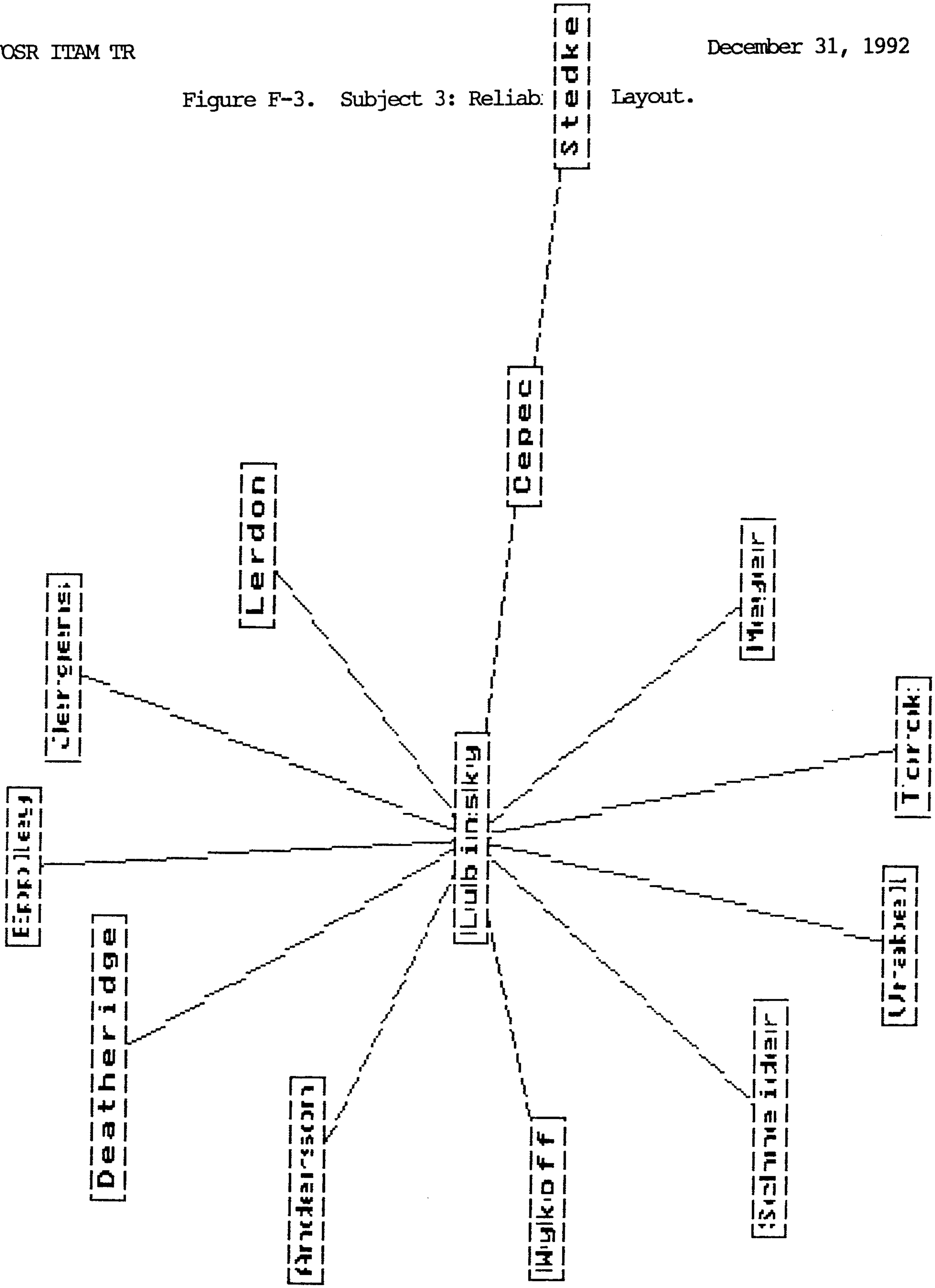


Figure F-4. Subject 4: Reliabi Layout.

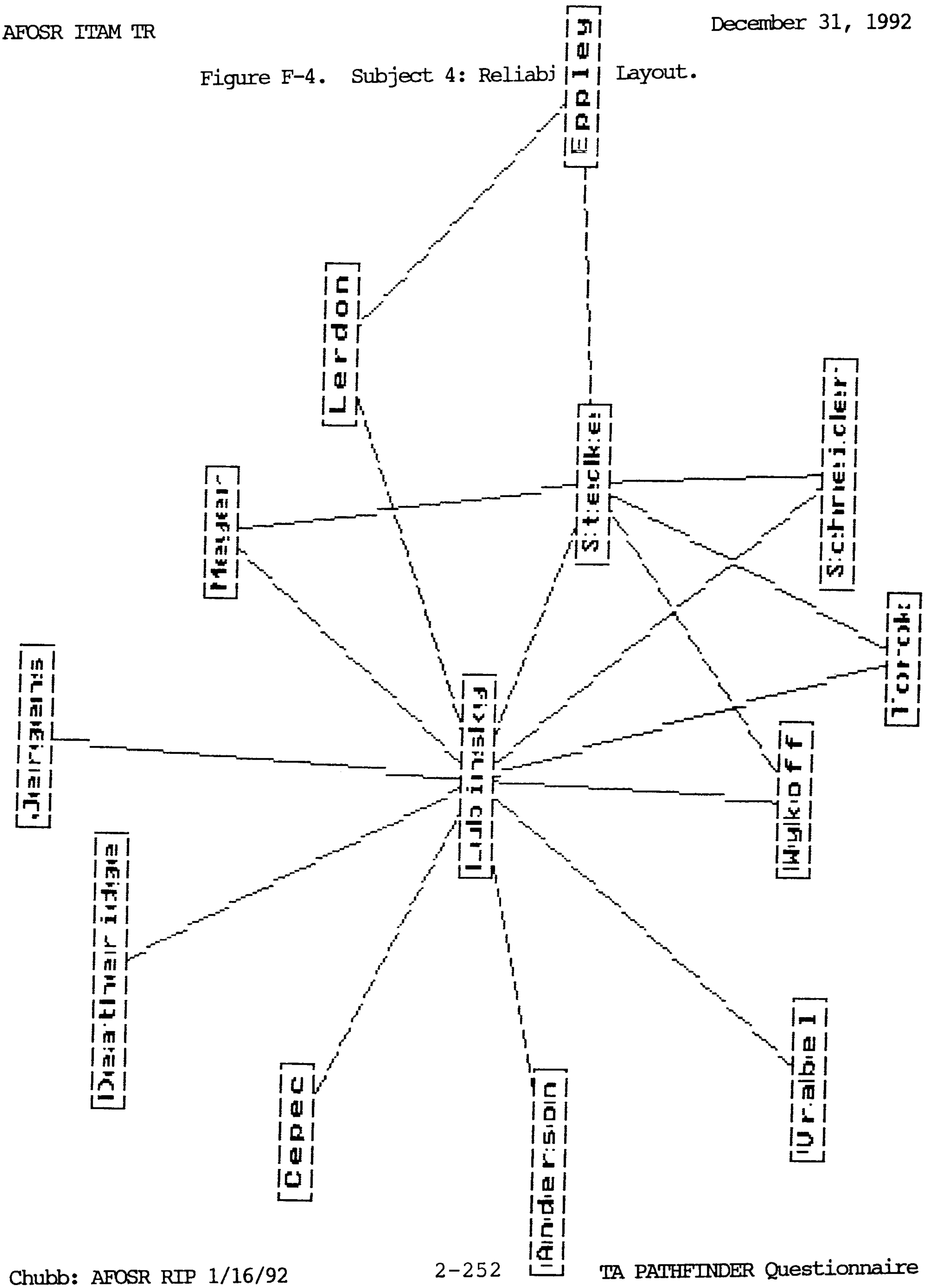


Figure F-5. Subject 5: Reliability Layout.

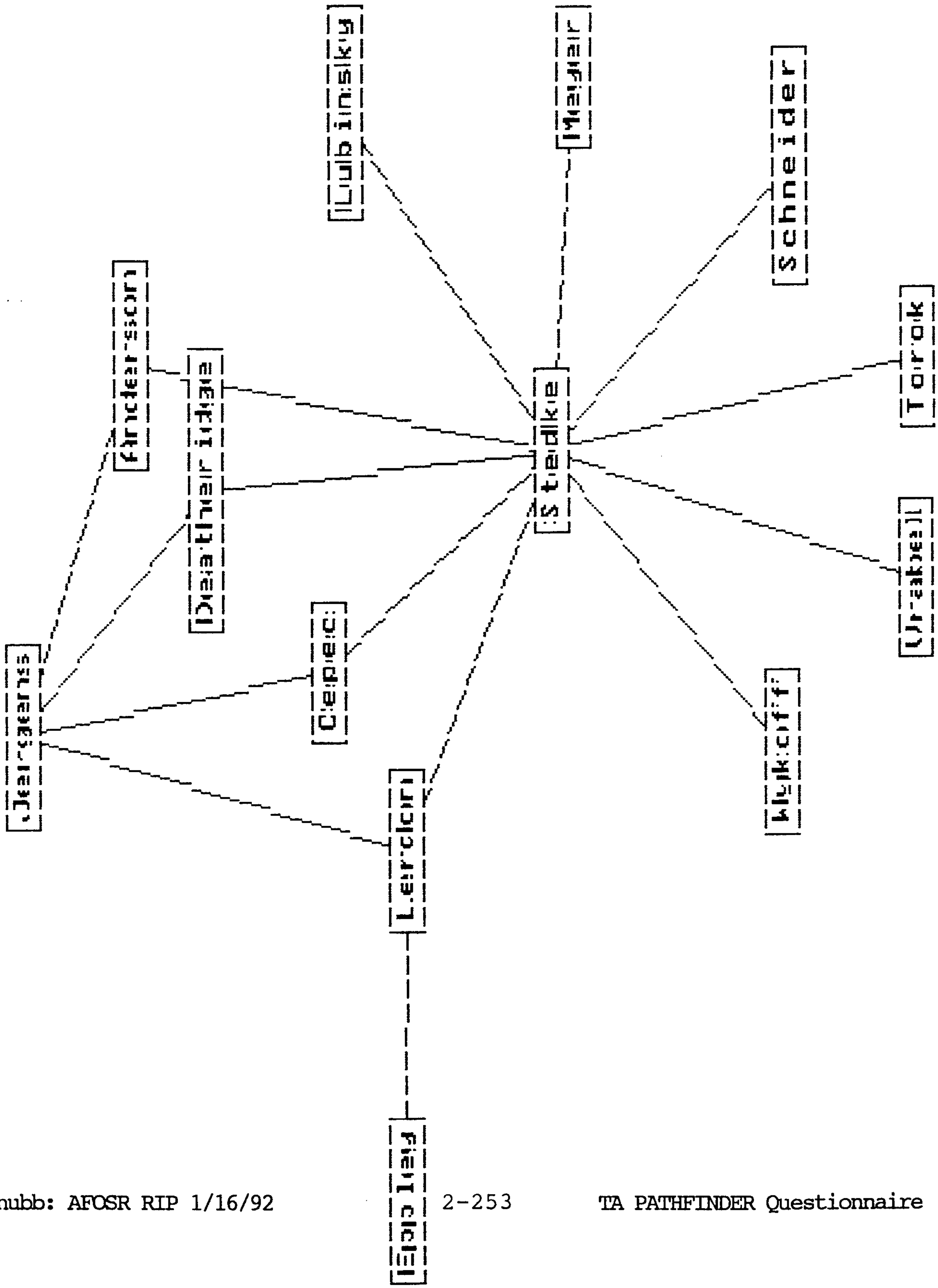


Figure F-6. Subject 6: Reliability Layout.

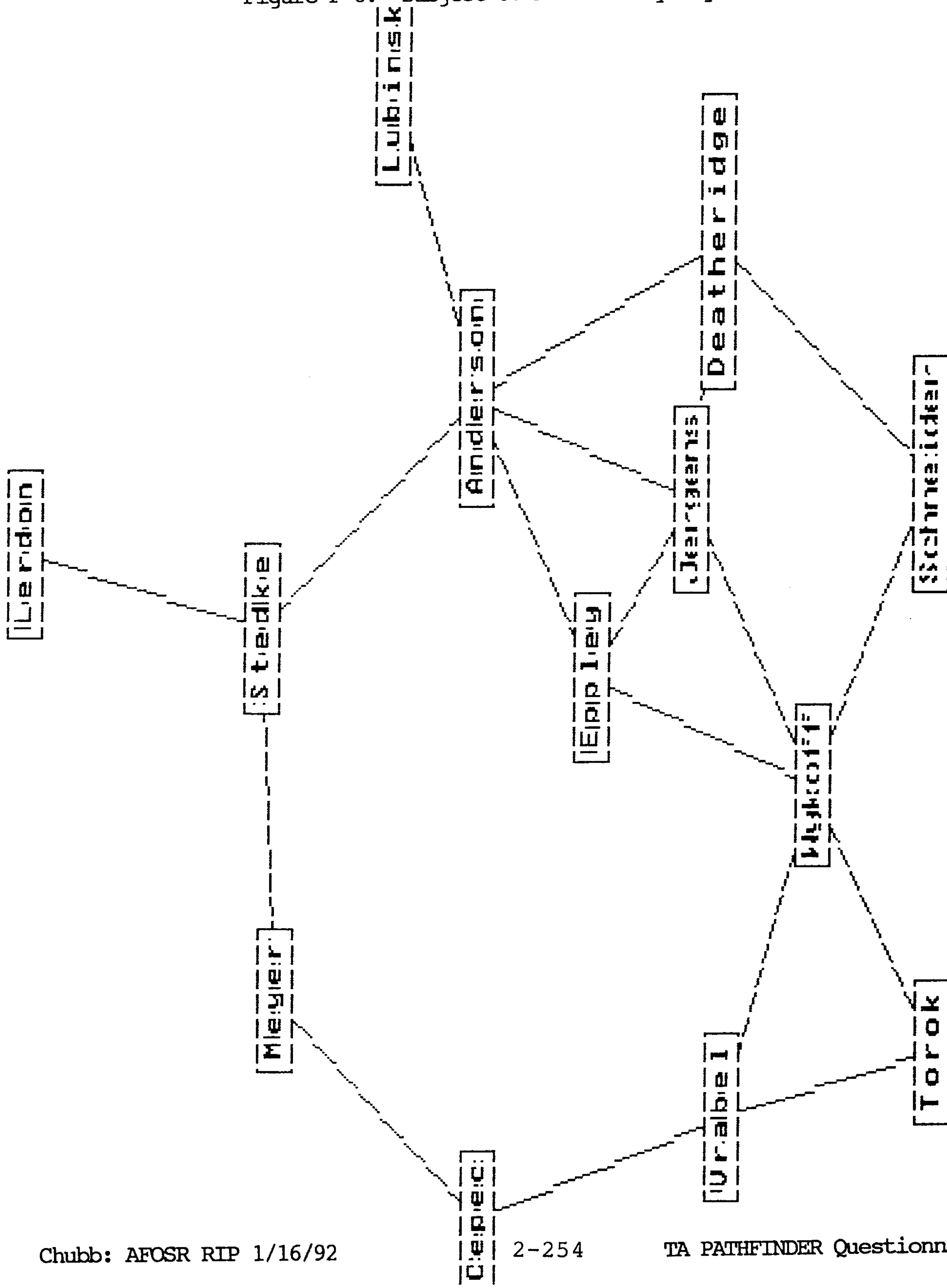


Figure F-7. Subject 7: Reliability La

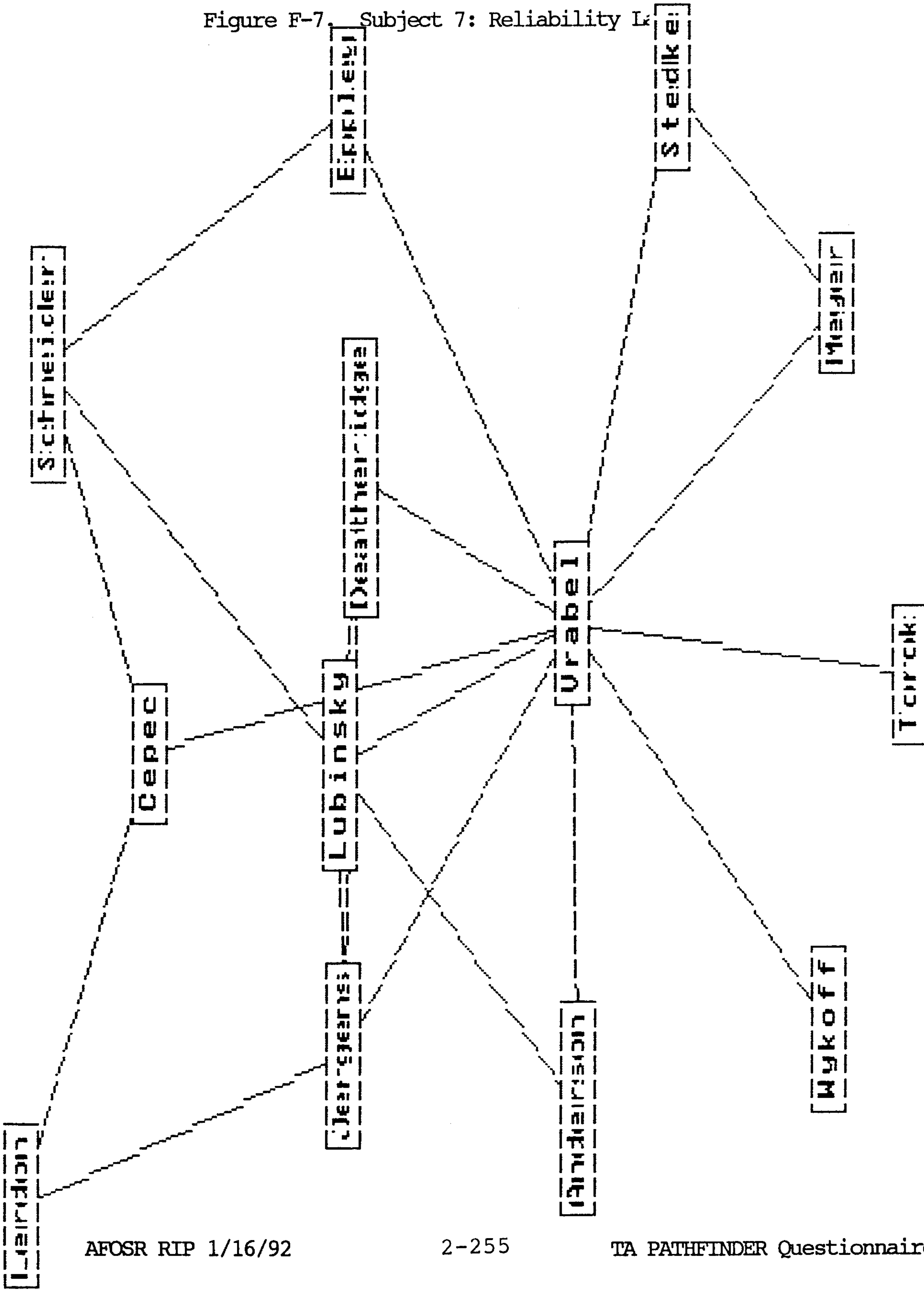
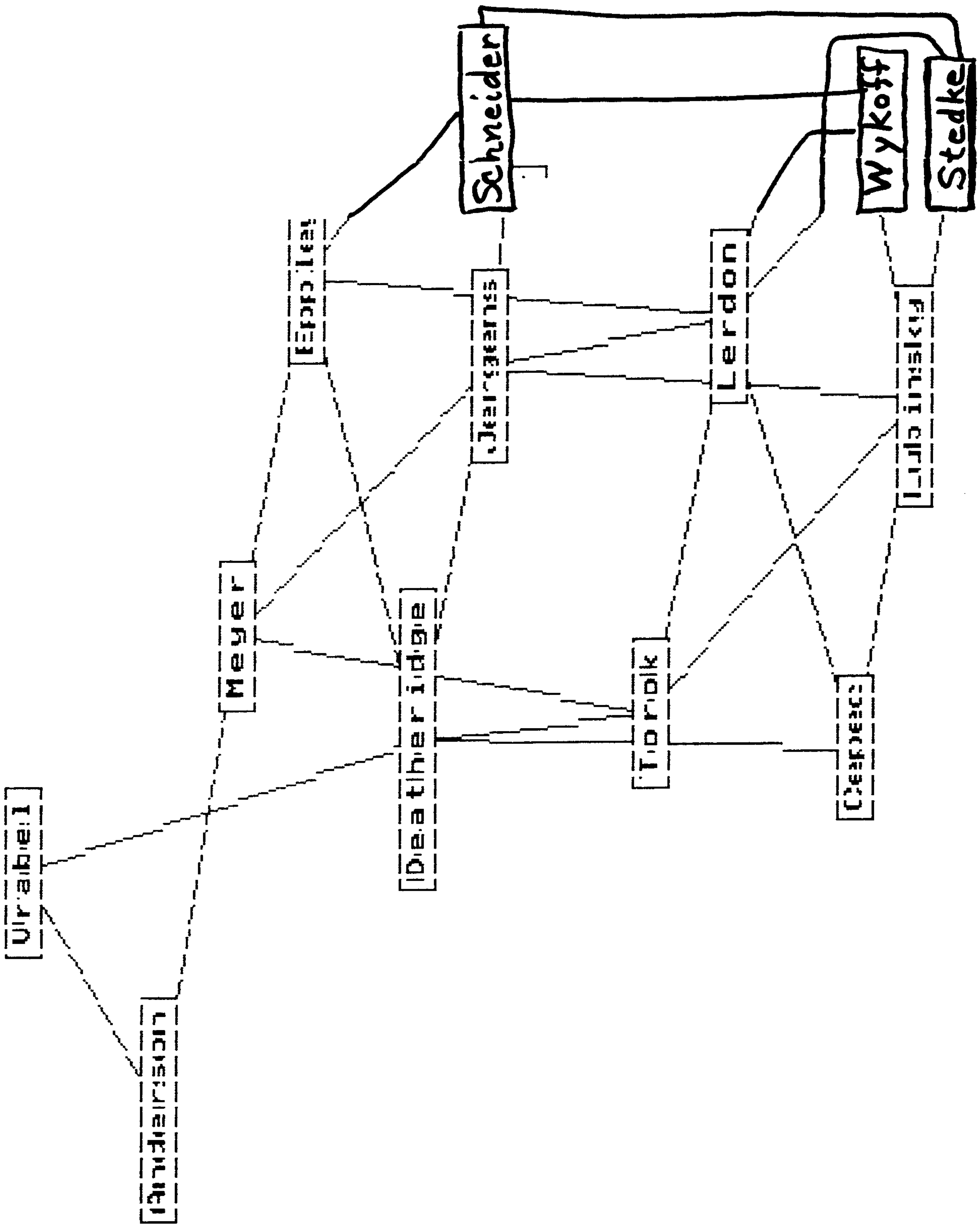


Figure F-8. Subject 8: Reliability Layout.



Appendix G: PATHFINDER Layouts of Analysis Similarities to CTA: Validity

The following eight diagrams are relatively uninteresting, in that the first is representative of the rest. The eight evaluators who compared the task analyses to each other also compared those task analyses with the one the FAA paid CTA to prepare: the criterion TRACON task analysis. The evaluators then compared the similarity of each task analysis with the CTA analysis (thirteen comparisons). These data were input to PATHFINDER, and the following diagrams are the product.

Figure G-1. Subject 1: Validity Layout.

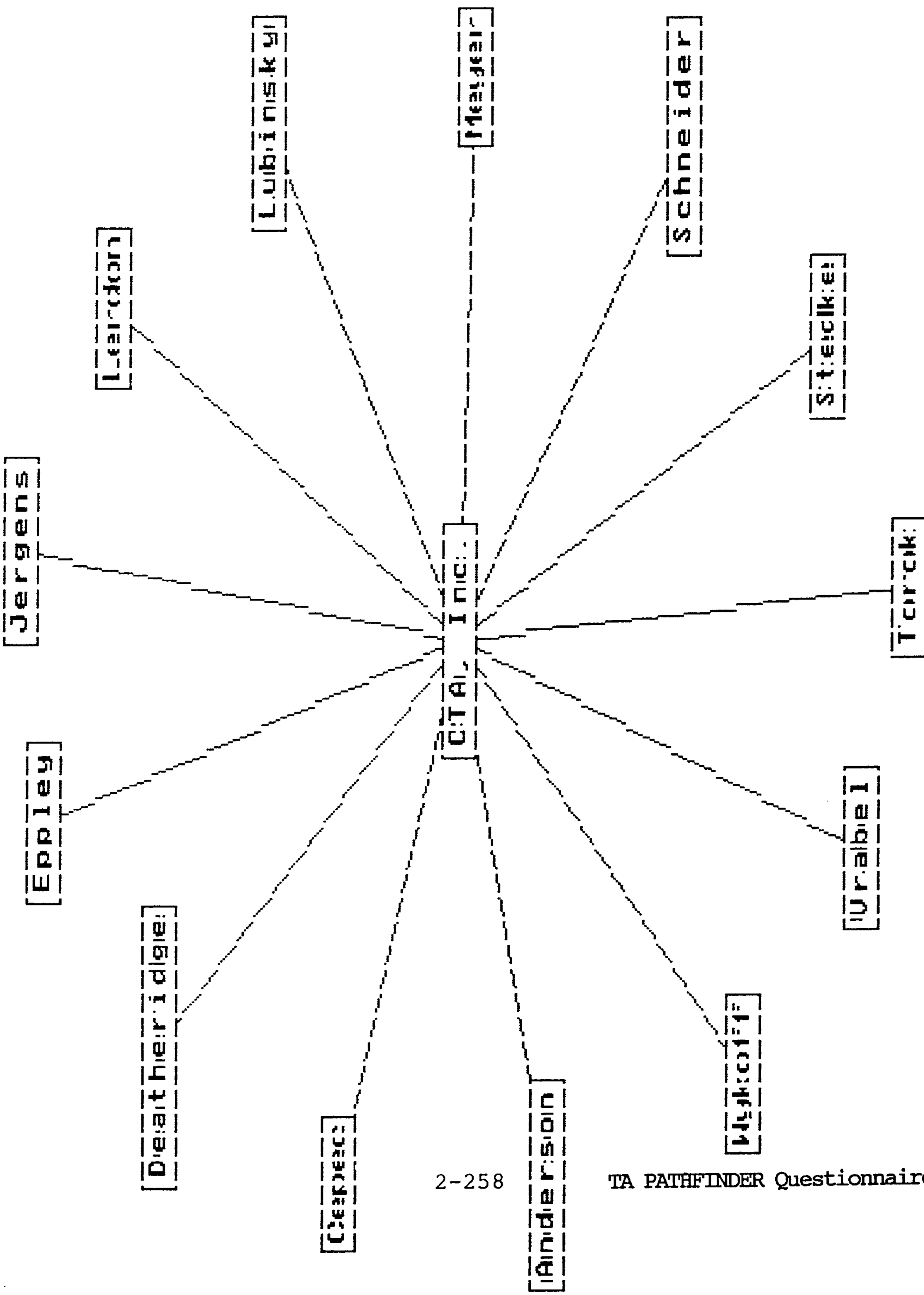


Figure G-2. Subject 2: Validity Layout.

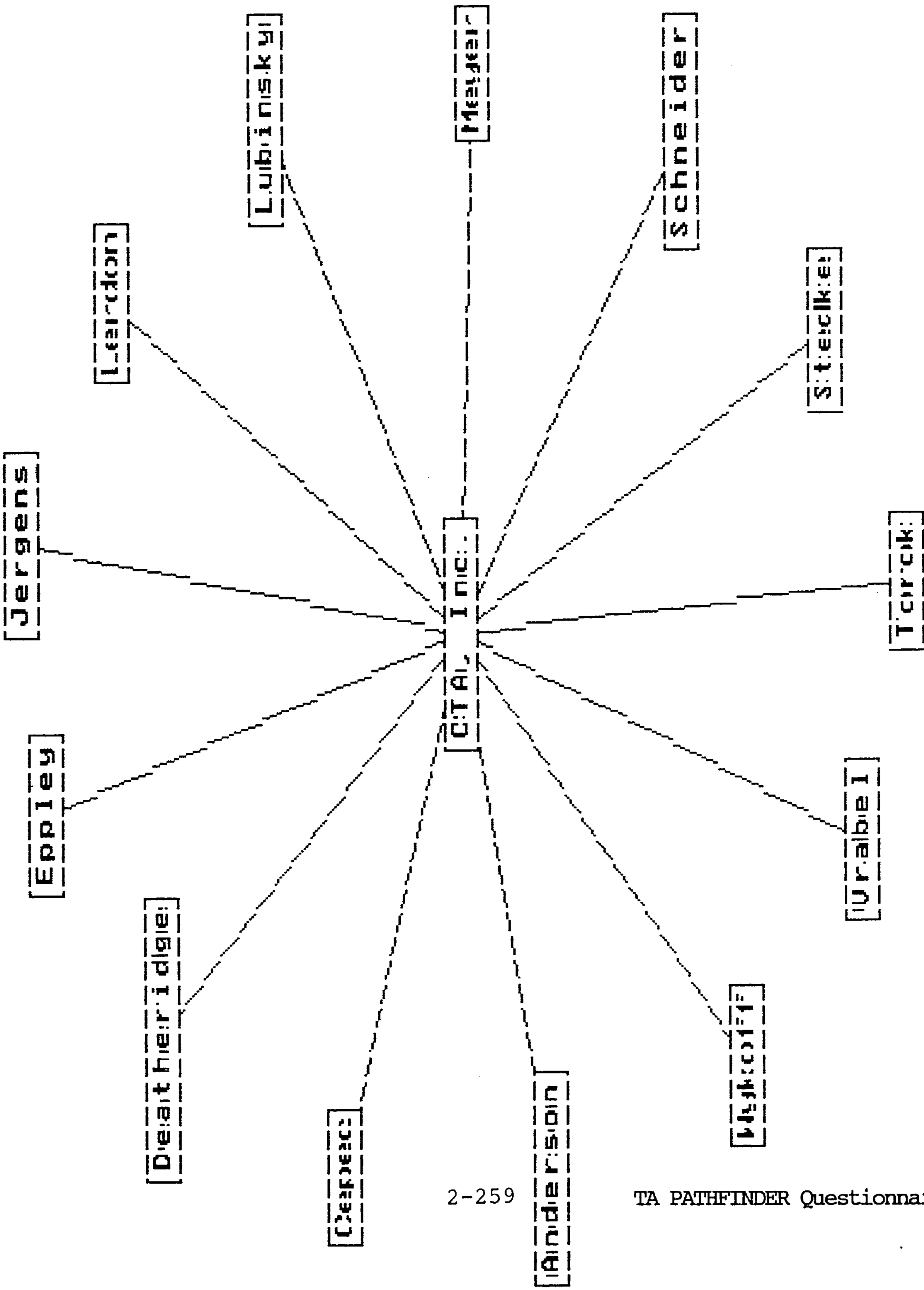


Figure G-3. Subject 3: Validity Layout.

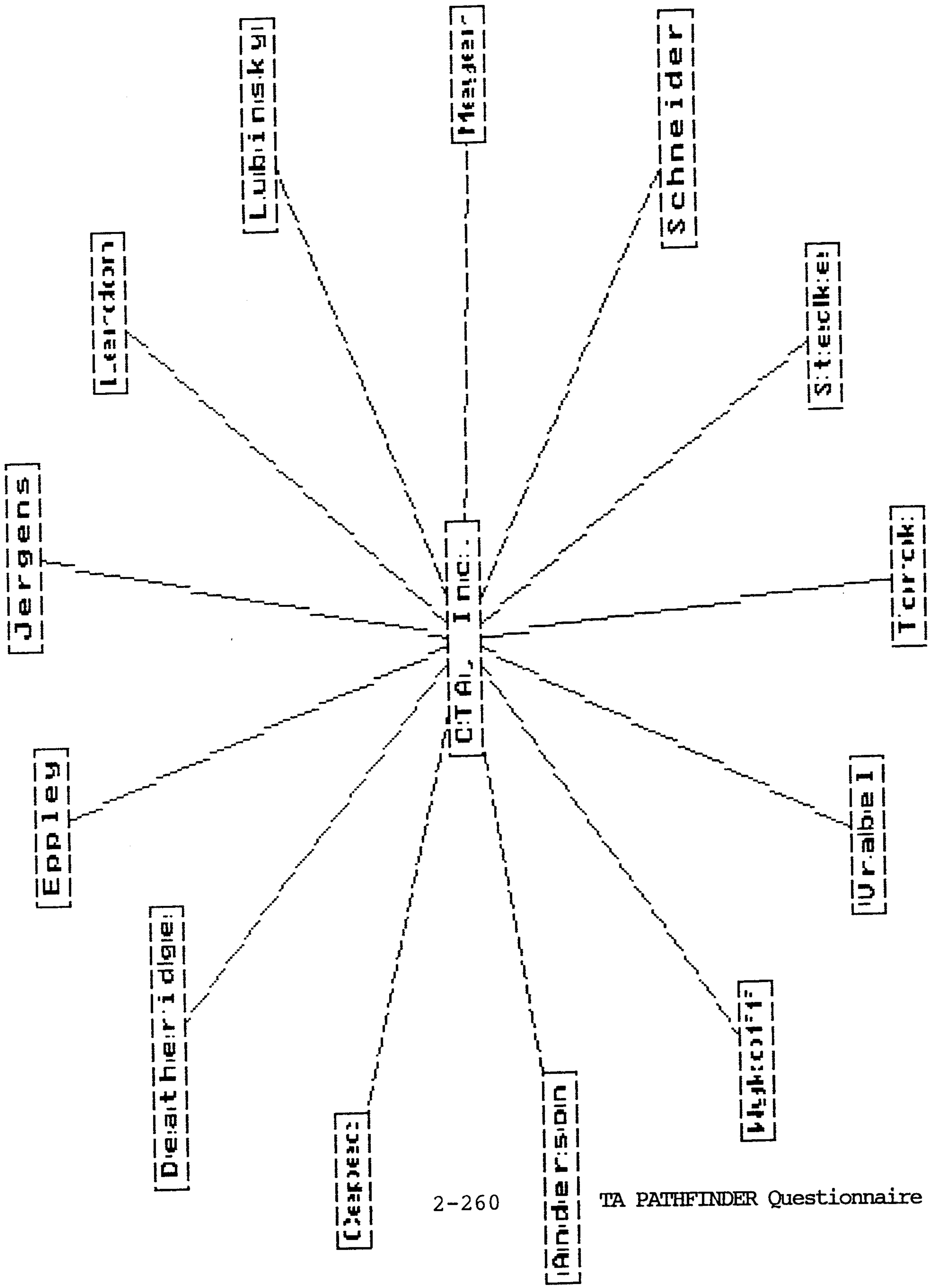


Figure G-4. Subject 4: Validity Layout.

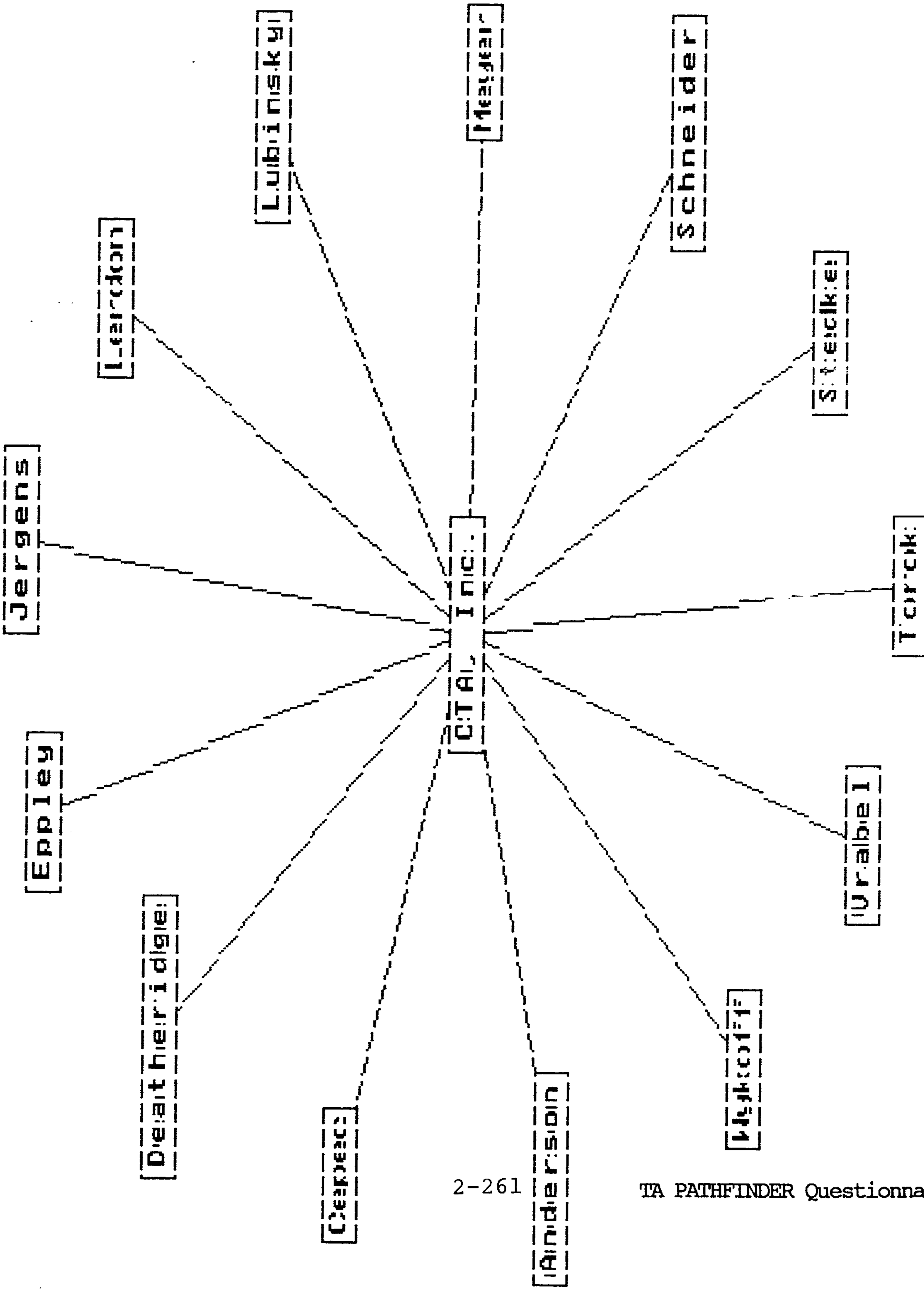


Figure G-5. Subject 5: Validity Layout.

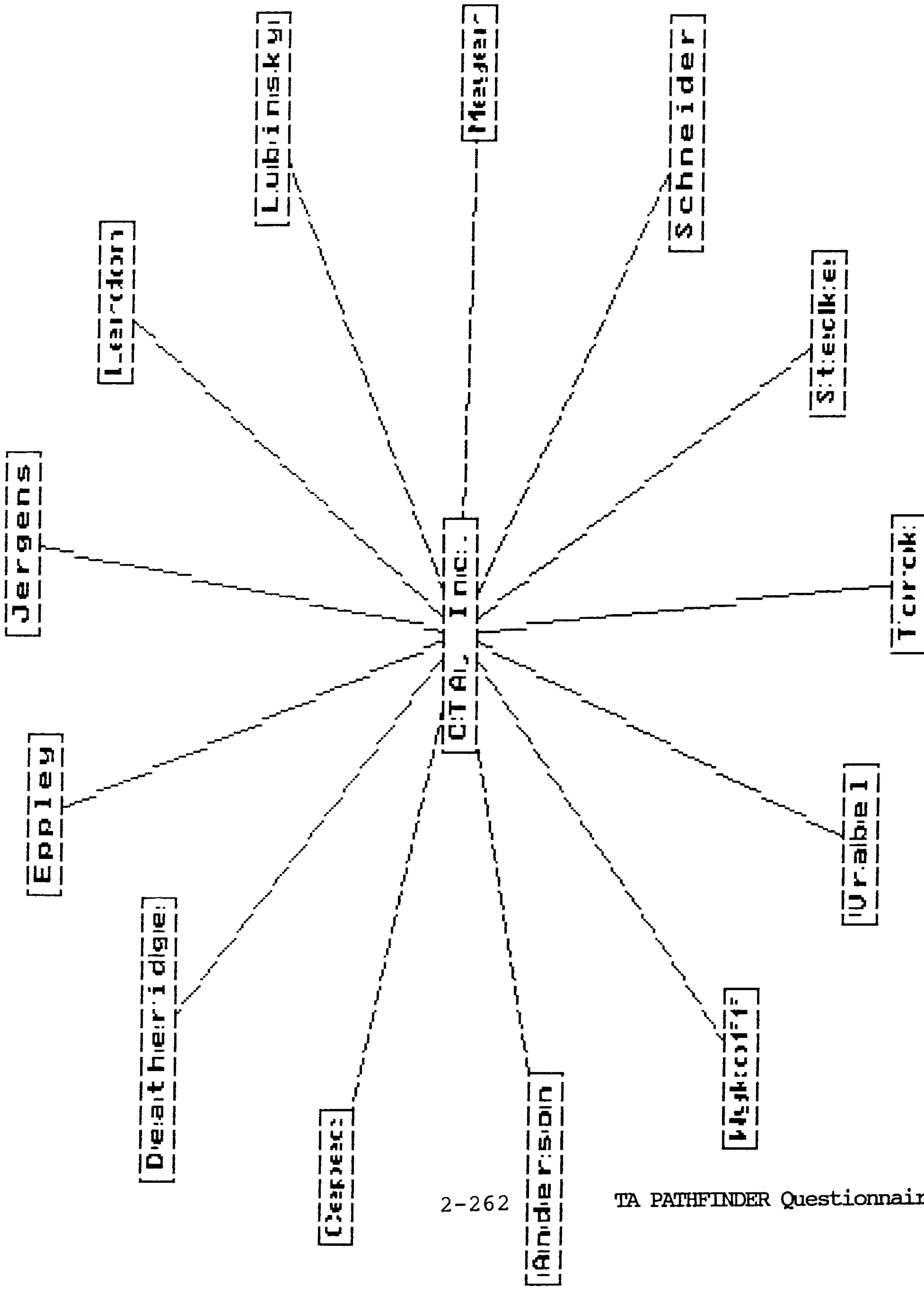


Figure G-6. Subject 6: Validity Layout.

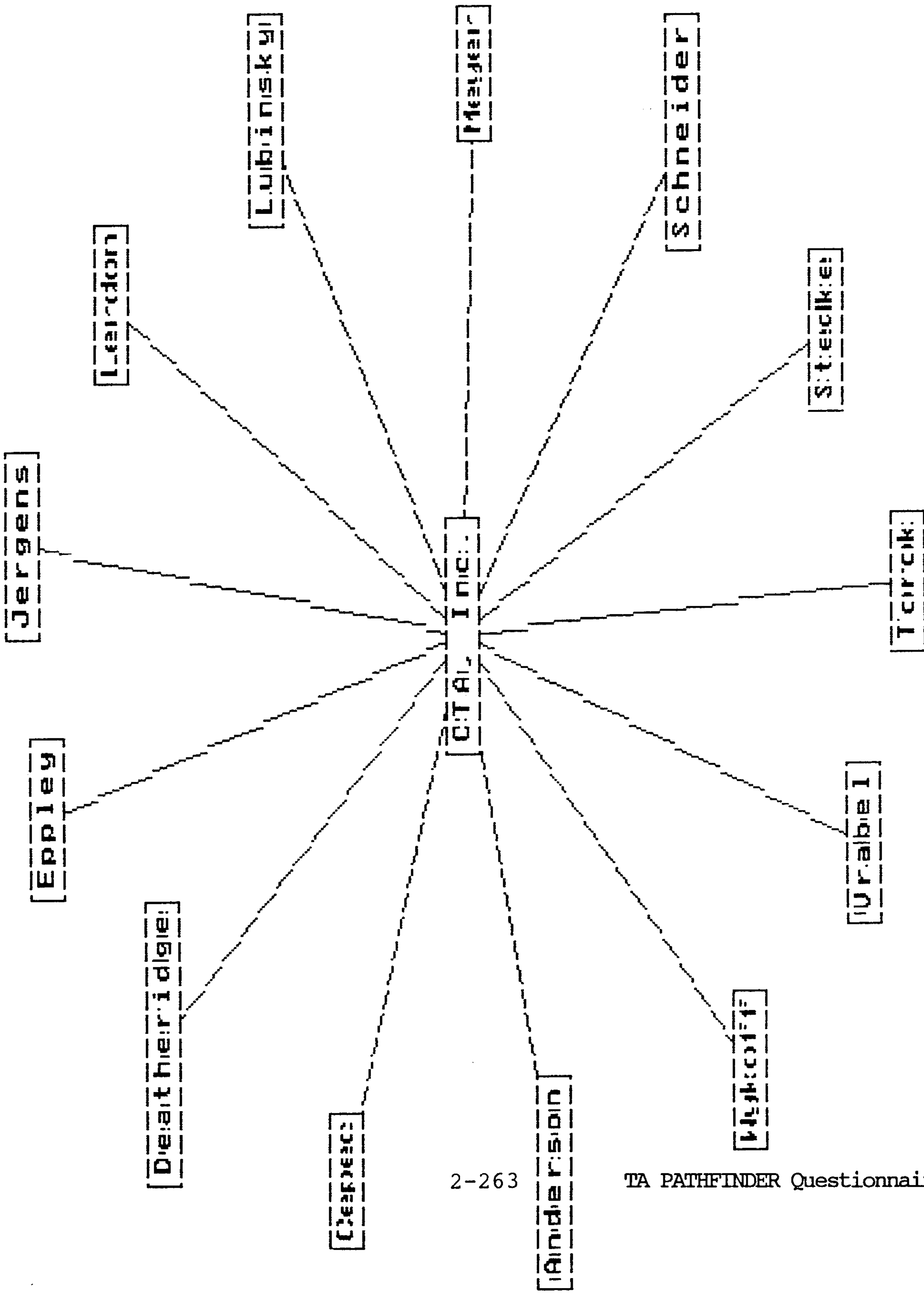


Figure G-7. Subject 7: Validity Layout.

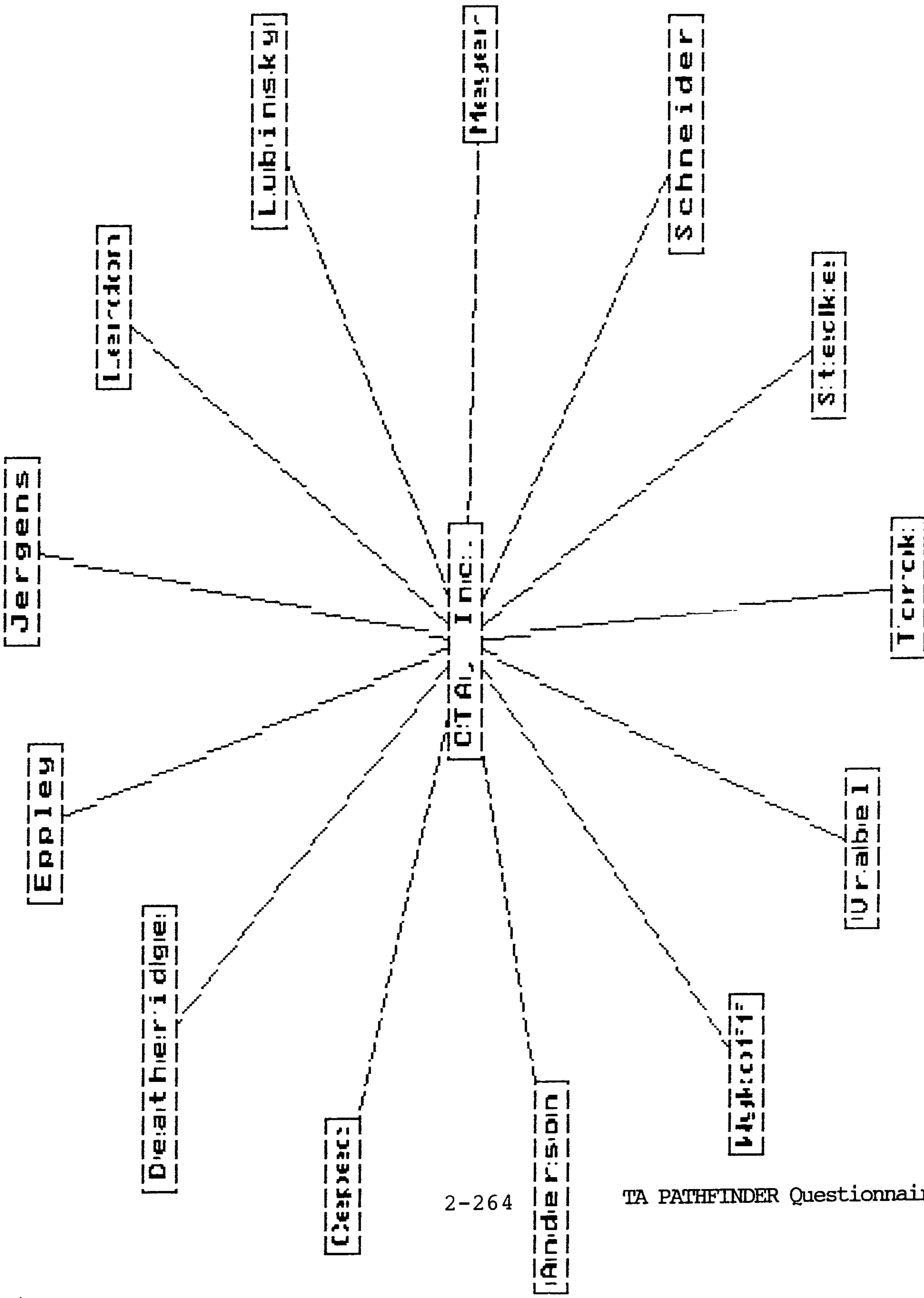
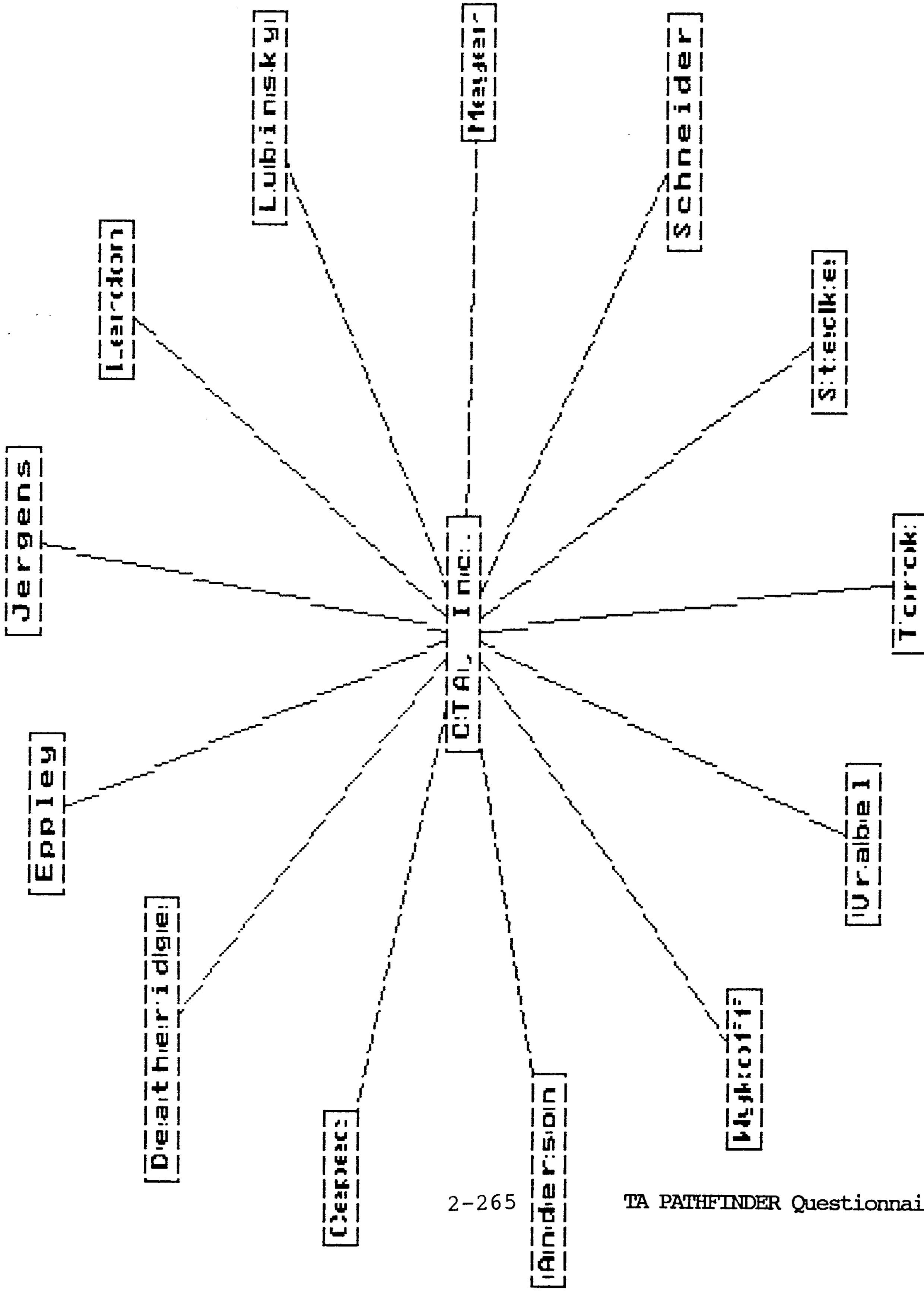


Figure G-8. Subject 8: Validity Layout.



NETWORK INTERFACE UNIT SOFTWARE STANDARDS

Arthur W. Draut
Associate Professor
Computer Science Department

Embry Riddle Aeronautical University
3200 Willow Creek Rd
Prescott, AX 86301

Final Report for:
Research Initiation Program
Armstrong Laboratory, Aircrew Training Division
Williams Air Force Base, Arizona

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, D.C.

and

Embry Riddle Aeronautical University

December 1992

NETWORK INTERFACE UNIT SOFTWARE STANDARDS

Arthur W. Draut
Associate Professor
Computer Science Department

Embry Riddle Aeronautical University

Abstract

Armstrong Laboratory has a network of aircraft simulators and other devices such as a threat generator, a GCI simulation and a controller's station. Each simulator interfaces with an Ethernet bus through a network interface unit. The software in these network interface units has been developed over a period of several years by different contractors. This software is continuing to evolve. The documentation for this software has been neglected. The purpose of this research contract was to develop documentation for these network interface units and a standard of software development and documentation according to DOD-STD-2167A.

NETWORK INTERFACE UNIT SOFTWARE STANDARDS

Arthur W. Draut

INTRODUCTION

It is assumed that the reader has a copy of DOD-STD-2167A and Associated DIDs and a copy of MIL-HDBK-287 readily available. The format of this documentation follows DOD-STD-2167A, tailored in accordance with MIL-HDBK-287. Quoting MIL-HDBK-287, section 4.3.1a, "...Tailoring is intended to eliminate unnecessary and duplicative requirements. For standards, a modified version of the requirement may be included in the SOW. For DIDs, requirements may be deleted or partially deleted, but not modified."

The documentation presented here is for part of the software in the simulator network at the Armstrong Laboratory, Williams AFB, AZ. The simulators involved are F-16 and F-15 simulators. The nodes on the network are aircraft simulators, a threat generator, a GCI site, and an controller's station. This software was not developed under 2167A standards. It is hoped that this research will provide a starting point for transforming the network software documentation to the format required by 2167A.

Acronyms and Abbreviations

CET	Combat Engagement Trainer (currently an F-16 simulator)
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSU	Computer Software Unit
DARPA	Defense Advanced Research Projects Agency
DID	Data Item Description
GCI	Ground Controlled Intercept
MULTIRAD	Multiship Research and Development
NIU	Network Interface Unit
PDU	Protocol Data Unit
RVA	Remote Vehicle Approximation
SIMNET	Simulator Network
SOW	Statement of Work

DISCUSSION OF THE PROBLEM

SIMNET was a network of ground vehicle simulators developed with and for DARPA by Bolt, Beranek, and Newman Systems and Technologies (BBN), Cambridge, MA. The first use of SIMNET was for a network of Army tank simulators. The Air Force wanted to adapt SIMNET to aircraft simulators and BBN was given a contract to apply SIMNET to a network of F-16 and F-15 simulators in the Armstrong Laboratory, Williams AFB, AZ. The author was first introduced to this network in the summer of 1991 under an AFOSR summer faculty program. At that time, BBN had one software engineer on site and he was not one of the developers of the modified SIMNET software. However he was a valuable source of information while he was there. In the fall of 1991, Loral, Inc., Akron, OH, replaced BBN as the contractor for this software.

There was little documentation for the software. The author began to create a design by studying the source code. When BBN left the project, they published a design specification, dated May 1991. This document was incomplete and in the author's opinion, it was poor. However, it did provide some useful information.

The development of the modified SIMNET software at Armstrong laboratory was not conducted in accordance with DOD-STD-2167A. The author was unable to find a Statement of Work, a System Specification, a Software Design, a Software Development Plan, and other documents required by 2167A. One can argue that this is a research project with continuously changing configurations and requirements and, since production software is not the goal of the project, 2167A does not apply. However, the author argues that even though the network configuration does change, the NIUs and the network protocols do not change. Therefore the network software should be treated as production software. This software will eventually be used at different sites by different groups and therefore should be developed according to government standards.

By using reverse engineering, a design was created that matched the software as it existed. A suggested design standard was also created. These were written in accordance with 2167A. The author strongly suggests that this software be placed under 2167A procedures and standards. Of course this would be done at some expense to the government. The current contractor would have to hire additional personnel to accomplish this. However, this would be a small cost compared to the costs that will result if this software continues to grow in a non standard fashion.

RESULTS

In this small project there was not time to recreate the entire software development process according to 2167A. Only a few sub-paragraphs are considered here. Although the software is written and running, it is feasible for the current contractor to place the software under 2167A standards.

2167A paragraph 5.1.2.2. To satisfy this, Appendix B contains a shell of a System Requirements Design, DID, DI-CMAN-80008A, SYSTEM SPECIFICATION. This is presented as a starting point for transforming the network software to 2167A standards.

2167A paragraph 5.3.2.1. and 5.4.2.1. These refer to preliminary design and detailed design. To satisfy this, Appendix A contains the DID, DI-MCCR-80012A, SOFTWARE DESIGN DOCUMENT. This was created so as to have a 2167A design document. It is for the software as it existed in December, 1991. The majority of the author's effort was spent in creating this document. It is suggested that the current contractor maintain it. Appendix C contains a suggested new version of this DID for the NIU software.

2167A paragraph 5.3.2.2. and 5.4.2.2. These refer to interfaces. Appendix D contains a shell of the DID, DI-MCCR-80027A, Interface Design Document, with references to in interface control document prepared by General Electric Government Services.

CONCLUSIONS

Industry and government have been slow to transition to 2167A. It is human nature to resist change. Software Engineering is a new discipline compare to disciplines such as Aeronautical Engineering or Electrical Engineering which have proven development procedures. There are many, well known examples of software delays, cost overruns and complete failures. It is urgent that software development standards be used. The author believes that 2167A is a reasonable and useful standard, and is anxious to see it implemented.

The development of the network software at Armstrong Laboratories provides a classic example of the problems that arise when software is developed by several contractors without a common standard. The author sites this project in his software engineering classes. It is not too late to standardize this project. Simulator networks will be extremely valuable in the future, saving the government millions of dollars in training pilots for combat.

As the author studied the network software, he became concerned that the goals of a nation-wide network of aircraft simulators might be unrealistic. At his suggestion, Armstrong Laboratory purchased network simulation software and he learned to use it, but did not have time to build a useful model. A proposal to build models of various network configurations is forthcoming.

ACKNOWLEDGEMENTS

The author attended a three day seminar on 2167A presented by David Maibor Associates in 1990. The seminar and travel expenses were paid by Embry Riddle Aeronautical University. The seminar was excellent and very useful. Attending this seminar provided the motivation for attempting this research. The author would like to thank Embry Riddle for sending him.

REFERENCES

DOD-STD-2167A, 29 February, 1988

MIL-HDBK-287, 11 August 1989

Report No. 7102, The SIMNET Network and Protocols, BBN Systems and Technologies Corporation, 31 July 1989

NIU Detailed Design Specification, BBN Systems and Technologies Corporation, 2 May 1991

Interface Control Document for the Combat Engagement Trainer/Network Interface, General Electric Government Services, 15 January 1991,

APPENDIX A
DI-MCCR-80012A

SOFTWARE DESIGN DOCUMENT

FOR THE

CET NIU CSCI

OF

SIMNET SYSTEM
Armstrong Laboratory
Williams AFB, AZ

CONTRACT NO. unknown

CDRL SEQUENCE NO. unknown

Prepared for:

Armstrong Laboratory

Prepared by:

Arthur W. Draut
Computer Science Department
Embry Riddle Aeronautical University

Table of Contents

Section 1. Scope

1.1 Identification

CSCI number : 1

CSCI title : CET NIU

System : SIMNET, Armstrong Laboratory, Williams AFB, AZ

1.2 System Overview

SIMNET (Simulator Network) is a protocol for a network of interconnected aircraft simulators. The physical layer of the network is an Ethernet. The nodes on the network are aircraft simulators, threat generators, and an operator station. Each node has an NIU (Network Interface Unit) between the device (simulator, threat generator, etc.) and the Ethernet. The purpose is to allow multi-ship training. The pilot in one simulator is presented with images of the simulated aircraft being flown by pilots in other simulators. There can also be computer generated threats displayed to the simulator pilot.

One of the aircraft simulators is a low fidelity F-16 cockpit, the CET (Combat Evaluation Trainer). The CSCI (Computer Software Configuration Item) described here is the software that resides in the CET NIU.

1.3 Document Overview

This document provides a description of the software in the CET NIU. It contains both preliminary design and detailed design information. It is the primary source for coders, testers, and modifiers of the CET NIU software.

This document is written in accordance with DOD-STD-2167A, Miliary Standard Defense System Software Development. More specifically, it is written in accordance with DID No. DI-MCCR-80012A, Software Design Document.

Section 2. Referenced Documents

BBN's Report No. 7102 The SIMNET Network and Protocols, July 1989
BBN's NIU Detailed Design Specification, May 1991

Section 3. Preliminary Design

3.1 CSCI Overview

This CSCI is the software that resides in the NIU for the CET. It is written in the C language. The following description of NIU software in general is extracted from the BBN NIU Detailed Design Specification, Section 2.1 NIU Description, page 6.

> The NIU provides the capability to connect dissimilar simulators to an existing SIMNET network. The NIU transforms the existing simulator data into SIMNET PDUs and provides additional overhead required to support the SIMNET protocol. The NIU allows simulators with non-homogeneous frame times to be interoperable on the same network.

This NIU is the interface between the CET simulator (host) and the SIMNET. Two data structures are passed between the NIU and the host: The Host-to-NIU buffer and the NIU-to-host buffer. There are several PDUs (Protocol Data Units) passed between the SIMNET and the NIU. These PDUs carry information from one simulator to another for the purpose of generating images. These are described in detail in the BBN Report 7102.

3.1.1 CSCI Architecture

This CSCI is subdivided into several CSCs. At the top level are two CSCs: MAIN and SIMULATION STATE MACHINE. At the second level are 4 CSCs: TIMERS, VEHICLE SIMULATION, NETWORK SIMULATION, and I/O SIMULATION. For identification, these CSCs are numbered as follows:

	CSC	Directory Path and File Name
1.1	MAIN	/u2/simnet/niu/src/newcet/niu_main.c
1.2	SIMULATION STATE MACHINE	/u2/simnet/niu/libcet/libstate/stt_machine.c
1.2.1	TIMERS	/u2/simnet/libsrc/libtimers/t_simul.c
1.2.2	VEHICLE SIMULATION	/u2/simnet/niu/src/newcet/niu_main.c
1.2.3	NETWORK SIMULATION	/u2/simnet/niu/src/newcet/n_net_simul.c
1.2.4	I/O SIMULATION	/u2/simnet/niu/src/newcet/nio_io_simul.c

These six CSCs are fairly clean modules. They pass no parameters between themselves. The relationships between them are simple. However, they are not cleanly separated among the various files. The directory organization and the naming of the files is confusing. It needs to be changed to correlate to the design of the source code. For example, note that the VEHICLE SIMULATION CSC is in the same file as the MAIN CSC. The directories for the source code are included in the appendix. The hierarchy of these first and second level CSCs is shown in figure 3.1.

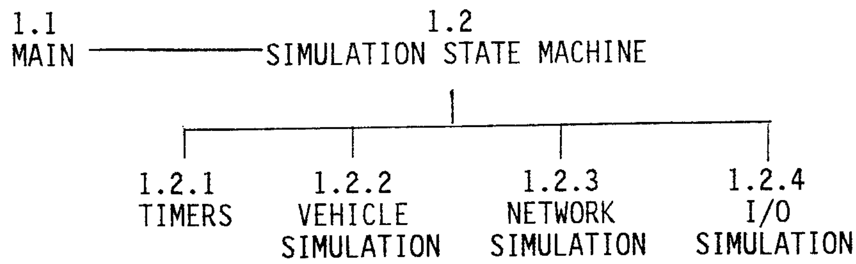


Figure 3.1

MAIN calls several initialization routines, then to determine the mode of operation it parses the command line entered by the operator who has started the NIU software, and then enters an infinite loop in which it repeatedly calls SIMULATION STATE MACHINE.

SIMULATION STATE MACHINE calls the four CSCs at the second level. It calls TIMERS to increment and test various timers. Next it calls VEHICLE SIMULATION which resets host buffer pointers, then receives and processes the host-to-NIU buffer, and then builds the NIU-to-host buffer. Next it calls NETWORK SIMULATION which updates the RVA. Lastly it calls I/O SIMULATION which polls the network, processes PDUs, and transmits the NIU-to-host buffer to the host.

There is no non-developmental software included in this CSCI.

3.1.2 System States and Modes

The system states are Startup, Idle, Initialize, and Simulate. Startup initializes those parameters that only need to be set once. In Idle the NIU waits for an activation signal from the host. Initialize sets those parameters that need to be reset at the start of each exercise. Simulate is the primary state for the running of an exercise.

The modes of operation are Debug, Use Ethernet, Keyboard Lock, and Verbose. In Debug certain parameters can be traced. The NIU can run with the host with or without an interface to SIMNET. The console keyboard can be locked or unlocked during a simulation. Messages can be displayed in a verbose mode during an exercise.

3.1.3 Memory and processing time allocation

This section lists the amount of memory and the processing time for each CSC.

3.2 CSCI Design Description

Since there is no System Specification or no Software Requirements Specification for this software, it is not possible trace the specified or derived requirements for this CSCI or for any of its CSCs. Such a trace is a requirement of DOD-STD-2167A for this section.

As mentioned above, the organization of the CSCs among the directories and files is confusing. Each of these first and second level CSCs resides in a single function. In some cases the function stands alone in a file; in others it is included along with several other functions in a file. Included in the following descriptions of them is the name of each function and the name of the file in which it resides. Hopefully, this will help the reader trace through the source code.

I suggest that the top level CSCs should be rearranged so that each CSC is in an individual file and the files should be logically arranged in directories.

3.2.1 CSC 1.1, MAIN

MAIN includes the startup state. It is in the file, niu_main.c, in the directory /u2/simnet/niu/src/newcet. It is near line number 211, and there are several other CSCs and CSUs in this file.

It initializes several parameters. It also parses the command line. The command line is entered by the operator at this NIU's console. During startup it calls a routine, sim_state_startup, which resides in the same file as SIMULATION STATE MACHINE. This violates the principle of single entry, single exit modules. This principle is violated many times in this software.

Input data for this CSC is the command typed at the console when the NIU software is started. This CSC produces no output data. After startup is complete, it calls a routine, simulation_state_machine in SIMULATION STATE MACHINE, repeatedly in an infinite loop.

3.2.2 CSC 1.2, SIMULATION STATE MACHINE

SIMULATION STATE MACHINE includes the Idle, Initialize and Simulate states. It is in the file, stt_machine.c, in the directory, /u2/simnet/niu/libcet/libstate. Its function name is simulation_state_machine, it is one choice in a switch (case) statement, it is near line number 160, and there are several CSUs in this file.

It is called from MAIN. It calls the four second level CSCs, TIMERS, VEHICLE SIMULATION, NETWORK SIMULATION, and I/O SIMULATION. It neither sends nor receives any data to or from these second level CSCs.

3.2.3 CSC 1.2.1, TIMERS

TIMERS is in the file, t_simul.c, in the directory, /u2/simnet/libsrc/libtimers. Its function name is timers_simul and it is the only function in this file.

It is called from SIMULATION STATE MACHINE. It increments an elapsed time count by a specified delta t and also increments a counter that counts ticks. It also tests the status of various other timers. There are no parameters passed between TIMERS and any of the other first or second level CSCs.

3.2.4 CSC 1.2.2, VEHICLE SIMULATION

VEHICLE SIMULATION is also in the file, niu_main.c, in the directory /u2/simnet/niu/src/newcet. Its function name is veh_spec_simulate, it is near line number 169, and there are several CSCs and CSUs in this file.

It is called from SIMULATION STATE MACHINE. It resets certain buffer pointers, processes the Host-to-NIU buffer and builds the NIU-to-Host buffer. There are no parameters passed between VEHICLE SIMULATION and any of the other first or second level CSCs.

3.2.5 CSC 1.2.3, NETWORK SIMULATION

NETWORK SIMULATION is in the file, n_net_simul.c, in the directory, /u2/simnet/niu/src/newcet. Its function name is net_simul and it is the only function in this file.

It is called by SIMULATION STATE MACHINE. It updates the RVA. There are no parameters passed between NETWORK SIMULATION and any of the other first or second level CSCs.

3.2.6 CSC 1.2.4, I/O SIMULATION

I/O SIMULATION is in the file, niu_io_simul.c, in the directory, /u2/simnet/niu/src/newcet. Its function name is io_simul, it is near line number 46, and there is also one CSU in this file.

It is called from SIMULATION STATE MACHINE. It polls the network, processes SIMNET PDUs, and sends the NIU-to-Host buffer to the host. There are no parameters passed between I/O SIMULATION and any of the other first or second level CSCs.

In five of the six CSCs described here, the functions rtc_start_time and rtc_stop_time are called. They are in the file, rtc_timing.c, in the directory, /u2/simnet/libsrc/librtc, near line numbers 128 and 134. These functions use the real time clock to measure elapsed time for certain actions. It appears that these elapsed times are never used, except in one case which is described in section 4.6. I suggest that these function calls be placed in conditional compilation blocks and that they write their results to files.

4.1 CSC 1.1, MAIN

MAIN calls several CSUs which are involved in initialization. I will describe these CSUs at a later date. However, one function call, add_rva_stat, passes the wrong number of parameters, appears to locally calculate garbage, and then returns nothing to MAIN. I suggest that this be removed. It resides in file, stats.c, in directory, /u2/simnet/niu/libcet/libnetwork. The call in MAIN is near line number 466.

4.2 CSC 1.2, SIMULATION STATE MACHINE

This CSC calls the above mentioned real time clock routines and calls the four second level CSCs. It contains no CSUs and is sufficiently described in Section 3, Preliminary Design

4.3 CSC 1.2.1, TIMERS

TIMERS calls several CSUs which I will describe at a later date.

4.4 CSC 1.2.2, VEHICLE SIMULATION

This CSC has the function name, veh_spec_simulate, residing in the file, niu_main.c, in the directory, /u2/simnet/niu/src/newcet. It calls two CSUs, MESSAGE PROCESS and MESSAGE BUILD.

4.4.1 CSU 1.2.2.1, MESSAGE PROCESS

MESSAGE PROCESS has the function name, msg_process_buffer. It resides in the file, prc_buf.c, near line number 17, in the directory, /u2/simnet/niu/libcet/libappmsg. It parses a buffer received from the host to determine the type of message and the required action.

Data Flow: The integer, RECEIVE_BUFFER_0, with a value of 10 is passed in the call from veh_spec_simulate, and is never changed. This integer is declared in the header file, libappmsg.h, in the directory /u2/simnet/niu/libcet/libappmsg.

4.4.2 CSU 1.2.2.2, MESSAGE BUILD

MESSAGE BUILD has the function name, msg_bld_vehicle_buffer, residing in the file, bld_updates.c, near line number 23, in the same directory as MESSAGE PROCESS. It builds a buffer of data to be passed to the host, viz., the NIU-to-host buffer.

Data Flow: The integer, SEND_BUFFER_0, with a value of 0, is passed in the call from veh_spec_simulate, and is never changed. This integer is declared in the header file, libappmsg.h, in the directory /u2/simnet/niu/libcet/libappmsg.

4.5 CSC 1.2.3, NETWORK SIMULATION

NETWORK SIMULATION has the function name, net_simul, residing in the file, n_net_simul.c, in the directory, /u2/simnet/niu/src/newcet. This is the only function in this file. This CSC calls the above mentioned real time clock routines. It calls two CSUs, NETWORK TRANSMIT and CHECK RANGE.

4.5.1 CSU 1.2.3.1, NETWORK TRANSMIT

NETWORK TRANSMIT has the function name, network_xmit, residing in the file, nwk_xmit.c, in the directory, /u2/simnet/niu/libcet/libnetwork. It compares the host's position with the RVA calculated position and, if a threshold is exceeded, transmits a PDU over the network.

Data Flow: No parameters are passed from the CSC, NETWORK SIMULATION, to the CSU, NETWORK TRANSMIT. NETWORK TRANSMIT sends a Vehicle Appearance PDU over the network.

4.5.2 CSU 1.2.3.2, CHECK RANGE

CHECK RANGE has the function name, rva_check_range, residing in the file, rva_range.c, in the directory, /u2/simnet/niu/libcet/librva. It determines the range to a vehicle, but does not return a value to the calling function.

4.6 CSC 1.2.4, I/O SIMULATION

I/O SIMULATION has the function name, io_simul, in the file niu_io_simul.c, in the directory, /u2/simnet/niu/src/newcet. This CSC does most of the processing in terms of CPU time. It calls the above mentioned real time clock routines, with one of the calls involving the parameter RTC_NIU_FRAME. This is the only case I've found in which one of the real time clock routines performs a useful function. This parameter is used to stop a loop in I/O SIMULATION when the elapsed time exceeds a pre-designated frame time. The primary purpose of this loop is to poll the network and process PDUs.

I/O SIMULATION calls one CSU, SEND BUFFER.

4.6.1 CSU 1.2.4.1, SEND BUFFER

SEND BUFFER had the function name msg_send_buffer_to_host, in the file msg_send.c, in the directory, /u2/simnet/niu/libcet/libappmsg. This CSU sends the NIU-to-host buffer. This buffer is built in the MESSAGE BUILD CSU described in section 4.4.2, above. I suggest that these two CSUs be brought together under the same CSC.

Data Flow: An integer, SEND_BUFFER_0, with a value of 0, is passed in the call from io_simul. This is the same integer passed in the call from veh_spec_simulate to msg_bld_vehicle_buffer described in section 4.4.1, above.

Section 5. CSCI Data

There are no global data elements passed among the first and second level CSCs.

Section 6. CSCI Data Files

/u2/simnet/niu/bin/load_name	NIU software described here
/u2/simnet/niu/data/nithresh.d	RVA thresholds
/u2/simnet/niu/data/niuprist.d	NIU range limits for targets
/u2/simnet/niu/data/niupars.d	Called from MAIN at startup

BBN's Detailed Design Specification refers to a data file, assoc.def, which I cannot find in the simnet directory or in any sub-directories.

Section 7. Requirements Traceability

The requirements documents do not exist.

Section 8. Notes

8.1 Acronyms

CDRL	Contract Data Requirements List
CET	Combat Evaluation Trainer
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSU	Computer Software Unit
DID	Data Item Description
NIU	Network Interface Unit
PDU	Protocol Data Unit
RVA	Remote Vehicle Approximation
SIMNET	Simulator Network

APPENDIX

Figure A.1 shows the directory hierarchy. The directories are listed in more detail on the following pages.

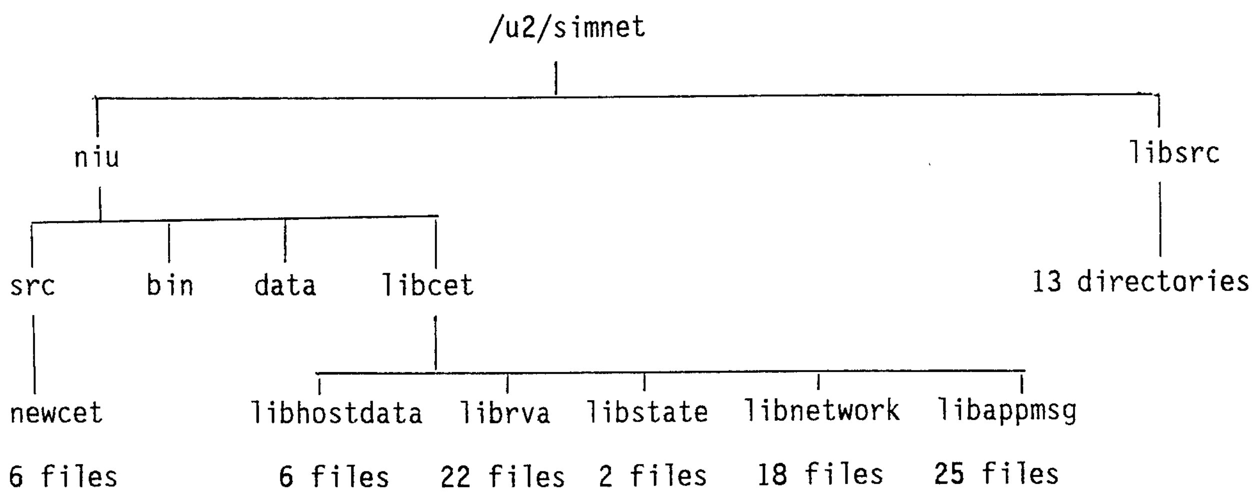


Figure A.1

```
/u2/simnet
total 31
drwxrwxr-x  2 root          512 Feb 15  1991 bin
-rw-rw-r--  1 root        13418 Dec  5 20:54 ctags.out
drwxrwxr-x  6 root         2048 Dec  2 13:11 include
drwxrwxr-x  8 root         3584 Oct 16  1990 include6.0
drwxrwxr-x  2 root          512 Oct 11  1990 lib
drwxrwxr-x 17 root          512 Apr 22  1991 libsrc  - - - - see below
-rwxr-xr-x  1 root          535 Dec  6 07:39 makeall
-rwxr-xr-x  1 root         1860 Dec  5 13:44 maketags
-rwxr-xr-x  1 root          368 Dec  5 13:37 makexref
drwxrwxr-x 13 root          512 Dec  5 13:31 niu   - - - - - - see below
drwxrwxr-x  2 root         1024 Dec  6 07:37 tagsdir
drwxrwxr-x  3 root          512 Dec  5 13:41 tools
drwxrwxr-x  5 root          512 Sep 12  1990 vehicle


/u2/simnet/libsrc
total 23
drwxrwxr-x  3 root          512 Dec 16 11:58 libapp
drwxrwxr-x  2 root         1536 Dec  2 13:11 libassoc
drwxrwxr-x  2 root         1024 Apr 22  1991 libassoc.old
drwxrwxr-x  2 root         1024 Dec  2 13:14 libcif
drwxrwxr-x  2 root         1024 Dec  2 13:15 libfilter
drwxrwxr-x  2 root          512 Dec  2 13:15 libkeybrd
drwxrwxr-x  2 root          512 Dec  2 13:17 libmap
drwxrwxr-x  2 root         3584 Dec  2 13:21 libmatrix
drwxrwxr-x  2 root          512 Dec  2 13:22 libmove
drwxrwxr-x  2 root         1536 Dec  2 13:25 libnetif
drwxrwxr-x  2 root         1536 Sep 12  1990 libnetif.old
drwxrwxr-x  2 root          512 Dec  2 13:26 librtc
drwxrwxr-x  2 root          512 Dec  2 13:26 libshm
drwxrwxr-x  2 root         1536 Dec  2 13:28 libtimers
drwxrwxr-x  2 root         1536 Dec  2 13:30 libutil


/u2/simnet/niu
drwxrwxr-x  2 root          512 Dec 17 09:16 bin
drwxrwxr-x  2 root          512 Sep 12  1990 data
drwxrwxr-x  2 root          512 Dec  2 13:44 include
drwxrwxr-x  8 root          512 Sep 26 15:27 lib
drwxr-xr-x  7 root          512 Aug  9 10:30 libcet  - - - - see below
drwxrwxr-x  7 root          512 Sep 26 15:27 libcstar
drwxrwxr-x  6 root          512 Oct 11  1990 libgci
drwxr-xr-x  7 root          512 Aug  9 10:25 libmdrc
drwxrwxr-x  7 root          512 Sep 12  1990 libsrc
drwxrwxr-x  7 root          512 Dec  4  1990 libtg
-rw-rw-r--  1 root         6344 Apr  9  1990 niunotes
-rw-rw-r--  1 root         6344 Apr  9  1990 niunotes%
drwxrwxr-x  9 root         1024 Dec  5 13:30 src   - - - - - - see below


/u2/simnet/niu/src/newcet
-rw-rw-r--  1 root         1868 Aug  9 10:36 n_net_simul.c
-rw-rw-r--  1 root         1692 Aug  9 10:36 niu_frm_ctl.c
-rw-rw-r--  1 root         3994 Dec 16 12:12 niu_io_simul.c
-rw-rw-r--  1 root         6643 Dec 11 13:08 niu_keybrd.c
-rw-rw-r--  1 root        13031 Dec 16 12:12 niu_main.c
-rw-rw-r--  1 root         3049 Aug  9 10:36 niu_network.c
-rw-rw-r--  1 root         4153 Aug  9 10:36 tmp.c
```

```
/u2/simnet/niu/libcet/libhostdata
-rw-rw-r--  1 root         907 Aug  9 10:30 hd_alloc.c
-rw-rw-r--  1 root        1631 Aug  9 10:30 hd_cet.c
-rw-rw-r--  1 root        1160 Aug  9 10:30 hd_event.c
-rw-rw-r--  1 root        4564 Aug  9 10:30 hd_host.c
-rw-rw-r--  1 root        1318 Aug  9 10:30 hd_update.c
-rw-rw-r--  1 root        1949 Aug  9 10:30 hd_veh.c

/u2/simnet/niu/libcet/librva
-rw-rw-r--  1 root        6539 Aug  9 10:29 rva_adjust.c
-rw-rw-r--  1 root        4613 Aug  9 10:29 rva_blades.c
-rw-rw-r--  1 root        2683 Aug  9 10:29 rva_debug.c
-rw-rw-r--  1 root        2839 Aug  9 10:29 rva_forget.c
-rw-rw-r--  1 root        6602 Aug  9 10:29 rva_hash.c
-rw-rw-r--  1 root        2124 Aug  9 10:29 rva_init.c
-rw-rw-r--  1 root        1848 Aug  9 10:29 rva_invis.c
-rw-rw-r--  1 root        4179 Aug  9 10:29 rva_lists.c
-rw-rw-r--  1 root         739 Aug  9 10:29 rva_loc.c
-rw-rw-r--  1 root        1864 Dec  2 13:51 rva_pr_get.c
-rw-rw-r--  1 root        9362 Aug  9 10:29 rva_pr_init.c
-rw-rw-r--  1 root         585 Aug  9 10:29 rva_pr_loc.c
-rw-rw-r--  1 root        2366 Aug  9 10:29 rva_pr_rm.c
-rw-rw-r--  1 root        1621 Dec 10 16:09 rva_range.c
-rw-rw-r--  1 root        1202 Aug  9 10:29 rva_setup.c
-rw-rw-r--  1 root       11738 Dec 10 16:12 rva_smooth.c
-rw-rw-r--  1 root        1424 Aug  9 10:29 rva_tick.c
-rw-rw-r--  1 root        7530 Dec 11 08:00 rva_update.c
-rw-rw-r--  1 root         978 Aug  9 10:29 rva_veh_app.c
-rw-rw-r--  1 cindy       1832 Dec 10 16:09 tmp_range.c
-rw-rw-r--  1 cindy       8855 Dec 10 14:45 tmp_smooth.c
-rw-rw-r--  1 cindy       7538 Dec 10 16:22 tmp_update.c

/u2/simnet/niu/libcet/libstate
-rw-rw-r--  1 root        5790 Aug  9 10:30 stt_machine.c
-rw-rw-r--  1 root        6450 Aug  9 10:30 stt_pars.c

/u2/simnet/niu/libcet/libnetwork
-rw-rw-r--  1 root        1632 Aug  9 10:29 nwk_act_ack.c
-rw-rw-r--  1 root        6659 Dec 11 13:46 nwk_activ.c
-rw-rw-r--  1 root        1052 Aug  9 10:29 nwk_appear.c
-rw-rw-r--  1 root        1778 Aug  9 10:29 nwk_datagram.c
-rw-rw-r--  1 root        3062 Aug  9 10:29 nwk_deact.c
-rw-rw-r--  1 root        1775 Aug  9 10:29 nwk_fire.c
-rw-rw-r--  1 root        1285 Aug  9 10:29 nwk_header.c
-rw-rw-r--  1 root        1629 Aug 13 18:06 nwk_impact.c
-rw-rw-r--  1 root        2465 Aug  9 10:29 nwk_init.c
-rw-rw-r--  1 root        9041 Dec 10 10:05 nwk_pkt.c
-rw-rw-r--  1 root        3303 Aug 13 17:37 nwk_radar.c
-rw-rw-r--  1 root        1658 Aug  9 10:29 nwk_rsp.c
-rw-rw-r--  1 root        5739 Aug  9 10:29 nwk_stats.c
-rw-rw-r--  1 root        5333 Dec  5 10:11 nwk_thresh.c
-rw-rw-r--  1 root        1882 Aug  9 10:29 nwk_trans.c
-rw-rw-r--  1 root        2889 Aug  9 10:29 nwk_xmit.c
-rw-rw-r--  1 root        1595 Aug  9 10:29 params.c
-rw-rw-r--  1 root        3374 Aug  9 10:29 stats.c
```

```
/u2/simnet/niu/libcet/libappmsg
-rw-rw-r--  1 root          2468 Aug  9 10:29 app_end.c
-rw-rw-r--  1 root          2774 Aug  9 10:30 app_init.c
-rw-rw-r--  1 root          4916 Dec  6 15:40 bld_status.c
-rw-rw-r--  1 root          3064 Aug 12 13:09 bld_updates.c
-rw-rw-r--  1 root          2539 Aug  9 10:29 buf_reset.c
-rw-rw-r--  1 root          2582 Aug  9 10:29 buf_setup.c
-rw-rw-r--  1 root           983 Aug  9 10:30 check_sizes.c
-rw-rw-r--  1 root           621 Aug  9 10:30 clr_n_mapped.c
-rw-rw-r--  1 root           488 Aug  9 10:30 get_init_buf.c
-rw-rw-r--  1 root           613 Aug  9 10:30 get_n_mapped.c
-rw-rw-r--  1 root          3257 Aug  9 10:30 host_prepare.c
-rw-rw-r--  1 root          1825 Aug  9 10:30 msg_loc.c
-rw-rw-r--  1 root          2855 Aug  9 10:29 msg_recv.c
-rw-rw-r--  1 root          1777 Aug  9 10:30 msg_send.c
-rw-rw-r--  1 root          3424 Aug  9 10:30 prc_buf.c
-rw-rw-r--  1 root          2596 Aug  9 10:30 prc_destroy.c
-rw-rw-r--  1 root          3197 Aug  9 10:30 prc_fire.c
-rw-rw-r--  1 root          2939 Aug  9 10:30 prc_impact.c
-rw-rw-r--  1 root          1116 Aug  9 10:30 prc_onlinereq.c
-rw-rw-r--  1 root          6872 Aug  9 10:30 prc_status.c
-rw-rw-r--  1 root          2467 Aug  9 10:30 pre_msg.c
-rw-rw-r--  1 root          1697 Aug  9 10:30 rawsend.c
-rw-rw-r--  1 root          1711 Aug  9 10:30 set_if.c
-rw-rw-r--  1 root          1124 Aug  9 10:30 setup_buf.c
-rw-rw-r--  1 root          1110 Aug  9 10:30 wait.c
```

SYSTEM SPECIFICATION

for the MULTIRAD SIMULATION SYSTEM

This is a suggested shell for a system specification for simulator network. This specification is being limited to simulators in a local area network. Specifications for wide area networks is beyond the scope of this short research project.
The title page and sections 1 and 2 are self-explanatory. See Appendix A for examples.

3. System Requirements

3.1 Definition

This is the system specification for the software in the network interface units in a network of aircraft simulators.

3.2 Characteristics

3.2.1 Performance Characteristics

The performance characteristics include a list of the states in which the system can exist and, for each state, a list of the modes of operation.

3.2.1.1 Startup State

Each simulator is started. The software in each simulator and in each NIU is booted up. Test data are passed through the network to test for a successful startup. There is only one mode of operation.

3.2.1.2 Inititialization State

This occurs prior to each simulation run. It is executed from the controller's station. Simulators are designated as active or inactive. Initial latitude, longitude, altitude, and velocity vectors are set for each active simulator. The configuration, e.g., weapons load, is set for each simulator. Terrain features and sun position is set for each simulator. Ground threat simulators are initialized in a similar manner. There is only one mode of operation.

3.2.1.3 Simulation State

The simulated combat scenario is played out in this mode.

3.2.1.3.X X-mode of Operation

A separate sub paragraph is written here for each mode. For the various modes, the value of X is numerically ordered beginning with 1. Suggested modes are:

- Debug, with and without network
- Research, with and without network
- Aircrew Training
- Aircrew Testing, with and without kill removal

3.2.2 System Capability Relationships - deleted

3.2.3 External Interface Requirements

3.2.3.X X-device External Interface Description

A separate sub paragraph is written here for each external device. These would include each aircraft simulator, threat generator, GCI station, and controller's station. For future expansion, the gateway to wide area networks would be described here.

3.2.4 through 3.2.9 - deleted

These requirements apply to hardware only.

3.3 Design and Construction

3.3.1 through 3.3.10 - deleted

These requirements apply to hardware only.

3.3.11 Computer Resource Reserve Capacity

Here the requirements for CPU speed and memory size shall be specified. The percentage of these parameters that are actually used for the deliverable system shall also be specified.

3.4 Documentation

The following DIDs will be deliverable documents

DI-CMAN-80008A	SYSTEM SPECIFICATION
DI-MCCR-80012A	SOFTWARE DESIGN DOCUMENT
DI-MCCR-80013A	VERSION DESCRIPTION DOCUMENT
DI-MCCR-80014A	SOFTWARE TEST PLAN
DI-MCCR-800A5A	SOFTWARE TEST DESCRIPTION
DI-MCCR-80017A	SOFTWARE TEST REPORT
DI-MCCR-80021A	SOFTWARE PROGRAMMER'S MANUAL
DI-MCCR-80025A	SOFTWARE REQUIREMENTS SPECIFICATION
DI-MCCR-80026A	INTERFACE REQUIREMENTS SPECIFICATION
DI-MCCR-80027A	INTERFACE DESIGN DOCUMENT
DI-MCCR-80029A	SOFTWARE PRODUCT SPECIFICATION
DI-MCCR-80030A	SOFTWARE DEVELOPMENT PLAN
DI-CMAN-80534	SYSTEM DESIGN DOCUMENT

APPENDIX C
DI-MCCR-80012A

SOFTWARE DESIGN DOCUMENT
for the
MULTIRAD NIU CSCI

This is a suggested shell for a software design document for simulator network. See Appendix A for examples of the title page and table of contents.

Section 1. Scope

1.1 Identification

CSCI number : 1

CSCI title : MULTIRAD NIU

System : MULTIRAD, Armstrong Laboratory, Williams AFB, AZ

1.2 System Overview

MULTIRAD (Multi-ship Research and Development) is a network of interconnected aircraft simulators. The protocol for the network is SIMNET, developed by BBN Corp. The physical layer of the network is an Ethernet. The nodes on the network are aircraft simulators, threat generators, and an operator station. Each node has an NIU (Network Interface Unit) between the device (simulator, threat generator, etc.) and the Ethernet. The purpose is to allow multi-ship training. The pilot in one simulator is presented with images of the simulated aircraft being flown by pilots in other simulators. There can also be computer generated threats displayed to the simulator pilot.

The CSCI (Computer Software Configuration Item) described here is the software that resides in the various NIUs.

1.3 Document Overview

This document provides a description of the NIU software. It contains both preliminary design and detailed design information. It is the primary source for coders, testers, and modifiers of the NIU software.

This document is written in accordance with DOD-STD-2167A, Miliary Standard Defense System Software Development. More specifically, it is written in accordance with DID No. DI-MCCR-80012A, Software Design Document.

Section 2. Referenced Documents

BBN's Report No. 7102 The SIMNET Network and Protocols, July 1989
BBN's NIU Detailed Design Specification, May 1991

Section 3. Preliminary Design

3.1 CSCI Overview

This CSCI is the software that resides in various NIUs. It is written in the C language. The following description of NIU software in general is extracted from the BBN NIU Detailed Design Specification, Section 2.1 NIU Description, page 6.

> The NIU provides the capability to connect dissimilar simulators to an existing SIMNET network. The NIU transforms the existing simulator data into SIMNET PDUs and provides additional overhead required to support the SIMNET protocol. The NIU allows simulators with non-homogeneous frame times to be interoperable on the same network.

This NIU is the interface between the host simulator and the SIMNET. Two data structures are passed between the NIU and the host: The Host-to-NIU buffer and the NIU-to-host buffer. There are several PDUs (Protocol Data Units) passed between the SIMNET and the NIU. These PDUs carry information from one simulator to another for the purpose of generating images. These are described in detail in the BBN Report 7102.

3.1.1 CSCI Architecture

This CSCI is subdivided into several CSCs. At the top level is the MAIN CSC. At the second level are 2 CSCs: INITIALIZE, and RUN. At the third level are four CSCs, NETWORK I/O, HOST I/O, RVA, and PDU. For identification, these CSCs are numbered as follows:

	CSC
1	MAIN
2.1	INITIALIZE
2.2	RUN
3.2.1	NETWORK I/O
3.2.2	HOST I/O
3.2.3	RVA
3.2.4	PROCESS PDU

HOST I/O consists of several versions of similar CSCs, one for each simulator, threat generator station, GCI simulator, and controller's station.

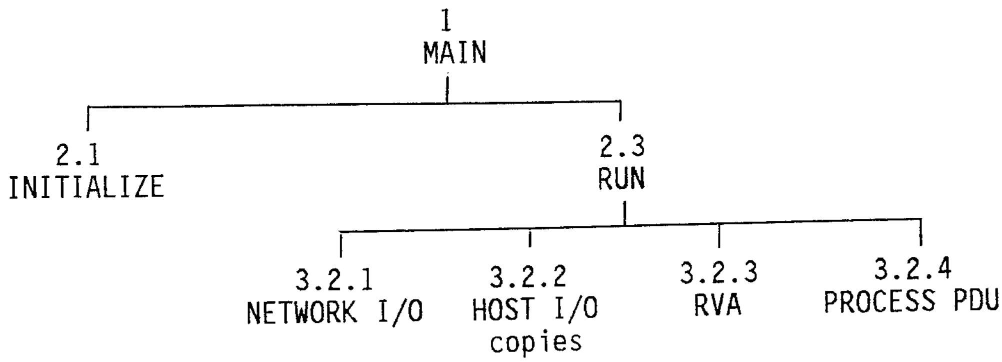


Figure 3.1

MAIN calls several initialization routines, and then enters an infinite loop in which it repeatedly calls RUN.

RUN calls the four CSCs at the second level.

There is no non-developmental software included in this CSCI.

3.1.2 System States and Modes

The system states are Startup, Initialize, and Simulate. Startup initializes those parameters that only need to be set once. Initialize sets those parameters that need to be reset at the start of each exercise. Simulate is the primary state for the running of an exercise.

The modes of operation are Debug with Ethernet, Debug without Ethernet, Aircrew training. In Debug certain parameters can be traced. The NIU can run with the host with or without an interface to SIMNET.

3.1.3 Memory and processing time allocation

This section lists the amount of memory and the processing time for each CSC.

3.2 CSCI Design Description

The following are descriptions of each of the four CSCs.

3.2.1 MAIN CSC

MAIN is numbered 1.1. It includes the startup state. After startup, it waits for a command from the controller's station to begin initialization. It calls INITIALIZE and RUN. Input data for this CSC is the command typed at the console when the NIU software is started. This CSC produces no output data. After initialization is complete, it calls RUN repeatedly in an infinite loop.

3.2.2 INITIALIZE CSC

INITIALIZE is numbered 2.1. It includes the initialization state. It derives requirements from the MULTIRAD system specification, paragraph 3.2.1.2.

It's function is to allow latitude, longitude, altitude, initial velocity vectors, and aircraft configuration to be set prior to a run.

It is called from MAIN and calls no other CSCs.

3.2.3 RUN CSC

RUN is numbered 2.3. It includes the simulate state. It derives requirements from the MULTIRAD system specification, paragraph 3.2.1.3.

It's function is to control its four subordinate CSC's during a run. This includes processing information necessary for this control.

It is called from MAIN. It calls NETWORK I/O, HOST I/O, RVA, and PDU.

3.2.4 NETWORK I/O CSC

NETWORK I/O is numbered 3.2.1. Its function is to communicate with the Ethernet. This includes transmitting and receiving PDUs. It is called from RUN. With respect to PROCESS PDU CSC, the PDUs must be global data elements.

3.2.5 HOST I/O CSC

HOST I/O is numbered 3.2.2. Its function is to communicate with the host simulator for the respective node on the network. This includes transmitting and receiving messages to and from the host simulator. This also includes encoding and decoding these messages.

3.2.6 RVA CSC

RVA is numbered 3.2.3. Its function is to calculate the position of all other vehicles on the network for presentation to its host simulator. This requires receiving old positions of the other vehicles from the network and using predictor corrector algorithms to calculate their current positions.

3.2.7 PROCESS PDU CSC

PROCESS PDU is numbered 3.2.4. Its function is to encode PDUs for transmission over the network and to decode PDUs received from the network.

Section 4. Detailed Design

It is not reasonable to continue with the detailed design until the government and the contractor approve the preliminary design.

For the same reason, Sections 5, 6, and 7 are omitted

Section 8. Notes

8.1 Acronyms

CDRL	Contract Data Requirements List
CET	Combat Evaluation Trainer
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSU	Computer Software Unit
DID	Data Item Description
NIU	Network Interface Unit
PDU	Protocol Data Unit
RVA	Remote Vehicle Approximation
SIMNET	Simulator Network

APPENDIX D
DI-MCCR-80027A

INTERFACE DESIGN DOCUMENT

for the CET/NIU INTERFACE

This is a suggested shell for an interface design document. It is suggested that the Interface Control Document for the Combat Engagement Trainer/Network Interface unit prepared by General Electric Government Services, 15 January 1991, be transformed to this format.

The title page and sections 1 and 2 are self-explanatory. See Appendix A for examples. The information on the title page and page one of GE's document can easily be changed to the 2167A format, since it already contains the required information.

3. Interface Design

3.1 Interface Diagrams

This requires a diagram showing the position of this interface in the simulator network.

3.2 CET/NIU Interface (identifier)

This requires a unique identifier for each interface in the simulator network.

3.2.1 Data Elements

This information is in section 3.2 of GE's document.

3.2.2 Message Description

This information is in section 3.4 of GE's document.

3.2.3 (deleted)

Interface priorities is not a factor here.

3.2.4 Communications Protocol

3.2.4.1 SIMNET

This information is in sections 3.2.4 and 3.3 of GE's document.

4. Notes

This section contains general information not covered above.

COMPONENTS OF SPATIAL AWARENESS:

VISUAL EXTRAPOLATION AND TRACKING OF MULTIPLE OBJECTS

Itiel E. Dror
Department of Psychology

Harvard University
33 Kirkland St.
Cambridge, MA 02138

Final Report for:
Research Initiation Program
Armstrong Laboratories

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, D.C.

and

Harvard University

December 1992

COMPONENTS OF SPATIAL AWARENESS: VISUAL EXTRAPOLATION AND TRACKING OF MULTIPLE OBJECTS

Itiel E. Dror
Department of Psychology
Harvard University

Abstract

Air Force pilots and control subjects participated in two experiments that assessed different processing components involved in spatial awareness. The experiments examined the ability to extrapolate and track motion of multiple objects. I found that all subjects were able to extrapolate the motion of three objects and track the motion of six objects at accuracy levels above chance. I found that although all subjects had comparable performance on both tasks, experienced fighter pilots extrapolated motion differently than novice pilots and the control subjects. Although experienced fighter pilots were more accurate at the most difficult condition, they were not more accurate across all conditions. I also found a "recency of flying" effect for motion tracking skill: pilots who recently flew were more accurate than pilots who did not. No such effect was found for motion extrapolating skills.

COMPONENTS OF SPATIAL AWARENESS: VISUAL EXTRAPOLATION AND TRACKING OF MULTIPLE OBJECTS

Itiel E. Dror

Spatial awareness involves being aware of one's own position relative to objects in the surrounding environment. Maintaining spatial awareness in a static environment is quite effortless, but in a dynamic environment it is very difficult. Maintaining spatial awareness in a dynamic environment requires not only that one account for one's own motion, which by itself changes the relative position of every object in the surrounding environment, but also that one take into account the motion of other objects. For example, when driving a car and preparing to come into an intersection and make a left turn, one's own motion is very important as it changes one's relative position to the upcoming intersection, but one must also account for the motion of cars that are entering the intersection. Maintaining spatial awareness is especially difficult when multiple objects are in motion at the same time. Tracking multiple objects in motion is very demanding on the attention system, and various models have been proposed to account for the way attention is allocated to multiple objects. One can quickly shift attention between the objects, or expand the region that one is attending to at one given moment. If attention is quickly shifted, then it can be guided by the location of the objects or by different regions of space (for a good review see Yantis, 1992).

In a dynamic environment one must occasionally disengage momentarily from direct visual contact with the surrounding objects. For instance, when one is driving and is trying to tune to a radio station, vision may shift to the radio dial. Thus, maintaining spatial awareness requires extrapolating the motion of objects when one cannot directly track them. The process of extrapolating motion seems to recruit various imagery mechanisms. Specifically, the processes of image scanning and image transformation may play an important role in extrapolation of motion. Image scanning can be used to scan along a trajectory of a moving object even when the object is not visible, which enables to anticipate its position. Image transformation can be used to shift and rearrange objects in an image, which enables to keep track of the spatial position of moving objects that are no longer visible.

Spatial awareness is particularly important in the tactical aviation environment in which highly agile fighter aircraft are designed to operate across the spectrum of flight altitudes and orientations. Pilots rely on spatial awareness for the successful accomplishment of missions as well as for safety. The high speed and maneuverability of fighter aircraft result in constant and rapid changes in the spatial positioning of the aircraft. The spatial relations to stationary ground targets as well as to other rapidly moving aircraft constantly change. Consequently, pilots must always assess and update their situation awareness. Pilots cannot constantly maintain direct visual contact with their surroundings because they frequently need to attend to cockpit displays; thus, they are required to track and to extrapolate motion. Failure to maintain spatial awareness can have disastrous consequences as evidenced by the large number of accidents that have been attributed to loss of situation awareness and spatial disorientation. Furthermore, maintaining spatial awareness during air combat is considered to be an essential element of success.

Such observations led me to examine pilots' abilities to track and extrapolate motion of multiple objects. In an earlier study on visual-spatial abilities of pilots (Dror, 1991; Dror, 1992; Dror, Kosslyn, & Waag, 1992), pilots were found to be better in some visual-spatial abilities but not in others. Specifically, they were found to be better at making metric spatial judgments, but not categorical spatial judgments; and they mentally rotated objects better than non-pilots. In contrast, pilots did not extrapolate motion of a single object, scan images, or extract visual features from objects obscured by visual noise better than non-pilots. It was of great interest to determine the relation between the results of the previous and present study. This was especially interesting because I previously found that pilots were better than non-pilots at image transformation, but not at image scanning, and both processes may by used in extrapolation of motion.

The goal of this study was to understand and characterize some of the processing subsystems involved in spatial awareness and the factors that influence them. Two tasks were designed to tap into the underpinnings of motion extrapolation and motion tracking. The difficulty of trials within each task was manipulated, so as to vary the amount of processing and thus affect accuracy rates and response times. The slope of increased response time and error rate as a function of difficulty reflects the processing ability of specific components *per se*- independent of the processes involved in encoding the stimulus and generating the response itself (which are reflected by the intercept of the function). This method allowed me to evaluate the ability of components that perform specific types of processing (Sternberg, 1969).

General Methods

The subjects were tested individually on each task in one testing session, which lasted on the average fifty minutes. Half the subjects were first tested on the motion tracking task and then on the motion extrapolation task; the other half of the subjects were tested in the reverse order. The subjects were given verbal instructions and then were asked to paraphrase them, and any misconceptions were corrected. The subjects began each task with a set of practice trials. During the practice trials, the computer gave feedback by beeping when the subject made an incorrect response, and the subjects were encouraged to ask questions. During the test trials, no feedback was provided and no talking was allowed. The tasks required the subjects to respond by pressing keys marked "yes" (the "b" key) and "no" (the "n" key) on the computer's keyboard. The subjects used two fingers of their dominant hand to press the keys. The tasks were administered on a Macintosh II ci computer with a high resolution video display card (24 bit video card). The computer was connected to a color 13 inch multiscan trinitron super fine pitch Sony monitor. The tasks were administered by a computer program that used the Shell and Macglib libraries of Micro M.L. Inc. All subjects sat so that their heads were approximately 50 centimeters from the computer screen. The subjects were asked to respond as quickly as possible while remaining as accurate as possible.

Subjects

Fifty-eight subjects were tested in both experiments. Thirty-four were pilots, who formed two distinct subgroups: 20 pilots were very experienced and highly trained fighter pilots, and 14 pilots were novice pilots who had never flown fighter aircraft. The mean flying hours in fighter aircraft (F-15 and F-16) of the first subgroup was 1650 (range 1000-3100). Their overall mean flying hours --including non-fighter aircraft-- was 2300 (range 1400-3400). Their mean age was 34.8 (range 29-41). The second subgroup of pilots included pilots who had never flown a fighter aircraft, but were assigned to be fighter pilots. They all had recently finished the undergraduate training of the Air Force and were waiting for their fighter aircraft training to began. The pilots in this group had approximately 300 flying hours on non-fighter aircraft. Their mean age was 26.1 (range 24-29). All the pilots were males and had completed at least a college education. The pilots were recruited and tested at Luke Air Force Base, AZ. The 24 control subjects were Harvard graduate students who were matched to the pilots by gender, age, and education. They were all males who have graduated college, their mean age was 27.5 (range 23-37).

Experiment 1: Visual Extrapolation

Visual imagery enables us to mentally manipulate the visual stimuli we receive from the outside world. We can manipulate and transform images in numerous ways. One such manipulation encompasses shifting and changing the spatial position of objects within a complex image (e.g., one can imagine rearranging items on the desk --shifting the phone and the computer to different locations). Indeed, such manipulations are often used in reasoning (Hayes, 1981). Motion can be accomplished in mental imagery by constantly shifting the spatial positions of items within the image. Motion extrapolation requires such changes, whereby the imagined motion is based on previously seen motion. Correct timing and spatial perception are also key factors that enable the imagined motion to be in the same speed and trajectory as the previously seen motion. Thus, motion extrapolation requires the use of imagery as well as perception of time and space.

I used a visual extrapolation task that was a variant of one devised by Dror (1991; see also Dror, Kosslyn, and Waag, 1992). In the previous study, subjects were required to extrapolate the motion of one object that was moving in a circular trajectory. I did not find that pilots were better at this type of task. In the present study subjects were required to extrapolate the motion of multiple objects moving in various straight trajectories. I assumed that it is more demanding when the motion of more objects had to be extrapolated. A group of moving balls was presented, and the subjects attended to a subgroup of them that were flashing. All balls were then removed from the display, and the screen remained blank. After a delay the balls reappeared, as if they had been in constant motion while the screen was blank. One ball was presented as a probe and the subjects were required to judge whether or not this ball was one of the balls that was flashing earlier.

Method

Materials. The stimuli were round disks (the "balls") 18 pixels in diameter – corresponding to 0.668 cm and 0.765 degrees of visual angle– which moved in straight trajectories on the screen. The balls' initial trajectories were random. The balls bounced off the walls of the screen but moved through each other. Twelve black balls were presented in motion on the screen while a subset of them (1, 2, or 3) flashed –changed colors from black to red and back to black every 60 ms. The speed was 4.33 cm per second, which corresponded to 4.95 degrees of visual angle. The motion was created by displaying a new screen every 60 ms with each ball advancing 7 pixels in its trajectory --corresponding to 0.260 cm and 0.298 degrees of visual angle. The computer generated the new position for each of the balls throughout the trial. While the screen was blank the computer continued to generate the motion of the balls, however the balls were not displayed on the computer screen.

A total of 96 trials were prepared. The trials were constructed in 8 blocks of 12 trials each. Half the trials in each block should have been evaluated as "yes" trials --presenting a ball probe that had flashed earlier-- and the other half were "no" trials --presenting a ball that had not flashed earlier. For each of the 6 "yes" and "no" trials in every block, half had a short time delay and half a long time delay. I increased the short and long time delays used in our previous study (Dror et al., 1992) as well as increasing the time difference between them. The short time delay was now 2 seconds and the long one 3.5 seconds. For each 3 trials in every block that had the same time delay and the same correct response, one required extrapolating the motion of 1 ball, one the motion of 2 balls, and one the motion of 3 balls. Thus, each block had 12 trials that included all possible combinations of variables. The trials within each block were randomly ordered with the constraint that the same number of balls, time delay, or response could not appear more than three times in succession. An additional 12 trials were prepared as practice trials. The block of practice trials had the same structure as the testing blocks and included all variable combinations.

Procedure. A trial began with an exclamation mark. When the subject was ready, he pressed the space bar, and then 12 moving balls appeared on the computer screen. The subject was told to track the motion of the flashing balls, and to keep tracking them after the balls disappeared, as if they were still moving on the computer screen. When the balls re-appeared, one of the 12 original balls was flashing. The subject was to respond "yes" if the flashing ball was one of the balls that was flashing earlier, and "no" if it was not. Immediately after the response, another exclamation mark appeared and a new trial began.

Results

The data were analyzed using an analysis of variance. Separate analyses were performed for response times and error rates. The data included 0.8% responses that were outliers, which were defined as times that were greater than 2.5 times the mean of the remaining scores in that cell; these times were replaced by the mean of the cell. Incorrect responses were excluded from the analysis of response times. The first analysis compared the control subjects to the pilots, and included the number of trajectories that were extrapolated (the number of balls) as a within-subjects variable, and profession (pilots vs. controls) as a between-subjects variable. I was interested in whether the within-subjects variable (which taps into the extrapolation process *per se*) interacted with the between-subjects variable (which represents training and experience of pilots). Such an interaction would show whether training and experience in piloting affects components involved in motion extrapolation. However, I first had to establish that our difficulty manipulation, i.e. the number of balls, did indeed effect performance, and that the subjects were indeed able to perform the task.

The subjects were able to perform the tasks at a relative high accuracy level of 74%, they even had an accuracy level well above chance, of 63%, at the most difficult condition of extrapolation the motion of three objects. The number of moving balls that were extrapolated affected both response times and error rates, $F(2,112)= 85.85$, $p< .01$ for response time (with means of 1227, 1443, and 1557 ms for 1, 2, and 3 balls, respectively), and $F(2,114)= 107.87$, $p<$

.01, for error rates (with means of 14.0, 27.1, and 37.1 percent error for 1, 2, and 3 balls, respectively). Indeed, linear contrasts revealed that response times increased linearly with the number of trajectories that were extrapolated, $F(1,112)= 177.56$, $p< .01$, as did error rates, $F(1,112)= 229.63$, $p< .01$. I then examined the statistical power of our main effect and calculated its effect size; the "r" was very large, .78 in response time, and .82 in error rate.

After establishing that the manipulation of the number of trajectories that were extrapolated did tap onto extrapolation performance, I proceeded to see whether piloting affects this ability. I found no interaction between the number of trajectories that were extrapolated and profession group (pilots vs. non-pilots), $F< 1$ for response time, and $F(2,112)= 1.26$, $p> .2$ for error rates. Pilots and the control non-pilots had comparable overall performance, $F< 1$ for response time and $F(1,56)= 1.26$, $p> .2$ for error rate.

I proceeded by separating the pilots' data into two distinct groups, according to training and experience. I assumed that if piloting affects the ability to extrapolate, then I have a better chance of picking it up if I use experience as a variable. The experienced pilots were all highly trained and highly experienced fighter pilots; the novice pilots were all pilots who graduated from basic pilot training and were found to be qualified to become fighter pilots, but had not yet began their fighter training. I thus replaced the between variable of profession (pilots vs. non-pilots) by an experience variable (no experience at all--the control subjects, basic pilot training, and experienced fighter pilots). I found no interaction in response time, $F< 1$, however, as illustrated in Figure 1, I found an interaction between experience and number of balls in the error rate, $F(4,110)= 3.28$, $p= .01$. This interaction reflected that the fighter pilots had a distinct pattern of errors, which was different from both the novice pilots and the control subjects. Further analysis confirmed this hypothesis, when I compared only the fighter pilots to the novice pilots I found an interaction, $F(2,64)= 5.15$, $p< .01$, as well as when I compared only the fighter pilots to the control subjects, $F(2,84)= 3.64$, $p= .03$. However, no interaction was observed when I compared only the novice pilots to the control subjects, $F(2.72)= 1.40$, $p> .25$. The distinct pattern of errors of the fighter pilots reflected that they were more accurate at the easiest and at the hardest conditions, however, they had higher errors in the intermediate condition. I found that the manipulation of the number of balls was significant, $F(2,110)= 85.60$, $p< .01$ for response time, and $F(2,110)= 121.80$, $p< .01$ for error rate. All three groups had comparable overall performance, $F< 1$ for both response time and error rate.

Finally I compared the two subgroups of pilots by adding an additional factor, how recently they had flown. Pilots who had more than 10 flying hours in the 2 weeks prior to the testing and over 15 in the 4 weeks prior to the testing were considered as pilots who had flown recently, and pilots who had not flown at all in the 2 weeks prior to testing and less than 4 hours in the 4 weeks prior to testing were considered as pilots who had not flown recently. I found that pilots who had flown recently and pilots who had not flown recently had comparable performance, $F< 1$ for both response time and comparable error rate.

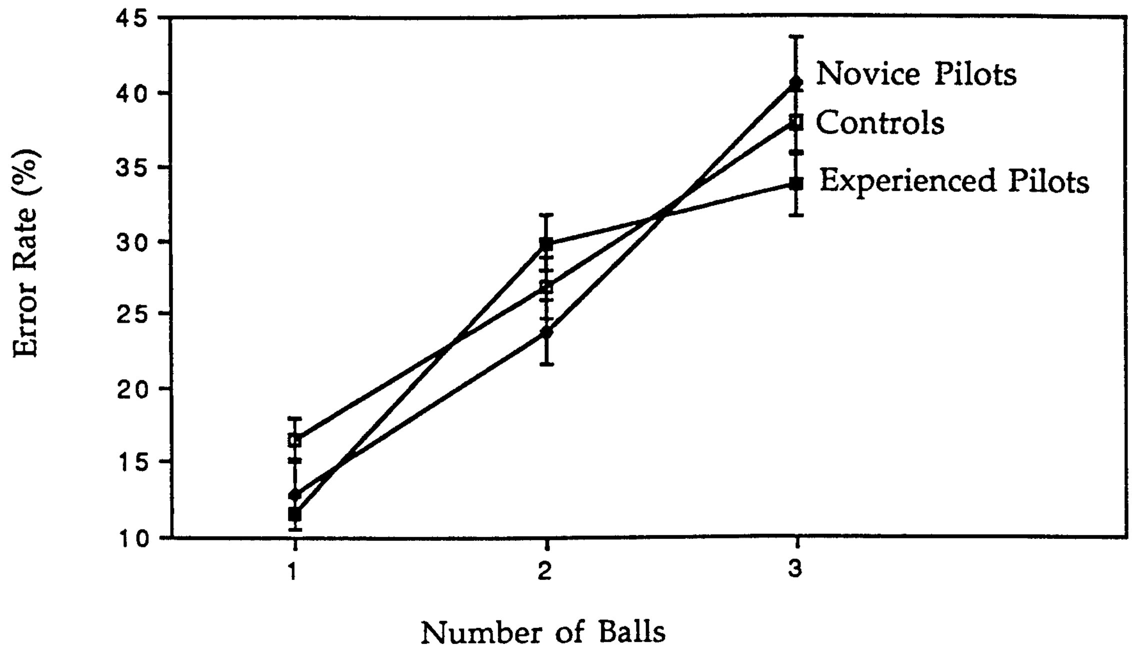


Figure 1: The results from the motion extrapolation task.

Discussion

I found that pilots, as well as control subjects, can extrapolate motion of multiple objects. They were able to extrapolate the motion of three moving objects at above-chance levels of accuracy. I found that increasing the number of trajectories that were extrapolated caused linear increases in response time and error rate.

The results suggest that fighter pilot training and experience affects extrapolation ability, but that this effect is not a simple enhancement of performance. I found that although fighter pilots had superior performance in the most difficult conditions, relative to novice pilots and control non-pilots, their performance was not better across all conditions. The importance of the specific training and experience that fighter pilots undergo is further emphasized by the fact that I found no difference between the novice pilots and the control non-pilots. In a previous study (Dror et al., 1992) I did not detect any differences between pilots and non-pilots in extrapolating motion. The present results are consistent with that finding. In the present study I also did not detect any difference when I compared the pilots to the controls, however, given that I tested 34 pilots who formed two distinct groups, I was able to detect a difference between the subgroup of experienced fighter pilots and the other subjects (controls and subgroup of novice pilots).

Experiment 2: Motion Tracking

Tracking a single ball that is moving at high speed is not an easy task. Professional baseball batters need to track a ball moving at speeds of up to 100 mph --producing angular velocities greater than 500 degrees per second. Indeed, professional baseball batters have superior skills at tracking a high-speed moving ball (Bahill and LaRitz, 1984). Tracking multiple objects is even more demanding because the attention system needs to cope with numerous objects moving in different direction at the same time. I set out to test pilots on a demanding task which required them to track up to 6 moving balls.

The motion tracking task was a variant of one used by Intriligator, Nakayama, & Cavanagh (1991), which was based on a task devised by Pylyshyn & Strom (1988). A group of moving balls were presented on the computer screen. The subjects were asked to track a subset of the balls that were flashing, and to continue to do so after they stopped flashing (and thus became indistinguishable from the other balls on the screen). The balls continued to move on the computer screen after the flashing was stopped; after a delay one of the balls was designated as a probe ball, and the subjects had to judge whether or not it was one of the balls that had flashed earlier.

Method

Materials. The same materials that were used in the extrapolation task were used here, except that 4, 5, or 6 balls were flashed and the balls did not disappear after the flashing was concluded. The balls remained on the screen and the pilots had to track the balls that flashed earlier. To force actual tracking and not enable the pilots to rely on extrapolation, the trajectory of each ball was changed so the motion was random --every 60 ms the balls changed trajectory by a random degree shift of up to 30 degrees. To avoid confusion between colliding balls, the balls now bounced off each other when they collided.

Procedure. The same procedure used in the extrapolation task was used here.

Results

The data were analyzed as in Experiment 1; 0.7% of these data were considered outliers. The first analysis compared the control subjects to the pilots, and included the number of trajectories that were tracked (the number of balls) as a within-subjects variable, and profession (pilots vs. controls) as a between-subjects variable.

First I had to establish that our difficulty manipulation (i.e., the number of balls) did indeed affect performance, and that the subjects were indeed able to perform the task. The subjects performed the task at an accuracy rate of 77%; at the most difficult condition of tracking 6 balls they performed at an accuracy level of 73%, both well above chance level.

I found that the number of moving balls that were tracked affected both response time and error rate, F(2,112)= 9.39, p< .01 for response time (with means of 1175, 1208, and 1236 ms for 4, 5, and 6 balls, respectively), and F(2,112)= 17.57, p< .01, for error rates (with means of 18.8, 24.6, and 26.9 percent error for 4, 5, and balls, respectively). Indeed, linear contrasts revealed that response times increased linearly with the number of trajectories that were tracked, F(1,112)= 19.00, p< .01, as did error rates, F(1,112)= 36.53, p< .01. I then examined the statistical power of our main effect by calculating its effect size; the "r" was of medium magnitude, .38 in response time, and .50 in error rate.

After establishing that the manipulation of the number of trajectories that were tracked did tap onto tracking performance, I proceeded to determine whether piloting affected this ability. I found no interaction between the number of trajectories that were tracked and profession group (pilots vs. non-pilots), F(2,112)= 1.69, p= .19 for response time, and F< 1 for error rates. Pilots and the control non-pilots had comparable overall performance, F< 1 for both response time and error rate.

I next separated the pilots data into two distinct groups, according to training and experience. I found no interaction in response time, F(4,110)= 1.52, p= .20, or in error rate, F(4,110)= 1.97, p= .10. I found that the manipulation of the number of balls was significant, F(2,110)= 9.67 p< .01 for response time, and F(2,110)= 21.26, p< .01 for error rate. All three groups had comparable overall performance, F< 1 for both response time and error rate.

Finally, as I did with the motion extrapolation task, I compared the two subgroups of pilots adding the additional factor of "recency of flying." I found a trend for the significance of flying recency in error rates, F(1,32)= 3.52, p= .07

(with means of 22.6 vs. 27.0 percent error for flying recently vs. not flying recently, respectively), no such trend was found in response time, $F < 1$.

Discussion

The most interesting results pertains to the "recency of flying effect" that reduced errors for pilots who had flown recently. Flying probably primes some of the underlying subsystems involved in motion tracking. Although I found that response times and error rates increased linearly with our manipulation, I did not find any differences in tracking abilities between experienced pilots, novice pilots, and controls. There was a very weak trend for an interaction in the error rate data, but because I had a relative large effect size and a relative large sample, I do not think that the lack of an interaction was caused by insufficient statistical power. However, I must be careful in rejecting the null hypothesis.

Overall Analysis

To examine further the claim that the two tasks do indeed tap into different components I performed a correlation analysis between the two tasks. It would further establish that the components recruited for extrapolating and tracking motion are indeed distinct. Because I wanted to examine the specific components involved in motion extrapolation and motion tracking *per se*, I did not want to correlate overall response time which includes artifacts such as, encoding the stimuli and pressing the response keys. I therefore correlated the slopes of how response time and error rate increased as a function of difficulty, which reflect the specific ability of extrapolating and tracking multiple objects in motion. I calculated such a slope for each subject for each task, and then examined the correlation between the slopes of each task. If both tasks tap into two distinct processes then performance will not be correlated, however, if both tasks share many components then performance will be correlated. I found no correlation in response time, $r^2 = .02$, nor in error rate, $r^2 = .02$.

General Discussion

I set out to investigate processing components that are involved in spatial awareness. This research explores the abilities to extrapolate and track motion, which seem to be important skills involved in spatial awareness. These skills are especially critical in the domain of piloting. Spatial awareness is a complex ability and probably involves many additional components. I found that two important abilities involved in spatial awareness, extrapolating and tracking multiple objects in motion, are two distinct abilities that are not correlated. I also found that Air Force fighter training and experience seems to affect extrapolation abilities more than tracking abilities. In addition, I found that pilots who flew recently were prone to track motion better than pilots who did not fly recently. This effect was not observed in motion extrapolation.

In order to measure spatial awareness ability correctly, one needs to first understand which component subsystems are involved in this process, and how they interact. Then, after having such a model, one needs to explore which factors affect each component and use these factors as a tool for quantifying the efficiency of the component subsystem. In this study I take a first step in this direction.

Footnotes

This research was supported by RIP research grant 92-76 awarded to the author by the Air Force Office of Scientific Research through contact F49620-90-C-0076 to Research and Development Laboratories. I would like to thank Stephen M. Kosslyn and Wayne L. Waag for their comments and suggests, as well as their support. I owe much thanks to James Intriligator for valuable advice regarding the designing of the tasks and comments on earlier versions of this manuscript, and to Raynald Comtois for the programming. Also I would like to thank William B. Raspotnik, L. Jet Jackson, and the personnel on Luke Air Force base for their help and time. Correspondence should be send to Itiel E. Dror, Department of Psychology, Harvard University, 33 Kirkland St., Cambridge, MA 02138.

References

Bahill, A.T. & LaRitz, T. (1984). Why can't batters keep their eyes on the balls?. *American Scientist*, 72, 249-253.

Dror, I.E. (1991). Visual cognitive profile for pilots. AFOSR, Research and Development Laboratories, TR-91-362.

Dror, I.E. (1992). Visual mental rotation: Different processes used by pilots. *Proceedings of the Human Factor Society 36th Annual Meeting. Vol. 2*, 1368-1372.

Dror, I.E., Kosslyn, S.M., and Waag, W. (1992). Visual-spatial abilities of pilots. Harvard University manuscript.

Hayes, J.R. (1981). *The Complete Problem Solver.* Hillsdale, NJ: Erlbaum.

Intriligator. J., Nakayama, K., & Cavanagh, P. (1991). Attentive tracking of multiple moving objects at different scales. *ARVO Annual Meeting Abstract Issue*, 1040.

Pylyshyn, Z.W. & Strom, R.W. (1988). Tracking multiple independent targets: Evidence for a parallel tracking mechanism. *Spatial Vision*, 3, 179-197.

Sternberg, S. (1969). The discovery of processing stages: Extensions of Doners' method. *Acta Psychologica*, 30, 276-315.

Yantis, S. (1992). Multielement visual tracking: Attention and perceptual organization. *Cognitive Psychology*, 24, 295-340.

THE ANALYSIS OF TWO DIMENSIONAL DISPERSIVE STRUCTURES USING THE FINITE-DIFFERENCE TIME-DOMAIN (FDTD) METHOD

Fred J. German
Post-Doctoral Fellow
Department of Electrical Engineering

Auburn University
200 Broun Hall
Auburn University, AL 36849

Final Report For:
Research Initiation Program
Armstrong Laboratory

Sponsored By:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, D.C.

December 1992

THE ANALYSIS OF TWO DIMENSIONAL DISPERSIVE STRUCTURES USING THE FINITE-DIFFERENCE TIME-DOMAIN (FDTD) METHOD

Fred J. German
Post-Doctoral Fellow
Department of Electrical Engineering
Auburn University

Abstract

The finite-difference time-domain method is formulated for the solution of electromagnetic scattering and propagation problems involving dispersive material bodies in two dimensions. A computer program was written and several simple cylindrical bodies were analyzed. Comparisons with analytic results have yielded excellent agreement. A user's guide for the programs and the source code is included.

THE ANALYSIS OF TWO DIMENSIONAL DISPERSIVE STRUCTURES USING THE FINITE-DIFFERENCE TIME-DOMAIN (FDTD) METHOD

Fred J. German

INTRODUCTION

The finite-difference time-domain (FDTD) technique is a well established numerical technique for obtaining solutions to a wide range of complex electromagnetic interaction problems [1]. The method is based on spatial and temporal discretizations of the Maxwell curl equations in differential form. Since the parameters defining a specific problem (material properties, conductor locations, sources, etc.) can be arbitrarily specified at any of the discrete spatial points and at any discrete time step, the method is very general and capable of accommodating many different types of structures with arbitrary geometric details and inhomogeneities.

This report is an investigation into the use of the FDTD method to model the penetration of electromagnetic radiation into two dimensional objects which are composed of inhomogeneous and dispersive materials. The problem of modeling electromagnetic penetration into this class of objects is of fundamental importance when investigating the effects of electromagnetic radiation on biological tissues. In dispersive media it is possible for field precursors to develop [2] which may lead to transient field magnitudes in excess of those predicted by time harmonic analysis. When determining the fields in biological bodies it is crucial to be able to accurately account for these transients since their magnitudes often determine the maximum field present in the body. Thus, a time domain analysis technique that is capable of giving accurate transient field values is required.

In the present work we restrict ourselves to the analysis of two dimensional objects. The FDTD method is however a fully three dimensional technique and the extension of the principles presented herein for two dimensions to three dimensions should be straightforward.

FORMULATION

Difference Equations

We wish to find solutions to the Maxwell curl equations subject to the boundary conditions imposed by the specific problem at hand. These equations are given by

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{1a}$$

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \sigma \mathbf{E} \tag{1b}$$

where the constituitive relations

$$\mathbf{B} = \mu(\omega)\mathbf{H} \tag{2a}$$

$$\mathbf{D} = \varepsilon(\omega)\mathbf{E} \tag{2b}$$

determine the relationship between the flux and field quantities. For the purposes of the present analysis we assume that $\mu(\omega) = \mu_o$, the permeability of free space ($4\pi \times 10^{-7}$ H/m) and that the conductivity, σ, does not vary with frequency. Thus, the equations of interest become

$$\nabla \times \mathbf{E} = -\mu_o \frac{\partial \mathbf{H}}{\partial t} \tag{3a}$$

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \sigma \mathbf{E} \tag{3b}$$

$$\mathbf{D} = \varepsilon(\omega)\mathbf{E} \tag{3c}$$

We wish to employ a discrete approximation for the solution of (3) in both space and time. In the present work we restrict ourselves to the two dimensional field problem of $Hz = 0$, $Ex = Ey = 0$ and no variation of fields in the z coordinate direction ($\partial/\partial z = 0$). For this case (TM to z) equations (3) become

$$\frac{\partial Dz}{\partial t} = \frac{\partial Hy}{\partial x} - \frac{\partial Hx}{\partial y} - \sigma Ez \tag{4a}$$

$$-\frac{\partial Ez}{\partial y} = \mu_o \frac{\partial Hx}{\partial t} \tag{4b}$$

$$\frac{\partial Ez}{\partial x} = \mu_o \frac{\partial Hy}{\partial t} \tag{4c}$$

$$Dz = \varepsilon(\omega)Ez \tag{4d}$$

Applying a central finite difference approximation in cartesian coordinates to (4a,b, and c) we obtain the following difference equations:

$$H_x^{n+1/2}(i,j+1/2) = H_x^{n-1/2}(i,j+1/2) - \frac{\Delta t}{\mu_o \Delta \ell}\left[E_z^n(i,j+1) - E_z^n(i,j)\right] \tag{5a}$$

$$H_y^{n+1/2}(i+1/2,j) = H_x^{n-1/2}(i+1/2,j) + \frac{\Delta t}{\mu_o \Delta \ell}\left[E_z^n(i+1,j) - E_z^n(i,j)\right] \tag{5b}$$

$$D_z^{n+1}(i,j) = D_z^n(i,j) + \frac{\Delta t}{\Delta \ell}\left[H_y^{n+1/2}(i+1/2,j) - H_y^{n+1/2}(i-1/2,j) - H_x^{n+1/2}(i,j+1/2) + H_x^{n+1/2}(i,j-1/2)\right] \tag{5c}$$

where i and j are the discrete spatial indices in the x and y coordinate direction, respectively, and n is the discrete time index. The spatial discretization is demonstrated in Figure 1 where the location of the different electric and magnetic field components in the cartesian lattice is shown. In these equations Δt is the discrete time step and $\Delta \ell$ is the discrete spatial separation between adjacent electric (or magnetic) field components.

The FDTD method requires that the electric field be specified at time $n\Delta t = 0$ at all electric field points in the lattice. Once this has been done equations (5a) and (5b) can be used to calculate the values of H_x and H_y at time $(n+1/2)\Delta t$ at all spatial points corresponding to these field quantities. Then, (5c) is used to calculate the value of the electric flux density, D_z at time $(n+1)\Delta t$ at all spatial points corresponding to this field component. After this operation is performed, we must be able to calculate the electric field E_z from the electric flux density D_z so that the algorithm can cycle back to (5a) and (5b) to repeat the procedure and thus advance the time.

The relation between E_z and D_z is given by (4d) in the frequency domain. There have been two methods proposed to deal with this relation in the time domain. The first has been formulated by Leubbers et. al. [3]. Since a multiplication in the frequency domain is equivalent to a convolution in the time domain, the method presented in [3] casts (4d) as a convolution integral which is performed numerically and updated at each time step as the algorithm progresses. The second technique proposed by Joseph et.

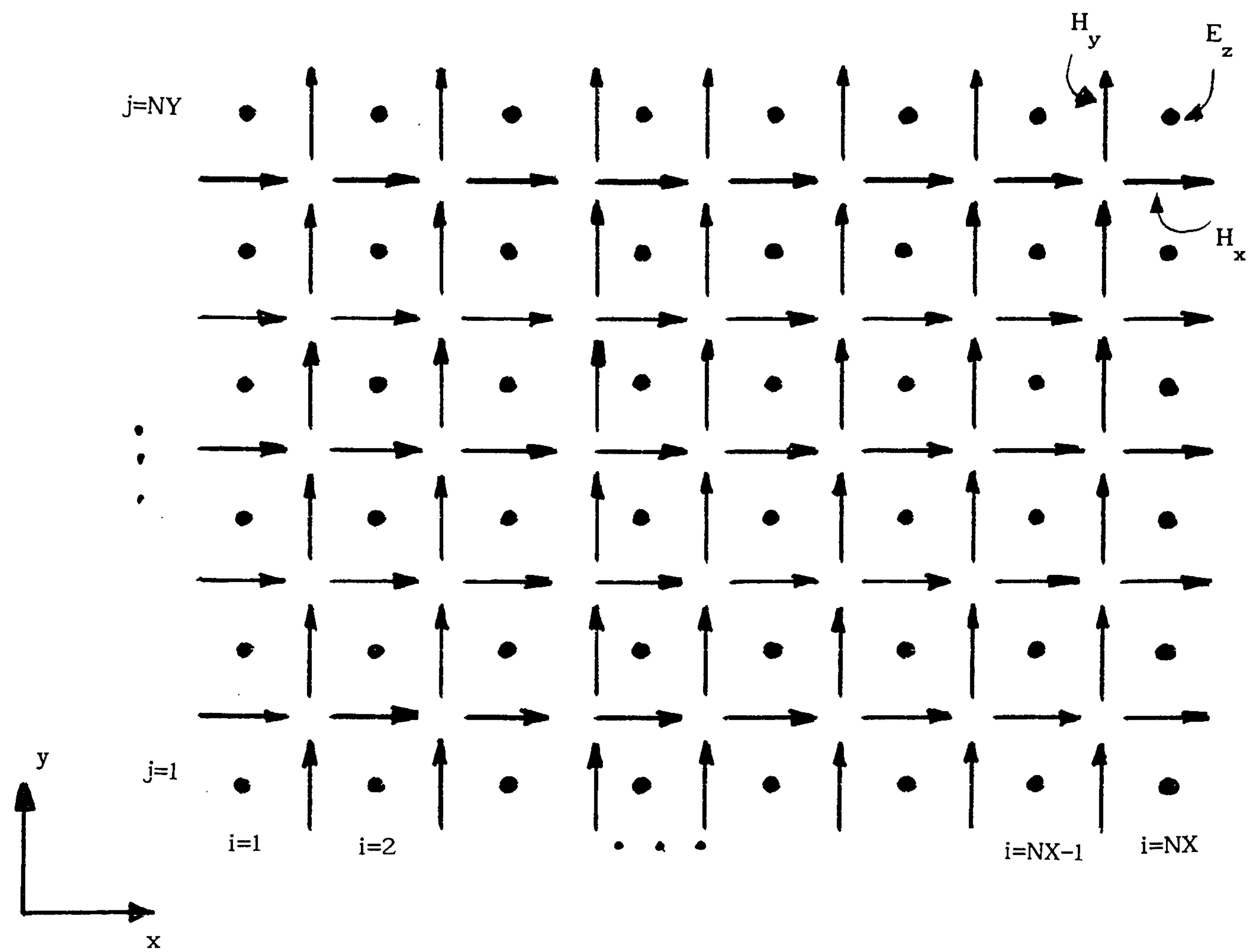


Figure 1. Discretization scheme of the two dimensional FDTD lattice showing the indexing (i,j) of the electric field components.

al. [4] uses an inverse Fourier transform to transform (4d) into a differential equation in the time domain which is then solved using central differencing. Applications of both these techniques to one dimensional problems has indicated that the differential equation method is more accurate than the convolution integral method [5] and is the method used in the present work.

Consider first the case of a Debye dispersive media where the permittivity is given by

$$\varepsilon(\omega) = \varepsilon_\infty + \frac{\varepsilon_s - \varepsilon_\infty}{1 + j\omega t_o} = \frac{Dz}{Ez} \qquad (6)$$

where

ε_s = DC permittivity

ε_∞ = infinite frequency permittivity

t_o = relaxation time of media

ω = radian frequency

Cross multiplying and taking the inverse Fourier transform of (6) yields the differential equation

$$D_z + t_o \frac{dDz}{dt} = \varepsilon_s E_z + t_o \varepsilon_\infty \frac{dEz}{dt} \qquad (7)$$

which, upon the application of central differencing becomes

$$E_z^{n+1}(i,j) = \frac{\Delta t + 2\tau}{2\tau\varepsilon_\infty + \varepsilon_s \Delta t} D_z^{n+1}(i,j) + \frac{\Delta t - 2\tau}{2\tau\varepsilon_\infty + \varepsilon_s \Delta t} D_z^{n}(i,j) + \frac{2\tau\varepsilon_\infty - \varepsilon_s \Delta t}{2\tau\varepsilon_\infty + \varepsilon_s \Delta t} E_z^{n}(i,j) \qquad (8)$$

which is the desired equation that relates the electric field (E_z) to the electric flux density at the current and previous time steps and the electric field at the previous time step - all known quantities.

While equation (8) is valid only for materials exhibiting Debye dispersion, the technique can be applied to any material provided the form of $\varepsilon(\omega)$ is such that equation (4d) can be inverse transformed to yield a differential equation relating D to E. For example, consider a Lorentz dispersive material which is defined by

$$\varepsilon(\omega) = \varepsilon_{\infty} - \frac{\omega_0^2 (\varepsilon_s - \varepsilon_{\infty})}{\omega^2 + 2j\omega\delta - \omega_o^2} \tag{9}$$

where

ω_o = resonant frequency of media

δ = damping coefficient

Using the inverse Fourier transform as before and applying central differencing to the resulting second order differential equation gives us

$$E_z^{n+1}(i,j) = \frac{1}{(\varepsilon_s \omega_o^2 \Delta t + 4\delta\varepsilon_{\infty}\Delta t + 2\varepsilon_{\infty})} \Big[(\omega_o^2 \Delta t^2 + 4\delta\Delta t + 2) D_z^{n+1}(i,j)$$

$$+ (\omega_o^2 \Delta t^2 - 4\delta\Delta t - 4) D_z^n(i,j) + 2 D_z^{n-1}(i,j)$$

$$- (\omega_o^2 \Delta t^2 \varepsilon_s - 4\delta\Delta t\varepsilon_{\infty} - 4\varepsilon_{\infty}) E_z^n(i,j) - 2\varepsilon_{\infty} E_z^{n-1}(i,j) \Big] \tag{10}$$

which allows us to advance E_z in terms of known present and previous field and flux quantities.

In the FDTD algorithm, after the electric flux density is updated via equation (5c), the new value for the electric field can be calculated from (8) for Debye media or (10) for Lorentz media. Extension to other types of dispersions, including those with multiple poles, should be easy to accomplish.

The enforcement of the boundary conditions which define the problem under consideration is accomplished automatically by simply specifying the material parameters at each electric field point in the lattice. In the case of perfect conductors where the conductivity, σ, is infinite, the conductivity can be set to a very large value or, alternatively, the electric field component at the affected node can be set to zero at each time step since the electric field tangential to a perfect conductor must be zero.

Lattice Termination

We now turn out attention to the electric field points at the edge of the FDTD lattice. Since we must use a finite FDTD lattice, the lattice must be truncated somehow to realistically simulate an infinite space. Examination of equation (5c) shows that for points at the extreme edges of the lattice, magnetic field values that fall outside the lattice are required to update the electric field along the lattice edges. Since these magnetic field values are unavailable, equation (5c) cannot be used to update the electric field values at the lattice edges and so we must employ an alternative update equation for these electric field values. This alternative update equation for the electric flux density at the lattice edges is usually referred to as an absorbing boundary condition (ABC) since its function is to absorb outward traveling waves in order to simulate propagation to infinity on the finite FDTD lattice. The number of ABC's that have been investigated for use with the FDTD method is far to extensive to warrant a detailed explanation here and the interested reader is referred to [6] for an excellent discussion of many of these ABC's.

In the work presented in this report a simple first order Mur ABC is employed [7]. Figure 1 shows a complete two dimensional

FDTD lattice with the field locations noted by the symbols shown in the legend. Consider the electric field points along the i = 1 lattice edge. The calculation of these electric field values can be achieved by the application of the following first order Mur ABC

$$E_z^n(1,j) = E_z^{n-1}(2,j) + \frac{1-\rho}{1+\rho}\left(E_z^{n-1}(1,j) - E_z^n(2,j) \right) \tag{11}$$

where

$$\rho = \frac{c\Delta t}{\Delta \ell} \tag{12}$$

and c is the speed of light (3×10^8 m/s).

Similar equations are easily derived for the remaining three lattice edges.

Scattered/Total Field Regions

The above ABC's will only work for waves that are traveling outward on the mesh. Thus, if we wish to simulate plane wave excitation originating at infinity within the FDTD lattice a technique is required to separate the outward traveling, or scattered, field component from the inward traveling, or incident, field component so that the ABC is applied only to the scattered fields. This is easily accomplished by dividing the FDTD lattice into a total field region and a scattered field region as shown in Figure 2 By doing this, only the outward traveling scattered wave is present in the regions that require application of the ABC, while in the total field region (which contains the structure under analysis) the total field (incident plus scattered) is known. At the boundary separating the total and scattered field regions slightly modified time stepping relations are required.

Lets require that the boundary separating the regions passes

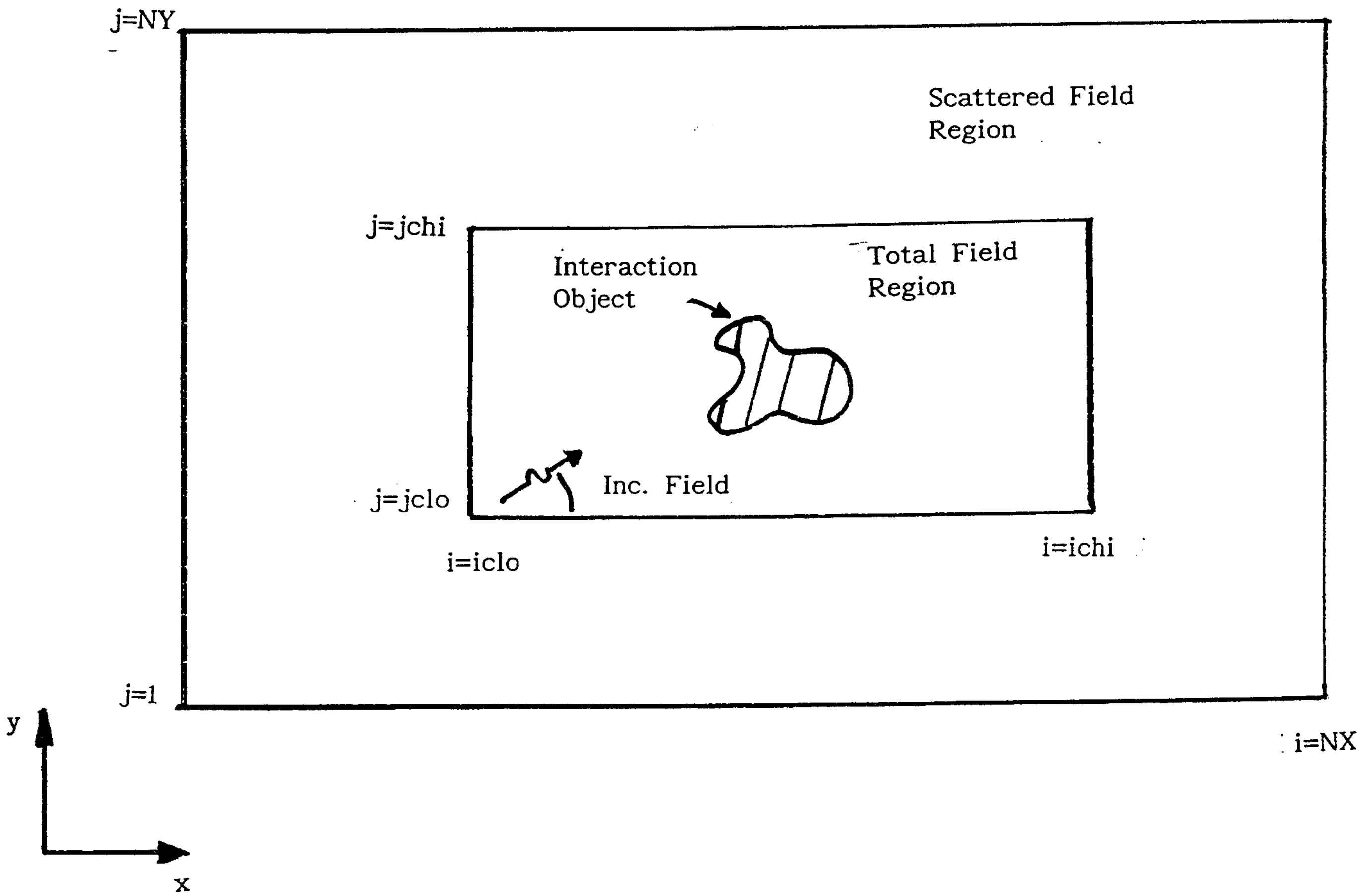


Figure 2. Separation of FDTD Lattice into total and scattered field regions.

through electric field points in the lattice. We also assume that the incident field is a known function of space and time. Consider the section of lattice shown in Figure 3 which contains the boundary between the two field regions. If we take the electric field values on the boundary to be total field values, then we must account for the fact that the H_x field component below the boundary is a scattered field. Thus, we should add the incident H_x, denoted $H_{x,inc}$ to it so that we have a scattered field. This is most easily accomplished by using the following equation to determine the electric field on the boundary:

$$E_z^{n+1}(i,j) = \hat{E}_z^{n+1}(i,j) + \frac{\Delta t}{\varepsilon_o \Delta \ell} H_{x,inc}^{n+1/2}(i,j-1/2) \quad (13)$$

where $\hat{E}$ is the electric field calculated with the usual FDTD update equations without accounting for the scattered field region.

In a similar fashion, the H_x field component in the scattered field region and adjacent to the boundary requires that the incident electric field be added to the updated H_x as follows:

$$H_x^{n+1/2}(i,j-1/2) = \hat{H}_x^{n+1/2}(i,j-1/2) + \frac{\Delta t}{\mu_o \Delta \ell} E_{z,inc}^{n}(i,j) \quad (14)$$

Similar equations are easily derived for the other three edges of the scattered/total field boundary.

Not only does the division of the mesh allow the ABC's to operate only on the scattered fields, but it also provides an efficient and easy way of specifying the initial excitation in the lattice. Since the incident field is a known function of space and time, the addition of the incident field in the update equations used on the scattered/total field boundary (equations 13 and 14) automatically excites the desired incident field

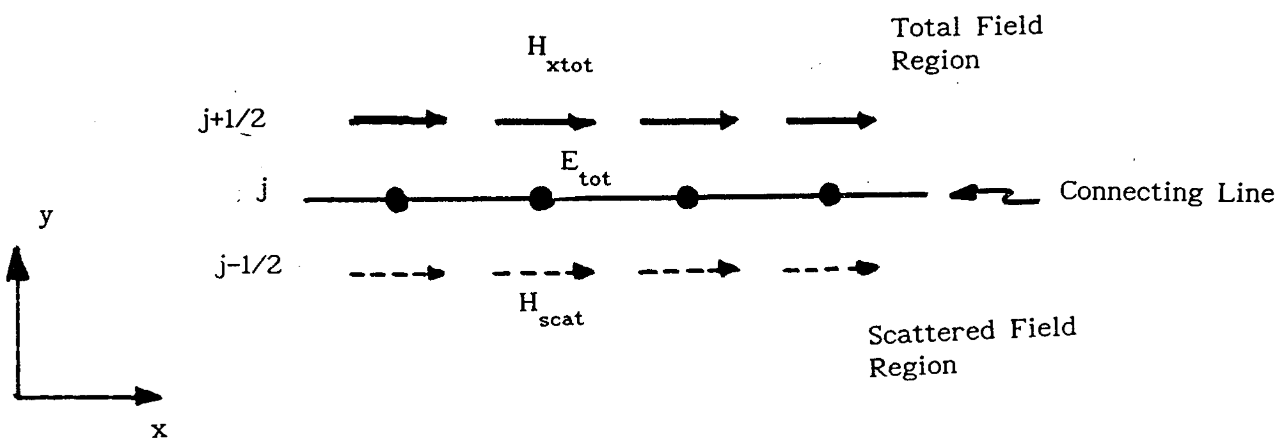


Figure 3. Details of the total/scattered field boundary separation.

excitation.

Numerical Accuracy

Since the FDTD equations are a discrete approximation to the continuous differential equations that govern electromagnetic wave propagation there is a certain amount of error associate with them. There are two main types of error associated with the FDTD solution of electromagnetic field problems; coarseness error and dispersion error.

Coarseness error is caused by the discrete spatial sampling used in the FDTD mesh. Since boundary conditions can be applied only at discrete points within the lattice, certain classes of objects (those with curved surfaces for example) can only be approximated in the mesh. This error can usually be reduced to acceptable values by the use of a sufficiently fine lattice. Of course the finer the lattice the more extensive the computer resources that will be required to solve the problem. Over the years, however, the regular cartesian FDTD lattice as applied in this work has been successfully applied to a wide range of interaction objects, including curved and irregularly shaped ones, with a high degree of accuracy and reasonable computer resources.

The second, and usually more critical, type of error is dispersion. The numerical propagation constant of waves in the FDTD lattice is not the same as the actual physical propagation constant of the waves. As the lattice spacing, $\Delta\ell$, becomes infinitesimally small, however, the numerical and the physical propagation constants become the same. Thus, a sufficiently small lattice is required so that the error caused by dispersion is within an acceptable limit. Detailed analyses of the dispersion properties of the FDTD lattice have been performed [6] and the error is well quantified. In the traditional (i.e., non-dispersive) FDTD formulation, the usual rule-of-thumb is to keep the lattice size less than one-tenth of the smallest wavelength in the lattice for the dispersion error to be negligible, that is

$$\frac{\Delta \ell}{\lambda_{min}} < 0.1 \tag{15}$$

A complete dispersion analysis has not been performed for the two dispersive FDTD formulations. In order to assess the accuracy of the results in this report, and assure that the dispersion error is negligible, all results that are to be presented were analyzed on increasingly fine lattices until convergence was observed. Since the dispersion error decreases with decreasing lattice spacing, these convergence studies assure that the results presented herein contain negligible error due to numerical dispersion.

A computer program package has been developed based on the formulation discussed above. The program is quite versatile and capable of analyzing a wide variety of interaction objects which are composed of: (1) perfect conductors, (2) non-dispersive lossy dielectrics, (3) Debye dispersive media, and/or (4) Lorentz dispersive media. Appendix 1 contains a user's guide for these programs and Appendix 2 contains a source listing of all the programs in the package. Though not necessary for operation of these programs, a basic familiarity with the FDTD method will certainly simplify their use.

In the next section numerical results for field penetration into dispersive cylinder obtained using the dispersive FDTD formulation is presented.

NUMERICAL RESULTS

In this section we present results for a homogeneous circular cylinders and for a multilayer circular cylinder. While the programs have been used to analyze a variety of objects, these benchmark cases have been selected for presentation here.

Figure 4 shows the first problem and Figure 5 the second. The first consists of a circular dielectric cylinder illuminated by a TM polarized plane wave. The radius of the cylinder, a, was taken to be 7 mm.

An exact series solution is known for this particular electromagnetic interaction problem [8]. The analytic solution for the electric field internal to the cylinder for sinusoidal time dependence of the incident field is given by

$$E_z = E_o \sum_{n=-\infty}^{\infty} j^n c_n J_n(k_d \rho) e^{jn\phi} \tag{16a}$$

where E_o is the magnitude of the incident electric field, ρ and ϕ ore the usual cylindrical coordinates with the origin taken to be at the center of the cylinder, $j = \sqrt{1}$, $k_d = \omega \sqrt{\mu_o \varepsilon_d}$ where ε_d is the complex permittivity of the cylinder, J_n is a Bessel function of the first kind of order n, and c_n is given by

$$c_n = \frac{1}{J_n(k_d a)} \left[J_n(ka) + a_n H_n^{(2)}(ka) \right] \tag{16b}$$

In this equation, $H_n^{(2)}$ is a n^{th} order Hankel function of the second kind and $k = \omega \sqrt{\mu_o \varepsilon_o}$. The coefficient a_n is given by the following:

FIG. d

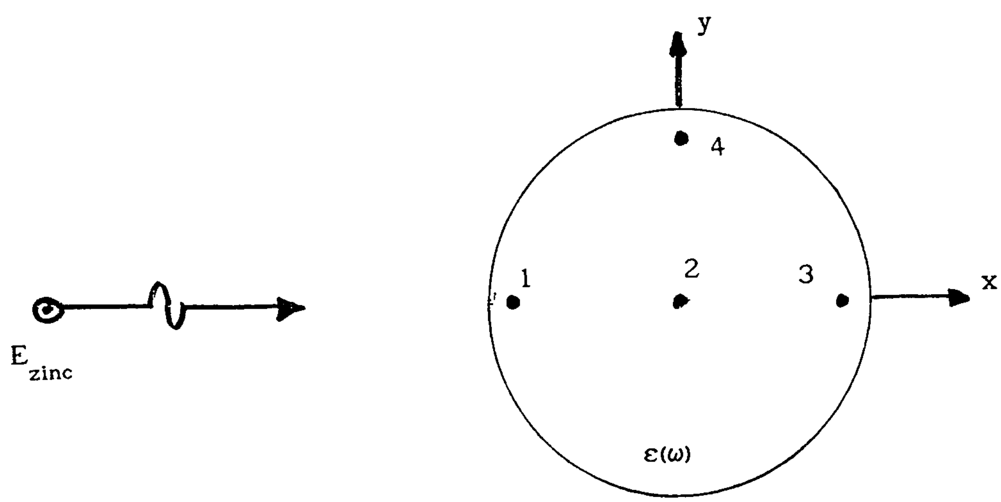


Figure 4. Geometry for the cylinder problem showing field observation point locations.

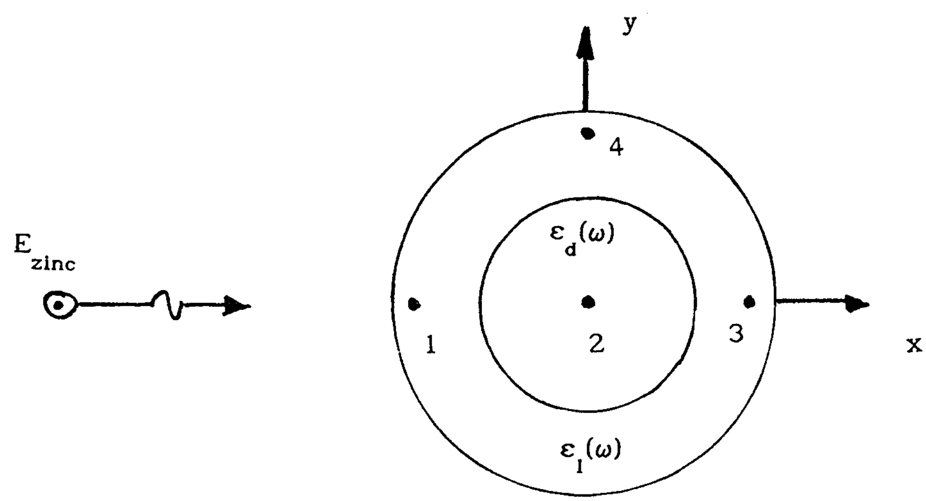


Figure 5. Geometry for the concentric cylinders showing field observation point locations.

$$a_n = \frac{-J_n(ka)}{H_n^{(2)}(ka)} \left[\frac{\varepsilon_d J_n'(k_d a)/\varepsilon_o K_d a J_n(k_d a) - J_n'(ka)/ka J_n(ka)}{\varepsilon_d J_n'(k_d a)/\varepsilon k_d a J_n(k_d a) - H_n^{(2)'}(ka)/ka H_n^{(2)}(ka)} \right] \quad (16c)$$

A computer program was written based on equations (16) so that comparison could be made with the numerical FDTD results. The series was summed to provide negligible truncation error (error < 1×10^{-5}).

For comparison to this time harmonic solution, the FDTD program was initialized using a Gaussian pulse plane wave. The algorithm was then stepped in time and the resulting magnitude of the Fourier transformed electric field observed. This steady state sinusoidal magnitude can then be compared to the analytic solution obtained from equations (16) by normalizing the spectrum of the incident Gaussian pulse.

The first example we examine is the case of a plane wave incident on a circular cylinder that is composed of a debye dielectric. The geometry for this problem was shown in Figure 4. The radius of the cylinder, a, was taken to be 7 mm. The FDTD mesh spacing was chosen as 350 μm which gives 40 FDTD cells across the diameter of the cylinder. The debye material parameters were $\varepsilon_s = 78.2$, $\varepsilon_i = 5.5$ and $t_o = 8.1$ ps. The electric field magnitude was observed at each lattice point along the x and y axes within the cylinder. The electric field magnitudes are plotted in Figure 6-9 for frequencies of 3, 5, 7.5 and 10 GHZ, respectively. (Note that only half of the y-axis data is shown due to symmetry.)

As can be seen in these figures, there is good agreement between the FDTD results and the analytically derived series solution. It is important to realize that for the series solution has to be calculated at each frequency point separately with the complex

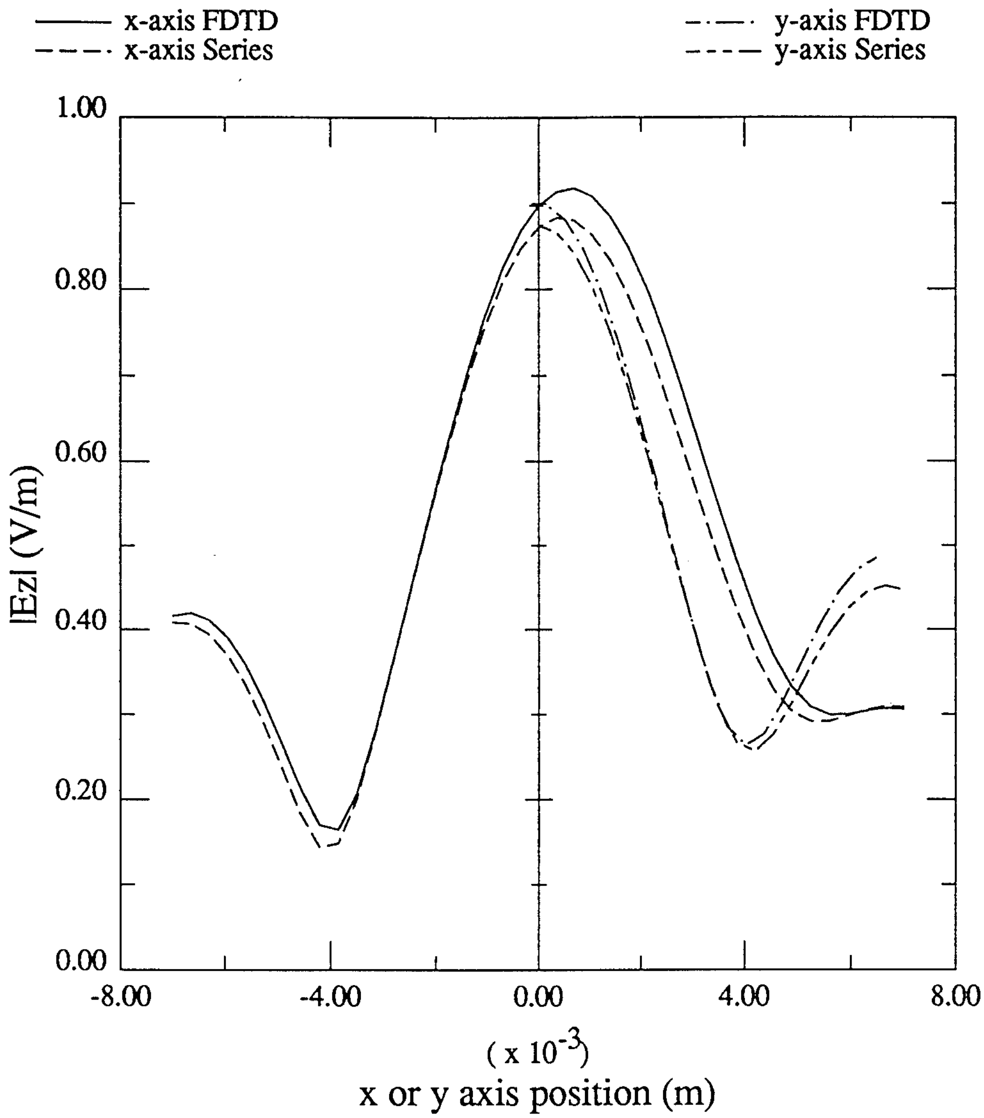


Figure 6. Magnitude of E_z vs. position for 7 mm radius Debye cylinder at 3-GHZ - Dispersive FDTD and analytic series solutions.

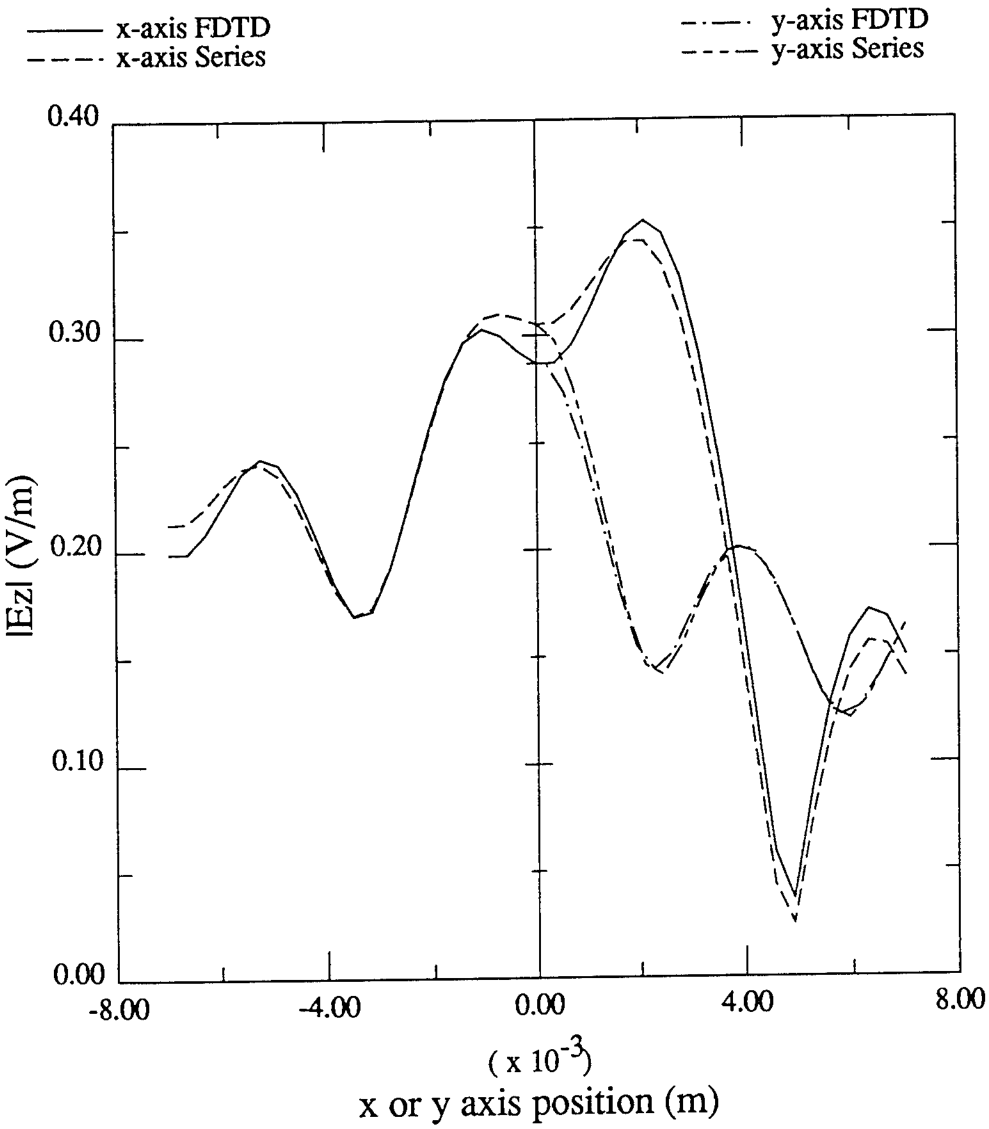


Figure 7. Magnitude of E_z vs. position for 7 mm radius Debye cylinder at 5-GHZ - Dispersive FDTD and analytic series solutions.

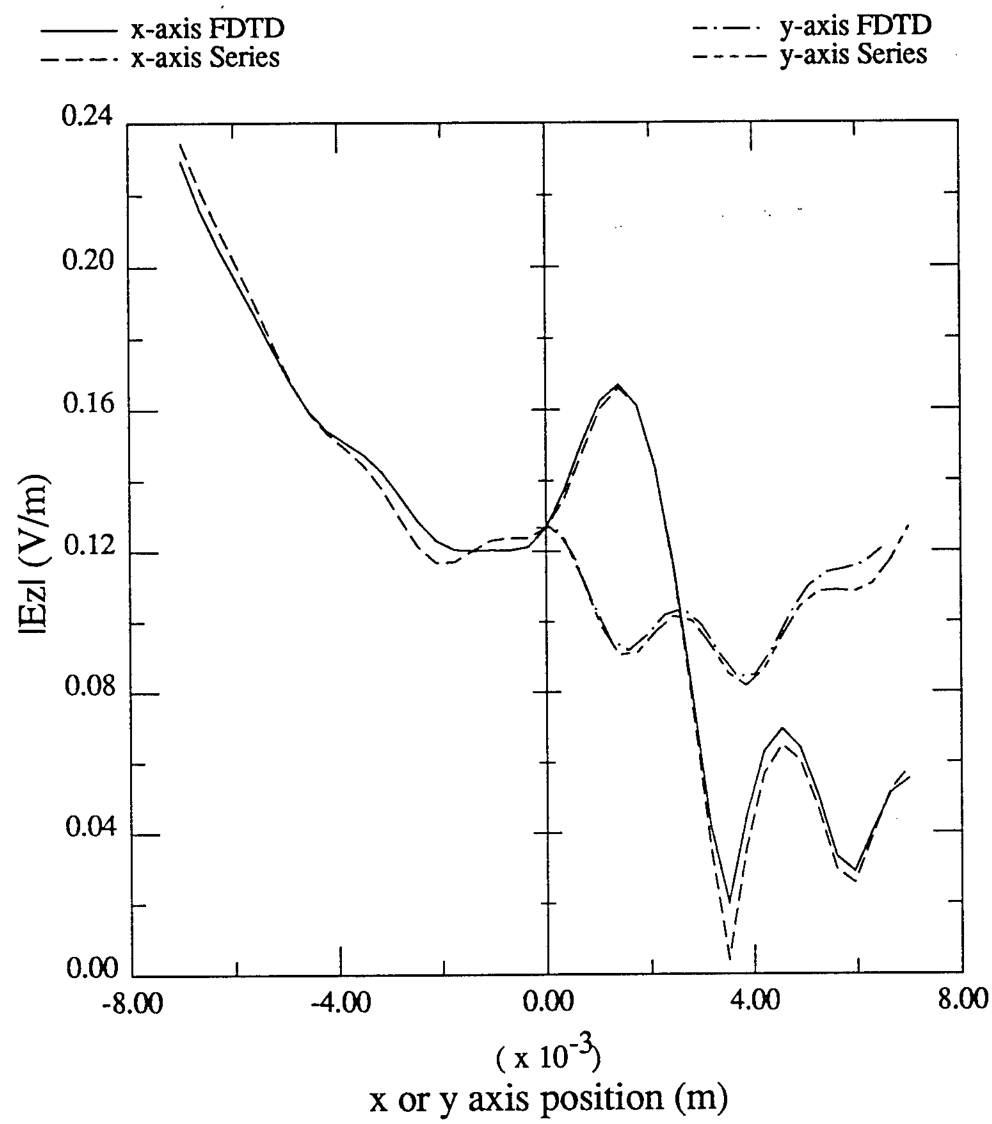


Figure 8. Magnitude of E_z vs. position for 7 mm radius Debye cylinder at 7.5-GHZ - Dispersive FDTD and analytic series solutions.

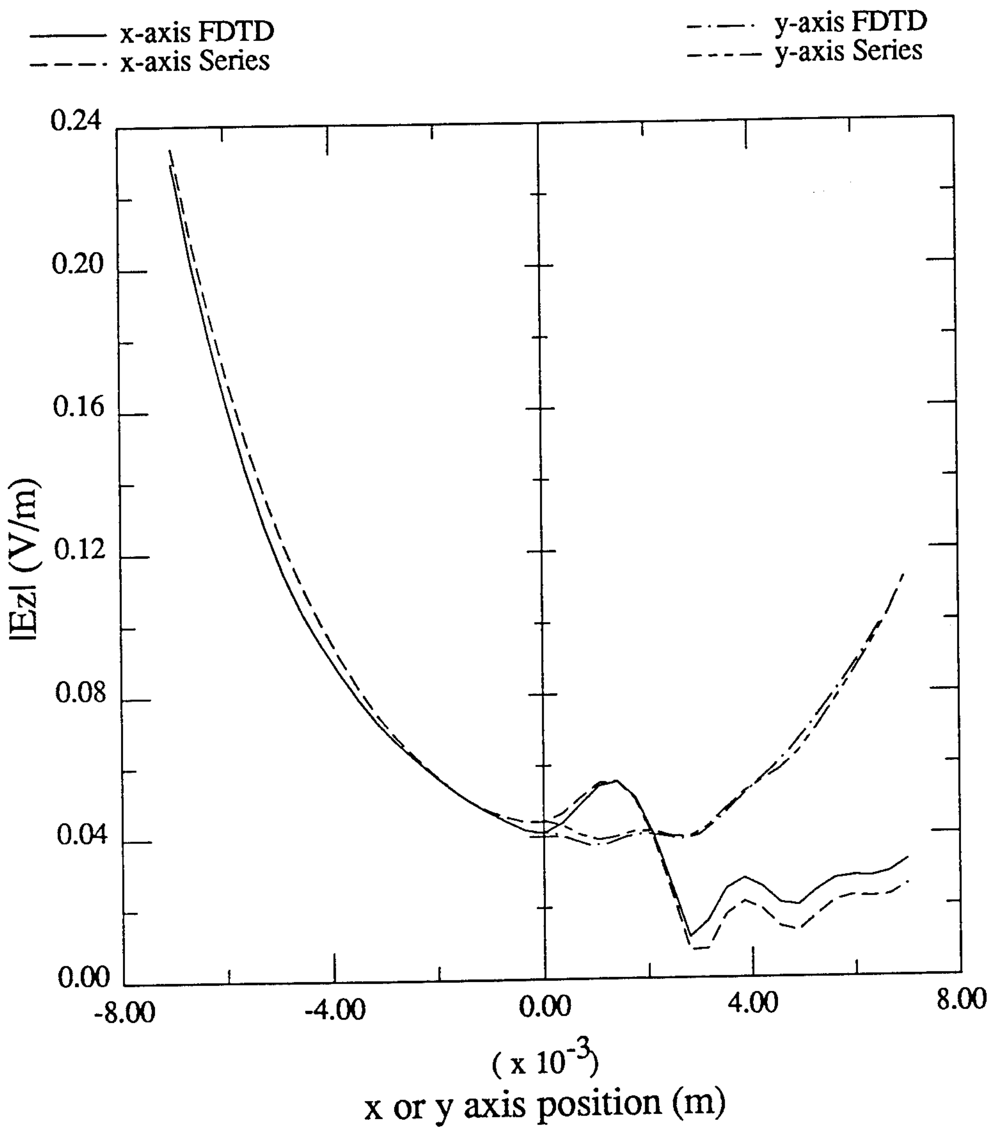


Figure 9. Magnitude of E_z vs. position for 7 mm radius Debye cylinder at 10-GHZ - Dispersive FDTD and analytic series solutions.

permittivity associated with that single frequency. On the other hand the FDTD solution, by using a wideband Gaussian excitation and the dispersive formulation presented in this report, calculates the field at all spatial points and at all frequencies simultaneously, thus making the method very efficient.

Having established the validity of the dispersive FDTD formulation in two dimensions and the proper working of the computer codes, we will examine how well the dispersive formulation compares with the traditional FDTD method. NOTE: As in the series solution, when using the traditional FDTD method for computations involving dispersive materials, each frequency point must be calculated separately using the material parameters appropriate at that frequency. Thus, if data is desired at N frequencies, the traditional FDTD method takes approximately N times as much computer time than the dispersive FDTD formulation since the dispersive formulation requires very little additional computation overhead. Thus, assuming the two techniques are equally accurate, the dispersive FDTD method is the choice for wideband or, as we shall see, transient computations.

Returning again to our debye cylinder, we now examine the electric field as a function of frequency at the four observation points indicated in Figure 4. The exact location of the points is as follows:

Point 1 - $(-18,0)\Delta\ell$

Point 2 - $(0,0)\Delta\ell$

Point 3 - $(18,0)\Delta\ell$

Point 4 - $(0,18)\Delta\ell$

In Figures 10 through 13 we plot the electric field at point 1, 2, 3 and 4, respectively. In all these figures the continuous curve was obtained from the dispersive FDTD formulation in a single computer run by utilizing a Gaussian pulse as the incident wave. For comparison, the field was computed for discrete frequency points using the traditional FDTD method (recalculating at each frequency). The agreement is excellent indicating that the dispersive FDTD method does not introduce any instabilities or accuracy degradation into the usual FDTD treatment of Maxwell's equations.

When using discrete numerical techniques such as FDTD it is important that a sufficiently fine grid resolution is used. In order to check the accuracy of the 350 μm lattice size used in these examples, the calculations were repeated for the same size cylinder on a FDTD mesh using a fine lattice spacing of 233.333 μm. These results are presented in Figures 14 through 17 for point 1, 2, 3 and 4 inside the debye cylinder. Since there is very little discrepancy between the results obtained on the course and fine meshes, it can be concluded that the 350 μm lattice spacing is sufficiently small to yield accurate results for this problem.

Next, in order to test the capability of the code to handle Lorentz media, the same cylinder geometry was run but for this case the cylinder was composed of a Lorentz dispersive material with parameters $\varepsilon_s = 78.2$, $\varepsilon_i = 5.5$, $\omega_o = \delta = 300\text{x}10^9$. The choice of these seemingly odd lorentz parameters requires some explanation.

Since in order to obtain results from the traditional FDTD (or

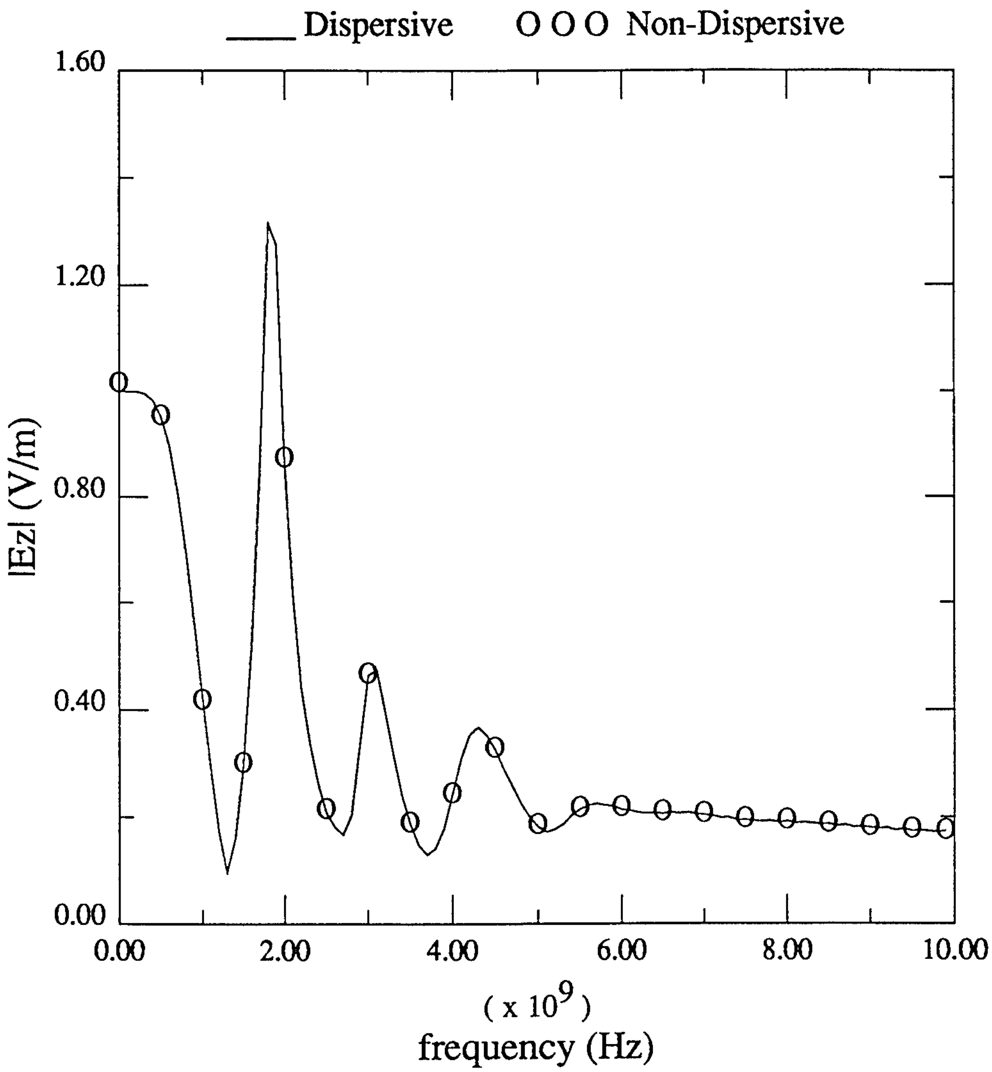


Figure 10. Magnitude of E_z vs. frequency at point 1 for Debye cylinder. ——— - Dispersive FDTD Formulation, 0 0 0 - Non-Dispersive FDTD formulation.

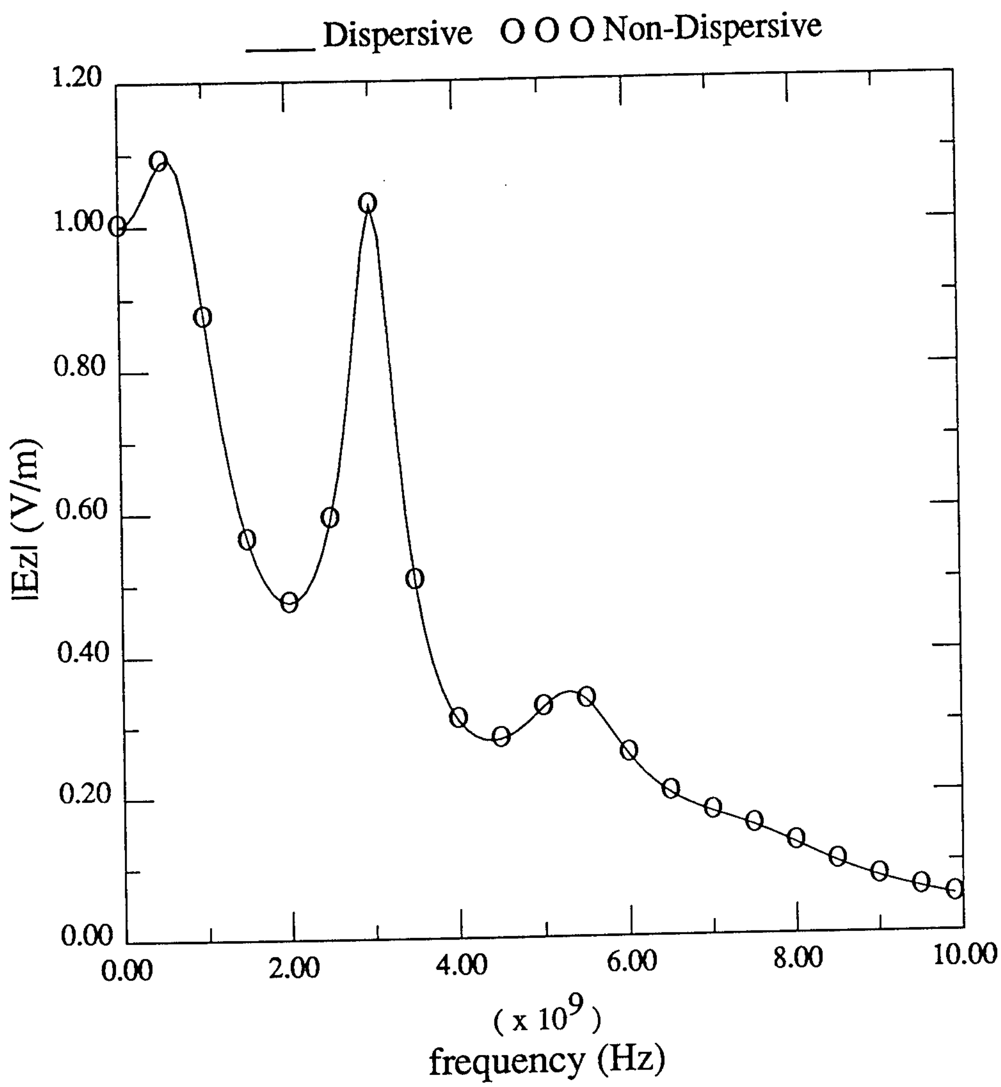


Figure 11. Magnitude of E_z vs. frequency at point 2 for Debye cylinder. ——— - Dispersive FDTD Formulation, 0 0 0 - Non-Dispersive FDTD formulation.

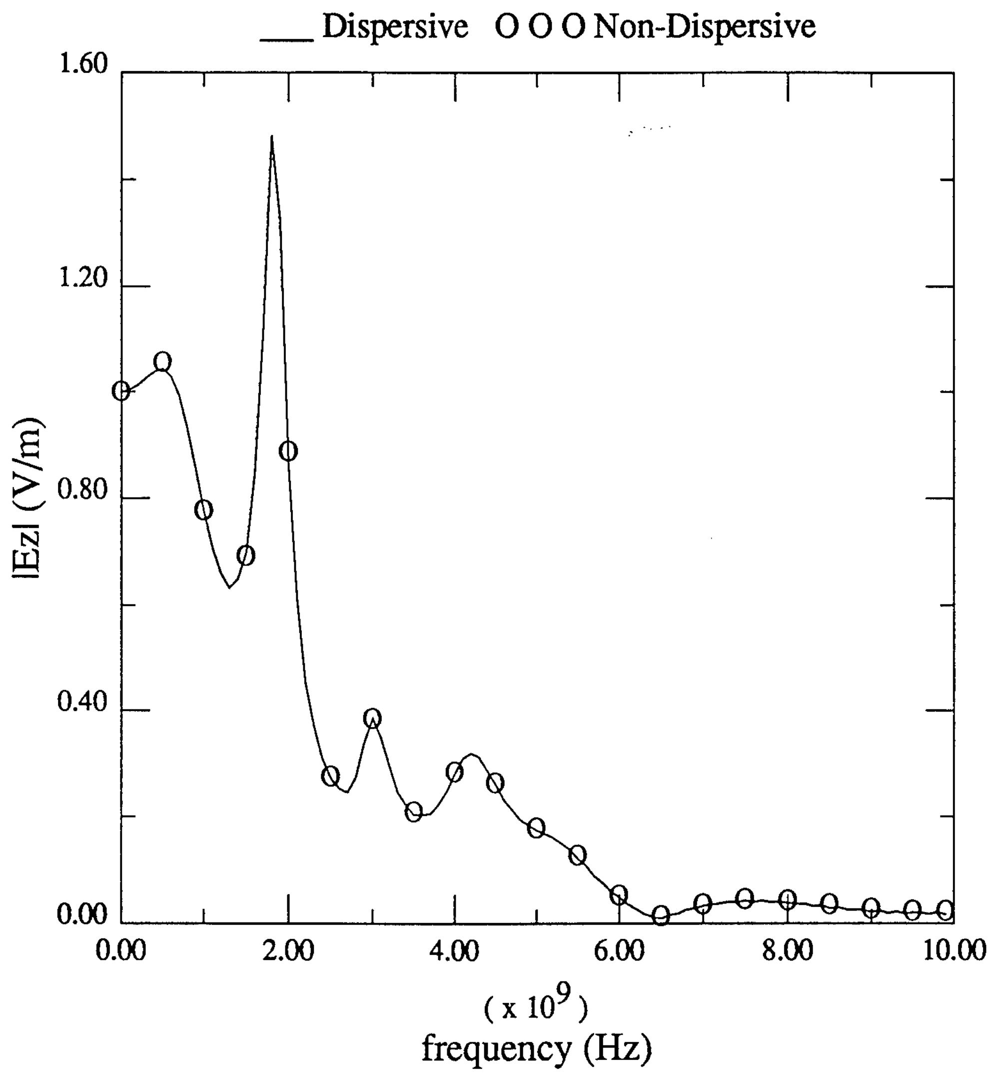


Figure 12. Magnitude of E_z vs. frequency at point 3 for Debye cylinder. ——— - Dispersive FDTD Formulation, O O O - Non-Dispersive FDTD formulation.

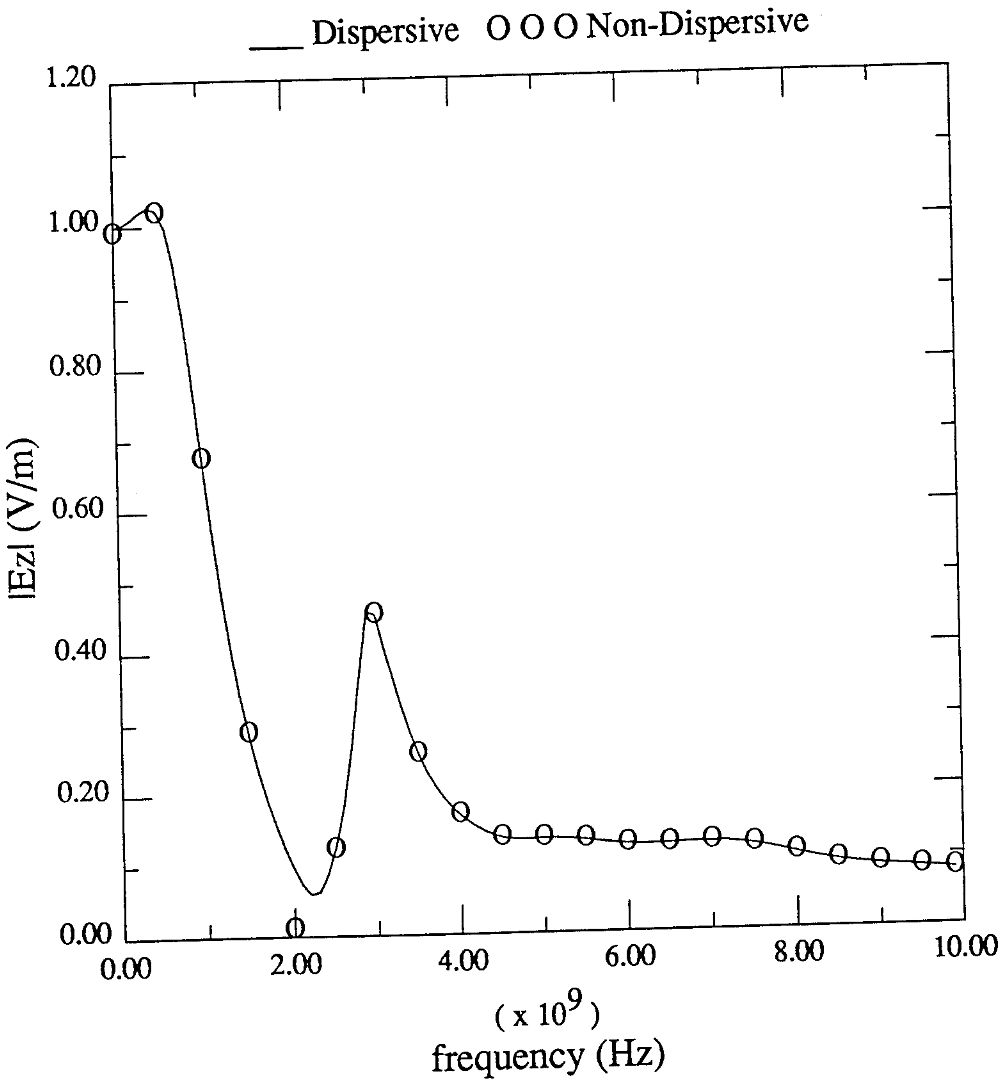


Figure 13. Magnitude of E_z vs. frequency at point 4 for Debye cylinder. ——— - Dispersive FDTD Formulation, 0 0 0 - Non-Dispersive FDTD formulation.

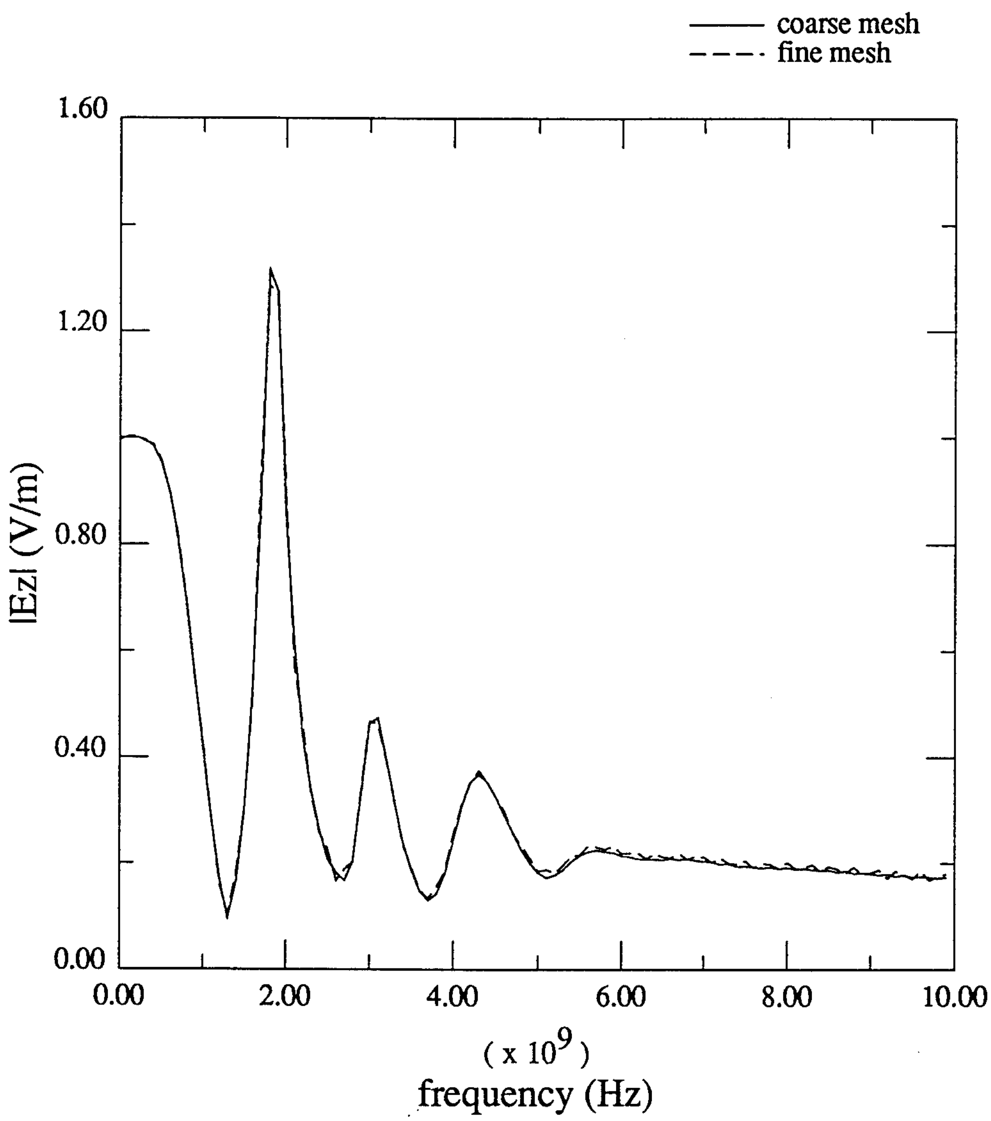


Figure 14. Magnitude of E_z vs. frequency for Debye cylinder at point 1 using course (350μm) and fine (233.333μm) mesh spacings.

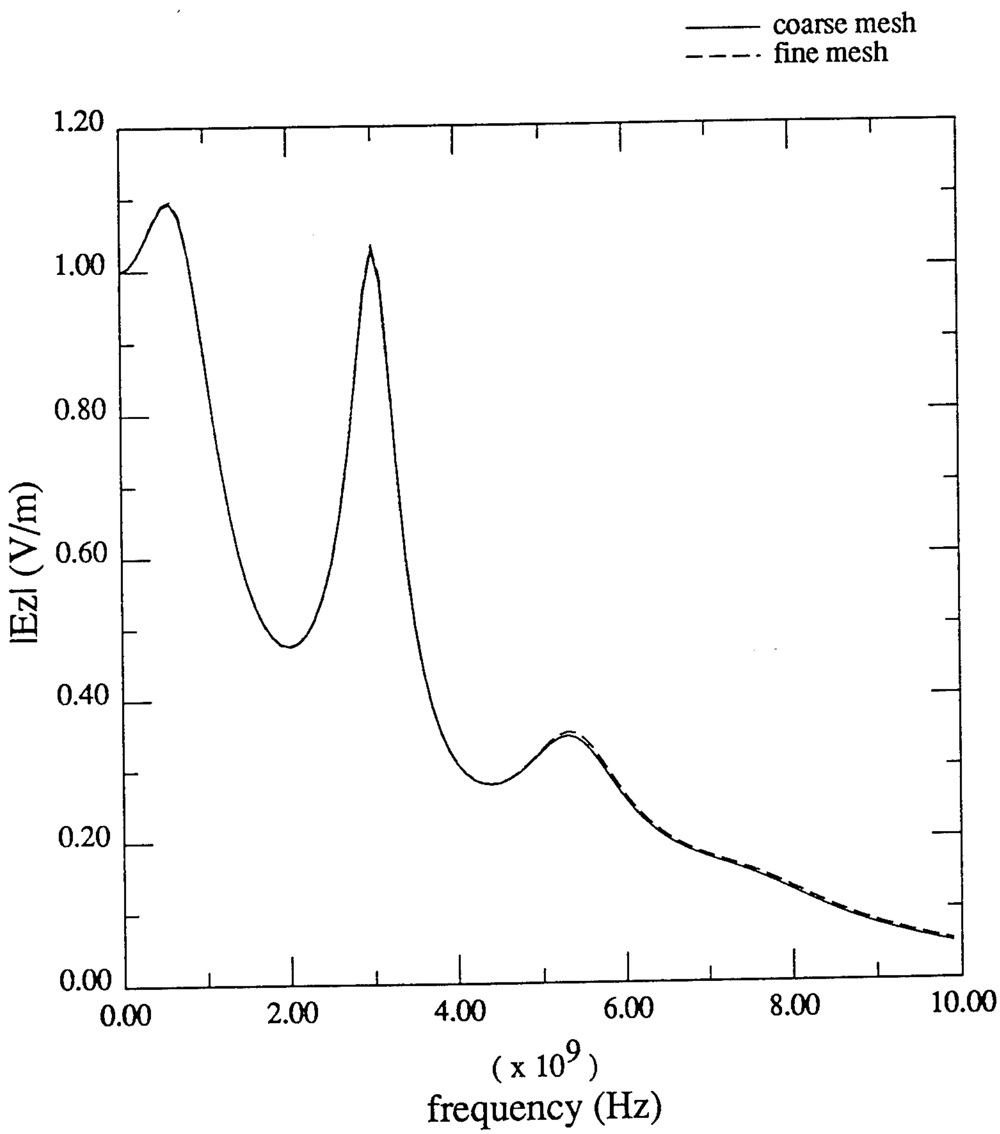


Figure 15. Magnitude of E_z vs. frequency for Debye cylinder at point 2 using course (350μm) and fine (233.333μm) mesh spacings.

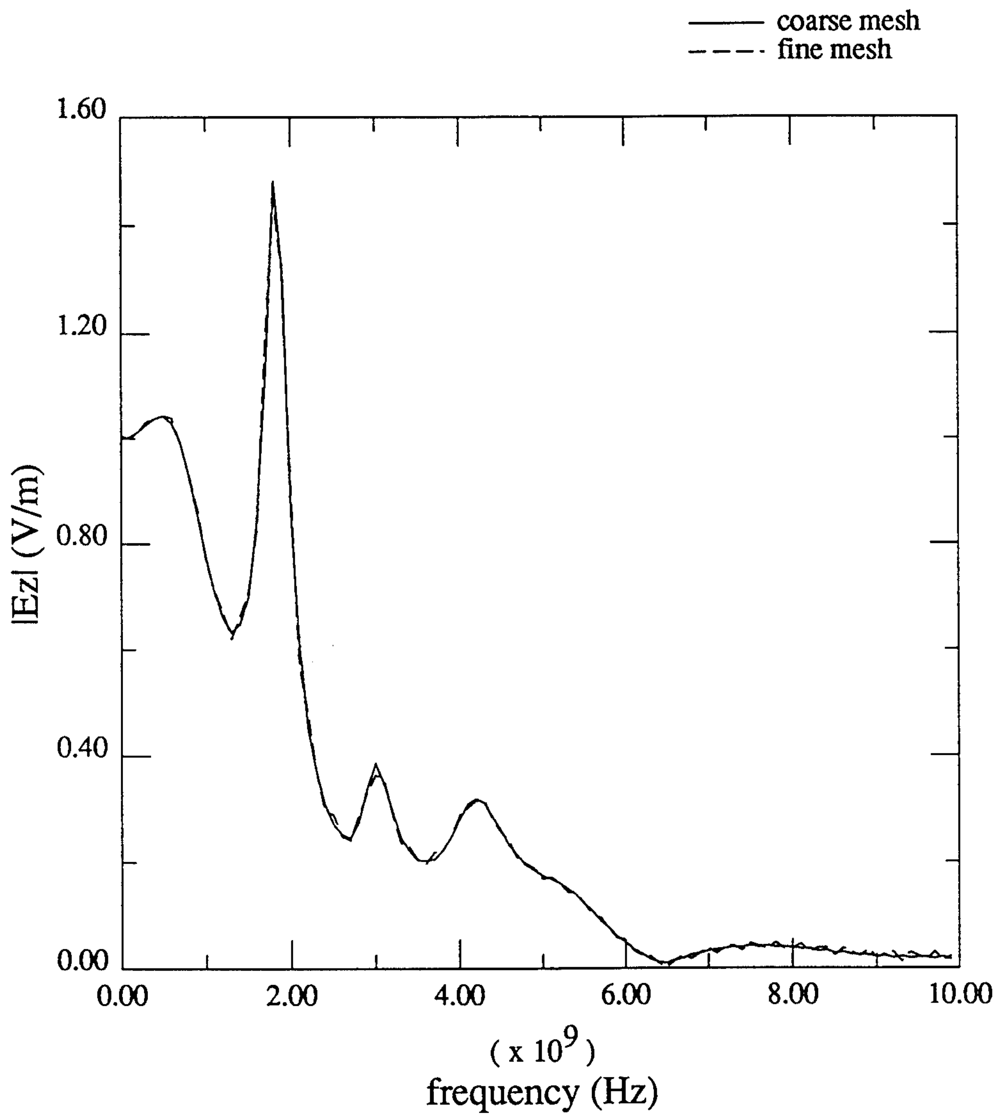


Figure 16. Magnitude of E_z vs. frequency for Debye cylinder at point 3 using course (350μm) and fine (233.333μm) mesh spacings.

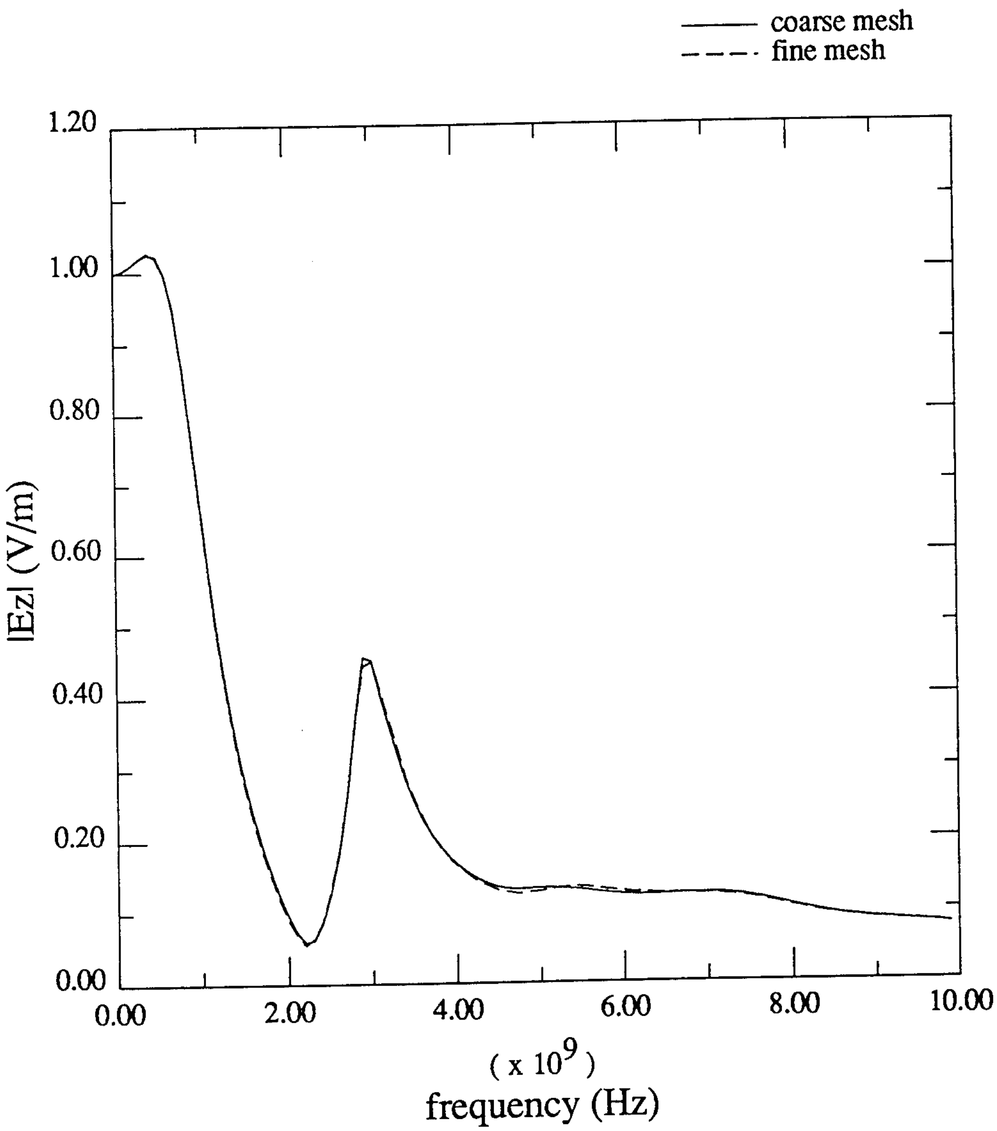


Figure 17. Magnitude of E_z vs. frequency for Debye cylinder at point 4 using course (350μm) and fine (233.333μm) mesh spacings.

series solution) the program must be run for each frequency point of data desired. In order to conserve computational expenditures, these lorentz parameters were chosen because, over the frequency band of interest, they yield approximately the same complex permittivity values as the debye media of the previous example, thus alleviating the need to rerun a complete set of data points for comparison. Since the purpose of the example is to check that the lorentz portions of the new FDTD programs are working properly, and not to gain physical insight into this particular problem, there is no loss of generality.

With that said, the results are presented in Figures 18 through 21 for the electric field at point 1, 2, 3 and 4. Once again we see good agreement between the data, thus indicating that the lorentz portions of the FDTD code have been properly implemented.

Next, we will examine the transient behavior of the electric field within the debye cylinder. The same cylinder as above was used with the 350 μm lattice spacing. The plane wave use to excite the problem was a unit step modulated 5-GHZ sinusoid. The sinusoid was turned on at time zero (taken to be when the wavefront crossed the i=iclo line) and allowed to remain on for the duration of the observation time.

Two sets of data were compared. First, the traditional FDTD method was used to obtain the time signature of the electric field at the four output points in the cylinder. In this case the material parameters were those calculated for the debye media at a frequency of 5-GHZ. Secondly, the dispersive FDTD formulation was used to produce the same set of data.

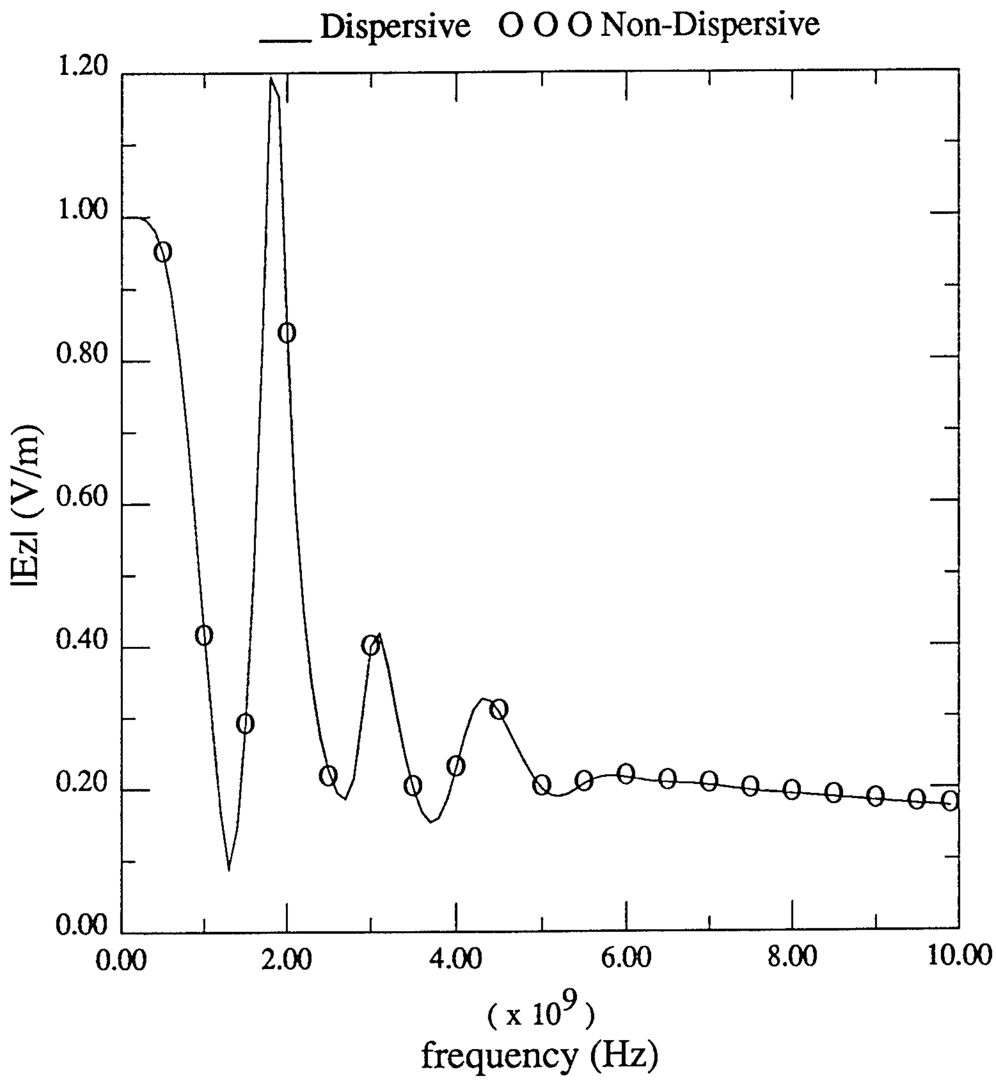


Figure 18. Magnitude of E_z vs. frequency at point 1 for Lorentz cylinder. ——— - Dispersive FDTD Formulation, 0 0 0 - Non-Dispersive FDTD formulation.

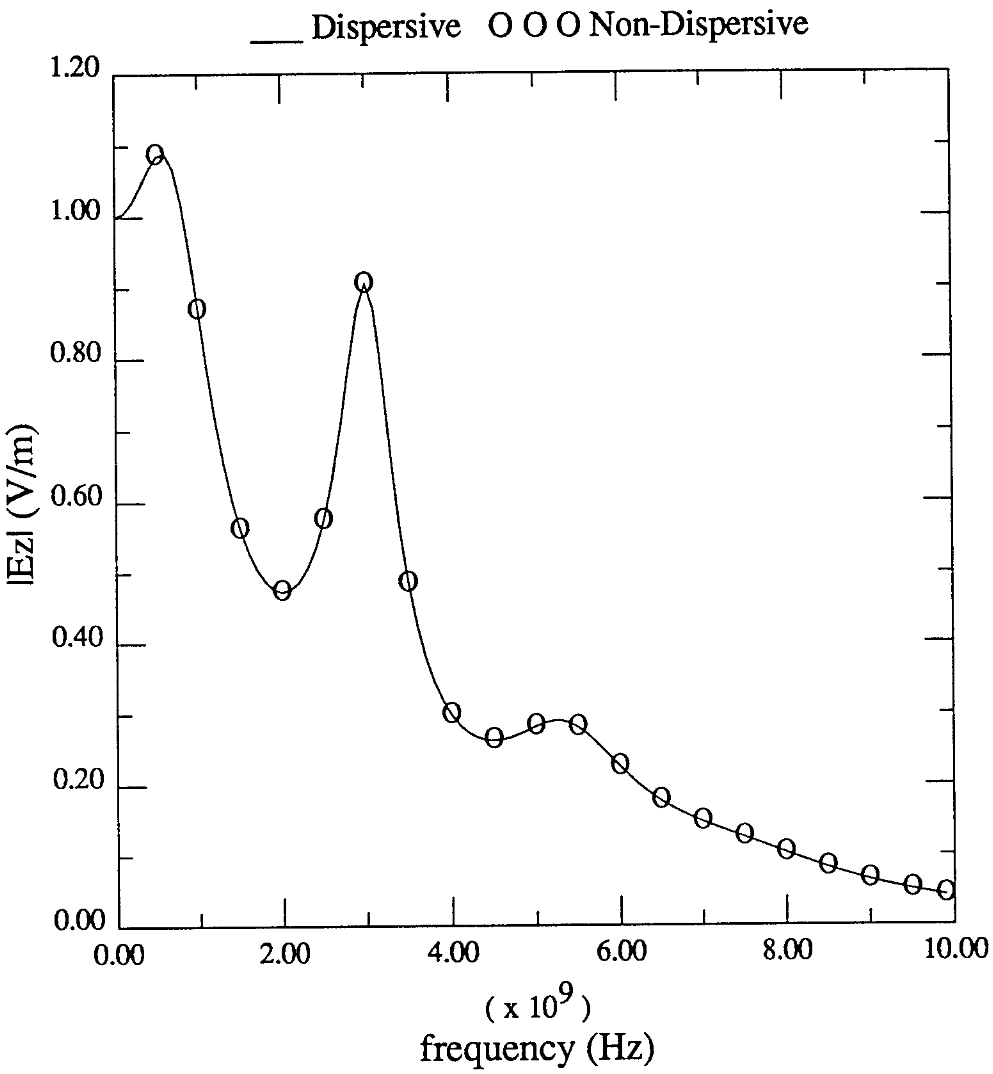


Figure 19. Magnitude of E_z vs. frequency at point 2 for Lorentz cylinder. ——— - Dispersive FDTD Formulation, O O O - Non-Dispersive FDTD formulation.

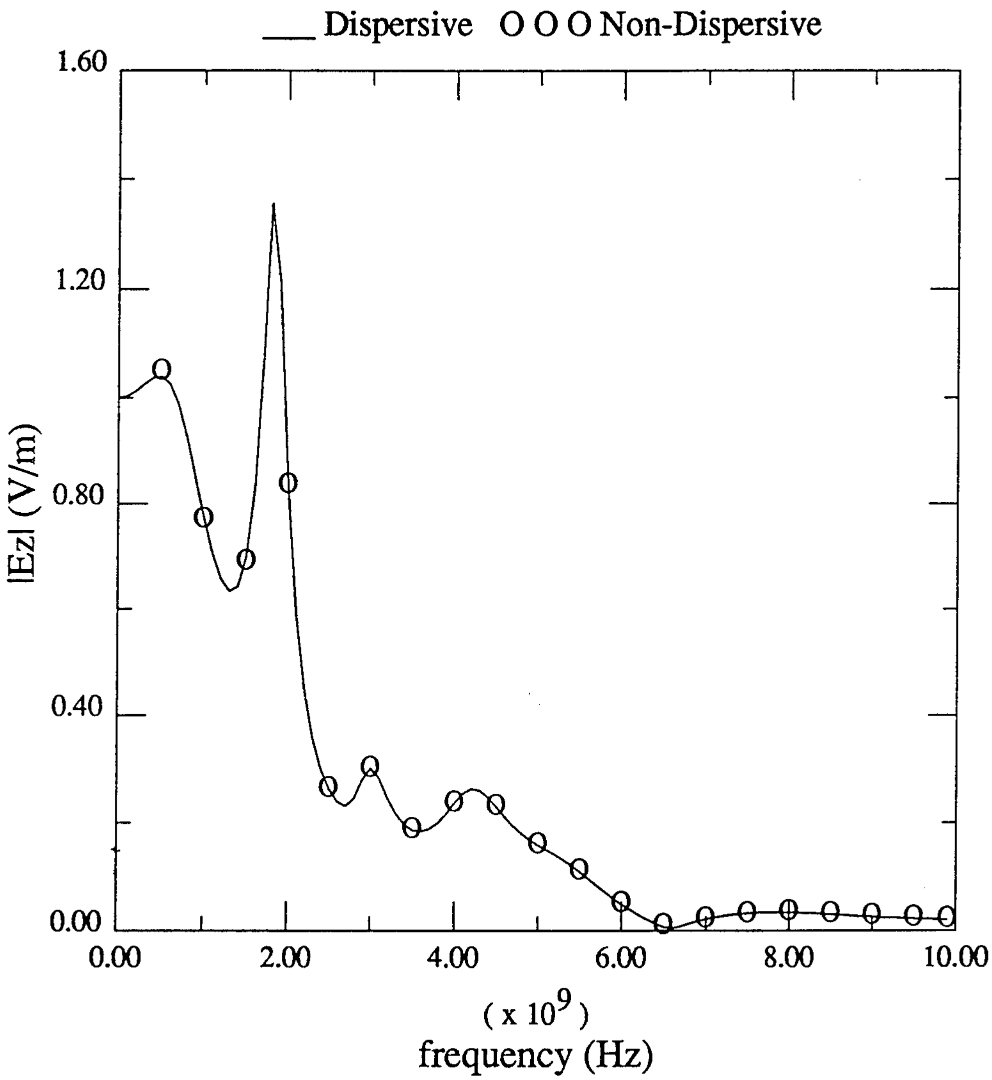


Figure 20. Magnitude of E_z vs. frequency at point 3 for Lorentz cylinder. ——— - Dispersive FDTD Formulation, 0 0 0 - Non-Dispersive FDTD formulation.

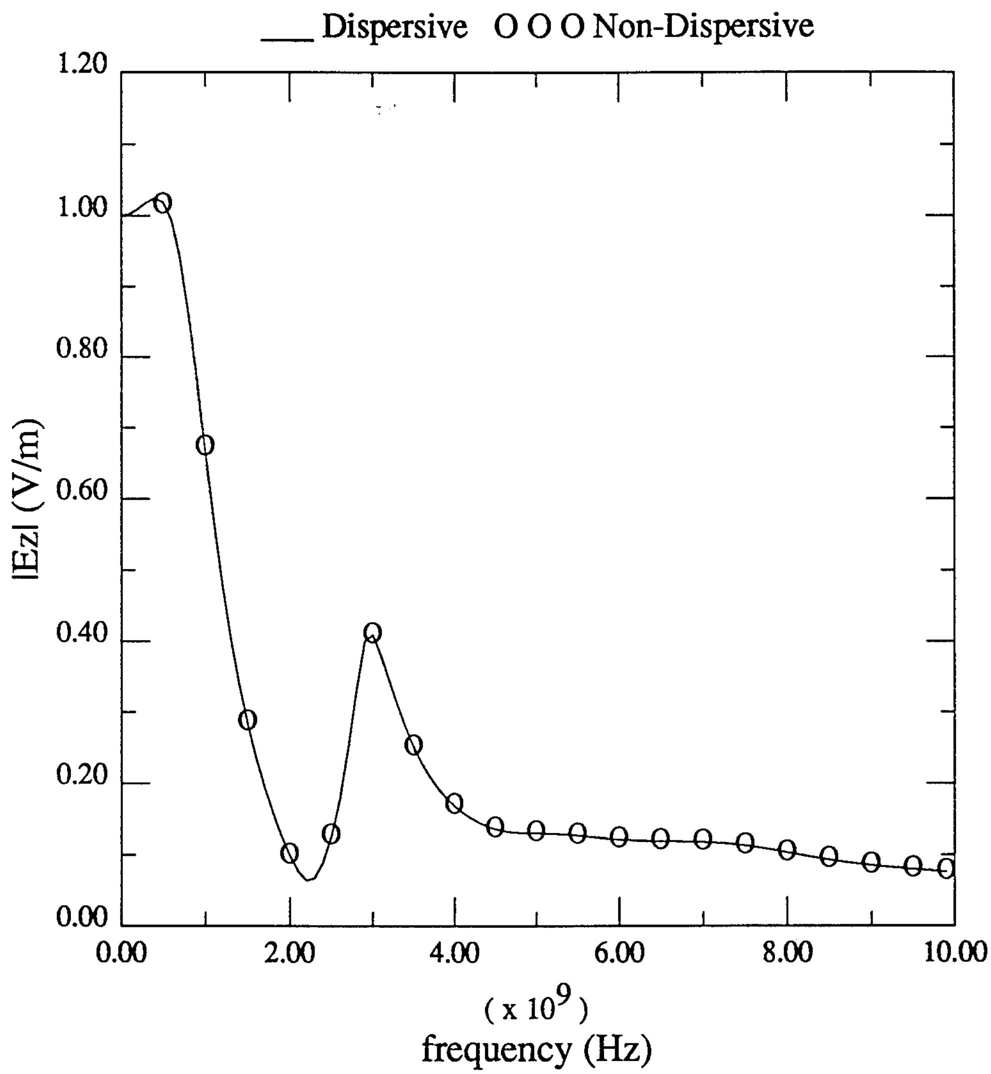


Figure 21. Magnitude of E_z vs. frequency at point 4 for Lorentz cylinder. ——— - Dispersive FDTD Formulation, 0 0 0 - Non-Dispersive FDTD formulation.

We would expect both sets of results to converge to the same solution as time approaches infinite (i.e., the same steady state solution). However, due to the high frequency components in the unit step use to modulate the sinusoid, we would expect to see differences in the early time (transient) response since the traditional FDTD method cannot account for the dispersive effects of the debye media.

The results for the time signature of the electric field at point 1-4 in the cylinder are shown in Figures 22-25. As expected, the solutions become practically identical after several cycles of the 5-GHZ sinusoid. It is also interesting to observe the differences in the early time response due to the high frequency components in the modulating waveform. These transient effects become more pronounced the farther the wave propagates in the material.

Of particular interest in the response in Figure 24 which is the electric field at output point 3. The early time difference in the two waveforms is quite pronounced. In fact, the magnitude for the dispersively modeled cylinder is significantly greater than when the dispersive effects are ignored, indicating the develop of something similar to a Brillioun precursor. These effects are very important to account for when assessing field levels in the human body.

In order to assess the numerical accuracy of the early time results we have computed the response using the 233.333 μm lattice spacing. The results are summarized in Figures 26-29. These figures show the early time response at outputs 1-4 for the course

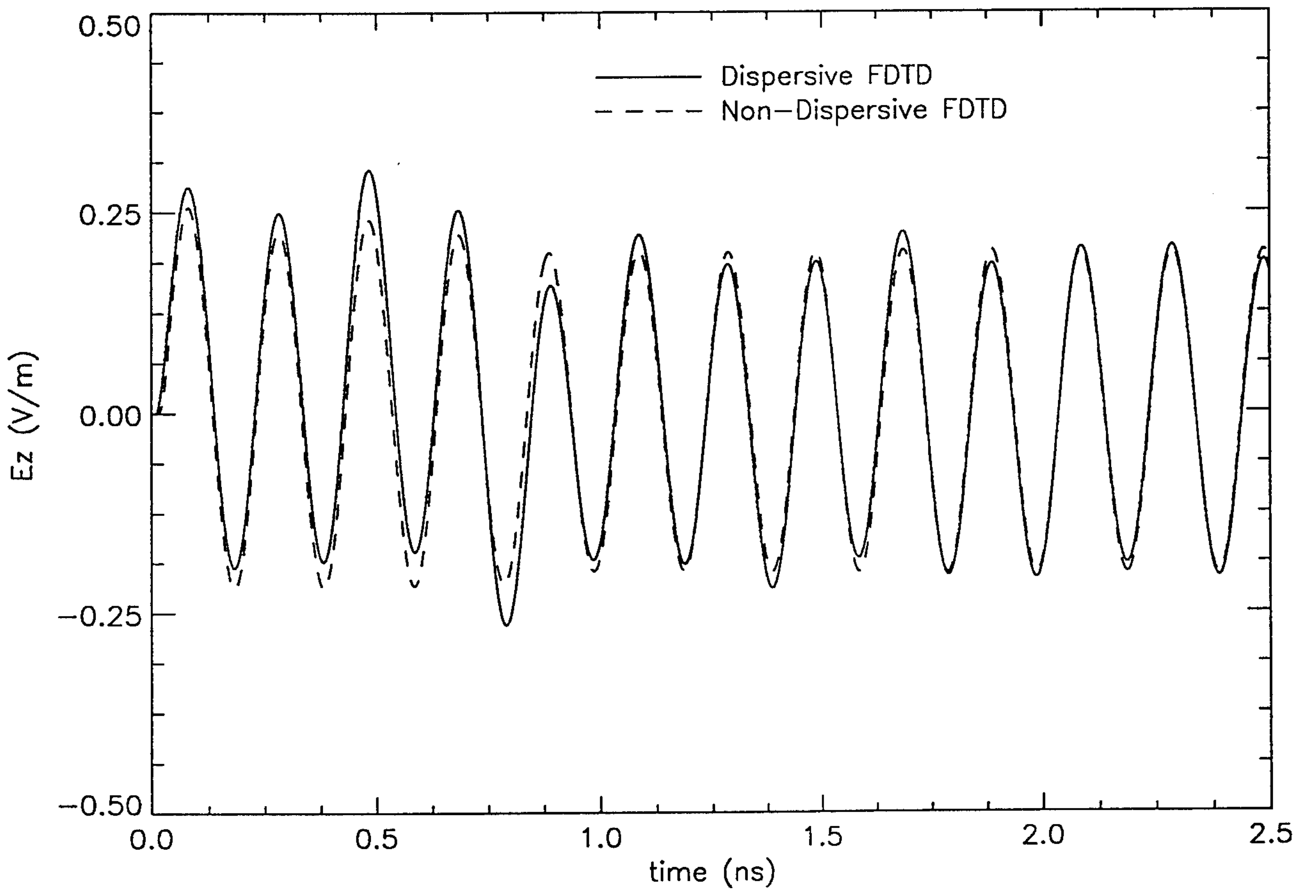


Figure 22. Time domain electric field response to a unit step modulated 5-GHZ sinusoid at observation point 1 for dispersive and non-dispersive FDTD schemes.

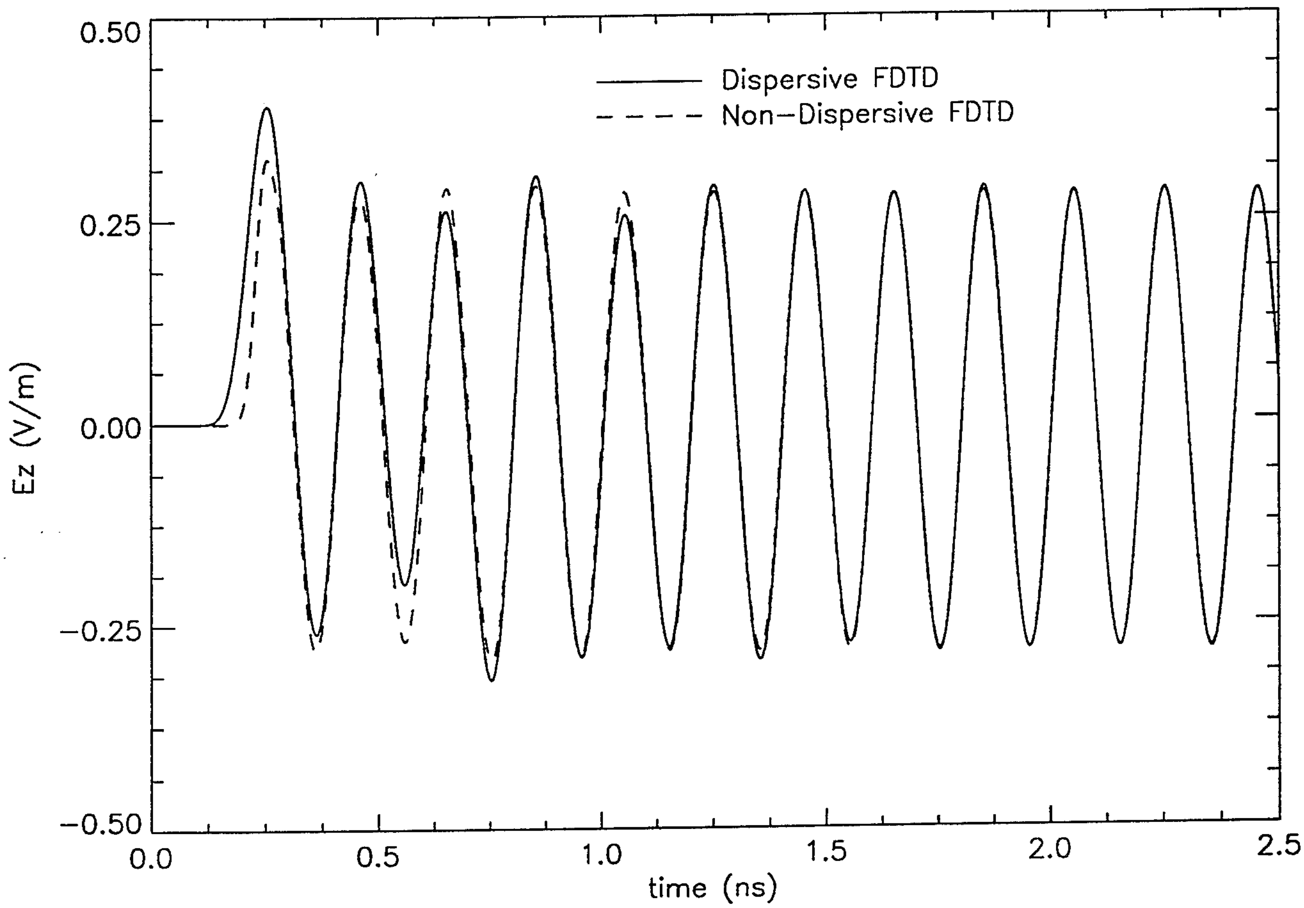


Figure 23. Time domain electric field response to a unit step modulated 5-GHZ sinusoid at observation point 2 for dispersive and non-dispersive FDTD schemes.

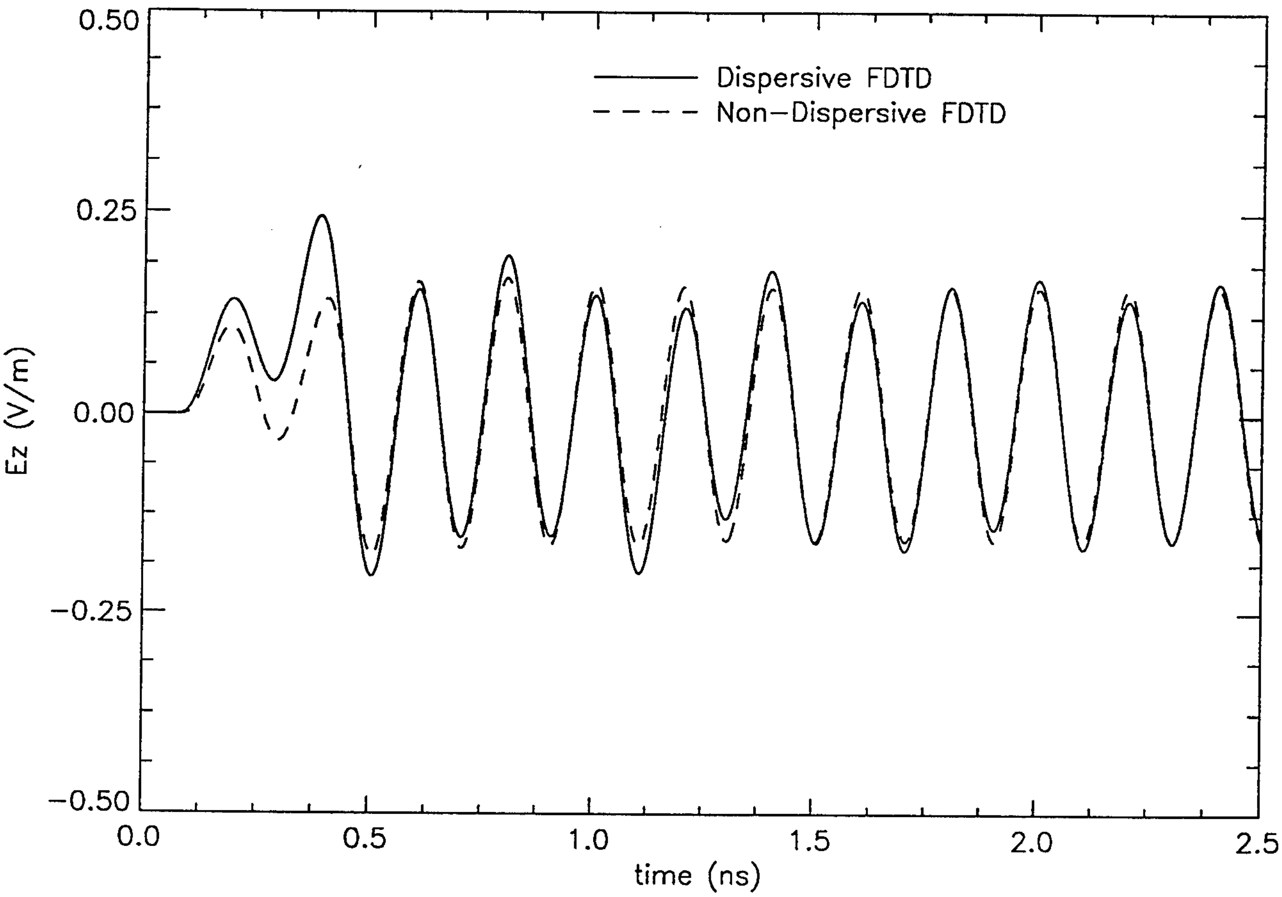


Figure 24. Time domain electric field response to a unit step modulated 5-GHZ sinusoid at observation point 3 for dispersive and non-dispersive FDTD schemes.

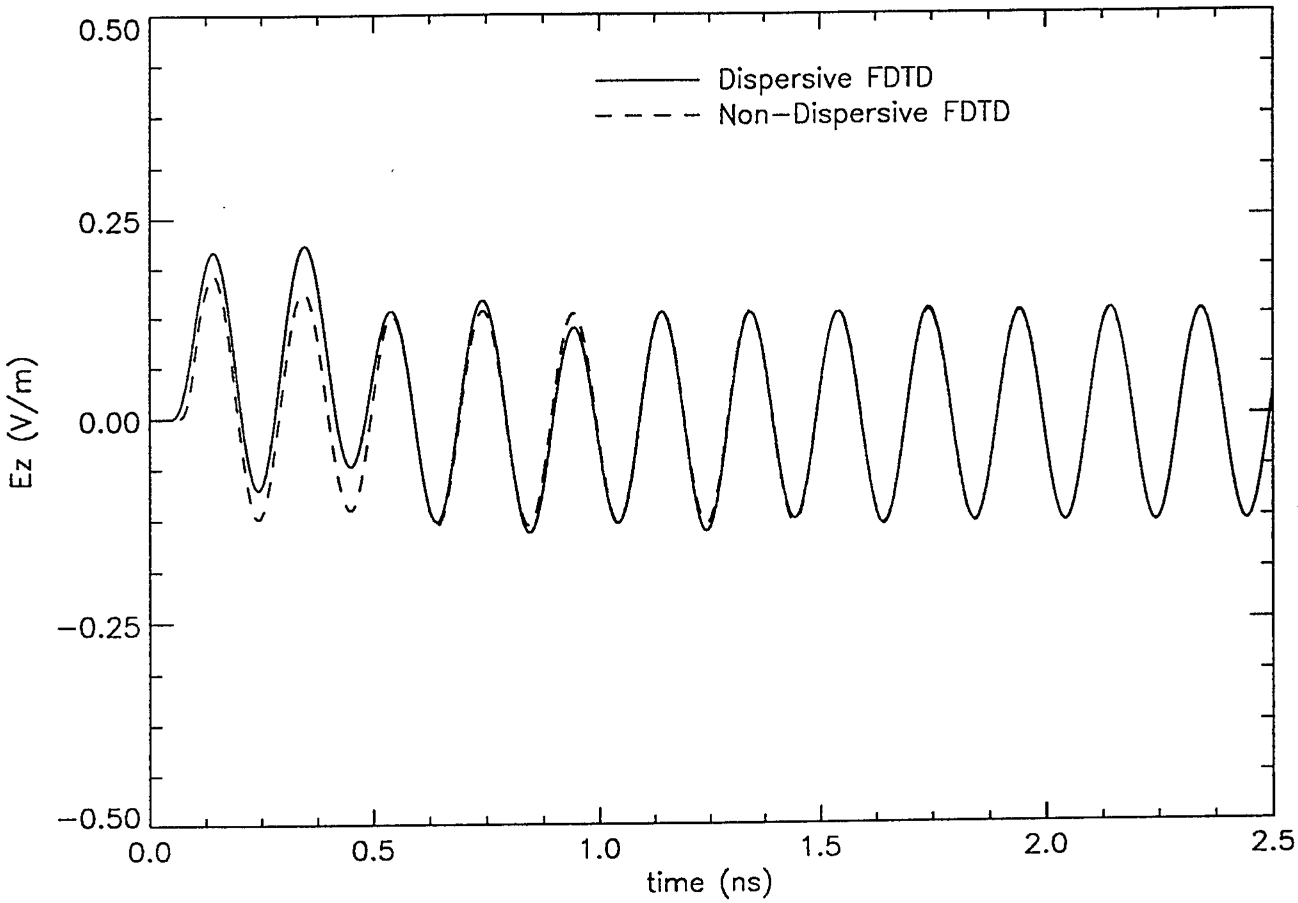


Figure 25. Time domain electric field response to a unit step modulated 5-GHZ sinusoid at observation point 4 for dispersive and non-dispersive FDTD schemes.

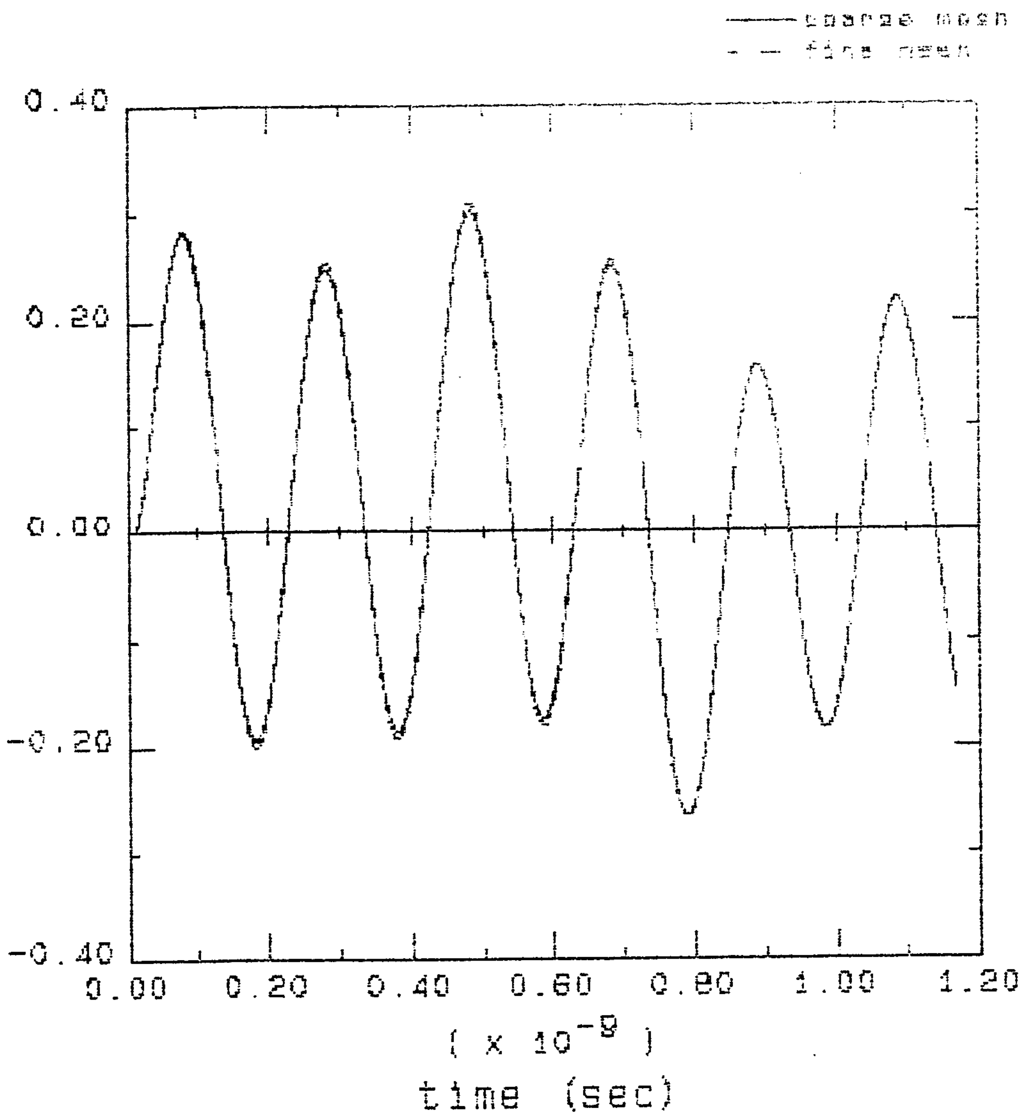


Figure 26. Time domain electric field response at observation point 1 for coarse (350μm) and fine (233.3333μm) FDTD meshes.

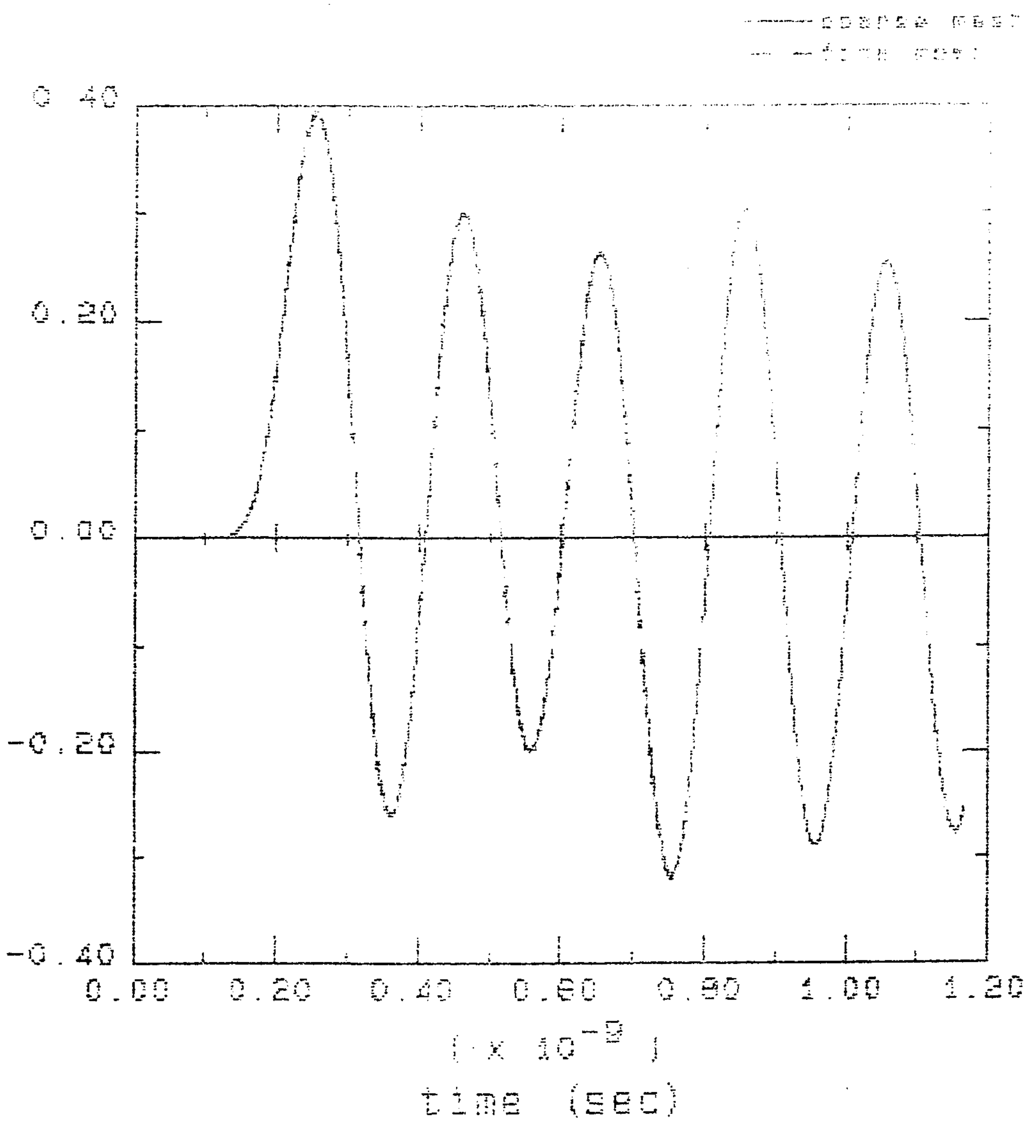


Figure 27. Time domain electric field response at observation point 2 for coarse (350μm) and fine (233.3333μm) FDTD meshes.

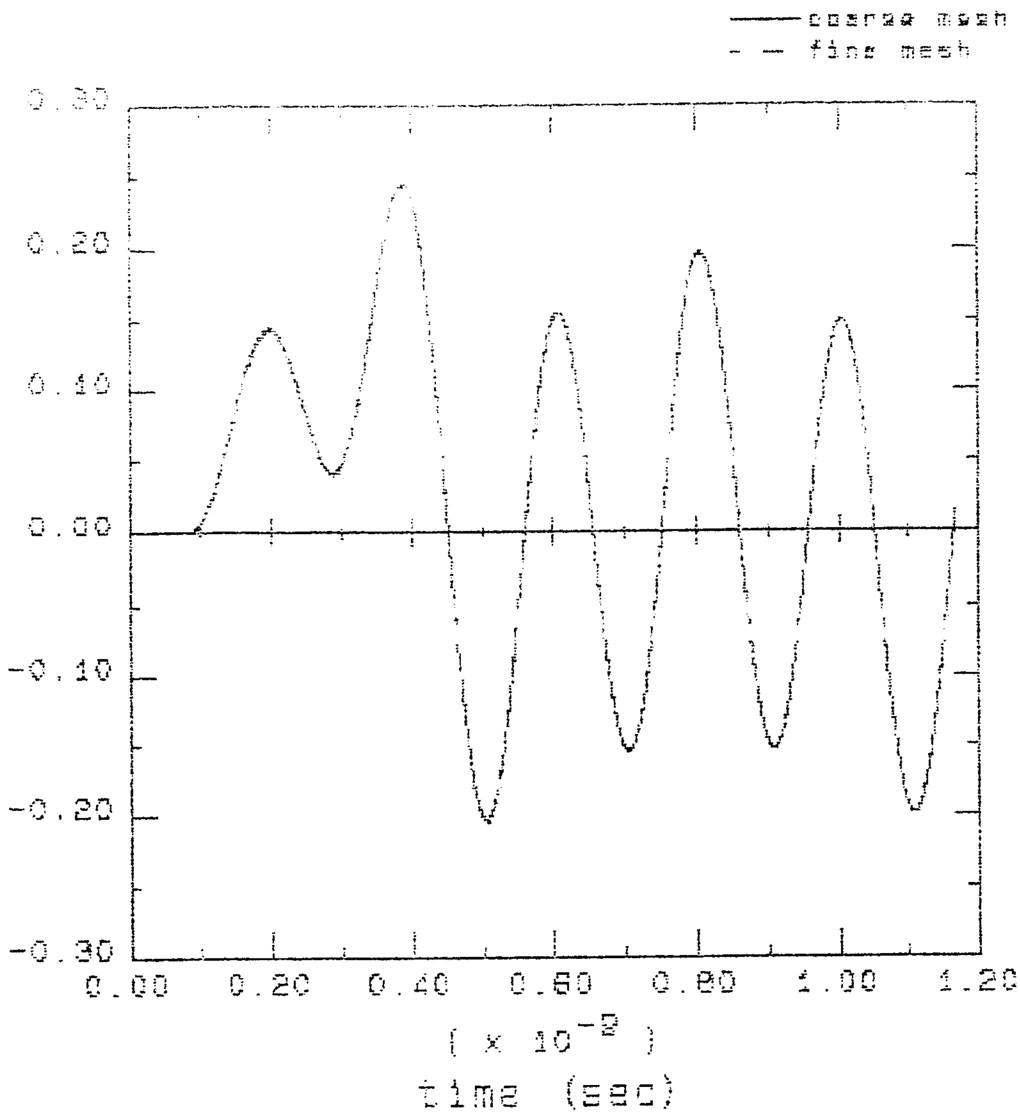


Figure 28. Time domain electric field response at observation point 3 for coarse (350μm) and fine (233.3333μm) FDTD meshes.

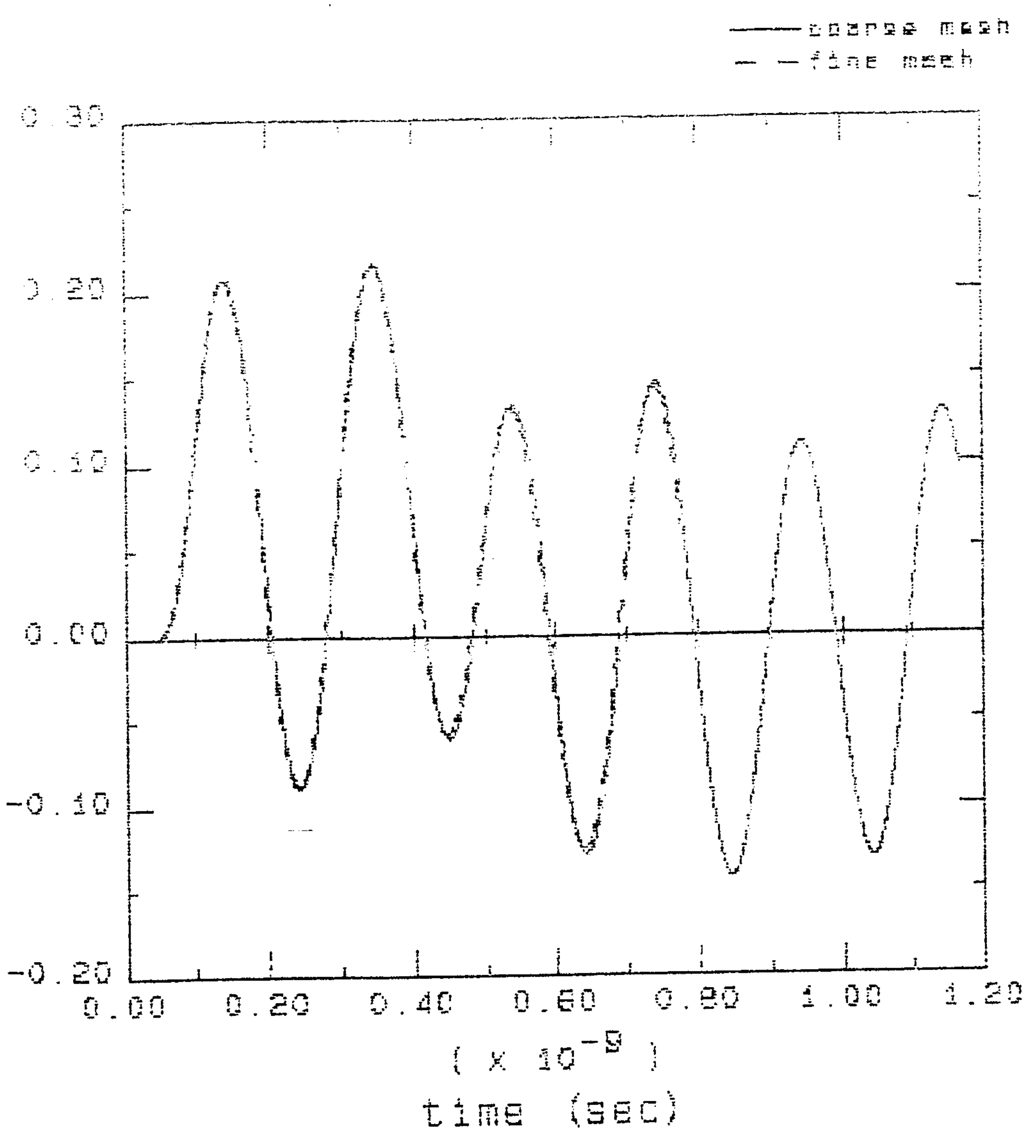


Figure 29. Time domain electric field response at observation point 3 for coarse (350μm) and fine (233.3333μm) FDTD meshes.

and fine FDTD meshes. Since the two sets of data are practically indistinguishable, we can assume that the results are numerically reliable.

As a final numerical example, we compute the transient fields in the concentric dispersive cylinders as shown in Figure 5 For this example the radius b is taken to be 7 mm and the radius is 3.5 mm. The inner cylinder is a debye material with the same parameters used above. The outer cylinder is taken to be a lorentz material with parameters $\varepsilon_s = 25$, $\varepsilon_i = 1$, $\omega_o = 250x10^9$ and $\delta = 50x10^9$. The same 5-GHZ waveform was used.

The results are presented in Figures 31 through 34 for output points 1 through 4 which have the same coordinates as used previously. From these results it is evident that there is significant transient phenomenon. We also note the convergence to a steady state value as would be expected.

CONCLUSIONS

In this report we have presented the formulation of the FDTD method for application to bodies involving dispersive media. A computer program package has been written based on this formulation and the source code as well as instructions for use are presented in the appendices to follow. Using this program we have computed some results for dispersive cylinders. The results presented in this report are very preliminary in the sense that the programs written are very general and capable of much more that the analysis of simple cylinders. It is unfortunate that time and budget constraints have prevented the application of the

31

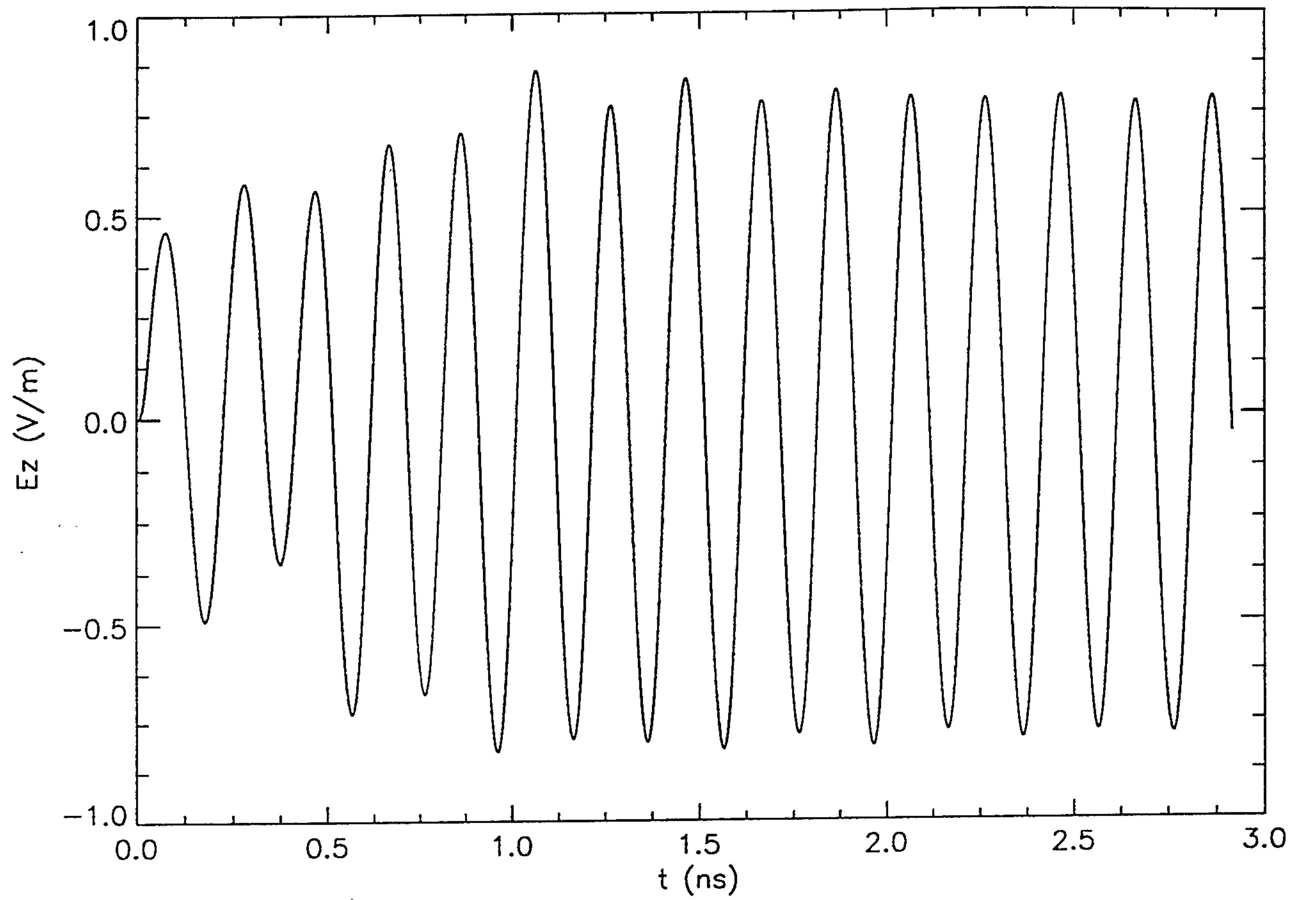


Figure 31. Time domain electric field response to a unit step modulated 5-GHZ sinusoid at observation point 1 for the concentric cylinder example.

32

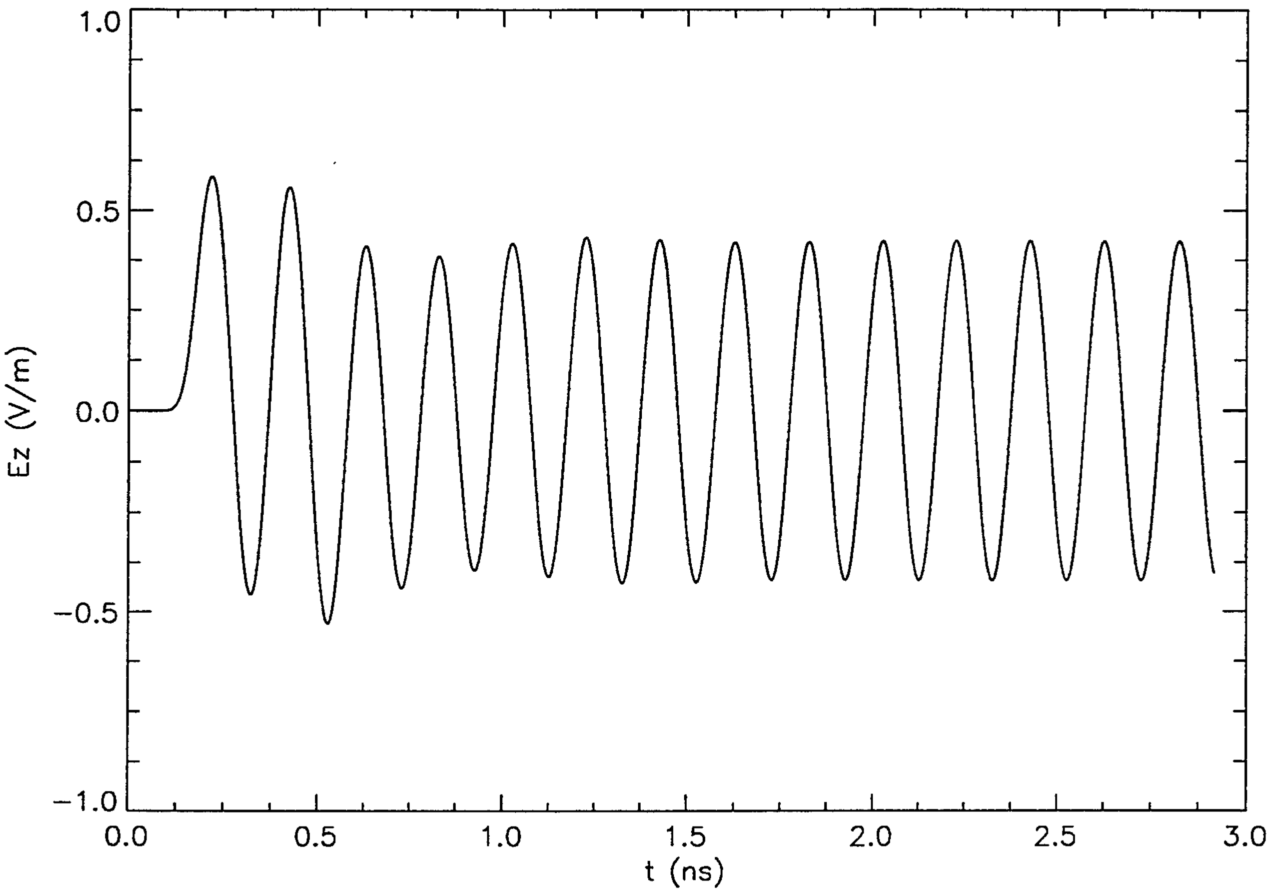


Figure 32. Time domain electric field response to a unit step modulated 5-GHZ sinusoid at observation point 2 for the concentric cylinder example.

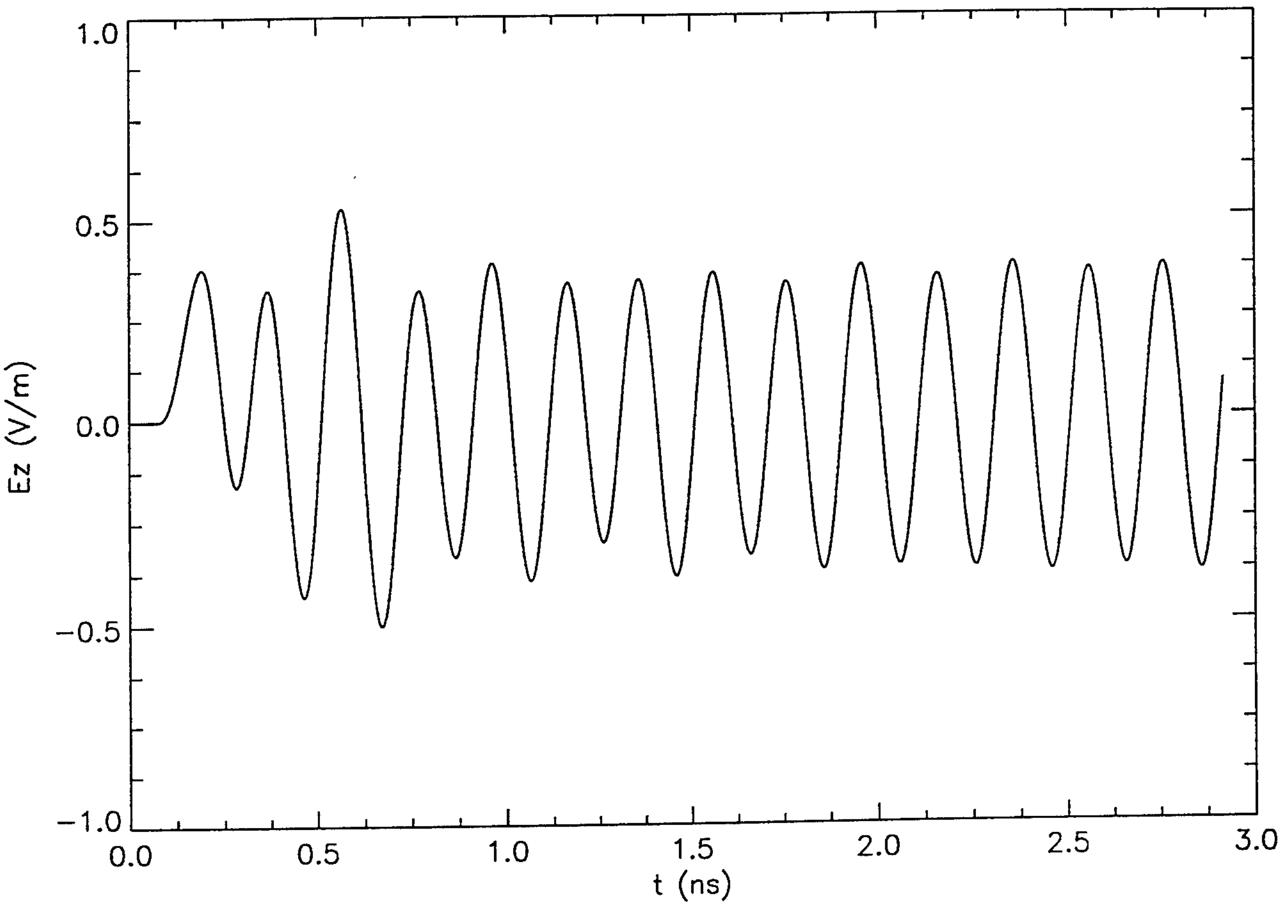


Figure 33. Time domain electric field response to a unit step modulated 5-GHZ sinusoid at observation point 3 for the concentric cylinder example.

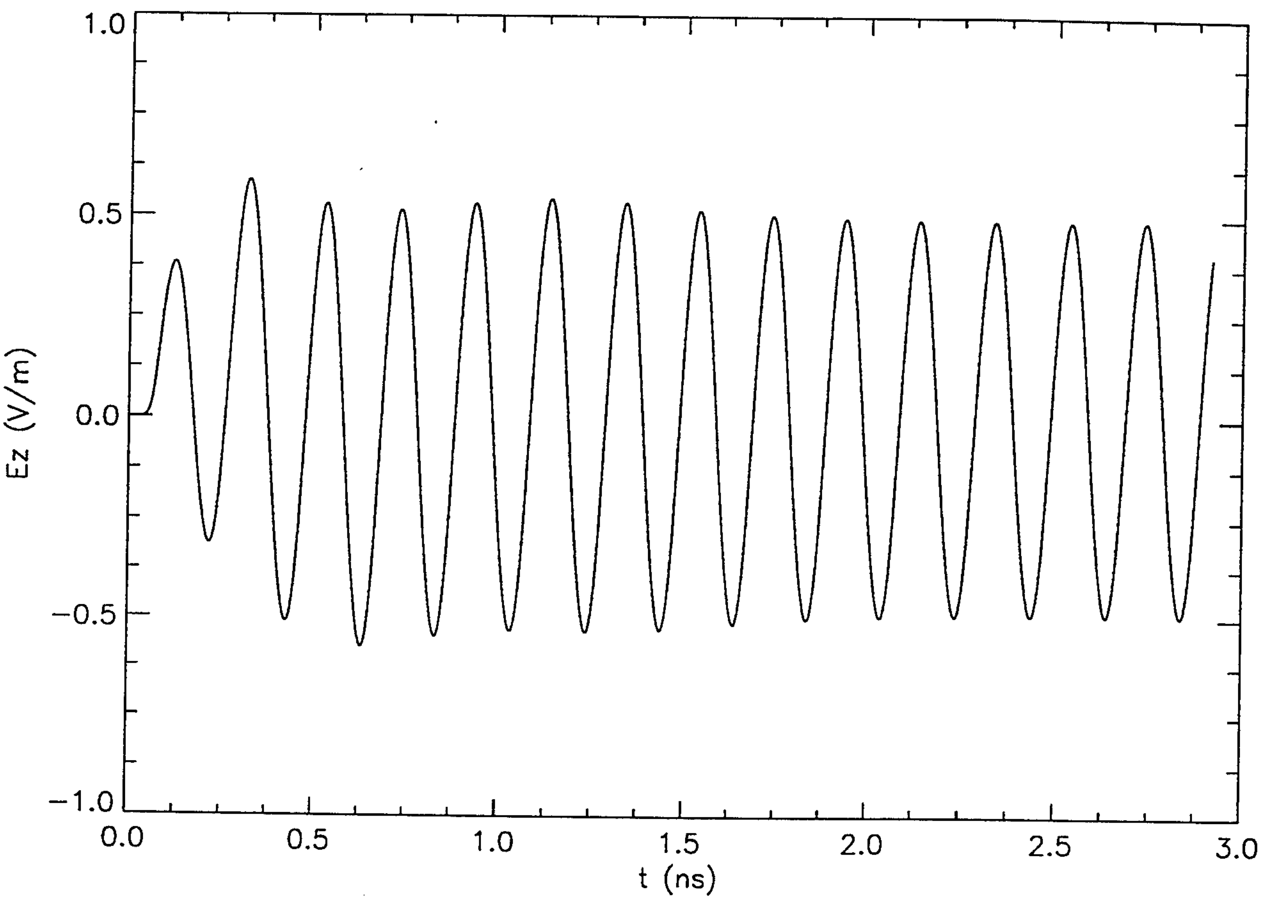


Figure 34. Time domain electric field response to a unit step modulated 5-GHZ sinusoid at observation point 4 for the concentric cylinder example.

codes to more challenging, practical problems.

It has been demonstrated that the programs are in working order and capable of predicting transient phenomena in dispersive media. The user's manual and code listings in the appendices will allow future users to investigate a wide variety of challenging two dimensional electromagnetic problems involving dispersive media.

While this work has examined two dimensional problems, the FDTD method is quite capable of calculations in three dimensions. The dispersive formulations presented here can easily be extended to three dimensions. The only limit to the type of problem that can be solved is that of computational resources. It is feasible that with the use of the new super-computing technologies to model a human body in three dimensions, including dispersive tissue effects, using the FDTD method and obtain the accurate prediction of transient fields.

APPENDIX 1

In this appendix, the computer programs used for the analysis presented in this report are described. It is important to become somewhat familiar with the location and numbering of the FDTD lattice points in order to use the programs intelligently. The numbering of the electric field point in the FDTD lattice are shown in Figure 1. The index "i" designates the location along the x-axis and the index "j" references points located on the y-axis. Since all points that must be specified as input data to the program are concerned with electric field nodes, the user need not be concerned with magnetic field node locations unless the magnetic field values are desired as output quantities. In this event, all the user has to know is that the H_y field component to the immediate right of any E_z point shares the same indices (i,j) as does the H_x field component immediately above the E_z node. For example, the E_z field point at (i,j) is referenced in the program as EZ(I,J), while $H_x(i,j+1/2)$ is indexed as HX(I,J) and $H_y(i+1/2,j)$ is indexed as HY(I,J).

There are six FORTRAN files that make up the entire program package. They are:

(1) ARRAY.F

(2) EXCITE.F

(3) OUTPUT.F

(4) BUILD.F

(5) FDTD.f

(6) FDTDS.F

The programs must be compiled and linked together (with the exception of FDTD.F and FDTDS.f which will be discussed shortly). In the following sections, the functions and required input data/program changes for each subroutine is described.

ARRAY.F

This section of code is not a program per se, but rather, a collection of parameter statements that are include in each subroutine via the INCLUDE statement. In this way, the user need not change the array dimensions in each section of code for larger mesh dimensions. The parameters specified in ARRAY.F are:

nsx = maximum number of electric field nodes in the x direction.

nsy= maximum number of electric field nodes in the y direction.

npec = maximum number of perfect electric conductors in the mesh.

nm = maximum number of dielectric (dispersive or otherwise) in the mesh.

nd = maximum number of different materials in the mesh.

Note that the arrays can be over-dimensioned without impeding proper program operation. The exception is nd which for larger values requires program modifications which will be discussed later.

EXCITE.F

This user modified subroutine specifies the time function to use as the exciting waveform. It is assumed that the excitation is a plane wave propagating in the direction specified by θ (see Figure

2). Time zero is referenced when the wavefront is at the mesh point (iclo,jclo). This subroutine specifies the time function for the plane wave. The parameter pw is the pulse width of the excitation. Currently it is set at 30000Δt which means that the plane wave excitation turn on at time zero and off at time 30000Δt. (In this case 30000Δt is very large to simulate a unit step modulation). The form of the time dependence must be defined by the user at five point within the program. The variables that require modification are ezinc, hxincl, hxinc2, hyincl and hyinc2.

Currently a sinusoid and a Gaussian (commented out) are programmed. When changing the excitation the user should change the time dependence only. The multiplying factors (e.g., costh/etao) MUST remain unchanged for the program to work properly. This arrangement, while requiring some program modifications by the user, has the advantage of allowing any arbitrary time function to be used in the program.

OUTPUT.F

This subroutine specifies the output format for the field data. Since so much data is generated by the FDTD method, this user modified subroutine allows for any output format of field values to be programmed. Currently it outputs EZ vs. time at four mesh points. Since all field information, i.e, Ez, Dz, Hx and Hy, at each mesh point and at each time step is passed to this subroutine, the user may tailor the output to any desired format - the possibilities are practically limitless.

BUILD.F

This subroutine takes the data read in from the input file and uses it to "build" the FDTD mesh. It's use is basically transparent to the user with one exception. Since, due to the cartesian nature of the grid, materials must be specified as rectangular blocks, complicated bodies, or those with curved surfaces, may require a large number of input data lines. In order to circumvent this a section of build may be used to program it to build the body automatically. The section of code that does this for the concentric cylinder example is included. This section is heavily commented with the required storage information to program it.

FDTD.F and FDTDS.F

These programs form the heart of the package. They read in the input data and perform the actual FDTD time-stepping procedure. These two codes differ in one respect. FDTDS.F employs a magnetic symmetry wall along the y-axis to reduce computational expenditure for objects possessing symmetry like the cylinders analyzed in this report. Figure 30 shows the geometry of the mesh when a symmetry boundary is included. The use of this code is the same as for FDTD.F, which does not use a symmetry boundary, with two exception: (1) When using FDTDS.F the incident angle MUST be specified as 0 degrees and (2) when using FDTDS.f the parameter iclo MUST be set equal to 1.

The only thing the user needs to know about the operation of these programs is the format of the input data. The program reads from a file called "FDTD.IN" which contains all the data describing the mesh parameters and the objects to be analyzed.

The locations of conductors and dielectrics in the mesh is

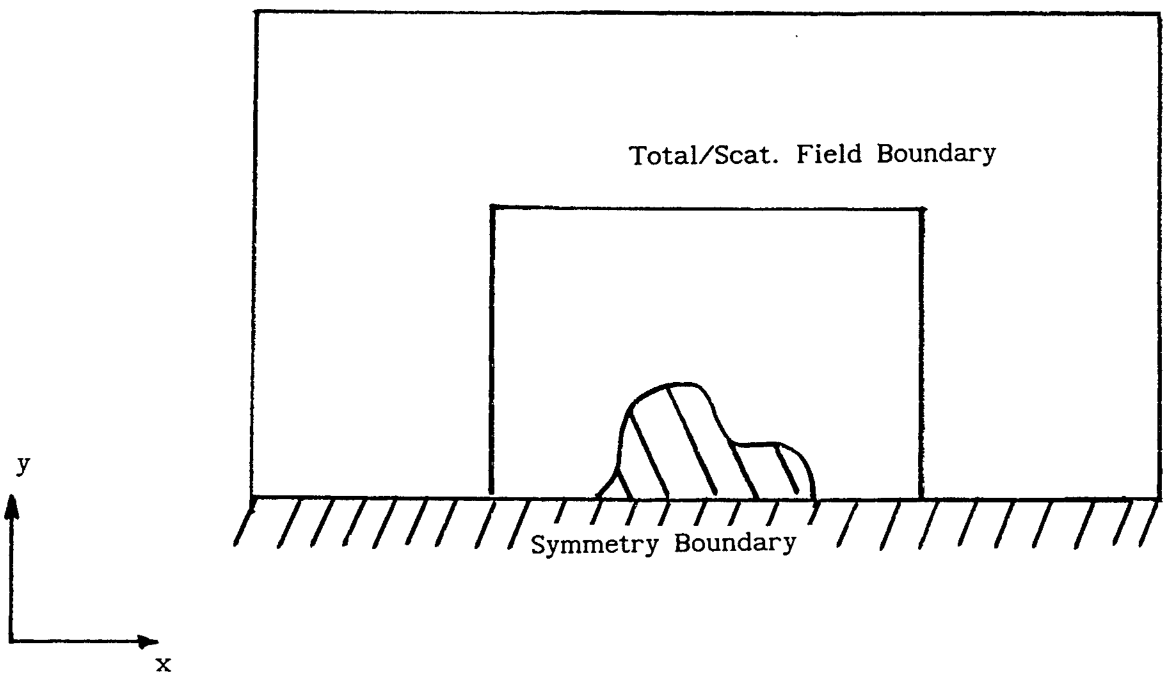


Figure 30. FDTD lattice layout including a magnetic symmetry boundary at $j = 0.5\Delta\ell$.

specified in terms of electric field nodes. The affected nodes are specified in the input file. Since the mesh is cartesian, the materials (or conductors) are specified as blocks. Following is the format of the input data file. In the actual file the data are separated by commas. A sample input file for the cylinder problem is included at the end of the source listings.

nx,ny	Mesh size
iclo,ichi,jclo,jchi	mesh division
pec_ilo,pec_ihi,pec_jlo,pec_jhi	PEC locations
. . . .	
. . . .	
pec_ilo,pec_ihi,pec_jlo,pec_jhi	
0,0,0,0	
med_ilo,med_ihi,med_jlo,med_jhi,icode	diel. locations
. . . .	
. . . .	
med_ilo,med_ihi,med_jlo,med_jhi,icode	
0,0,0,0,0	
$\varepsilon(1),\sigma(1)$	diel. properties
. .	
. .	
$\varepsilon(5),\sigma(5)$	
deb_ilo,deb_ihi,deb_jlo,deb_jhi,icode	debye locations
. . . .	
. . . .	
deb_ilo,deb_ihi,deb_jlo,deb_jhi,icode	
0,0,0,0,0	
$\varepsilon_s(1),\varepsilon_i(1),\sigma(1),t_0(1)$	debye properties
. .	
. .	
$\varepsilon_s(5),\varepsilon_i(5),\sigma(5),t_0(5)$	
lor_ilo,lor_ihi,lor_jlo,lor_jhi,icode	lorentz locations
. . . .	
. . . .	
lor_ilo,lor_ihi,lor_jlo,lor_jhi,icode	
0,0,0,0,0	
$\varepsilon_s(1),\varepsilon_i(1),\omega_o(1),\delta(1)$	lorentz properties
. .	
. .	
$\varepsilon_s(5),\varepsilon_i(5),\omega_0(5),\delta(5)$	
ni	number of time steps
theta	incident angle
dl	lattice spacing

This input format isn't as daunting as it looks. To define blocks of conductor or material the coordinates of opposite corners must be defined. The coordinate of the lower left corner is (xx_ilo,xx_jlo) and the upper right corner is (xx_ihi,xx_jhi), where xx is one of the following:

pec - perfect electric conductor
med - non-dispersive dielectric
deb - debye dielectric
lor - lorentz dielectric

The maximum number of different dielectrics is 15. Five non-dispersive, five debye and five lorentz. The number icode specifies which parameters are used. The zeros are an indicator to the program that the data set is complete for the present material type. The actual material parameters are read after the locations. As mentioned, there can only be five sets of parameters for each dielectric type. If there are less than five, only the number that there are is input. They MUST be in order from 1 to 5. The number of dielectric blocks allowed is determined in ARRAY.F by nm. Note that this can be any number. The limit of five is just for the different *types* of materials.

Thus, it is seen that the input each type of block consists of the location and which number parameter line to use - icode -(except PEC's which do not require any parameter lines). After the row of zeros indicating the end of the current media type, the material parameters (up to five lines) are input.

The number icode specifies which parameters to use for each block of material. It can only be form 1 to 5.

APPENDIX 2

What follows is a source listing of the programs described above. A copy of the programs can be e-mailed to any interested parties by the author. He can be reached at:

Dr. Fred J. German
Texas Instruments, Inc.
P.O. Box 801 M/S 1809
McKinney, TX 75075

Phone: 214-952-3723

```
parameter (nsx=225,nsy=112)
parameter (npec=10,nm=20)
parameter (nd=16)
```

```
      subroutine excite(i,j,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)
c______________________________________________________________________
c
c This subroutine supplies the excitation function for program fdtd
c______________________________________________________________________

      common/excite_com/theta,iclo,ichi,jclo,jchi,dl

      pi = 3.141592654
c     c = 2.997682e8
      c = 3.e8
      w = 2.*pi*5.e9
      etao = 377.
      dtr = pi/180.

      pw = 30000.*dt


      if (ic.eq.1) then
        costh = cos(theta*dtr)
        sinth = sin(theta*dtr)
      end if


c Sinusoidal plane wave propagating in +x direction
c Incident Electric Field
      x = float(i-iclo)*dl
      y = float(j-jclo)*dl
      t = float(ic-1)*dt
      tr = t - (x*costh+y*sinth)/c
      if (tr.ge.0. .and. tr.le.pw) then
        ezinc = sin(w*tr)
c       ezinc = EXP(-(4./pw * (tr-pw/2.))**2.)
      else
        ezinc = 0.0
      end if


c Incident y Directed Magnetic Field
      x = (float(i-iclo)-0.5)*dl
      y = float(j-jclo)*dl
      t = (float(ic)-0.5)*dt
      tr = t - (x*costh+y*sinth)/c
      if (tr.ge.0. .and. tr.le.pw) then
        hyinc1 = -sin(w*tr)*costh/etao
c       hyinc1 = -costh*EXP(-(4./pw * (tr-pw/2.))**2.)/etao
      else
        hyinc1 = 0.0
      end if

      x = (float(i-iclo)+0.5)*dl
      y = float(j-jclo)*dl
      t = (float(ic)-0.5)*dt
      tr = t - (x*costh+y*sinth)/c
      if (tr.ge.0. .and. tr.le.pw) then
        hyinc2 = -sin(w*tr)*costh/etao
c       hyinc2 = -costh*EXP(-(4./pw * (tr-pw/2.))**2.)/etao
      else
        hyinc2 = 0.0
      end if


c Incident x Directed Magnetic Field
      x = float(i-iclo)*dl
      y = (float(j-jclo)-0.5)*dl
```

```
      t = (float(ic)-0.5)*dt
      tr = t - (x*costh+y*sinth)/c
      if (tr.ge.0. .and. tr.le.pw) then
        hxincl = sin(w*tr)*sinth/etao
c       hxincl = sinth*EXP(-(4./pw * (tr-pw/2.))**2.)/etao
      else
        hxincl = 0.0
      end if


      x = float(i-iclo)*dl
      y = (float(j-jclo)+0.5)*dl
      t = (float(ic)-0.5)*dt
      tr = t - (x*costh+y*sinth)/c
      if (tr.ge.0. .and. tr.le.pw) then
        hxinc2 = sin(w*tr)*sinth/etao
c       hxinc2 =  sinth*EXP(-(4./pw * (tr-pw/2.))**2.)/etao
      else
        hxinc2 = 0.0
      end if


      return
      end
```

```
      SUBROUTINE OUTPUT(IC,DT,nx,ny)
C______________________________________________________________________
C
C                              OUTPUT
C OUTPUTS THE DESIRED FIELD DATA FROM FDTD1
C______________________________________________________________________

      INCLUDE 'array.f'
      dimension ez(nsx,nsy),hx(nsx,nsy),hy(nsx,nsy),dz(nsx,nsy)

      COMMON/OUTPUT_COM/ez,hx,hy,dz,ni

      PI = 3.141592654
      eo = 8.854e-12

      IF (IC.EQ.1) THEN
        OPEN(UNIT=25,STATUS='NEW',NAME='ez_1.dat')
        OPEN(UNIT=26,STATUS='NEW',NAME='ez_2.dat')
        OPEN(UNIT=27,STATUS='NEW',NAME='ez_3.dat')
        OPEN(UNIT=28,STATUS='NEW',NAME='ez_4.dat')
      END IF

c
C TEMPORAL OUTPUT AT FIXED LOCATIOZN
        WRITE(25,*) dt*ic,ez(57,2)
        WRITE(26,*) dt*ic,ez(75,2)
        WRITE(27,*) dt*ic,ez(93,2)
        WRITE(28,*) dt*ic,ez(75,17)
        RETURN

      END
```

```
      subroutine build(id)
c_____________________________________________________________________
c This subroutine is a user generated program which sets up the geometry
c of the interaction object(s) within the mesh.
c
c Written by: Dr. Fred J. German
c       Date: February 1992
c_____________________________________________________________________

      include 'array.f'

      dimension id(nsx,nsy),f1(nd),f2(nd),f3(nd),f4(nd),f5(nd)
      dimension ipec(npec,4),im(nm,5),idm(nm,5),ilm(nm,5)
      dimension rmatc(4,nd)

      common/build_com/f1,f2,f3,f4,f5,dt,ipec,im,idm,ilm,
     &        rmatc,nx,ny,kpec,km,kdm,klm

c useful constants
      eo = 8.854e-12


c initialize entire mesh for free space
      do 10 j=1,ny
      do 11 i=1,nx
        id(i,j) = 1
        f1(id(i,j)) = eo
        f2(id(i,j)) = 0.0
 11   continue
 10   continue

c fill conductor and material blocks that are specified as input data

c fill conductors
      do 12 l=1,kpec
      write(*,*) '...conductors ...'
        do 13 j=ipec(l,3),ipec(l,4)
        do 14 i=ipec(l,1),ipec(l,2)
          id(i,j) = 16
 14     continue
 13     continue
 12   continue


c fill non-dispersive dielectrics
      do 15 l=1,km
      write(*,*) 'non-dispersives ...',l
        do 16 j=im(l,3),im(l,4)
        do 17 i=im(l,1),im(l,2)
          id(i,j) = im(l,5) + 1
          f1(id(i,j)) = rmatc(1,id(i,j))*eo
          f2(id(i,j)) = rmatc(2,id(i,j))
 17     continue
 16     continue
 15   continue


c fill debye dispersive dielectrics
      do 18 l=1,kdm
      write(*,*) 'debye - dispersives ...'
        do 19 j=idm(l,3),idm(l,4)
        do 20 i=idm(l,1),idm(l,2)
          id(i,j) = idm(l,5) + 5
          es = rmatc(1,id(i,j))*eo
          ei = rmatc(2,id(i,j))*eo
```

```
            to = rmatc(4,id(i,j))
            f1(id(i,j)) = (dt + 2.*to)/(2.*to*ei + es*dt)
            f2(id(i,j)) = (dt - 2.*to)/(2.*to*ei + es*dt)
            f3(id(i,j)) = (2.*to*ei - es*dt)/(2.*to*ei + es*dt)
            f4(id(i,j)) = rmatc(3,id(i,j))
 20       continue
 19       continue
 18     continue

c fill lorentz dispersive dielectrics
        do 21 l=1,klm
        write(*,*) 'lorentz - dispersives ...'
          do 22 j=ilm(l,3),ilm(l,4)
          do 23 i=ilm(l,1),ilm(l,2)
            id(i,j) = ilm(l,5) + 10
            es = rmatc(1,id(i,j))*eo
            ei = rmatc(2,id(i,j))*eo
            wo = rmatc(3,id(i,j))
            dmp = rmatc(4,id(i,j))
            den = 1./(es*(wo*dt)**2 + 4.*dt*dmp*ei + 2.*ei)
            f1(id(i,j)) = ((wo*dt)**2 + 4.*dmp*dt + 2.)*den
            f2(id(i,j)) = ((wo*dt)**2 - 4.*dmp*dt - 4.)*den
            f3(id(i,j)) = 2.*den
            f4(id(i,j)) = -((wo*dt)**2*es - 4.*dmp*dt*ei - 4.*ei)*den
            f5(id(i,j)) = -2.*ei*den
 23       continue
 22       continue
 21     continue


c***************************************************************************
c In many cases it is easier to fill in material blocks using a few lines
c of code than entering many input data lines in fdtd.in.  One case is the
c dielectric cylinder below because the equation for the cylinder can be
c used to fill in instead of entering the many lines of input data
c required for the cartesian approximation.  The material codes are listed
c below for filling blocks.  Sections of code can be used to fill blocks
c in addition to input data.
c***************************************************************************
c build dielectric bodies
c_________________________________________________________________________
c       * id=2->5  => non-dispersive dielectric
c                   f1 = er*eo
c                   f2 = sigma
c                   f3-f5 not used
c_________________________________________________________________________
c       *  id=6->10 => debye dispersive material
c                   f1 = (dt + 2.*to)/(2.*to*ei + es*dt)
c                   f2 = (dt - 2.*to)/(2.*to*ei + es*dt)
c                   f3 = (2.*to*ei - es*dt)/(2.*to*ei + es*dt)
c                   f4 = sigma
c
c                       to = relaxation time of media
c                       ei = permittivity @ infinite frequency
c                       es = permittivity @ zero frequency (dc)
c_________________________________________________________________________
c       *  id=11->15 => lorentz dispersive material
c                  f1 = ((wo*dt)**2 + 4.*dmp*dt + 2.)*den
c                  f2 = ((wo*dt)**2 - 4.*dmp*dt - 4.)*den
c                  f3 = 2.*den
c                  f4 = -((wo*dt)**2*es - 4.*dmp*dt*ei - 4.*ei))*den
c                  f5 = -2.*ei*den
c  where
c               den = 1./(es*(wo*dt)**2 + 4.*dt*dmp*ei + 2.*ei)
```

```
c
c                       wo = resonant frequency
c                       dmp = damping coefficient
c                       ei = permittivity @ infinite frequency
c                       es = permittivity @ zero frequency (dc)
c_____________________________________________________________
c      *  id = 16 => perfect electric conductor (pec)
c_____________________________________________________________

c Half of Concentric Dielectric Cylinders
c  r1 = 10*dl
c  r2 = 20*dl
        to = 8.1e-12
        eid = 5.5*eo
        esd= 78.2*eo
        eil = eo
        esl = 25.0*eo
        wo = 250.e9
        dmp = 50.e9

        do 120 j=1,25
        do 130 i=50,100
          cyl = sqrt((float(i)-75.)**2 + (float(j)-1.5)**2)

c Fill Outer Ring With Lorentz Material
          if (cyl.le.20. .and. cyl.gt.10.) then
            id(i,j) = 11
            den = 1./(esl*(wo*dt)**2 + 4.*dt*dmp*eil + 2.*eil)
            f1(id(i,j)) = ((wo*dt)**2 + 4.*dmp*dt + 2.)*den
            f2(id(i,j)) = ((wo*dt)**2 - 4.*dmp*dt - 4.)*den
            f3(id(i,j)) = 2.*den
            f4(id(i,j)) = -((wo*dt)**2*esl - 4.*dmp*dt*eil - 4.*eil)*den
            f5(id(i,j)) = -2.*eil*den
          end if

c Fill Inner Core With Debye Material
          if (cyl.le.10.) then
            id(i,j) = 6
            f1(id(i,j)) = (dt + 2.*to)/(2.*to*eid + esd*dt)
            f2(id(i,j)) = (dt - 2.*to)/(2.*to*eid + esd*dt)
            f3(id(i,j)) = (2.*to*eid - esd*dt)/(2.*to*eid + esd*dt)
            f4(id(i,j)) = 0.0
          end if

 130    continue
 120    continue



c lorentz media
c       wo = 4.e16
c       ei = eo
c       es = 2.25*eo
c       dmp = 0.28e16
c
c       den = 1./(es*(wo*dt)**2 + 4.*dt*dmp*ei + 2.*ei)
c       do 201 j=1,jchi-2
c       do 200 i=16,21
c         id(i,j) = 11
c         f1(id(i,j)) = ((wo*dt)**2 + 4.*dmp*dt + 2.)*den
c         f2(id(i,j)) = ((wo*dt)**2 - 4.*dmp*dt - 4.)*den
c         f3(id(i,j)) = 2.*den
c         f4(id(i,j)) = -((wo*dt)**2*es - 4.*dmp*dt*ei - 4.*ei)*den
c         f5(id(i,j)) = -2.*ei*den
c200    continue
c201    continue
```

```
c build perfect electric conductors (PEC)
c    open box from fang
c       do j=1,30
c          id(120,j) = 16
c       end do
c
c       do i=120,180
c          id(i,30) = 16
c       end do

        return
        end
```

```
c
c
c                                        fdtd
c
c This program implements a two dimensional (Hz=0) FDTD algorithm for
c inhomogeneous dispersive space
c
c Written by: Dr. Fred J. German
c       Date: January - August 1992
c

        include 'array.f'

        dimension dz(nx,ny),ez(nx,ny),hx(nx,ny),hy(nx,ny)
        dimension eps(nd),id(nx,ny)
        real mo

        COMMON/OUTPUT_COM/ez,hx,hy,dz
        common/build_com/eps

c define useful constants
        c = 3.e8
        pi = 3.141592654
        mo = 4.e-7*pi
        eo = 8.854e-12
        eta = 377.

c set up geometry in the mesh
        call build(id)

c calculate the maximum allowable time step
        dt = dl/(2.*c)
        hc = dt/(mo*dl)
        dc = dt/dl
        ec = dc/eo
        bc = (1.-c*dc)/(1.+c*dc)

c time loop
        do 20 ic=1,ni
        write(*,*) 'time step = ',ic

c output the desired field data (t = to)
        call output(ic,dt)

c ADVANCE THE H-FIELDS
        do 26 j=1,ny-1
        do 25 i=1,nx-1

c   advance hx (t = to + dt/2)
          if (i.ne.1) hx(i,j) = hx(i,j) - hc*(ez(i,j+1) - ez(i,j))

c   advance hy (t = to + dt/2)
          if (j.ne.1) hy(i,j) = hy(i,j) + hc*(ez(i+1,j) - ez(i,j))
  25    continue
  26    continue

c CORRECT H-FIELDS OUTSIDE TOT/SCAT FIELD BOUNDARY
        do 36 i=iclo,ichi
          call excite(i,jclo,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)
```



```
          hx(i,jclo-1) = hx(i,jclo-1) + hc*ezinc
          call excite(i,jchi,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)

          hx(i,jchi) = hx(i,jchi) - hc*ezinc
```

```
 36     continue

        do 37 j=jclo,jchi
          call excite(iclo,j,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)
          hy(iclo-1,j) = hy(iclo-1,j) - hc*ezinc
          call excite(ichi,j,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)
          hy(ichi,j) = hy(ichi,j) + hc*ezinc
 37     continue


c APPLY ABC's TO SCATTERED ELECTRIC FIELD ON OUTER BOUNDARY
        do 27 i=2,nx-1
          ezt = ez(i,2)
          dz(i,2) = dz(i,2) + dc*(hy(i,2)-hy(i-1,2)-hx(i,2)+hx(i,1))
          ez(i,2) = dz(i,2)/eps(id(i,2))
          ez(i,1) = ezt + bc*(ez(i,1) - ez(i,2))
          if(i.eq.2) ez(1,2) = ezt + bc*(ez(1,2)-ez(i,2))
          if(i.eq.(nx-1)) ez(nx,2) = ezt + bc*(ez(nx,2)-ez(i,2))
          ezt = ez(i,ny-1)
          dz(i,ny-1) = dz(i,ny-1) + dc*(hy(i,ny-1)-hy(i-1,ny-1)
     &                 -hx(i,ny-1)+hx(i,ny-2))
          ez(i,ny-1) = dz(i,ny-1)/eps(id(i,ny-1))
          ez(i,ny) = ezt + bc*(ez(i,ny)-ez(i,ny-1))
          if(i.eq.2) ez(1,ny-1) = ezt + bc*(ez(1,ny-1)-ez(i,ny-1))
          if(i.eq.(nx-1)) ez(nx,ny-1) = ezt + bc*(ez(nx,ny-1)-ez(i,ny-1))
 27     continue

        do 28 j=3,ny-2
          ezt = ez(2,j)
          dz(2,j) = dz(2,j) + dc*(hy(2,j)-hy(1,j)-hx(2,j)+hx(2,j-1))
          ez(2,j) = dz(2,j)/eps(id(2,j))
          ez(1,j) = ezt + bc*(ez(1,j) - ez(2,j))
          ezt = ez(nx-1,j)
          dz(nx-1,j) = dz(nx-1,j) + dc*(hy(nx-1,j)-hy(nx-2,j)
     &                 -hx(nx-1,j)+hx(nx-1,j-1))
          ez(nx-1,j) = dz(nx-1,j)/eps(id(nx-1,j))
          ez(nx,j) = ezt + bc*(ez(nx,j) - ez(nx-1,j))
 28     continue


c UPDATE REMAINING Ez & Dz (t=to + dt)
        do 30 j=3,ny-2
        do 29 i=3,nx-2

c non-dispersive dielectric (id = 1-5)
        if (id(i,j).le.5) then
          dz(i,j) = dz(i,j) +
     &          dc*(hy(i,j) - hy(i-1,j) - hx(i,j) + hx(i,j-1))
          ez(i,j) = dz(i,j)/eps(id(i,j))
        end if
c       end if

c debye dispersive dielectric (id = 6-10)
        if (id(i,j).ge.6 .and. id(i,j).le.10) then
          dz(i,j) = dz(i,j) +
     &          dc*(hy(i,j) - hy(i-1,j) - hx(i,j) + hx(i,j-1))

        end if

c perfect electric conductor (id = 11)
        if (id(i,j).eq.11) then
          dz(i,j) = 0.0
          ez(i,j) = 0.0
        end if
```

```
  29    continue
  30    continue

c CORRECT E-FIELDS ON TOT/SCAT FIELD BOUNDARY
        do 39 i=iclo+1,ichi-1
          call excite(i,jclo,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)
          dz(i,jclo) = dz(i,jclo) + dc*hxinc1
          ez(i,jclo) = dz(i,jclo)/eps(id(i,jclo))
          call excite(i,jchi,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)
          dz(i,jchi) = dz(i,jchi) - dc*hxinc2
          ez(i,jchi) = dz(i,jchi)/eps(id(i,jchi))
 39     continue

        do 40 j=jclo+1,jchi-1
          call excite(iclo,j,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)
          dz(iclo,j) = dz(iclo,j) - dc*hyinc1
          ez(iclo,j) = dz(iclo,j)/eps(id(iclo,j))
          call excite(ichi,j,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)
          dz(ichi,j) = dz(ichi,j) + dc*hyinc2
          ez(ichi,j) = dz(ichi,j)/eps(id(ichi,j))
  40    continue

        call excite(iclo,jclo,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)
        dz(iclo,jclo) = dz(iclo,jclo) + dc*(-hyinc1 + hxinc1)
        ez(iclo,jclo) = dz(iclo,jclo)/eps(id(iclo,jclo))
        call excite(iclo,jchi,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)
        dz(iclo,jchi) = dz(iclo,jchi) - dc*(hyinc1 + hxinc2)
        ez(iclo,jchi) = dz(iclo,jchi)/eps(id(iclo,jchi))
        call excite(ichi,jclo,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)
        dz(ichi,jclo) = dz(ichi,jclo) + dc*(hyinc2 + hxinc1)
        ez(ichi,jclo) = dz(ichi,jclo)/eps(id(ichi,jclo))
        call excite(ichi,jchi,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)
        dz(ichi,jchi) = dz(ichi,jchi) + dc*(hyinc2 - hxinc2)
        ez(ichi,jchi) = dz(ichi,jchi)/eps(id(ichi,jchi))


c ADVANCE THE TIME STEP
  20    continue

        stop
        end
```

```
c______________________________________________________________________
c
c                              fdtd
c______________________________________________________________________
c This program implements a two dimensional (Hz=0) FDTD algorithm for
c inhomogeneous dispersive space w/ a symmetry boundary @ y = 0
c
c Written by: Dr. Fred J. German
c       Date: January - August 1992
c______________________________________________________________________

      include 'array.f'

      dimension dz(nsx,nsy),ez(nsx,nsy),hx(nsx,nsy),hy(nsx,nsy)
      dimension dzsave1(nsx,nsy),dzsave2(nsx,nsy)
      dimension ezsave2(nsx,nsy),ezsave1(nsx,nsy)
      dimension id(nsx,nsy),f1(nd),f2(nd),f3(nd),f4(nd),f5(nd)
      dimension ipec(npec,4),im(nm,5),idm(nm,5),ilm(nm,5)
      dimension rmatc(4,nd),emax1(nsx),emax2(nsx)
      real mo

      common/output_com/ez,hx,hy,dz,ni
      common/build_com/f1,f2,f3,f4,f5,dt,ipec,im,idm,ilm,
     &          rmatc,nx,ny,kpec,km,kdm,klm
      common/excite_com/theta,iclo,ichi,jclo,jchi,dl

c define useful constants
      c = 3.e8
      pi = 3.141592654
      mo = 4.e-7*pi
      eo = 8.854e-12
      eta = 377.

c read in problem data from input file
      open (unit=24,status='old',file='fdtd.in')

c     mesh dimensions
        read(24,*) nx,ny
        write(*,*) nx,ny

c     scattered/total field boundary
        read(24,*) iclo,ichi,jclo,jchi
        write(*,*) iclo,ichi,jclo,jchi


c     conductor locations
        kpec = 0
  1     kpec = kpec + 1
        read(24,*) (ipec(kpec,m),m=1,4)
        write(*,*) (ipec(kpec,m),m=1,4)
        if (ipec(kpec,1)) 2,2,1
  2     kpec = kpec - 1

c     non-dispersive material blocks
        km = 0
        kmax = 0
  3     km = km + 1
        read(24,*) (im(km,m),m=1,5)
        write(*,*) (im(km,m),m=1,5)
        if (im(km,5).gt.kmax) kmax = im(km,5)
        if (im(km,1)) 4,4,3
  4     km = km - 1
        if (kmax.gt.4) then
          write(*,*) '**** TOO MANY NON-DISPERSIVE DIELECTRICS ****'
          stop
        end if
```

```
          do l=1,kmax
            read(24,*) rmatc(1,l+1),rmatc(2,l+1)
            write(*,*) rmatc(1,l+1),rmatc(2,l+1)
          end do

c       debye dispersive material blocks
          kdm = 0
          kmax = 0
  5       kdm = kdm + 1
          read(24,*) (idm(kdm,m),m=1,5)
          write(*,*) (idm(kdm,m),m=1,5)
          if (idm(kdm,5).gt.kmax) kmax = idm(kdm,5)
          if (idm(kdm,1)) 6,6,5
  6       kdm = kdm - 1
          if (kmax.gt.5) then
            write(*,*) '**** TOO MANY DEBYE MATERIALS ****'
            stop
          end if
          do l=6,kmax+5
        write(*,*) 'reading debye material parameters'
            read(24,*) rmatc(1,l),rmatc(2,l),rmatc(3,l),rmatc(4,l)
            write(*,*) rmatc(1,l),rmatc(2,l),rmatc(3,l),rmatc(4,l)
          end do

c       lorentz dispersive material blocks
          klm = 0
          kmax = 0
  7       klm = klm + 1
          read(24,*) (ilm(klm,m),m=1,5)
          write(*,*) (ilm(klm,m),m=1,5)
          if (ilm(klm,5).gt.kmax) kmax = ilm(klm,5)
          if (ilm(klm,1)) 8,8,7
  8       klm = klm - 1
          if (kmax.gt.5) then
            write(*,*) '**** TOO MANY LORENTZ MATERIALS ****'
            stop
          end if
          do l=11,kmax+10
            read(24,*) rmatc(1,l),rmatc(2,l),rmatc(3,l),rmatc(4,l)
            write(*,*) rmatc(1,l),rmatc(2,l),rmatc(3,l),rmatc(4,l)
          end do

c       number of time steps
          read(24,*) ni
          write(*,*) ni
c       angle of incidence
          read(24,*) theta
          write(*,*) theta
          theta = 0.0
c       mesh size
          read(24,*) dl
          write(*,*) dl


c calculate the maximum allowable time step
        dt = dl/(2.*c)
        hc = dt/(mo*dl)
        dc = dt/dl
        ec = dc/eo
        bc = (1.-c*dc)/(1.+c*dc)

c set up geometry in the mesh
        call build(id)


c       do i=50,100
```

```
c       write(21,*)
c         write(21,*) (int(f1(id(i,j))/eo),j=1,25)
c       end do

c time loop
        do 20 ic=1,ni
        write(*,*) 'time step = ',ic

c output the desired field data (t = to)
        call output(ic,dt,nx,ny)

c ADVANCE THE H-FIELDS
        do 26 j=1,ny
        do 25 i=1,nx


c   advance hx (t = to + dt/2)
          if (i.ne.nx) hx(i,j) = hx(i,j) - hc*(ez(i,j+1) - ez(i,j))

c   advance hy (t = to + dt/2)
          if (j.ne.ny) hy(i,j) = hy(i,j) + hc*(ez(i+1,j) - ez(i,j))
 25     continue
 26     continue

c CORRECT H-FIELDS OUTSIDE TOT/SCAT FIELD BOUNDARY
        do 36 i=iclo,ichi
          call excite(i,jchi,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)
          hx(i,jchi) = hx(i,jchi) - hc*ezinc
36      continue

        do 37 j=jclo,jchi
          call excite(iclo,j,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)
          hy(iclo-1,j) = hy(iclo-1,j) - hc*ezinc
          call excite(ichi,j,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)
          hy(ichi,j) = hy(ichi,j) + hc*ezinc
 37     continue


c APPLY ABC's TO SCATTERED ELECTRIC FIELD ON OUTER BOUNDARY
        do 27 i=2,nx-1
          ezt = ez(i,ny-1)
          dz(i,ny-1) = dz(i,ny-1) + dc*(hy(i,ny-1)-hy(i-1,ny-1)
     &                 -hx(i,ny-1)+hx(i,ny-2))
          ez(i,ny-1) = dz(i,ny-1)/f1(id(i,ny-1))
          ez(i,ny) = ezt + bc*(ez(i,ny)-ez(i,ny-1))
          if(i.eq.2) ez(1,ny-1) = ezt + bc*(ez(1,ny-1)-ez(i,ny-1))
          if(i.eq.(nx-1)) ez(nx,ny-1) = ezt + bc*(ez(nx,ny-1)-ez(i,ny-1))
 27     continue

        do 28 j=2,ny-2
          ezt = ez(2,j)
          dz(2,j) = dz(2,j) + dc*(hy(2,j)-hy(1,j)-hx(2,j)+hx(2,j-1))
          ez(2,j) = dz(2,j)/f1(id(2,j))
          ez(1,j) = ezt + bc*(ez(1,j) - ez(2,j))
          ezt = ez(nx-1,j)
          dz(nx-1,j) = dz(nx-1,j) + dc*(hy(nx-1,j)-hy(nx-2,j)
     &                 -hx(nx-1,j)+hx(nx-1,j-1))
          ez(nx-1,j) = dz(nx-1,j)/f1(id(nx-1,j))
          ez(nx,j) = ezt + bc*(ez(nx,j) - ez(nx-1,j))
 28     continue
```



```
c UPDATE REMAINING Ez & Dz (t=to + dt)
        do 30 j=2,ny-2
        do 29 i=3,nx-2
```

```
c save previous electric fields & flux's
        dzsave2(i,j) = dzsave1(i,j)
        dzsave1(i,j) = dz(i,j)
        ezsave2(i,j) = ezsave1(i,j)
        ezsave1(i,j) = ez(i,j)

c non-dispersive dielectric (id = 1-5)
        if (id(i,j).le.5) then
          dz(i,j) = dz(i,j) - dt*f2(id(i,j))*ez(i,j) +
     &          dc*(hy(i,j) - hy(i-1,j) - hx(i,j) + hx(i,j-1))
          ez(i,j) = dz(i,j)/f1(id(i,j))
        end if

c debye dispersive dielectric (id = 6-10)
        if (id(i,j).ge.6 .and. id(i,j).le.10) then
          dz(i,j) = dz(i,j) - dt*f4(id(i,j))*ezsave1(i,j) +
     &          dc*(hy(i,j) - hy(i-1,j) - hx(i,j) + hx(i,j-1))
          ez(i,j) = f1(id(i,j))*dz(i,j) + f2(id(i,j))*dzsave1(i,j)
     &          + f3(id(i,j))*ezsave1(i,j)
        end if

c lorentz dispersive dielectric (id = 11-15)
        if (id(i,j).ge.11 .and. id(i,j).le.15) then
          dz(i,j) = dz(i,j) +
     &          dc*(hy(i,j) - hy(i-1,j) - hx(i,j) + hx(i,j-1))
          ez(i,j) = f1(id(i,j))*dz(i,j) + f2(id(i,j))*dzsave1(i,j)
     &          + f3(id(i,j))*dzsave2(i,j) + f4(id(i,j))*ezsave1(i,j)
     &          + f5(id(i,j))*ezsave2(i,j)
        end if


c perfect electric conductor (id = 16)
        if (id(i,j).eq.16) then
          dz(i,j) = 0.0
          ez(i,j) = 0.0
        end if

  29    continue
  30    continue

c enforce symmetry condition
        do 60 i=1,nx
          dz(i,1) = dz(i,2)
          ez(i,1) = ez(i,2)
  60    continue

c correct e-fields on tot/scat field boundary
        do 39 i=iclo+1,ichi-1
          call excite(i,jchi,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)
          dz(i,jchi) = dz(i,jchi) - dc*hxinc2
          ez(i,jchi) = dz(i,jchi)/f1(id(i,jchi))
 39     continue

        do 40 j=1,jchi-1
          call excite(iclo,j,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)
          dz(iclo,j) = dz(iclo,j) - dc*hyinc1
          ez(iclo,j) = dz(iclo,j)/f1(id(iclo,j))
          call excite(ichi,j,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)
          dz(ichi,j) = dz(ichi,j) + dc*hyinc2
          ez(ichi,j) = dz(ichi,j)/f1(id(ichi,j))
  40    continue

        call excite(iclo,jchi,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)
        dz(iclo,jchi) = dz(iclo,jchi) - dc*(hyinc1 + hxinc2)
        ez(iclo,jchi) = dz(iclo,jchi)/f1(id(iclo,jchi))
```

```
      call excite(ichi,jchi,ic,dt,ezinc,hxinc1,hxinc2,hyinc1,hyinc2)
      dz(ichi,jchi) = dz(ichi,jchi) + dc*(hyinc2 - hxinc2)
      ez(ichi,jchi) = dz(ichi,jchi)/f1(id(ichi,jchi))


c advance the time step
 20   continue



      stop
      end
```