A Common-Format Database for Modeling Strategic Communications

The research described in this report was sponsored by the Joint Staff and was conducted in RAND's National Defense Research Institute, a federally funded research and development center supported by the Office of the Secretary of Defense and the Joint Staff, Contract No. MDA903-90-C-0004.

Library of Congress Cataloging in Publication Data
A common-format database for modeling strategic communications / John H. Craigie . . . [et al.].
p. cm.
"Prepared for the Joint Staff."
"R-4012-JS"
Includes bibliographical references.
ISBN 0-8330-1203-7
UB212.C653 1992
355.3'3041'0285—dc20 91-40698
CIP

RAND is a nonprofit institution that seeks to improve public policy through research and analysis. Publications of RAND do not necessarily reflect the opinions or policies of the sponsors of RAND research.

Published 1992 by RAND
1700 Main Street, P.O. Box 2138, Santa Monica, CA 90407-2138
A Common-Format Database for Modeling Strategic Communications


Prepared for the Joint Staff

RAND

Approved for public release; distribution unlimited
A Common-Format Database for Modeling Strategic Communications


Prepared for the Joint Staff
PREFACE

This report describes the results of a study in which RAND researchers evaluated the viability of a common-format strategic command, control, and communications (C3) database by developing and assessing a prototype. The report discusses the content of the database and the development of its architecture and recommends follow-on work (which would primarily be done by other organizations). Also included is a compendium of useful information concerning strategic C3 models (for network functional assessment and communications engineering).

The principal audience for this work consists of C3 analysts in the following organizations:

- The Joint Staff (J-6, J-6, J-8, and J-36)
- The Assistant Secretary of Defense (Program Analysis and Evaluation)
- The Office of the Assistant Secretary of Defense (Command, Control, Communications, and Intelligence)
- The Defense Information Service Agency
- The Nuclear Command and Control Systems Support Staff
- The Commander in Chief, Strategic Air Command/JSTPS
- The Commander in Chief, Air Force Space Command
- The Air Force Center for Studies and Analysis
- NOP-941D/APL, Naval Ocean Systems Center
- The Defense Nuclear Agency
- The Defense Intelligence Agency

This report was prepared for the C4 Systems Directorate, Joint Staff. The research was carried out in the Applied Science and Technology Program of the National Defense Research Institute, RAND’s federally funded research and development center sponsored by the Office of the Secretary of Defense and the Joint Staff.
SUMMARY

Several years ago, the Joint Staff noted significant discrepancies between the results of command, control, and communications (C³) network functional assessments made using different models, even though the networks were similar and the databases appeared to be identical. The consensus arising from a Joint Staff workshop convened in 1984 to resolve these discrepancies was that (1) the databases were not, in fact, identical and (2) development of common databases and methods could minimize differences between the functional assessments and aid in understanding any remaining differences. The strategic C³ community has made it clear that a baseline, standard common-format database is necessary if they are to meet the requirements of current DoD directives expeditiously and efficiently.

This report describes the results of a project that the Joint Staff initiated to develop and test the usefulness of a baseline common-format database. In conjunction with the Defense Communications Agency (now the Defense Information Systems Agency), the Defense Nuclear Agency, the Air Force Center for Studies and Analysis, and other government agencies, the Joint Staff also conducted a validation and verification effort to ensure that the models’ calculation methods are both correct and compatible.

Although the strategic C³ analysis community is not large, our first survey identified some forty network functional and communications engineering models. And that first survey was incomplete. Furthermore, new models are being introduced all the time. Each organization has its own view of how the strategic C³ network operates; these different views translated into different models of the network, each with unique input data requirements. Diagnosing discrepancies is difficult because these models and databases are seldom merged, compared, or subjected to any review or validation, except by the individual users and possibly their consumers.

Even when the database is on line, differences will persist. This planning study and prototype development effort is intended to provide an expeditious way to resolve technical differences when they occur. The Standard Strategic Common Format Database would provide a valid baseline description of the current strategic C³ network for use by specific government organizations. These organizations would all benefit from the development, maintenance, and use of this database.

Project Approach

We conducted on-site surveys of strategic C³ analysts to determine the nature of their analyses, the analysis tools in use, and the available computer environments. We defined the data required for a standard strategic C³ database and identified data sources. The bulk of our effort involved developing a common-format database approach, including prototype software, and using this common format for the representation of communications procedures. Finally, we loaded data in this common format into SIMSTAR-85A and ran a test case.

While performing these tasks, we cohosted with the Joint Staff a series of workshops, bringing together all of the strategic C³ analysis organizations and most of the key personnel.
Information Sources

To learn about the strategic C³ analysis community and the tools of the trade, we first conducted a broad survey, visiting the analysts and obtaining documentation for the computer models in use. In developing the common-format database approach, we canvassed the computer science literature and discussed the problem with specialists in the computer science field and with the strategic C³ analysis community to establish their needs in light of individual working conditions and analysis requirements.

Most of the information pertaining to the operations of strategic C³ systems—and the database we seek to define and develop—was, because of its sensitivity, very difficult to obtain. We were able, however, to identify two classes of data sources:

- Documentation by the Joint Staff, Unified and Specified Commands, Services, and defense agencies. The basic sources are documents that describe strategic C³ systems and operations. They constitute top-level input data to the electronic databases for the models of interest. There are also technical data sources that describe systems and mission requirements in sufficient detail to allow their performance to be quantified.

- Electronic Databases. The most efficiently assimilated sources for large amounts of data are electronic databases, usually those created for running the relevant models. These electronic databases represent the outputs of preprocessors that define the specific data used to run network or engineering models. Electronic databases are very useful when one needs to acquire a large database under time and resource constraints, or when one wishes to use approved and quality-controlled data. Since they are highly processed, however, they must be fully understood and used with caution.

Since strategic C³ system operation during nuclear war conditions must be evaluated, attack scenarios are required. The analyst must obtain and/or develop scenarios that reflect the various contingencies or situations in which the systems must perform to support national policy. The analyst requires a threat database to complete and/or modify the attack scenario to meet the specific needs of the analysis task at hand. For example, master planning analyses involving new systems or procedures must allow those new options to be attacked in a manner consistent with the baseline attack.

An important aspect of this database is its security classification, which can vary from unclassified to Top Secret, and occasionally Special Access Required. The threat data usually come from Special Compartmented Information sources. The result is a highly but variably classified and very sensitive database that has very significant database management ramifications for this effort.

RECOMMENDATIONS

Although the strategic C³ analysis community would clearly benefit from the development, maintenance, and use of a standard, common-format database, the main effort we describe here would cost over $1M; even an austere effort would cost at least half that figure. Today's funding environment suggests instead a more deliberate development effort. In contrast with a hardware production program, proceeding at a slower pace will actually be more eco-
nomical than initiating an immediate, full-scale effort. The Department of Defense can keep its options open with the following building-block approach:

- **Block 1**: Document and promulgate an approved *standard* strategic C³ database (this step does not depend on the development of a *common format*).
- **Block 2**: Develop *specifications* for the procurement of the common-format database, support software, and documentation.
- **Block 3**: Develop an *austere* version of the common-format database and support software development.
- **Block 4**: Develop the *Standard Strategic Common-Format Database* to include a software support and documentation package.

Blocks 2, 3, and 4 need not be completed for Block 1 to be useful, nor would initiation of either Block 1 or 2 require large start-up costs. Each block will build upon its predecessor; downstream blocks can be implemented, redirected, delayed, or dropped based on the best information available at the time.

The benefits of this action will accrue to the Office of the Secretary of Defense; the Joint Staff; the Commander in Chief, Strategic Air Command; the Joint Strategic Target Planning Staff; the Defense Information Systems Agency; the Defense Nuclear Agency; the Office of the Chief of Naval Operations; and the Air Force Center for Studies and Analysis. The burden should likewise be shared. We suggest initiating Block 1 immediately and initiating informal efforts by the Joint Staff Strategic C³ Working Group to develop a preliminary Block 2 specification.
ACKNOWLEDGMENTS

A large number of people have contributed to this project. The informal working group that evolved out of our workshops has played a significant role in shaping our concepts concerning database content, organization, and format. Many have contributed by responding to surveys, speaking at the workshops, and commenting on the five working drafts that preceded this report. It is difficult to select an appropriate number of names from the more than one hundred participants, but the following are among those who have had a major impact on this study:

John Bird       Y. S. Fu       Richard Paul
Richard Buckner Burt Gambill Balram Prasad
Glenn Butcher   Robert Graney Royden Rutherford
Donald Butts    David Hunsuck Eugene Schmidt
Eugene Chen     Francis J. Kelly Howard Seguine
Bronius Cikotas David Kitterman Babu Singaraju
Patrick Crowley Raymond Lunnen, Jr. Bradley Stubbs
Stanley Czajowski Mary McCully Bruce Thiemann
Lauren Decker   Allyson Miller Anthony Virgilio
Jerry Ferguson  Lawrence Miller Gary Zank
Edward Field, Jr. David Niemoller

We are very grateful to Willis Ware and Gaylord Huth for their thoughtful review and insights. The authors, of course, are responsible for any errors of fact, concept, or judgement.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Chapter Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Preface</td>
<td>iii</td>
</tr>
<tr>
<td></td>
<td>Summary</td>
<td>v</td>
</tr>
<tr>
<td></td>
<td>Acknowledgments</td>
<td>ix</td>
</tr>
<tr>
<td></td>
<td>Figures</td>
<td>xiii</td>
</tr>
<tr>
<td></td>
<td>Tables</td>
<td>xv</td>
</tr>
<tr>
<td></td>
<td>Abbreviations</td>
<td>xvii</td>
</tr>
<tr>
<td>1.</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Objective—The Need for a Standard, Formatted Database</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Approach</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Central Findings from the Project</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Purpose and Contents of This Report</td>
<td>4</td>
</tr>
<tr>
<td>2.</td>
<td>C³ Systems and Methods of Analysis</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Analysis of Systems Operations</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Functional Analysis of Networks</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Engineering Analysis of Communications Links</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>Recommended Code Support</td>
<td>30</td>
</tr>
<tr>
<td>3.</td>
<td>The Strategic C³ Database</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>The SSCF Database Concept</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>Database Content—System Description</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>Procedures and Operational Data</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>Environment</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>Prototype Database Representation</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>Which Language to Use for Development</td>
<td>58</td>
</tr>
<tr>
<td>4.</td>
<td>Prototype Common-Format Database—Design Issues</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>Database Architecture Development</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>Database Administration—Technical Implementation</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>Keyword Language Concept for C³</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>Software Tools and Data Set Production</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>The Database Language</td>
<td>95</td>
</tr>
<tr>
<td>5.</td>
<td>Support Software for Prototype Common-Format Database</td>
<td>104</td>
</tr>
<tr>
<td></td>
<td>Keyword Glossary Manager</td>
<td>104</td>
</tr>
<tr>
<td></td>
<td>Database Manager</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td>Data-Set Translators</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td>Report Generators</td>
<td>114</td>
</tr>
<tr>
<td></td>
<td>Reference Data—The REF_DATA</td>
<td>116</td>
</tr>
<tr>
<td></td>
<td>Development Cost Estimates</td>
<td>117</td>
</tr>
</tbody>
</table>
6. PROTOTYPE COMMON-FORMAT DATABASE RECORDS FOR PROCEDURES ........................................ 121
   Network and Scenario ........................................ 121
   Fixed Command Center ...................................... 124
   Airborne Command Post .................................... 128
   Fixed Relay Station ........................................ 130
   Command Center Succession ............................... 130
   Other Strategic Database Elements ....................... 133

7. DEMONSTRATING THE COMMON-FORMAT DATABASE .................. 135
   Demonstration Objectives ................................ 135
   Test Case C8 System ...................................... 136
   Test Case Representation in Common Format ............. 141
   Conversion From Common Format to SIMSTAR Format .... 156

8. CONCLUSIONS AND RECOMMENDATIONS .............................. 167
   Conclusion 1: Concept ..................................... 167
   Conclusion 2: Implementation and Control ................ 167
   Conclusion 3: Organization and Content ................ 168
   Conclusion 4: Format ...................................... 168
   Conclusion 5: Development Effort ......................... 169
   Conclusion 6: Applications ................................ 169
   Recommendations .......................................... 170

Appendix
A. PROJECT ADMINISTRATION ................................... 173
B. A FORMAL SYNTAX FOR THE SSCP DATABASE ................. 183

REFERENCES .................................................. 187
# FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1. Control of Strategic Nuclear Forces</td>
<td>6</td>
</tr>
<tr>
<td>2.2. General Nuclear War Information Flow</td>
<td>7</td>
</tr>
<tr>
<td>2.3. Strategic C3 Engineering and Functional Analysis</td>
<td>8</td>
</tr>
<tr>
<td>2.4. SCAP Computer Analysis System Overview</td>
<td>18</td>
</tr>
<tr>
<td>2.5. Frequency Band Coverage of Codes</td>
<td>27</td>
</tr>
<tr>
<td>3.1. Extraction of a Data Set from the Database</td>
<td>38</td>
</tr>
<tr>
<td>3.2. SSCF Database Development Using &quot;Raw&quot; Inputs From Model B and External Data Sources</td>
<td>38</td>
</tr>
<tr>
<td>4.1. The Database Utilization Processes</td>
<td>67</td>
</tr>
<tr>
<td>4.2. Presentation and Representation</td>
<td>69</td>
</tr>
<tr>
<td>4.3. Overview of Database Administration</td>
<td>71</td>
</tr>
<tr>
<td>4.4. The SSCP Database Content Verification</td>
<td>72</td>
</tr>
<tr>
<td>4.5. Database Options</td>
<td>73</td>
</tr>
<tr>
<td>4.6. Prototype STRATCAM Keyword Format</td>
<td>79</td>
</tr>
<tr>
<td>4.7. Data Flows for Scenario Language</td>
<td>80</td>
</tr>
<tr>
<td>4.8. WEDCOM Example Exchange Format</td>
<td>84</td>
</tr>
<tr>
<td>4.9. SIMSTAR Format (Fixed Field, Ordered-Record Format)</td>
<td>92</td>
</tr>
<tr>
<td>4.10. Data-Set Building Concept</td>
<td>93</td>
</tr>
<tr>
<td>5.1. Database Manager</td>
<td>107</td>
</tr>
<tr>
<td>5.2. Generic Database Manager Screen</td>
<td>109</td>
</tr>
<tr>
<td>5.3. Data-Set Translator Structure</td>
<td>112</td>
</tr>
<tr>
<td>5.4. Report Generator Structure</td>
<td>115</td>
</tr>
<tr>
<td>6.1. Example Network and Key Nodes</td>
<td>123</td>
</tr>
<tr>
<td>6.2. Database Records for Command Center 1</td>
<td>125</td>
</tr>
<tr>
<td>6.3. Database Records for Command Center 2</td>
<td>129</td>
</tr>
<tr>
<td>6.4. Data Records for Relay 1</td>
<td>131</td>
</tr>
<tr>
<td>6.5. Database Records for Command Center Succession</td>
<td>132</td>
</tr>
<tr>
<td>6.6. Miscellaneous Operational and Technical Node Data</td>
<td>134</td>
</tr>
<tr>
<td>7.1. Example Test-Case Wiring Diagram</td>
<td>137</td>
</tr>
<tr>
<td>7.2. Test-Case Data for the NEACP Node</td>
<td>141</td>
</tr>
<tr>
<td>7.3. Test-Case Data for the SACHF (HF Net Control) Node</td>
<td>142</td>
</tr>
<tr>
<td>7.4. Test-Case Data for HF Relays 1, 2, and 3</td>
<td>143</td>
</tr>
<tr>
<td>7.5. Test-Case Data for the Jim Creek Node</td>
<td>143</td>
</tr>
<tr>
<td>7.6. Test-Case Data for the TACAMO Node</td>
<td>144</td>
</tr>
<tr>
<td>7.7. Test-Case Data for the Grand Forks ICBM Wing Node</td>
<td>145</td>
</tr>
<tr>
<td>7.8. Test-Case Data for the SSBN Atlantic Ocean Node</td>
<td>145</td>
</tr>
<tr>
<td>7.9. Database Records for Nodes: Part 1 of the Test-Case Database</td>
<td>146</td>
</tr>
<tr>
<td>7.10. Database Records for Transmitters: Part 2 of Database Equipment Description</td>
<td>150</td>
</tr>
<tr>
<td>7.11. Database Records for Receivers: Part 2 of Database Equipment Description</td>
<td>151</td>
</tr>
<tr>
<td>7.12. Database Records for NEACP</td>
<td>153</td>
</tr>
<tr>
<td>7.13. Database Records for SACABN</td>
<td>153</td>
</tr>
<tr>
<td>7.14. Database Records for JIMCRK</td>
<td>154</td>
</tr>
<tr>
<td>7.15. Database Records for HFRLY1</td>
<td>154</td>
</tr>
<tr>
<td>7.16. Database Records for HFRLY2</td>
<td>154</td>
</tr>
<tr>
<td>7.17. Database Records for HFRLY3</td>
<td>155</td>
</tr>
<tr>
<td>7.18. Database Records for TACWL (TACWL + TWLLF)</td>
<td>155</td>
</tr>
</tbody>
</table>
7.19. Database Records for GNDFKS .................................................. 158
7.20. Database Records for SSBNL .................................................. 157
7.22. Flow Diagram of the Extended Uniform Distribution Approximation . 161
7.23. Flow Diagram of the Shunt Distribution Approximation ............... 162
7.24. SIMSTAR 84A Representation of the SACABN-Like Node via Three Pseudo Nodes (sacr, sachf, and saclf) .............................................. 165
7.25. SIMSTAR 85A Representation of the TACAMO-Like Node via Two Pseudo Nodes (tacr and tact) .............................................. 165
TABLES

2.1. Threat/Attack/Weapons Effects Codes/Databases ........................................ 10
2.2. Typical Analysis Activities of Commands and Military Services .................. 11
2.3. C³ Network Codes ......................................................................................... 13
2.4. Link Codes Summary ..................................................................................... 20
2.5. Codes Contributing to Support of SIMSTAR, WAAM, and SCAP/NSCS .......... 28
2.6. Recommended Code Compatibility with SSCF Database ............................. 30
3.1. HF Communication Input Requirements for IONCAP and SIMSTAR ............ 41
3.2. Classes of Procedural and Operational Information ..................................... 43
3.3. Communication Actions ................................................................................ 47
3.4. Input Data Requirements for SIMBAL II and WEFCOM ............................ 56
4.1. WEDCOM Data Set and C³ Database Translation ......................................... 81
4.2. WEDCOM Keyword Glossary .................................................................... 82
4.3. Keyword Glossary ....................................................................................... 98
5.1. Lines of Code Needed for Major Support Software Tools .......................... 118
5.2. Effort Required for Keyword Glossary Development ................................. 119
6.1. Example Keyword Definitions ..................................................................... 122
7.1. Test-Case Node Mnemonics ......................................................................... 138
7.2. Test-Case Communications Equipment ...................................................... 139
7.3. Intraneo Process Description Specification ................................................ 140
7.4. Proliferation of Receiver and Transmitter Entries in Translating from
    Common Format to SIMSTAR-85A ................................................................. 163
A.1. Strategic C³ Working Group Organization and Membership ...................... 177
B.1. Primitive Vocabulary .................................................................................... 183
B.2. Database Grammar ....................................................................................... 184
## ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABNCP</td>
<td>airborne command post</td>
</tr>
<tr>
<td>ADI</td>
<td>Air Defense Initiative</td>
</tr>
<tr>
<td>AF</td>
<td>Air Force</td>
</tr>
<tr>
<td>AFCSA</td>
<td>Air Force Center for Studies and Analysis</td>
</tr>
<tr>
<td>AFSAT</td>
<td>Air Force Satellite</td>
</tr>
<tr>
<td>AFSPACECOM</td>
<td>Air Force Space Command</td>
</tr>
<tr>
<td>AFWAL, AFWL</td>
<td>Air Force Wright Laboratory</td>
</tr>
<tr>
<td>APL</td>
<td>Applied Physics Laboratory</td>
</tr>
<tr>
<td>ASAT</td>
<td>antisatellite</td>
</tr>
<tr>
<td>ATTACK</td>
<td>code</td>
</tr>
<tr>
<td>AUTODIN</td>
<td>Automatic Digital Information Network</td>
</tr>
<tr>
<td>AUTOVON</td>
<td>Automatic Voice Network</td>
</tr>
<tr>
<td>AWAA</td>
<td>Attack Warning/Attack Assessment</td>
</tr>
<tr>
<td>BMO</td>
<td>Ballistic Missile Office</td>
</tr>
<tr>
<td>C³</td>
<td>command and control communications; command, control,</td>
</tr>
<tr>
<td></td>
<td>and communications</td>
</tr>
<tr>
<td>CS²I</td>
<td>command, control, communications, and intelligence</td>
</tr>
<tr>
<td>CCIR</td>
<td>International Radio Consultations Committee</td>
</tr>
<tr>
<td>CINC</td>
<td>commander in chief</td>
</tr>
<tr>
<td>CINCLANT</td>
<td>Commander in Chief, Atlantic Fleet</td>
</tr>
<tr>
<td>CINCPAC</td>
<td>Commander in Chief, Pacific Fleet</td>
</tr>
<tr>
<td>CINCSAC</td>
<td>Commander in Chief, Strategic Air Command</td>
</tr>
<tr>
<td>COMSEC</td>
<td>communications security</td>
</tr>
<tr>
<td>CSC</td>
<td>Communications Science Corporation</td>
</tr>
<tr>
<td>CSCI</td>
<td>computer software configuration item</td>
</tr>
<tr>
<td>DASIA</td>
<td>DoD Nuclear Information and Analysis Center</td>
</tr>
<tr>
<td>DBMS</td>
<td>database management system</td>
</tr>
<tr>
<td>DCA</td>
<td>Defense Communications Agency (now DISA)</td>
</tr>
<tr>
<td>DETEC</td>
<td>a C³ network code</td>
</tr>
<tr>
<td>DIA</td>
<td>Defense Intelligence Agency</td>
</tr>
<tr>
<td>DISA</td>
<td>Defense Information Systems Agency</td>
</tr>
<tr>
<td>DMMS</td>
<td>a C³ network code</td>
</tr>
<tr>
<td>DNA</td>
<td>Defense Nuclear Agency</td>
</tr>
<tr>
<td>DoD</td>
<td>Department of Defense</td>
</tr>
<tr>
<td>DSCS</td>
<td>Defense Satellite Communications System</td>
</tr>
<tr>
<td>DSSO</td>
<td>Defense System Support Organization</td>
</tr>
<tr>
<td>EAM</td>
<td>emergency action message</td>
</tr>
<tr>
<td>EAP</td>
<td>emergency action plan</td>
</tr>
<tr>
<td>EDAC</td>
<td>error detection and correction</td>
</tr>
<tr>
<td>EDB</td>
<td>electronic database</td>
</tr>
<tr>
<td>EHF</td>
<td>extra high frequency</td>
</tr>
<tr>
<td>ELF</td>
<td>extra low frequency</td>
</tr>
<tr>
<td>ELF PCA</td>
<td>ELF Propagation Computational Aid</td>
</tr>
<tr>
<td>ESD</td>
<td>Electronic Systems Division</td>
</tr>
<tr>
<td>EUR</td>
<td>Europe</td>
</tr>
<tr>
<td>FCC</td>
<td>Federal Communications Commission</td>
</tr>
<tr>
<td>FIFO</td>
<td>first in, first out</td>
</tr>
<tr>
<td>FLTSAT</td>
<td>Fleet Satellite (Navy)</td>
</tr>
<tr>
<td>FTD</td>
<td>Foreign Technology Division</td>
</tr>
<tr>
<td>GSP</td>
<td>General Strike Plan</td>
</tr>
<tr>
<td>GTE</td>
<td>General Telephone and Electric</td>
</tr>
</tbody>
</table>
GWEN  Ground Wave Emergency Network
HAB  high-altitude burst
HEMP  high-altitude electromagnetic pulse
HF  high frequency
HQ  headquarters
ICBM  intercontinental ballistic missile
ITS  Institute of Telecommunications Sciences
IWG  intelligence working group
JCSAN  Joint Chiefs of Staff Alert Network
JHU  Johns Hopkins University
JRAD  Joint Resources Assessment Database
JSTPS  Joint Strategic Target Planning Staff
LANL  Los Alamos National Laboratory
LANT  Atlantic
LF  low frequency
LOS  line of sight
MEECN  Minimum Essential Emergency Communications Network
MILSTAR  Military Strategic and Tactical Relay System
MINEFIELD  C$ network code
NATO  North Atlantic Treaty Organization
NCA  National Command Authority
NCMC  NORAD Cheyenne Mountain Complex
NEACP  National Emergency Airborne Command Post
NETSIM  C$ network code
NMCC  National Military Command Center
NMCS  National Military Command System
NORAD  North American Air Defense
NOSC  Naval Ocean Systems Center
NRA  Nuclear Regulatory Agency
NRL  Naval Research Laboratories
NSCS  Navy Strategic Communications Simulator
NSNF  nonspecific nuclear forces
NTIA  National Telecommunications and Information Administration
NUCOM  Nuclear Effects on Joint Force Communications
NWEM  Nuclear Weapons Effects Model
OASD  Office of the Assistant Secretary of Defense
OJCS  Office of the Joint Chiefs of Staff
OPAREA  operational area
OPORDER  operational order
OSD  Office of the Secretary of Defense
OTH  over the horizon
PA&E  Program Analysis and Evaluation
PAC  Pacific
PACCS  Post-Attack Command and Control System
PNAC  C$ network code
PPB  Planning, Programming, and Budgeting
PPBS  Planning, Programming, and Budgeting System
PRPSIM  code
RANC  radar absorption and noise clutter
RANC IV  a DNA code
RISOP  Red Integrated Strategic Operations Plan
SAC  Strategic Air Command
SAIC  Science Applications International Corporation
SCAMP  SAC Command and Control Architecture Master Plan
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCAP</td>
<td>small communications augmentation package code</td>
</tr>
<tr>
<td>SCENARIO</td>
<td>special compartmentalized information code</td>
</tr>
<tr>
<td>SCI</td>
<td>Strategic Defense Initiative</td>
</tr>
<tr>
<td>SDI</td>
<td>systems engineering and technical assistance</td>
</tr>
<tr>
<td>SETA</td>
<td>super high frequencies</td>
</tr>
<tr>
<td>SIDAC</td>
<td>Single Integrated Damage Analysis Capability code</td>
</tr>
<tr>
<td>SIMBAL</td>
<td>Simulation of Multiple Bursts and Links code</td>
</tr>
<tr>
<td>SIMBIP</td>
<td>code</td>
</tr>
<tr>
<td>SIMSTAR</td>
<td>code</td>
</tr>
<tr>
<td>SINBAC</td>
<td>code</td>
</tr>
<tr>
<td>SIOP</td>
<td>Single Integrated Operating Plan code</td>
</tr>
<tr>
<td>SKYMAP</td>
<td>submarine launched ballistic missile</td>
</tr>
<tr>
<td>SLBM</td>
<td>Space and Warfare Systems Command</td>
</tr>
<tr>
<td>SPAWAR</td>
<td>Stanford Research Institute (needs International)</td>
</tr>
<tr>
<td>SRI</td>
<td>fleet ballistic missile submarine (nuclear powered)</td>
</tr>
<tr>
<td>SSBN</td>
<td>Strategic Standard Common-Format code</td>
</tr>
<tr>
<td>STRIKE</td>
<td>six-year defense program</td>
</tr>
<tr>
<td>SYDP</td>
<td>take charge and move out</td>
</tr>
<tr>
<td>TACAMO</td>
<td>Threat Assessment Model</td>
</tr>
<tr>
<td>TAM</td>
<td>tactical warning and attack assessment</td>
</tr>
<tr>
<td>TW/AA</td>
<td>very high frequency</td>
</tr>
<tr>
<td>VHF</td>
<td>very low frequency</td>
</tr>
<tr>
<td>VLF</td>
<td>vulnerability number</td>
</tr>
<tr>
<td>WAAM</td>
<td>WWMCCS Allocation and Assessment Model</td>
</tr>
<tr>
<td>WAAMAG</td>
<td>WAAM attack generator</td>
</tr>
<tr>
<td>WECOM</td>
<td>Weapons Effects on Communications</td>
</tr>
<tr>
<td>WEDCOM</td>
<td>Weapons Effects on D-Region Communications</td>
</tr>
<tr>
<td>WEFCOM</td>
<td>Weapons Effects on F-Region Communications</td>
</tr>
<tr>
<td>WESCOM</td>
<td>Weapons Effects on Satellite Communications</td>
</tr>
<tr>
<td>WWMCCS</td>
<td>Worldwide Military Command and Control System</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

RAND has been supporting the Command, Control, and Communications Systems Directorate, Joint Staff/J5, in a project to investigate the feasibility and structure of a strategic command and control communications (C3) database—a common, authoritative, documented source of information to support selected C3 analysis efforts within the Department of Defense. We will refer to this product as the Standard Strategic Common-Format (SSCF) Database. This report presents the results of this project.

OBJECTIVE—THE NEED FOR A STANDARD, FORMATTED DATABASE

The strategic C3 community uses many network functional and communications engineering models. These models and their supporting databases are seldom compared or subjected to review or validation except by their users and occasionally their consumers. Prior to the initiation of this project, the Joint Staff noted discrepancies between the results of functional assessments of similar networks using different models, but ostensibly the same database. The Joint Staff convened a workshop [1] to find ways to compare methodologies and models to resolve differences in results. The consensus was that differences could be minimized and more easily understood if model input was based on common data. Accordingly, the Joint Staff initiated this project to develop a prototype strategic C3 database and a method for its definition, control, and dissemination. Also, the Joint Staff, the Defense Communication Agency (now the Defense Information System Agency), the Defense Nuclear Agency, the Air Force Center for Studies and Analysis, and other government organizations have conducted the validation and verification of several important models, which was another recommendation of the Joint Staff workshop.

The basic concept provides two things that are new:

- A standard strategic C3 database, an approved set of information designated and centrally controlled by the Joint Staff to serve as a baseline for analyses by the commanders in chief (CINC’s), Services, and agencies; and
- A common-format version of this standard strategic database, including support software and documentation.

The combination of these new items is the SSCF Database. It has not been clear until now that a common-format database that the various Department of Defense strategic C3 analysis entities could all use would be feasible. The purpose of this prototype effort has been to clearly define the problem, determine feasibility, and scope the effort involved.

1In this report, C3 will stand for command and control communications, meaning communications in support of command and control. C3 can also stand for command, control, and communications—three nouns rather than two adjectives and a noun—which is the meaning in the name of the Joint Staff organization for whom this study was performed. Both uses are correct. When we refer to strategic C3, we mean C3 in support of the nuclear triad.
A standard strategic C³ database effort is important for several reasons:

1. A validated standard description of the vital processes governing the control of nuclear weapons has long been a needed, but lacking, aspect of the Worldwide Military Command and Control System (WWMCCS) Evaluation Program [1]. Use of a standard, validated database could enhance current analyses supporting the Functional Analysis and Consolidation Review Panel; Joint actions; Service actions and recommendations; CINC programs and recommendations; operational plans; and Planning, Programming, and Budgeting System recommendations. The specificity and lack of ambiguity of a standard database should improve both the advocacy and accountability aspects of the WWMCCS Evaluation Program.

2. In the current environment, however, discrepancies will again arise that will require resolution. The use of a standard database could reduce the frequency, and probably the magnitude, of unintended differences between the analyses of such organizations the Joint Staff, the Office of the Assistant Secretary of a Defense, the Defense Information System Agency, the Strategic Air Command/JSTPS, NOP-941, the Air Force Center for Studies and Analysis, and others.

3. In the event of significant unintended differences, the combination of the SSCP database² and the Joint Staff Strategic C³ Working Group (which is a natural outgrowth of the SSCP Database effort) will allow for better diagnostic capabilities and more expeditious correction of the anomalies that will occur.

4. Dissemination of the SSCP database among the Working Group can also provide enough cost savings to offset most, if not all, of the costs of developing, maintaining, and disseminating the Database.

APPRAACH

The project has consisted of several tasks:

1. Canvassing strategic C³ analysts to determine the nature of their analyses, the analysis tools in use, and the available computer environments;

2. Defining the data required for a standard strategic C³ database and identifying data sources;

3. Developing a common-format database approach, including some prototype software;

4. Applying the common-format database approach to the representation of communications procedures; and

5. Loading data represented in this common format into SIMSTAR-85A and running a test case.

²If the terms "database" and "data set" are initial capped, they refer to the SSCP Database or a part of it, which is called a Data set. Lower case will be used for generic use of these terms.
We conducted a survey of strategic C³ models and databases used by or for the DoD. The initial survey involved identifying the most relevant codes³ appropriate for the use of DoD strategic C³ analysts. We identified the data fields associated with the C³ codes, e.g., bandwidth, data rate, modulation, and antenna gain. Since the tools (analytical programs, computers, and operating systems) the various groups use changed during the course of our study, we updated this survey.

We compared the data fields identified in the survey and made note of hard-wired data (e.g., antenna patterns) that are internal to some codes, special load-time input files (e.g., the CCIR atmospheric noise data), input logical branches (i.e., where inputs may redefine subsequent entries), alternative data specifications (e.g., different choices of orbital elements), and differences in modeling level of detail. We examined ways to capture electronically, merge, and organize the databases the C³ analysis organizations use to permit a more coherent database set. We examined the various interfaces between the C³ programs (which currently access their own separate subsets of data) and an integrated database. We identified ways to efficiently disseminate and control the database (for both technical and security purposes).

We constructed a prototype common-format database to demonstrate the feasibility of interfacing the various codes to the common format. Both technical and practical issues were addressed in this formulation. The response to this first demonstration was essentially that we had accomplished the "easy part"; i.e., the facilities and hardware technical parameters. To completely establish the feasibility and utility of our common-format approach, we converted the much less intuitive, much more ambiguous procedural information into common format.

CENTRAL FINDINGS FROM THE PROJECT

We found that there is a relatively small, but important community of specialists in strategic C³ analysis that could benefit from and contribute to a standard database. This database should be centrally controlled by or for the Joint Staff. For ease, reliability, and fidelity of transmission and use, a common format for the database would be desirable, but not required. We developed a prototype of this common format to the extent required to establish its feasibility and to roughly scope the effort required to develop it. Based on these findings, we believe the concept is not only feasible, but viable, and suggest that the Joint Staff incorporate aspects of this concept into current programs.

We estimated that the development and documentation of the support software would require from $800K to $1200K. The conversion of the strategic communications database itself could probably be accomplished for an additional $300K to $400K. At this cost and in today's funding environment, we are not likely to see the initiation of the effort described here. We therefore considered an austere effort involving the development of the keyword glossary editor, data editor, and translators to support only the WWMCCS Allocation and Assessment Model and—to a degree—STRATCAM. This could be accomplished for $300K to $400K, and much of this cost would be matched by savings in the strategic C³ analysis community's database development efforts.

³In this report we will use the term codes to refer to computer models that are used in the analyses the Strategic C³ analysis community performs. This term is most common in the nuclear effects area, but crops up elsewhere from time to time. Some consider this use of the term code to be improper. In any event, it merits explanation.
Section 8 contains detailed conclusions and recommendations.

PURPOSE AND CONTENTS OF THIS REPORT

This report is intended to provide the reader with

- An enumeration of the tools in use by the strategic C³ analysis community;
- A description of a prototype common-format database approach; and
- An estimate of the scope of the effort required to provide the database, support software, and documentation.

This RAND study has been a foundation for subsequent efforts, which, if carried out, would be performed by other organizations. These organizations would (1) further develop the database design; (2) complete the presentation and translation interface facilities; (3) develop the user documentation; and (4) perform validation and verification of the SSCF Database.

Section 2 is a description of our survey, including a review of strategic C³ systems analysis methodology, including functional analysis of networks and engineering analysis of communications links. Section 3 describes the SSCF Database with regard to its development, content, distribution, and application. Of special concern and emphasis is the discussion of procedures and operational data.

Section 4 is an extensive treatment of common-format database design issues, including database architecture and administration, and a keyword language concept. Section 5 is a description of a set of common format database software tools, including estimates of the effort required for their development.

Section 6 contains, for the SSCF Database described in Sec. 3, example database records using the prototype described in Sec. 4. Section 7 lays out the test case that we used to demonstrate the feasibility of this approach. Section 8 contains our conclusions and recommendations.

Appendix A contains a discussion of database administration, project administration, and security considerations. Appendix B contains a formal syntax for the SSCF database.
2. C³ SYSTEMS AND METHODS OF ANALYSIS

This section identifies the models, input data, and data sources used in analyses of strategic command and control communications (C³). To compile this information, we surveyed model users and developers, focusing on network functional analysis models, such as SIMSTAR and the Worldwide Military Command and Control System (WWMCCS) Allocation and Assessment Model (WAAM), and on communications engineering models, such as the Defense Nuclear Agency (DNA) Weapons Effects on Communications (WECOM) models. Using these models, we identified the information required to quantify C³ performance and vulnerability for developing a Standard Strategic Common-Format (SSCF) Database. It is important to note that these descriptions refer to the community's activities and tools at the time of our surveys. The specifics will always be changing.

ANALYSIS OF SYSTEMS OPERATIONS

The problem the models address is control of the strategic nuclear triad. We therefore begin with a definition of this problem in terms of the communications required to guarantee continuity of command and control. Subsequent sections describe the methods and computer programs that analyze these communications.

Command and Control of Strategic Nuclear Forces

The essence of the strategic C³ problem has, for decades, been the control of the strategic nuclear offensive force triad, pictured in Fig. 2.1. Traditionally, this has meant implementation of the Strategic Integrated Operations Plan (SIOP). For most of the relevant analyses that the strategic C³ community performs, this has come to mean dissemination of emergency action messages (EAMs) through the Minimum Essential Emergency Communications Network (MEECN). Analyses involving the North Atlantic Treaty Organization's General Strike Plan (GSP) have sometimes been integrated into strategic C³ analyses, but more often the strategic analyses have been limited to the control of SIOP weapons. The Strategic Defense Initiative (SDI) and perhaps the Air Defense Initiative could be important parts of this picture, depending on the degree to which these programs are funded.

A complete analysis of strategic command, control, communications, and intelligence (C³I) includes attack warning, attack assessment, decision support, conferencing, EAM generation and dissemination, force management, and a reconstitution and regeneration system to manage and execute the Department of Defense's (DoD's) triad of strategic nuclear weapon systems. The analytical emphasis in the community, and therefore the focus of this study, has been EAM dissemination, although it is acknowledged that the remainder of the problem needs greater emphasis than it has received to date.

General Nuclear War Information Flow

Figure 2.2 depicts the flow of information involved during a general nuclear war. When the United States' national policy was limited to an assured retaliatory capability, the critical in-
formation flow was provided by the MEECN—the assured communications required to execute the surviving triad forces after a large-scale attack on the United States. Over the years, our national policy has evolved to include selective response and the ability to carry out a protracted conflict, both capabilities intended to make deterrence more credible. This has resulted in the requirement for sophisticated force management. Conferencing and EAM dissemination are still vital, but are now joined by decision support, report-back, and other forms of secure, two-way communications. Changes in national policy, the availability of newer and better C³ systems, and the evolving threat tend to produce a constantly changing menu of systems and scenarios that must be evaluated. Maintaining up-to-date methodology and databases to analyze general nuclear war information flow is essential. Information flow is an essential part of the responsibilities that the DoD has assigned to the WWMCCS Evaluation Program [1].

**Strategic C³ Engineering and Functional Analysis**

Figure 2.3 depicts a typical strategic C³ engineering and functional analysis flow. Block 1 represents both the threat (based on objective assessments of the enemy's capabilities) and
specific attacks (based on the enemy's capability plus a presumed intent). The analyst may (1) develop hypothetical attacks based on these specified threats, (2) use specified attacks (e.g., the Red Integrated Strategic Operations Plan [RISOP]) obtained from another analyst, or (3) use the specified attack (e.g., the RISOP) as a baseline and develop hypothetical attacks as excursions.

The nuclear effects and benign propagation models (Blocks 2 and 3 of Fig. 2.3) are used in the evaluation of link performance; i.e., both the ability of blue C³ links to support the network and the ability of red threats (jammers, physical attack, high-altitude nuclear bursts, etc.) to deny or disrupt blue communications. These link models usually require input data that are narrower in scope, but more detailed, than the network codes.

Block 4, the characterization of C³ networks, includes defining, in terms suitable for analysis, the facilities, equipment, operations, and procedures that collectively form a strategic C³ (or C³I) network.

In two of the major network computer codes of interest today—SIMSTAR and WAAM—the link performance calculations (Block 5) and the network performance calculations (Block 6) are performed within the discrete event simulation. SIMSTAR and WAAM both rely on simplified algorithms and/or look-up tables in performing link calculations.
System Operations

The following key documents describe strategic C³ systems and operations:

- Strategic system descriptions
- Nuclear C³ assessment
- Strategic C³ system performance
- NSNF Master Plan
- Current SYDP
- EAP Vol. IV
- EAP Vol. VII
- SAC Regulations 55–14, 100–7, 100–20
- Navy OPORDER 32
- Tactical warning and attack assessment (TW/AA) 2000
- Test reports, e.g., POLO HAT
- Force location files
These documents constitute much of the top-level input data to Block 1 of Fig. 2.3. The information sources are quite diffused and the database is very large. Reading through a few documents such as these—no matter how well the documents are put together—though necessary, is never sufficient. There are "second tier" documents which describe, for example, new systems in sufficient technical detail to allow the analyst to quantify the inputs required to run the computer codes of interest. These documents are, for the most part, not identified here. They will, however, be important inputs for the analyst. Some of them would be identified in a database description and user's guide document. The analyst must use even more second-tier documents when developing excursions from the baseline.

Documents such as these vary from unclassified to top secret. Some contain sensitive information (e.g., SIOP or funding information). The range of classification and sensitivities, and the large number of sources, poses a database management problem. Database management systems coupled with appropriate security procedures can address some of the control issues. This topic is addressed in more detail in App. A to this report.

**Threats to Blue Strategic C³**

Two aspects of the development of attack scenarios (Block 1 of Fig. 2.3) have database management implications. First, in development of Red attacks against Blue C³ networks, the analyst has to consider the effects of those attacks:

1. Our C³ facilities could be damaged or destroyed. This usually consists of a laydown of nuclear weapons, but could also consist of conventional high explosives or even unconventional warfare attacks on the facilities by conventional forces and/or covert activity by enemy agents. The bottom line for all these types of attack would be for each facility the probability of damage (or survival) as a function of time in the scenario.

2. Radio links could be disrupted or denied (jamming, blackout, and scintillation).

3. Red Forces and/or agents could usurp the Blue network.

It is fairly common for a specified attack to include the details of the physical damage or destruction while leaving disruption, denial, or usurpation up to the analyst to develop as he sees fit. This is an important and often overlooked degree of freedom that has, on a number of occasions, produced substantial differences between results from organizations that were using essentially the same attack on essentially the same network.

Second, an analyst needs an extensive database describing the Red resources in order to integrate that threat into a scenario. The laydown involves attacks by intercontinental ballistic missiles (ICBMs), submarine launched ballistic missiles (SLBMs), bombers, cruise missiles, and, potentially, future space weapons. In the strategic C³ arena, virtually all jammers are known radio transmitters. Exoatmospheric bursts could come from ICBMs, SLBMs, or possibly Soviet spacecraft. Although not usually incorporated into the hypothetical Red attacks, unconventional warfare, selectively applied, could represent a significant threat.

Analysts in Joint Staff J-8 and the Defense Information Systems Agency (DISA) regularly construct attacks that are known as the RISOP. The attacks form a very important benchmark threat application. From time to time, however, analysts need to consider extensions to the RISOP attacks. For example, if an analyst is evaluating a number of candidate future
C₃ options, each excursion of the C₃ system logically requires an excursion of the RISOP to attack the new Blue C₃ system or configurations.

In scenario development, the strategic analyst must consider the Red-Blue defense posture coupling. If the Soviets were assumed to attack from a generated posture (a high state of readiness, involving deployment of forces and C₃ systems for survival), the analyst would usually assume that the United States would have, to some degree, matched the Soviets' deployments. Thus, the U.S. forces and C₃ assets would operate in a different mode and presumably suffer less damage from the hypothetical Soviet first strike.

The above threat and attack scenario information can fall in a wide range of security classifications. The threat data tend to come from special compartmented information intelligence sources (and some threat data will remain at that classification level). The basic classification can range all the way to top secret, special access required.

**Attack/Weapons Effects Codes and Databases**

Table 2.1 lists a number of threat, attack, and weapons effects codes and databases. The RISOP has already been discussed. SIDAC, NUCWAVE, and CLUSTER are tools that can be used to construct a laydown and determine the effects of that laydown on a set of communications nodes. The WAAM Attack Generator (WAAMAG) is used to develop attacks on C₃ systems, and DISA has "married" it to the SIMSTAR code, since SIMSTAR lacks an attack generation facility. The Joint Resources Assessment Database (JRAD) is a compendium of

<table>
<thead>
<tr>
<th>Threat/Attack/Weapons Effects Codes/Databases</th>
<th>User/Proprietor</th>
</tr>
</thead>
<tbody>
<tr>
<td>RISOP</td>
<td>Joint StaffJ-8</td>
</tr>
<tr>
<td>JRAD</td>
<td>DISA</td>
</tr>
<tr>
<td>SIDAC</td>
<td>DISA</td>
</tr>
<tr>
<td>NUCWAVE</td>
<td>DISA</td>
</tr>
<tr>
<td>CLUSTER</td>
<td>DISA</td>
</tr>
<tr>
<td>WAAMAG</td>
<td>DISA</td>
</tr>
<tr>
<td>WAAAMAG</td>
<td>SAIC</td>
</tr>
<tr>
<td>SINBAC</td>
<td>LOGICON</td>
</tr>
<tr>
<td>STRIKE</td>
<td>AF Ballistic Missile Office, TRW</td>
</tr>
<tr>
<td>NWEM</td>
<td>Air Force Center for Studies and Analysis, DNA, TRW</td>
</tr>
<tr>
<td>HEMP</td>
<td>HQ SAC</td>
</tr>
<tr>
<td></td>
<td>NRA</td>
</tr>
<tr>
<td></td>
<td>AFWL</td>
</tr>
<tr>
<td>VNs (RED)</td>
<td>DIA/DE-4C</td>
</tr>
<tr>
<td>(BLUE)</td>
<td>DISA</td>
</tr>
<tr>
<td>ATTACK/TW/AA</td>
<td>HQ Space Command</td>
</tr>
<tr>
<td></td>
<td>Teledyne Brown</td>
</tr>
<tr>
<td>EVOLVING THREAT TO WWMCCS</td>
<td>DIA/DT-4</td>
</tr>
<tr>
<td></td>
<td>DISA</td>
</tr>
<tr>
<td></td>
<td>TRW</td>
</tr>
</tbody>
</table>
material on Blue C³ facilities, including estimated hardness as measured by the vulnerability number (VN) system. Change 4, dated 1 June 84, to the DIA “Green Book” introduced a number of variations in the VN categories. DNA then sponsored an effort to produce a code, “Nuclear Weapons Effects Software for Small Computers,” which produces a PC-based damage probability calculation capability.

**Unified and Specified Command and Services Analysis Activities**

Table 2.2 lists highlights of strategic C³ analytical activities that the nuclear CINCs and Services perform. Our survey was sufficiently broad in scope to give us an appreciation of the general nature and scope of these activities, but we have by no means acquired an in-depth or up-to-date understanding of the wide variety of efforts referred to here. The purpose of this brief description is to set in perspective the discussions of network functional analysis and link engineering analysis codes in Secs. 3 and 4 of this report.

The Goldwater-Nichols Defense Reorganization Act of 1986 gives the Unified and Specified Commands a greater role in the Planning, Programming, and Budgeting process. However, even before 1986, the Unified and Specified Commands and the military services always developed their own positions on C³ system acquisitions and operations.

SAC performs SIMSTAR-based analyses that support funding and operational recommendations for CINCSAC, e.g., the SAC Command and Control Architecture and Master Plan (SCAMP). The database requirements for this work reflect the planning function, with an orientation toward both current and future systems. Engineering-level codes are used for more detailed connectivity assessments. For example, SKYMAP has been used to study the robustness of the MILSTAR network vis-à-vis high-altitude burst excursions on the RISOP. SKYMAP is a graphical tool that displays blackout absorption regions, which are calculated based upon RANC IV phenomenology.

The Joint Staff J-8 and JSTPS were both conducting SIOP and RISOP wargaming analysis. JSTPS performs Blue C³ targeting studies in connection with SIOP generation. Monte Carlo trials of the SIOP and RISOP are run using a multimodule discrete event simulation, which computes message delays. Database requirements reflect the focus of JSTPS’s work on deployed systems.

**Table 2.2**

<table>
<thead>
<tr>
<th>Command/Service</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>CINCSAC</td>
<td>SCAMP</td>
</tr>
<tr>
<td>Joint Strategic Target</td>
<td></td>
</tr>
<tr>
<td>Planning Staff (JSTPS)</td>
<td>SIOP vs. RISOP (focussed on current capabilities)</td>
</tr>
<tr>
<td>CINCLANT/CINCPAC</td>
<td>Fleet ballistic missile submarine (SSBN) connectivity</td>
</tr>
<tr>
<td>AFSPACECOM</td>
<td>Exercises, diagnostics, and planning (TWIAA)</td>
</tr>
<tr>
<td>Service</td>
<td>AFCSA: AP programs</td>
</tr>
<tr>
<td></td>
<td>NOP-941D: SSBN connectivity</td>
</tr>
</tbody>
</table>
AF Space Command has developed the Tactical Warning/Attack Assessment (TW/AA) model (see the next section) for use in systems evaluations and exercises; however, full-scale application has not, to our knowledge, taken place. In 1986, the Systems Integration Office of AF Space Command tasked DISA to perform AW/AA\textsuperscript{1} analyses to support system design and, we believe, the Planning, Programming, and Budgeting recommendation process.

AFCSA is charged with answering Air Staff connectivity questions, typically addressing marginal (performance versus cost) utility. AFCSA has conducted a number of analyses over recent years based on the use of SIMSTAR, notably Sabre Connector studies and an evaluation of GWEN for OASD/C\textsuperscript{3}I and, more recently, MILSTAR studies.

The Director, Space Command and Control, under the Chief of Naval Operations; the Naval Space and Warfare Systems Command; and CINCLANT and CINCPAC perform and/or sponsor work by such organizations as the Johns Hopkins University's Applied Physics Laboratory, Naval Research Laboratories (NRL), and the Naval Ocean Systems Center (NOSC) to evaluate SSBN connectivity. DNA also sponsors studies of strategic naval communications in nuclear environments.

**FUNCTIONAL ANALYSIS OF NETWORKS**

In this subsection, we will discuss the analytical tools that are used to assess the performance of communications networks as a whole. This type of analysis almost always uses discrete-event simulations of network survivability, vulnerability, and functional performance.

**Network Codes**

Table 2.3 depicts a number of C\textsuperscript{3} network codes and the users or proprietors of these codes. During our surveys, we were able to contact most of the key individuals involved to discuss their code applications and database requirements. We have found that the information on who is using what network codes, computer systems, operating systems, and database management systems is highly perishable, lasting no longer that it takes to document it. The fact that "the only constant is constant change" is indeed the message. Our approach must be flexible enough to accommodate changes in codes, computers, and software support.

SIMSTAR [2-9] and WAAM [10-12], developed by AFCSA and DISA (formerly DCA), respectively, are the dominant network-analysis codes within the strategic C\textsuperscript{3} community. Both are discrete-event Monte Carlo simulations possessing comprehensive pre- and post-processing capabilities. Each covers the principal communications media, including radio frequencies from VLF to EHF. The codes differ primarily in their approaches to message handling and scenario specification.

SIMSTAR requires specification of message handling procedures at each node, and each message type is routed separately. This requires a large, detailed database. However, the user has considerable flexibility in the definition of protocols and can model resource competition at each node (though self-jamming is not normally accounted for). Scenarios are generated externally. The Threat Assessment Module (TAM) Computer Software Configuration

\textsuperscript{1}In some organizations, the term "Attack Warning and Attack Assessment (AW/AA)" replaced the term "Tactical Warning and Attack Assessment (TW/AA)," but both are still in use.
<table>
<thead>
<tr>
<th>Code</th>
<th>User or Proprietary</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMSTAR</td>
<td>Joint Staff</td>
</tr>
<tr>
<td>STRATCAM</td>
<td>AFCSA</td>
</tr>
<tr>
<td>SIMSTAR</td>
<td>OASD (PA&amp;E)</td>
</tr>
<tr>
<td>SIMSTAR</td>
<td>DSSO</td>
</tr>
<tr>
<td>SIMSTAR</td>
<td>AF Electronic Systems Division</td>
</tr>
<tr>
<td>SIMSTAR</td>
<td>AF Ballistic Missile Office, TRW, Inc.</td>
</tr>
<tr>
<td>SIMSTAR</td>
<td>Intelligence Working Group</td>
</tr>
<tr>
<td>SIMSTAR</td>
<td>Los Alamos National Laboratories</td>
</tr>
<tr>
<td></td>
<td>(LANL)</td>
</tr>
<tr>
<td>SIMSTAR</td>
<td>HQ SAC/XP</td>
</tr>
<tr>
<td>SIMSTAR</td>
<td>JSIPS</td>
</tr>
<tr>
<td>STRATCAM</td>
<td>SAIC</td>
</tr>
<tr>
<td>SIMSTAR</td>
<td>Martin Marietta</td>
</tr>
<tr>
<td>SIMSTAR</td>
<td>AF Foreign Technology Division</td>
</tr>
<tr>
<td>WAAM</td>
<td>DISA</td>
</tr>
<tr>
<td>WAAM</td>
<td>DSSO</td>
</tr>
<tr>
<td>WAAM</td>
<td>TRW</td>
</tr>
<tr>
<td>WAAM</td>
<td>AF Ballistic Missile Office, TRW, Inc.</td>
</tr>
<tr>
<td>NSCS/SCAP</td>
<td>NOP-841D</td>
</tr>
<tr>
<td>NSCS/SCAP</td>
<td>Johns Hopkins University Applied</td>
</tr>
<tr>
<td></td>
<td>Physics Laboratory</td>
</tr>
<tr>
<td>DETEC</td>
<td>Los Alamos National Laboratory</td>
</tr>
<tr>
<td>FNAC</td>
<td>CSC/AFWL</td>
</tr>
<tr>
<td>MINEFIELD</td>
<td>NSCSC</td>
</tr>
<tr>
<td>MINEFIELD</td>
<td>Intelligence Working Group</td>
</tr>
<tr>
<td>TW/AA</td>
<td>HQ Space Command</td>
</tr>
<tr>
<td>TW/AA</td>
<td>Teledyne Brown</td>
</tr>
<tr>
<td>TW/AA</td>
<td>RDA</td>
</tr>
<tr>
<td>DMMS</td>
<td>SAIC</td>
</tr>
<tr>
<td>NETSIM</td>
<td>GTS Sylvania</td>
</tr>
</tbody>
</table>

NOTE: SIMSTAR is being replaced by STRATCAM.

Item of the STRATCAM simulator is a stand-alone module that allows the user to preprocess and calculate the probability of damage, probability of upset, and recovery from upset of communication nodes as a result of multiple nuclear bursts. The main STRATCAM simulator then uses these preprocessed damage and upset data from the nuclear threat to calculate communications network connectivity.

WAAM uses generic protocols in the form of a procedures file at each node, which establishes the handling of each message type. This approach is more efficient in terms of database requirements than SIMSTAR. Routing is specified either by predefined networks or by connectivity files. Resource competition occurs only at the receiver (where self-jamming is accounted for). WAAM can accept externally prepared scenarios, or the analyst can create damage timelines using the WAAMAG.

AF Space Command developed a near-emulation-level model of the TW/AA network to support its own exercises, diagnostics, and planning, although other potential users, particularly the Army Strategic Defense Command, were involved in the development. The code includes a ballistic missile attack generator, very detailed models of the tactical warning sensors and
communications network, a nuclear weapons effects module, and an SDI interface. The sensor model generates formatted messages, the contents of which are finally displayed as they might appear in the North American Defense (NORAD) Cheyenne Mountain Complex.

Medium- or high-fidelity versions of the model are selectable at the user's option. These differ primarily in the treatment of nuclear effects and communications protocols. Both versions treat protocols in considerable detail and possess an exhaustive array of data-entry menus. The medium-fidelity communications stress module is based upon WESCOM and SIMBAL, while the high-fidelity stress module is closer in structure to SCENARIO and PRPSIM. (The engineering codes are described in the section below on engineering analysis of communications links.) The radar-disturbed environment is an elaboration of NORSE. The database compiled for the code is extensive and unique, focusing on the TW/AA sensors and communications assets.

Issues have been raised concerning the TW/AA model that leave its future somewhat in question. The lengthy run time of the code precludes Monte Carlo trials, except perhaps on a Cray. The code has not been validated, but we understand that further distribution has been approved.

STRATCOMMAND was a network code that was the predecessor to SIMSTAR, but only treated a single message at a time.

NOSC used MINEFIELD for the analysis of Navy MEECN connectivity. This code is a distant cousin of SIMSTAR, but message-authentication techniques have been modeled in detail to support communications-security studies.

The Johns Hopkins University Applied Physics Laboratory uses the Navy Strategic Communications Simulator/Strategic Communications Continuing Assessment Program (NSCS/SCAP) to analyze Navy MEECN connectivity. This code computes network flow and message delays, using SIMBAL to assess link performance. The SIMBAL results are supplemented with statistics based on real-world data. The network simulation includes effects of node outages due to physical attack, jamming, and equipment failure. Message handling procedures are specified on a node-by-node basis (à la SIMSTAR), and include message piecing, handling time, and retransmission sequences. The propagation model is restricted to the VLF, LF and HF bands. The SCAP postprocessor produces contour maps of message acceptance probability, message acceptance time, etc., over the SSBN operational area.

The Propagation Network Analysis Code (PNAC), which CSC developed under AFWL sponsorship, is a discrete-event simulation resembling SIMSTAR, but is specialized to satellite and microwave line-of-sight links. A unique feature of PNAC is the rigorous treatment of disturbed propagation. The nuclear environment is specified by an output file generated with the SCENARIO code. The network model treats a variety of protocols, routing techniques, acknowledgement, prioritization and buffering, which are specified by input menus.

DETEC is under development at Los Alamos National Laboratory. It is envisioned as a national modeling resource, which would reside at the laboratory but serve a wider community over secure lines. Although intended primarily for Strategic Defense Initiative (SDI) analysis, the scope of the model encompasses communications, sensors, and attack generation, in addition to SDI weapons and battle management. Unique features of the code will include
the capability to handle rule-based and interactive decision processes, and the extensive use of graphical interfaces. DETEC will require a Cray to run.

Network Code Use and Status

The various network codes listed in Table 2.3 can be categorized according to their use and status, as follows:

- **Operational, primary**
  - SIMSTAR/STRATCAM/TAM
  - WAAM
  - NSCS/SCAP
- **Operational, available and/or in use**
  - STRATCOMMAND, DMMS, SIMSTAR-B
  - MINEFIELD
  - PNAC
- **In development or new concepts**
  - TW/AA
  - DETEC
  - Fuzzy sets, expert systems, etc.

SIMSTAR, WAAM, and NSCS/SCAP are in place, and members of the strategic C³ community are using these codes to perform important analyses (described below). These particular efforts address the vital issues of

- SIOP versus RISOP wargaming.
- PPBS recommendation process, and
- Operations planning.

Although the analyses that other organizations perform are considered worthwhile, the community of interest that the Joint Staff originally envisioned emphasizes the vital wargaming and Planning, Programming, and Budgeting processes.

The second set of network codes are in place, checked out, and available for use but for various reasons are not considered primary for this effort. SIMSTAR 85-A and, more recently, STRATCAM, have virtually replaced STRATCOMMAND and SIMSTAR B. STRATCAM itself is undergoing change.

MINEFIELD is quite specialized. Analyses using MINEFIELD have focused on communications security and thus highlight cryptographic equipment and procedures and “spoofing” techniques. The MINEFIELD analyses are performed using kill probabilities taken from a RISOP attack tape. The propagation statistics were taken from the MBECN Master Plan and from connectivity study runs performed for DCA. MINEFIELD is not considered transportable, but is available for use by the strategic C³ community.
PNAC is highly oriented toward satellite communications and can be valuable for certain applications, but its use is limited at this time. The TW/AA program is far too detailed to fit in the “network functional analysis” family treated in this survey. Future codes—such as DETEC, other SDI codes, new approaches (e.g., fuzzy-set algorithms) and very simple network codes—are items in the “tool box” that should be examined from time to time and considered for future use.

SIMSTAR, WAAM, and the NSCS/SCAP are discrete-event simulations that evaluate stochastically the flow of messages through a network that is degraded by enemy action. The organization of these three codes, however, is significantly different. SIMSTAR represents the network using the following generic “entities”:

1. Nodes
2. Messages
3. Receivers
4. Transmitters
5. Jammers
6. Packet switching
7. Bursts
8. Events

The inputs the analyst provides create the organization, structure, and substance of the network.

WAAM is a very different model. WAAM associates the following classes of files with the threat allocation and physical damage assessments:

- Ballistic missile allocation
  - Ballistic missile assets
  - Target data file
  - Allocation history file
- Antisatellite weapon allocation
  - Antisatellite database (Red)
  - Satellite database (Blue)
- Jammer allocation
- Bomber allocation
- Damage assessment
  - Blast effects
  - High-altitude electromagnetic pulse effects
  - Radioactive fallout.
As indicated before, SIMSTAR does not have a parallel to the WAAMAG process. One might expect the discrete event simulations to be organized in much the same way, since they were created to do essentially the same job. Not so. The WAAM functional assessment files are organized around specific force elements and specific National Military Command System and WWMCCS assets and procedures, not by a generic database-driven entities, as in SIMSTAR:

1. Force element and command center structure
2. Current MEECN procedures
3. Equipment inventories
4. ICBMs, SLBMs, and bombers
5. Postattack Command and Control System, Take Charge and Move Out (TACAMO), and Airborne Command Posts (ABNCPs) and their bases
6. Command centers
7. Communications relay sites
8. Emergency rocket communications systems
9. Ground entry points
10. Satellites
11. Hostile jammers
12. Automatic Voice Network, Automatic Digital Network, and Joint Chiefs of Staff Alerting Network switches
13. LF, HF, satellite, and terrestrial connectivities

The NSCS/SCAP organization is shown in Figure 2.4. The following are the major elements of SCAP:

1. SIMBIP, a preprocessor that allows the analyst to make inputs to the “Simulation of Multiple Bursts and Links” (SIMBAL) code, which Kaman/Tempo developed for DNA.
2. SIMBAL II, which generates propagation data for Navy communications links.
3. The SCAP preprocessor, which integrates data from SIMBAL II with predicted benign CCIR atmospheric noise tables.
4. The NSCS, a discrete event simulation of dissemination of EAMs through a limited communications network.
5. A postprocessor and graphics package.

The NSCS, like the earlier DCA MEECN Systems Model Attack Assessment Module, takes a complex set of files from an engineering code (in this case SIMBAL II) and processes those files for use in the discrete event simulation. SIMSTAR and WAAM, on the other hand, have built-in algorithms or look-up tables that represent simplified versions of the communications engineering codes. The NSCS also assumes that nodes are either in or out (i.e., they ei-
ther have been destroyed or have survived an attack), rather than accepting time-varying probabilities of node survival, as do SIMSTAR and WAAM.

The SIMSTAR, WAAM, and NSCS/SCAP simulations must be supported. Not only are they involved in the performance of the important SIOP versus RISOP wargaming and Planning, Programming, and Budgeting process, their different structures offer the strategic C³ community the opportunity for useful comparison checks.

**Network Code Issues**

Several issues come up in comparisons of the results of different C³ network analyses. First, the network representation is very different among the major codes with respect to their treatment of message handling and routing, link availability, node (facility) destruction or damage, reliability, and human factors. Second, the network codes, in addition to handling these very stochastic processes, are usually required to address a fairly wide range of threat and attack scenarios. This leads to long computation times and, even more important, to long setup and analysis times. Third, complexity of the database and the scope of the problems to be solved make database management an important issue. The sensitivity and classification of these databases are also of concern.
Fourth, validation and verification of the network codes, using test data where applicable, are important elements in ensuring their utility. One can characterize this process using the scientific method:

- Observation of the system,
- Formulation of a mathematical model that attempts to explain the observation of the system, and
- Prediction of the system behavior on that basis, i.e., applying mathematical transformation and logical deduction to obtain solutions to problems run with the model.

Carrying through the first step of the scientific method—for a strategic C3I system involving, as it would, nuclear combat—is absurd to contemplate. On the other hand, many of the functions that are performed in military operations can be, and indeed are, exercised. Thus, it is possible to approximate the functions that take place and the timing of those functions in terms that lend themselves to computer-based analyses. The computer can generate results that one could call “pseudo-observations” of the system under study. Indeed, that is what these discrete-event simulations are designed to do. To succeed, both the model and the input data must be sufficiently accurate and precise to allow meaningful comparisons. DCA directed a study [13] to compare the network representation of of SIMSTAR and WAAM, to each other and to “ground truth,” i.e., the best information available (test data, detailed scientific calculations). Other validation and verification efforts were, at the time of publication of this report, taking place and under consideration. This project will address database and database management aspects of both network representation and the development and/or specification of attack scenarios.

ENGINEERING ANALYSIS OF COMMUNICATIONS LINKS

Some computer codes allow analysis of portions of a network in greater detail and with higher fidelity than is appropriate for full network functional simulations. Engineering codes can be used to “design” a system or parts of a system, and are often used to analyze in detail critical parts of a network in support of the functional simulations described in the prior sections.

Table 2.4 lists the key engineering codes that we identified in our survey. In the following subsections, we will discuss the codes developed or sponsored by the DNA, the National Telecommunications and Information Administration (NTIA), and codes developed at the NOSC and NRL.

DNA Codes

The Atmospheric Effects Division of DNA sponsors the development of numerous disturbed-environment codes covering the frequencies from ELF to EHF. Many analysts continue to use codes that DNA no longer supports (e.g., RANC). The DoD Nuclear Information and Analysis Center (DASIA) has summarized the supported codes in the “DASIA Computer Codes Summary.” DASIA maintains a nuclear effects reading room and bibliographical service. Its holdings are compiled in the “DASIA Shelf List,” which is accessible in hard copy or by computer terminal on the premises.
Table 2.4
Link Codes Summary

<table>
<thead>
<tr>
<th>Code</th>
<th>Frequency</th>
<th>Propriator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multilink</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIMBAL II</td>
<td>VLF/LF/HF</td>
<td>DNA</td>
</tr>
<tr>
<td>VLFSIM</td>
<td>VLF/LF/HF</td>
<td>DCA CAS</td>
</tr>
<tr>
<td>Link</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WEDCOM</td>
<td>ELF-LF</td>
<td>DNA</td>
</tr>
<tr>
<td>WEFCOM</td>
<td>HF</td>
<td>DNA</td>
</tr>
<tr>
<td>NUCOM</td>
<td>HF</td>
<td>Stanford Research Institute</td>
</tr>
<tr>
<td>WEMCOM</td>
<td>VHF</td>
<td>DNA</td>
</tr>
<tr>
<td>WETCOM</td>
<td>VHF-SHF</td>
<td>DNA</td>
</tr>
<tr>
<td>WESCOM</td>
<td>VHF-EHF</td>
<td>DNA</td>
</tr>
<tr>
<td>WESCOM</td>
<td>VHF-EHF</td>
<td>DNA</td>
</tr>
<tr>
<td>SIGDAT</td>
<td>VHF-EHF</td>
<td>DNA</td>
</tr>
<tr>
<td>Propagation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODEFNDR</td>
<td>ELF-LF</td>
<td>NOSC</td>
</tr>
<tr>
<td>WAVEGUID</td>
<td>ELF-LF</td>
<td>NOSC</td>
</tr>
<tr>
<td>SECM WG</td>
<td>ELF-LP</td>
<td>NOSC</td>
</tr>
<tr>
<td>VLFSIM</td>
<td>VLF/LF</td>
<td>NRL</td>
</tr>
<tr>
<td>ANIHOP</td>
<td>LF</td>
<td>NTIA/ITS</td>
</tr>
<tr>
<td>WAVEHOP</td>
<td>LF</td>
<td>NRL</td>
</tr>
<tr>
<td>GWAFA</td>
<td>LF-EHF</td>
<td>NTIA/ITS</td>
</tr>
<tr>
<td>MFSIPM</td>
<td>MF</td>
<td>NTIA/ITS</td>
</tr>
<tr>
<td>IONCAP</td>
<td>HF</td>
<td>NTIA/ITS</td>
</tr>
<tr>
<td>HFMUFE84</td>
<td>HF</td>
<td>NTIA/ITS</td>
</tr>
<tr>
<td>PROPHET</td>
<td>HF</td>
<td>NOSC</td>
</tr>
<tr>
<td>ABCOM</td>
<td>HF</td>
<td>Stanford Research Institute</td>
</tr>
<tr>
<td>EPM/85</td>
<td>HF-EHF</td>
<td>DECAC</td>
</tr>
<tr>
<td>SRICOM</td>
<td>HF-EHF</td>
<td>Stanford Research Institute</td>
</tr>
<tr>
<td>LONGLEY</td>
<td>HF-EHF</td>
<td>NTIA/ITS</td>
</tr>
<tr>
<td>SKYMAP</td>
<td>VHF-EHF</td>
<td>AFCSA</td>
</tr>
<tr>
<td>TROPO</td>
<td>VHF-SHF</td>
<td>MAXIM</td>
</tr>
<tr>
<td>MPM</td>
<td>SHF-EHF</td>
<td>NTIA/ITS</td>
</tr>
<tr>
<td>Environment</td>
<td>Type</td>
<td></td>
</tr>
<tr>
<td>RANC</td>
<td>Nuclear</td>
<td>MRC</td>
</tr>
<tr>
<td>WEIH</td>
<td>Nuclear</td>
<td>Kaman Tempo</td>
</tr>
<tr>
<td>SPREAD DEBRIS</td>
<td>Nuclear</td>
<td>PSR</td>
</tr>
<tr>
<td>NTIA</td>
<td>Noise</td>
<td>NTIA/ITS</td>
</tr>
<tr>
<td>CCIR</td>
<td>Noise</td>
<td>CCTR</td>
</tr>
<tr>
<td>WGL</td>
<td>Noise</td>
<td>NRL</td>
</tr>
<tr>
<td>GLAMP</td>
<td>Nuclear</td>
<td>DNA</td>
</tr>
<tr>
<td>SCENARIO</td>
<td>Nuclear</td>
<td>DNA</td>
</tr>
</tbody>
</table>

The nuclear-effects models fall into three categories: (1) first principle (e.g., MICE/MELT), (2) engineering (e.g., WECOM), and (3) multiburst/multilink (e.g., SIMBAL II). The WECOM family [14] includes WESCOM (satellite), WETCOM (troposcatter and tropospheric LOS), WEMCOM (meteor burst), WEFCOM (HF, F-Layer) and WEDCOM (ELF/VLF/LF, D-Layer).

WEMCOM became available in late 1987. DNA is also distributing WECOM's ambient environment propagation module, MBLINK, separately. Planned improvements in the WECOM
D-layer chemistry model will particularly benefit WEDCOM, since D-region ionization is the predominant nuclear effect on meteor-burst links. A 1-minute execution time for a single-link, single-time calculation is typical for all codes except WEMCOM, which is slower. WECOM is based on the TROPO code, which was updated and merged with WECOM and NORSE\(^2\) phenomenology modules. It has not been released. WEFCOM is derived in a similar fashion from NUCOM. It was originally distributed in 1986 and, along with WESCOM and WEDCOM, has undergone some recent revisions. All but WETCOM are available at this time. Extensions are being made in the WEDCOM waveguide model, an improved raytrace algorithm and F-layer depletion model are under development for WEFCOM, and the scintillation and propagation models in WESCOM have been revised. WEDCOM was also revised to predict horizontal field and multiple dipole loop antenna reception.

SIMBAL uses a tabular lookup and database-plus-interpolation VLF/LF/HF code that can simulate scenarios involving as many as 200 high-altitude bursts, and thousands of low-altitude bursts. It has undergone a major revision, completed validation, and was released early in 1989 as SIMBAL II. Further efforts are under way to extend SIMBAL's VLF database.

The WECOM codes and SIMBAL are all designed to run on VAX and CYBER computers; however, users have successfully transported them to other machines. PC versions of the WECOM codes have been distributed that consist of 1 to 2 MB of source code (plus PC input editors) and run on the IBM 386/286. Kaman Tempo provides some user assistance to authorized users. In the same vein, DNA has recently released a semiquantitative program for the IBM PC/XT/AT, called the ELF Propagation Computational Aid, that calculates the effect of a “uniform-spread debris” nuclear environment on ELF transmissions from the Navy Wisconsin Test Facility.

The nuclear phenomenology modules for the WECOM codes have been standardized so that a single copy of the object file can be linked to any of the codes. The underlying physics and chemistry for the nuclear and benign environments are detailed in Reference 14, which characterizes the atmospheric density, temperature, pressure, neutral species, precipitation, and winds; the ground conductivity, the ionospheric electron density profile; the prompt radiation energy release, transport, and deposition; and the fireball and debris region electron and ion densities, delayed energy release, transport, and deposition.

For the engineering level codes to be computationally efficient and accessible to systems analysts, it is necessary to compromise somewhat on accuracy, relative to what is provided in the first principles codes. Approximations identified in the WECOM phenomenology include the assumption of simple fireball shapes and ionization changes within the fireball, the use of interpolation to compute fireball parameters as a function of yield and burst altitude, the adoption of simplified models for electron density fluctuations and striation scale dimensions, the neglect of striations outside of the fully ionized fireball region, the forcing of disturbed atmospheric chemistry back to ambient after 12–24 hours, modeling of acoustic gravity waves as outwardly propagating density perturbations (i.e., no tilts or reflections at discontinuities), the simplified treatment of low-altitude multiburst effects, the use of parabolic fits to the upper ionospheric electron density profiles, and the neglect of nonuniformities in plume structure when tracing rays through the fireball plume at HF/VHF.

\(^2\)The Nuclear Optical and Radar System Effects code developed under DNA sponsorship.
Uncertainties in WECOM phenomenology are due, primarily, to gaps in the atmospheric test series database. It is felt, however, that the engineering codes provide representative environments for the purpose of assessing strategic C$^3$ connectivity. Uncertainties include the atmospheric chemistry reaction rates (under revision), the physics describing striation formation and development, multiburst interactions affecting fireballs and dust clouds, ionospheric acoustic gravity waves, geomagnetic field distortion, plume reflections, and small-scale ionization structure. Additional uncertainties arise from stochastic elements, such as winds (both tropospheric for dust and ionospheric for debris and plume), and from uncertainties in Red weapon design.

For those instances where environments are highly stressed and warrant greater concern for fidelity, nuclear codes are available whose treatment lies intermediate in detail between the first principle and engineering codes. SCENARIO predicts high-altitude nuclear environments (>100 km) produced by single or multiple explosions. Although it models hydrodynamics in a fashion somewhat similar to that of the first principle codes, the simplified magnetic field effects increase speed. SCENARIO has been modified to handle cases with up to 1000 bursts. C/LAMP uses simplified models of the flow field rather than full hydrodynamics and uses very detailed atmospheric chemistry to compute low-altitude (<100 km) nuclear environments. C/LAMP will be included as the nuclear environments module in the next release of the NORSE code. To ensure continuity across the limit of 100 km, C/LAMP is currently being extended to cover high altitudes as well.

PRPSIM is a high-fidelity, transionospheric propagation model for analyzing satellite communications in a disturbed environment. The environments for PRPSIM are generally obtained from SCENARIO. PRPSIM has recently been updated with the new two-component power spectrum model for scintillation promulgated by DNA that is also being installed in WESCOM. SIGDAT is a database-oriented code, developed from PRPSIM, specifying the disturbed signal predicted for the DNA-approved "reasonable worst-case scenario." This code has been revised to incorporate the new scintillation model and a richer selection of satellite orbits beyond the original geosynchronous case.

SIMBAL operates at a coarser level than either the physics or engineering codes. Run time is significantly reduced by performing look-ups on data tables wherever possible. Many of these tables have been revised for SIMBAL II. Other improvements are inclusion of mode-conversion coefficients for VLF propagation, treatment of beta-h ionospheric profiles, and inclusion of both the NTIA and localized-source interference models.

Because of the higher level of aggregation in SIMBAL, approximations prevail to a greater degree than in WECOM. Jammers are treated as all-frequency, omnidirectional sources. Disturbed E- and F-layer ionization calculations are not performed. Debris from surface bursts is assumed to stabilize at a 15-km altitude. The receiver model does not include internal noise. Ionospheric profiles for LF propagation are chosen by matching the disturbed profile to a reference at the approximate reflection altitude (computed assuming an exponential profile). VLF propagation is calculated using look-up tables for attenuation over homogeneous paths as a function of frequency, range, ionospheric profile type, etc. For realistic paths, these segments are pieced together using interpolated mode-conversion coefficients. The HF propagation model precomputes a family of ray paths over discrete frequencies, using a simplified quasi-parabolic ray trace model. Interpolation is used to obtain results for intermediate frequencies and takeoff angles.
NTIA Codes

Although NTIA pioneered the development of LF wavehop propagation codes in the early 1970s, their ANIHOP/ANIREF codes have not been supported for nearly a decade. The waveguide codes NOSC developed allow treatment of disturbed environments involving inhomogeneous paths, but ANIHOP does handle the uniform spread debris environment and requires minimal user interaction.

GWAPA is a common user interface for two earlier groundwave codes, WAGSLAB and GWSNR. WAGSLAB is the latest version of the irregular-terrain groundwave model that was previously called WAGNER. The code numerically solves the Hufford integral equation over paths with varying elevation and conductivity. This version treats ground cover as a dielectric slab, which allows one to model forests and cultural features in an idealized way. WAGSLAB is the only generally available program that deterministically computes propagation over irregular terrain. Availability of terrain data (conductivity, elevation and vegetation electrical constants) and CPU time required per path run are the major constraints on applicability of this code. GWSNR is a smooth-earth groundwave model that also computes atmospheric noise. GWAPA computes signal-to-noise ratios for both regular and irregular terrain and computes antenna performance over lossy earth for a variety of antennas.

IONCAP is the HF propagation program that NTIA currently supports to perform long-term predictions. It computes field strength, mode reliability, maximum usable frequency, and noise field at any location on the earth. It has recently been upgraded to include the new NTIA noise database. HFBC is a version prepared for the IBM PC/AT and XT and a variant; this version incorporates slightly faster algorithms to generate area-coverage contours. IONCAP will accept user-defined antenna patterns, but does have some internal antenna options. Planned improvements include representation of loss parameters for each propagation mode and frequency, installation of an interactive input processor and inclusion of an area coverage prediction technique, updated FOF2 coefficients, and improved polar ionospheric and ionospheric electron density models. In addition to communications applications, IONCAP is configured for over-the-horizon radar analysis. The mainframe and PC versions of IONCAP are both available from NTIS. However, it should be noted that only the mainframe version includes source code.

HFMUFES is an earlier NTIA HF code that is no longer supported. Although not necessarily inferior to IONCAP, it exists in many unauthorized, and possibly adulterated, versions. It is still in wide use, however, in part because the internal antenna package is extensive and caters to tactical applications. HFMUFES treats sporadic-E and spread-F differently from IONCAP and does not incorporate the latest noise data. In addition, propagation modes are computed differently, and HFMUFES does not allow user-defined electron density profiles, while IONCAP does.

AMBCOM is an HF code from SRI that is suitable for analyzing both telecommunications links and over-the-horizon radar geographical coverage. This model differs from IONCAP in that it uses a two-dimensional ray tracing algorithm like NUCOM and WEFCOM, which accounts for horizontal electron density gradients. Thus, although longer running, it is more accurate in computing paths through naturally disturbed regions, such as the auroral zone and across the day-night terminator.
NTIA has developed an atmospheric attenuation and propagation delay prediction code, called MFM, which covers the frequencies from 1 to 1000 GHz. The current version treats altitudes up to 30 km, but in the future this may be extended to 120 km. The code is available for the IBM PC on diskette, from Dr. Hans Liebe of NTIA/ITS. A mainframe version is also available from NTIS.

The ITS Irregular Terrain Model (Longley-Rice) makes statistical "area predictions" of attenuation and variability of radio signals for frequencies between 20 MHz and 20 GHz. The model is based on electromagnetic theory and on statistical analysis of terrain features and radio measurements. SRICOM is another NTIA-sponsored model, from SRI, that predicts tropospheric radio communications performance over irregular terrain. This model covers the same frequencies as the ITS code and includes the effects of man-made non-Gaussian radio noise. The radio system, propagation, and noise parameters are treated statistically, except for the frequency, range and antenna height, which are discrete inputs. The user may choose any of three area-type propagation models: Longley-Rice, EPM-73, or modified EPM-73.

E. Haakinson of NTIA has developed a meteor burst propagation code, based on Oetting's model. This code is currently available from NTIS.

MFSPM is a collection of empirical MF skywave codes that grew out of a critical survey of available prediction methods at NTIA/ITS. The package includes the Federal Communications Commission, CCIR, and Wang models and others, as described in Reference 15.

The NTIA skywave codes were generally not developed for application to the disturbed environment—the algorithms were not configured to handle steep ionization-density gradients. However, the codes should be valid for a "uniform-spread debris" environment.

**Navy Codes**

NOSC is the principal developer of waveguide propagation models for the ELF, VLF, and LF bands. One of the earliest modal codes, WAVEGUID, is still in use (it features double precision). MODEFNDR is the most recent product of this lineage (earlier versions were called MODESRCH). Both WAVEGUID and MODEFNDR solve for modes over homogeneous paths and need to be coupled with mode-summing and mode-conversion codes to compute attenuation over paths with varying conductivity or ionospheric profiles. NOSC supports two mode-summing codes. FASTMC, as its name suggests, was designed for computational speed. FULLMC does integrations through the ionosphere prior to calculating mode-conversion coefficients and is therefore slower.

SEGMENTED WAVEGUIDE is a waveguide propagation model that treats laterally inhomogeneous paths. It uses the WKB method for transitioning between segments, as did the earlier version of the code, called IPP. The WKB method is restricted to smooth variations along the path and is therefore not suitable for severely disturbed environments. For the latter cases, it is recommended that MODEFNDR be used. A new unified interface, LWPC, is available for MODEFNDR, FASTMC, and SEGMENTED WAVEGUIDE. LWPC simplifies the process of defining the propagation path and obtaining the fields along the path.

The principal VLF analysis tool in use at NRL is the VLF Automated Computation Method (VLPACM), which is also called the Navy Coverage Prediction Program. The model provides
graphical output of signal and noise thresholds and of signal-to-noise and signal-to-jam ratios for specified time availabilities. VLFACM is written in FORTRAN and currently runs on the VAX, Cray, or IBM PC. The propagation portion of the code uses a "look-up table" approach, based on data collected for paths to submarine patrol areas. The data cover frequencies between 15 and 30 kHz, but extrapolations are performed to obtain results up to 60 kHz. The code is very fast and simple and does not rely on ionospheric models. The limitations of the code include those attributable to the database (e.g., the lack of transverse-electric-mode data, gaps in some geographical regions) and the inability to handle disturbed environments beyond the uniform-spread-debris case. Data are currently being collected to cover the Arctic.

The noise portion of the code consists of both the CCIR noise model and an NRL-modified version of the Westinghouse Geosynchronous Laboratory (WGL) model. The WGL model computes interference at a receiver location by combining noise signals propagated from sources distributed according to the worldwide thunderstorm database. This approach has the advantage of providing both noise direction and amplitude. The results depend on knowledge of the lightning spectra (e.g., poor for TE-mode and higher frequencies) and the quality of the thunderstorm distributions. It should be noted that WGL was calibrated with the "old" CCIR noise database and is therefore slightly incompatible with the latest NTIA model. Ongoing efforts sponsored by DNA, NRL, Office of Naval Research, and the Rome Air Development Center are aimed at improving the knowledge of TE source spectra (and the distribution of horizontal lightning, which is believed to contribute to TE noise) and incorporating extensive new worldwide ELF/VLF noise data that Stanford University has collected into the CCIR/NTIA database.

NRL has developed a modified version of ANIHOP, WAVEHOP, for propagation in the LF band. The differences between this code and ANIHOP appear to be minor, although NRL has independently supported the WAVEHOP version for nearly a decade.

Engineering Code Database Issues

The issues relevant to engineering codes can be grouped under three broad questions:

What should they be used for?

• Providing inputs to network codes, e.g.,
  —Data (SCAP/NSCS, WAAM)
  —Subroutines (SIMSTAR)
  —Look-up tables, curve fits (SIMSTAR, WAAM)
  —Code development calibration (SIMSTAR)
• Validation of outputs of network codes
• Case studies
  —High fidelity
  —Optimization, excursions
  —Planning
Which code should be used?

- Types
  - Propagation
  - Link
- Fidelity
- Supported or unsupported
- Simplicity, transportability, speed, etc
- Are decision aids needed?

Are they compatible with functional codes?

- Data
  - Internal
  - Load-time
  - Input logic
  - Equivalent representation
- Modeling approach
  - Should variances be documented?

Database support for engineering codes is necessary, since they serve important functions in assessing C^3 networks. These functions include:

- Providing inputs to network codes,
- Validating results obtained with network codes, and
- Performing special case studies, typically ones requiring more precision or more sophisticated results than available with the network codes, or smaller link studies not requiring a full network simulation. An example combining these two features would be a study to optimize placement of airborne resources.

Fig. 2.5 shows the frequencies covered by the various network and link codes "as advertised" by the code developer. The breadth that SIMSTAR and WAAM display is simply a reflection of the fact that the networks they have been developed to model have communications systems in most of the frequency bands. The link codes are more specialized, but collectively they span the principal media. It is noteworthy that "old" codes are not required to fill gaps; they have in every instance been supplanted by newer versions.

As indicated in Table 2.5, there has been an ongoing support of the network codes by the link codes, wherein the latter have functioned as standards for calibration purposes or have provided subroutines or data to the network codes. This process is essential to maintaining the quality and timeliness of the network models; however, there is usually a lag between upgrades of the link codes and the subsequent feeding of results or modules to the network level. This is evident from the preponderance of "old" link codes that have been involved in
the process to date. It is anticipated that the release of the WECOM codes, and others, will spur the incorporation of improved subroutines and modules into the network codes. Thus, although the provision is a fait accompli to the network analyst at the time of his use of the network code, the database support of the engineering codes is still useful.

Although strategic C3 analysts are compelled, generally, to curtail detailed link calculations, there are no established priorities for deciding when these calculations should be undertaken. A useful set of guidelines would need to reflect the network code limitations, the scenario, and the communications media used. The following paragraphs illustrate the key factors involved in validation and case studies.

The need for validation arises from the unavoidable reliance on aggregated data in the functional level codes and the less-than-rigorous treatment accorded higher order effects. Thus, for example, if a large network simulation revealed that connectivity depended critically on an adaptive HF link, it would be prudent to verify closure using WEFCOM. Two considerations are apparent. The first is that a critical link merits further attention. The second is that some classes of communications systems might have characteristics that put them in the "zone of unreliability" of the analyst's network code. On this account, he may wish to take out some technical insurance. For example, if his network code computed ray paths by interpolating between stored ionospheric profiles, it may not fully account for disturbances in the E and F layers.
Table 2.5
Codes Contributing to Support of SIMSTAR, WAAM, and SCAP/NSCS

<table>
<thead>
<tr>
<th>Link codes</th>
<th>SIMSTAR</th>
<th>WAAM</th>
<th>SCAP/NSCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub-routine</td>
<td>Lookup</td>
<td>Calibration</td>
<td>Sub-routine</td>
</tr>
<tr>
<td>WEDCOM</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>WESCOM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IONCAP</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SIMBAL II</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Old link codes</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SIMBAL I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WESCOM/SATL</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>WAVEGUID</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>ANIHOP</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>HFNET</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>GPSNR</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WREC3 V</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>RANC IV</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUCOM/BREM</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>WEPH</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SPREAD DEBRIS</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WGL</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Case studies comprising excursions, optimizations, and planning assessments provide a strong rationale for enlisting engineering codes.

Once the analyst has concluded that an engineering code is required, the question arises, "Which code?" There are two categories of engineering codes: propagation and link.

Propagation codes compute signal level or signal-to-noise level over a specified link. Typically, they include benign environment and antenna models. Although these codes do not accept nuclear scenario input in the form of burst events, one can enter the standard "uniform spread debris" ionization profiles. The propagation codes that NTIA, NRL, and NOSC have developed fall into this category.

Link codes compute bit, character, or message error rates over specified links (usually, several links are defined). These codes include both benign- and stressed-environment models, antennas, and modems. The DNA WECOM codes are of this type.

The analyst's approach for handling the nuclear environment is clearly an important factor in selecting a code. Other factors that may enter the decision process include fidelity, simplicity, speed, accuracy, transportability, output format, user friendliness, and reliability. Should some means be provided to help the analyst make these choices? At minimum, the analyst would benefit by consulting available documentation and experts at DNA, NTIA, etc.

Since one envisions the strategic C3 community using engineering models primarily in a backup role, it is desirable that the results of those models be comparable with the network models, at least to the degree achievable without modifying source code. Clearly, some measure of equality is enforceable at the input stage, but there are several obstacles to achieving the desired level of compatibility:
• Internal databases
• Load-time databases
• Input logic and data specifications
• Modeling approach.

Each of the engineering codes has internal databases, which may differ from those in the network codes or in other engineering codes. For example,

• WEFCOM, IONCAP and HFUMFES have different internal HP antenna databases.
• The modem models in WECOM are not always identical with those in the network codes, except possibly for the classical modems. There are several reasons for this. Many of the fielded systems have undergone a series of upgrades, in which EDAC codes, interleavers, etc., were tweaked, and the models have not caught up. Modem performance data for prospective systems, such as MILSTAR, are closely held and typically are treated differently depending on the level of classification.
• Although the nuclear phenomenology databases are now consistent across the WECOM codes (since they share the same stress modules), the environmental data in the network codes are decidedly different.
• The propagation codes have no internal data for disturbed environments at all. The user has the option to supply a digitized form of the uniform spread debris ionization profiles at load-time.

Sponsors or proprietors of codes must face such questions as whether codes should be modified to allow user control of important internal databases and how sensitive or proprietary data should be handled.

Many of the engineering codes have load-time databases in addition to their internal databases. These are more directly under user control, but any changes must mimic the formats used in the data tapes supplied by the developer. Therefore, another question arises: What SSMF Database architecture should be adopted to simplify reformatting of load-time data for each code? Examples of load-time data include:

• Atmospheric noise models. These are of different vintage in HFUMFES and IONCAP; VLFACM computes noise using the WCL model, so the load-time data consist of thunderstorm distributions rather than noise statistical moments. These distributions are not fully compatible with the NTIA model.
• Disturbed-ionosphere data. In the future, WECOM may accept user-defined disturbed ionospheres at load-time.

Another problem that will impact the choice of database management system is the presence of logical branches and equivalent data specifications in input lists. Logical branches involve the use of data values that redefine subsequent entries. These, and other symptoms of "crafty" coding, would require that a knowledge engine be allied with the database management system if it were desired to automate the generation of data files. Similarly, translation between alternative representations of ephemerides, antenna gain, etc., for code inputs would require complex data dictionaries.
Some discrepancies in model results will persist in spite of efforts to standardize data inputs. These discrepancies are due to differences in modeling approach. For example, noise values computed using the NTIA and WGL models will generally disagree to some extent, although not often seriously. Modal and wavehop longwave propagation models have been known to produce different results when in principle they should agree. Ionospheric anomalies are treated differently in IONCAP and HFMFUES, which generally produce different results. In some instances, it may be advisable for the SSCF Database documentation to include expected variances in link performance predictions that are due to modeling differences.

RECOMMENDED CODE SUPPORT

Table 2.6 shows our recommendations concerning which computer codes the Database effort should serve. As DNA has pointed out, the community would be better off if a number of the codes:

<table>
<thead>
<tr>
<th>Codes</th>
<th>Support Now</th>
<th>Monitor</th>
<th>Do Not Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network codes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIMSTAR/STRATCAM</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WAAM</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TW/AA</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DETECC</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PNC</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>MINEFIELD</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>NSCS/SCAP</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link codes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WEDCOM</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WEFCOM</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WEMCOM</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MBLINK</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WSHCOM</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WETCOM</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIMBAL II</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LWPC</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODEFNDR</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEGMENTED W/G</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IONCAP</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>HFBC</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GWAPA</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRCOM</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MNP</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ELF PCA</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Old link codes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HFMFUES</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPP</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANIHOP</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WAGNER</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EFM.73</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIMBAL</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUCOM</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other codes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WAAMAG</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SKYMAP, SIDAC</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
older codes were no longer used. We recommend serving the SIMSTAR/STRATCAM, WAAM, and NSCS/SCAP simulations; the WECOM codes; and SIMBAL II. Other codes, such as DETEC, IONCAP, and SKYMAP, should be monitored to determine when and if their needs should be addressed in terms of a standard database. The WAAMAG and other attack codes that the Joint Staff and DISA use should likewise be served. Omission of a particular model or code does not mean that we believe that code serves no useful purpose; rather, it simply means that we feel that compatibility between it and the SSCF Database is not required.

These various computer simulations, engineering models, and scientific computer codes are continually being updated and replaced. That makes the code support process more difficult, but it also provides an opportunity. These tools can be developed or modified with the notion of commonality and common-format interoperability as a part of the design philosophy.
3. THE STRATEGIC C³ DATABASE

In Sec. 2, we reviewed the strategic command and control communications (C³) analysis community—what they do and how they do it. From that survey, we have drawn conclusions concerning their input data requirements, i.e., what the strategic C³ database should include to support operations with the recommended codes. The subsections below describe the development of the common-format database concept, including the process of deciding what the database should and should not include; provide more detailed descriptions of the major elements of the database; summarize our findings concerning the prototype database representation; and discuss briefly the Ada and C languages, as they apply to the database design.

THE SSCF DATABASE CONCEPT

Here we will discuss the content, distribution, and applications for the Standard Strategic Common Format (SSCF) Database. This discussion includes some of the issues and thought processes that allowed RAND and the Joint Staff Strategic C³ Working Group to determine what the database should be and, for RAND, how it should be developed and formatted.

Determining the Database Content

In the Second Joint Staff/RAND Strategic C³ Workshop [16], we adopted the Joint Staff definition of Strategic C³ database:

A set of agreed upon, verified data to be used by the Strategic C³ community in conducting C³ analyses, or from which excursions can be made.

This definition has three key elements:

• Some form of quality control to arrive at “agreed upon, verified data”
• Community use
• Excursions from a standard baseline.

The first two elements have significant implications for database production and dissemination that are described here. The final element is significant because it shows the intent of Joint Staff and the other members of the strategic C³ analysis community to use variations in the C³ system configurations and the threat. This has implications with respect to database content.

There was general concurrence that the central “hard” physical data would make up the major portion of the database. It became clear, however, in our workshops and in the development of the database organization, that one person’s facts were another person’s assumptions. For example, it sounds rather straightforward to identify specific equipment at specific physical locations or on specific aircraft. But what does the analyst do when confronted with a retrofit program that involves a large fleet of aircraft or a number of ground facilities and might stretch out over two or three years? The analyst will seldom
know in advance—probably because it is unknowable—the actual retrofit and checkout rate. Typically, the analyst would assume either that all of the equipment has been retrofitted or that none of it has. The question of actual versus potential communication links between nodes is another case in point. If Emergency Action Plan (EAP) Volume VII calls for a radio operator with two receivers to monitor four frequencies in a given order of priority, how does the analyst take this into account? It was first suggested that the database need not contain such information, that it would be the responsibility of the analyst to decide message processing rules and delay statistics. The consensus of the working group, though, was that the procedural data must be a part of the standard database. Variations in procedural data are the largest source of differences between study results. It was agreed that the database should be organized such that procedural and scenario information would not be interspersed with, or buried in, the "hard" physical data. The different parts of the database should maintain their discrete identities. The database organization discussed below must also take into account distribution of parts of the database (we call them data sets) as opposed to the entire database.

The Working Group initially agreed that an Office of Primary Responsibility (OPR) and an "as of" date would accompany each item or database record. However, it was suggested that doing this on an item-by-item basis would load up the database and—more importantly—the analyst quite substantially. A more pragmatic approach would be to assign an OPR and an "as of" date to blocks of data. Although data storage is becoming so cheap that tagging items rather than blocks of items is quite feasible, the pacing item is the effort required by the analyst in performing the tagging operations. The database structure provides flexibility so that the degree of blocking in various data items can be accommodated to the structure of the attribution sources. In any case, the files can be so manipulated that these "overhead" items are invisible to the analyst in most operations, called up only when needed. It is clear that this information could be of great value to an analyst attempting to assimilate and integrate this common database into the analysis at hand, but can be a distraction in the presentation of data to a sponsor.

We all agreed that a standard database, if it does nothing else, should lessen the analyst's data collection efforts. We also acknowledged that there is no substitute for clearly defining the problem, that having a standard database does not relieve the analyst of the responsibility of fully understanding what he is modeling. Nonetheless, some important uses of the database require running different models under assumptions that are as similar as possible. For this reason, we recommend that the database support a common definition of message processing and functional relationships between nodes. These factors are the focus of Sec. 6 of this report.

Clearly, the database cannot be open and must be bounded. For example, the current Five-Year Defense Plan (or the configuration at the end of a particular such plan) represents a bounded set, but including all contemplated new systems and procedures would render the database open ended and therefore unmanageable. On the other hand, the database manager should understand what complete input is required to run each of the models, the standard database supports and should support at least those inputs that are explicitly or implicitly common or comparable among models. Not only will this aid each analyst using that model, it will identify important degrees of freedom. If different assumptions were made by different analysts using the same model (or using different models), failure to account for such
variables would result in the same kind of confusion that led to the 1984 workshop in the first place.

The following issues served as a checklist in our design process:

- What items should be included and at what level of detail?
- Who will be responsible for initial construction and ongoing maintenance?
- What database management system (DBMS) should be used (if any)?
- Where and on what type of machine(s) should the database reside?
- Should there be separate versions of the database at multiple levels of classification, or should access controls be embedded in the DBMS?
- How will users access the information?
  - By distributing multiple copies of the database and requiring the user to have the appropriate DBMS?
  - By "dial up"?
  - By requesting generation of specific reports?
- How should information be collected and verified both initially and on a periodic basis?
- How can access be controlled without compromising utility to a wide group of users?

In our internal deliberations and analysis, workshops, and technical interchange, we and the user community developed a shared view of the problem.

**Database Content**

In Sec. 2, we identified the strategic communications computer codes used by the strategic C³ analysis community (the Working Group). For these codes (and a few others that were still under consideration at the time), we extracted or constructed input data tables; i.e., we determined the specific input information required to use each computer code or program. Our initial cut at the SSCF Database essentially consisted of the union of those input files. From this initial review and a series of discussions with various members of the Working Group, we developed a candidate list of SSCF Database categories.

One finding of our study has been that no two modelers think alike concerning the arrangement of their input data. Almost any arrangement is satisfactory, but for a database intended for a wide variety of users, some order must be attempted. The following rough outline is typical and covers all categories we encountered on this study, but could be expected to grow (and perhaps to warp) during the development and maintenance of the database over an extended period of time:

1. Nodes
   1.1. Node Name
   1.2. Attributes:
   1.2.1. Function Classes: Message Originator(s), Command Centers/Command Posts, Communications Relays, Force Elements
1.2.2. Location Classes: Fixed Surface, Mobile Surface, Mobile Airborne, Mobile Exoatmospheric

1.2.3. Specific Identification: World Aeronautical Chart Identification Number, Category Code

1.3. Location Data

1.4. Hardness Data

1.5. Complement (Equipment Lists, Characterizations)

1.5.1. Transmitters

1.5.2. Receivers

1.5.3. Antennas

1.5.4. Modems

1.5.5. Power Supplies

1.6. Node-Specific Hardware Parameters

1.6.1. Message Processing Rules

1.6.2. Hardware Priorities

1.6.3. Buffer Sizes

1.6.4. Packet Parameters

2. Equipment

2.1. Nomenclature (Name, Class)

2.2. Transmitter Parameters

2.3. Receiver Parameters

2.4. Antenna Parameters

2.5. Modem Parameters

2.6. Power Requirements

3. Network Topology

3.1. Connectivity

3.2. Message Structure

4. Procedural Data (Including Timing)

4.1. Emergency Action Messages and other Formatted Messages

4.1.1. Origination and Injection

4.1.2. Transmission

4.1.3. Reception

4.1.4. Processing Delays, Reactions (Message Effect)

4.2. Unformatted, Nonstandard Communications

4.3. Conferencing

4.4. Resource Deployment and Employment

4.5. Queueing Protocols and Endogenous Events

5. Environment

5.1. Benign

5.1.1. Dates, Time, or Epochs

5.1.2. Atmosphere

5.1.3. Ground

5.1.4. Electronic Interference

5.2. Threat (Including such Nomenclature as Weapon Type and Class)

5.2.1. Ballistic Missiles

5.2.2. Bombers
5.2.3. Cruise Missiles
5.2.4. Space Weapons
5.2.5. Jammers
5.2.6. Ground Forces
5.2.7. Unconventional Forces
5.3. Attack
5.3.1. Nuclear Bursts (Laydown and Damage)
5.3.2. Jammer Allocation and Electronic Attack (Description and Damage)
5.3.3. Other Physical (e.g., Spetsnaz)
5.4. Other Exogenous Events

6. Simulation Parameters (case name, runtime flags, report generation requests, etc.) (NOT IN DATABASE)

The Database “Package”

We visualize that the database manager would send out a database “package” at the request of the user and upon receiving approval by the database executive. This package would consist of

- All or selected parts of the database;
- A glossary;
- Support software (data translation and utility files); and
- User reference documentation, the Database Description and User’s Guide.

It is not unusual for an analyst to receive a magnetic tape containing several megabytes of data, but without any accompanying specific information concerning what the tape contains or how to read it. Our recommended package would therefore contain a source document that would explain what the analyst needs to know about the data.

The Database Description and User’s Guide should provide the analyst with an understanding of the nature, capabilities, limitations, and use of the SSCF Database.

The following is a suggested outline:

1. Introduction
   1.1. Background and Purpose
   1.2. Strategic C³ System Analyses
   1.3. SSCP Database Overview
   1.4. Contents
2. Strategic C³ Systems and Operations
   2.1. Control of Nuclear Weapons
   2.2. C³ Functional, Engineering Analyses
   2.3. Threats to C³ Operations
   2.4. Data Sources
3. Functional Assessment of C³ Networks
   3.1. Description of Functional Assessment Codes
3.2. Network Database Acquisition
3.3. Network Database Description
3.4. Ground Rules and Assumptions
4. Engineering Analysis of C³ Links
4.1. Description of Engineering Analysis Codes
4.2. Link Analysis Database Acquisition
4.3. Link Analysis Database Description
4.4. Ground Rules and Assumptions
5. The Strategic C³ Database
5.1. Formation of the SSCF Database
5.2. Description of the SSCF Database
5.3. Application of the SSCF Database
5.4. Database Management Philosophy
5.5. Ground Rules and Assumptions
6. SSCF Database—Operating Instructions
6.1. Obtaining the SSCF Database
6.1.1. Procedures
6.1.2. Database/Computer/Operating System Specification
6.2. Excursions and Configuration Control
6.3. Security

At some time, it may make sense to split this document into two volumes but a single volume appears appropriate for now. The Database Description and User's Guide would be a companion to the data tape. As a matter of fact, the latest version of the guide should probably be put on the data tape itself.

It has been suggested that Secs. 2, 3, and 4, and parts of Sec. 5 of the Guide seem to be out of place in a document intended to guide the database user. That is not the sole—or even the primary—function of this document; however, it was clearly the original Joint Staff intent to make this a source document that would provide the analyst with a broad understanding of the SSCF Database. That approach still appears sensible to us.

If a user is receiving the Database Description and User's Guide for the eighth time, it probably will not be necessary to read all the fine print. On the other hand, there is a fairly large turnover in most of the strategic C³ analysis organizations, and this source document should serve as a useful teaching (or self-learning) tool.

Items 3.4, 4.4, and 5.5, ground rules and assumptions, are included. Differences in ground rules and assumptions are major sources of modeling and analysis discrepancies. As an alternative, a separate section or even a separate document should contain a discussion of ground rules and assumptions.

Database Application

Figure 3.1 depicts the use of the database and support software to extract a data set for use in a particular model. The analyst would log on and access the database support software (Step 1). Using these tools, he would read (and perhaps modify) various elements of the
database and construct a partial version of the database (Step 2). (We call the product a “data set.”) The analyst would then use the database support software to extract “Data set A” from the database, sending it over for use with Model A (Step 3).

The software support tools should also be used in database development. Step 1 of Figure 3.2 shows the analyst accessing the database support software. Using these tools, the analyst can extract data sets (Step 2) from those used to run Model B. The database support software would also be of value to the analysts even when manually extracting information.

Fig. 3.2—SSCF Database Development Using “Raw” Inputs from Model B and External Data Sources
from hard copy documents (Step 3M). The analysts (Step 3E) could also use the software support tools to enter the data in the common format for entry into the database itself.

An analyst whose primary concern is entering a particular data set into a model for an important analysis will tend to look at the database from the standpoint of "getting the model-specific inputs right." On the other hand, an analyst who is constructing the procedural part of the Database from a plan or regulation is more likely to want to verify that the database accurately reflects that input document. The nature of the database and database support software the analyst desires might well differ somewhat, depending on the relative importance of the two tasks. Both tasks are necessary and the software development should take both into account with about equal emphasis. (It might be pointed out that Step 2 of Figure 3.2 offers an opportunity for the analyst to "close the loop," checking to see if the data he would "pull out" of Model B looks like what he sent over to it in Step 3.)

DATABASE CONTENT—SYSTEM DESCRIPTION

Here we will treat the "hard core" data for the SSCF Database: nodes/facilities, communications equipment, and network topology. This will be followed by discussions of procedural data and environment, with emphasis on the threat and attack. Much of this description is adapted from Reference 17, which is a logical representation of the data-generation process and data sources. The up-to-date, classified source material was not available at the time of this study, but the database can be characterized adequately at an unclassified level.

Facilities Data

The facilities data are derived primarily from the periodically updated Joint Staff description documents, as well as from detailed commander-in-chief (CINC) and Service documentation. Modifications to the data associated with system changes not yet documented in the system description or changes to the node list associated with hypothetical changes (e.g., new programs for the Planning, Programming, and Budgeting System, Joint Staff, and CINC master plans) are obtained from various sources, such as reports published by the appropriate systems program offices (e.g., Advanced High Frequency [HF], MILSTAR). Force element locations are specified by the user organizations, although a standard should be specified by the Joint Staff. Node vulnerabilities, which are used in damage models, such as the SIDAC [18], are developed by the Defense Intelligence Agency (DIA). A central repository exists at the Defense Information Systems Agency (DISA), but it is not clear that these data are compared or correlated between working group members. A current DISA effort is underway to improve this situation, however. The specific latitude and longitude of each of the C2I facilities can be obtained from the Joint Resources Assessment Database (JRAD). The physical attacks used in Joint Staff analyses use JRAD latitudes and longitudes as aim points. If another group uses a different set of latitudes and longitudes, it can skew the physical damage assessment results somewhat. The JRAD also provides the category code and World Aeronautical Chart identification number. The category code is typically used together with the installation name to choose the correct JRAD record for the specific type of C2I capability under examination, e.g., an HF transmitter site as opposed to a low frequency (LF) or very low frequency (VLF) transmitter site, all three of which may be located in the same complex.
We discuss threat and attack in more detail later in Sec. 3, but it should be noted here that the damage files (i.e., the expected damage as a function of time for each facility) usually become part of the facility file when input into discrete event simulations.

**Communications Systems Data**

In this section, we will discuss communications equipment and systems and also the environment in which the communications systems will operate—both benign and hostile. The hostile environments can be invoked by nature (e.g., noise) and by man (e.g., jamming and nuclear detonation). We will discuss the operations of radio communications systems from the "Blue side" here. We will view the attackers' problem, i.e., "link attacks," later. The node/facility equipment complement should list transmitters, receivers, antennas, modems, and power supplies.

For each of these types of equipment we need to know the nomenclature (name, class); the technical parameters, e.g., transmitter-effective isotropic radiated power, noise figure, antenna gain, and modulation technique; and the power requirements.

The analyst requires data that describe the propagation medium and the technical characteristics of the radio equipment. The propagation medium may include the ionosphere, whose properties undergo frequency-dependent, spatial, and temporal variations. Equipment characteristics are not limited to hardware design parameters, but also include, to some extent, the structure of the signal being processed. For many links, transmission frequencies are specified by operation plans, but because of the temporal and geographic variability of ionospheric support of HF propagation, HF frequencies are often assigned on the basis of propagation prediction calculations (as functions of time and link) to attempt to optimize communications performance. The usual assumption is that the operator selects the optimum HF link frequency.

The engineering codes typically represent radio link performance more accurately than do the faster running simulations but usually require more sophisticated inputs. As an example, input requirements for IONCAP and SIMSTAR for HF communications are shown in Table 3.1.

**PROCEDURES AND OPERATIONAL DATA**

At the beginning of the study, we were not sure that procedural and operational data would or could fit into the standard database. In this section, we will examine this complex set of data in three steps:

1. Classification
   - Types of actions
   - Relationships between actions
2. Description
   - Specific Actions
### Table 3.1

**HF Communication Input Requirements for IONCAP and SIMSTAR**

<table>
<thead>
<tr>
<th></th>
<th><strong>IONCAP</strong></th>
<th><strong>SIMSTAR</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ionospheric environment</td>
<td>Beginning and end times of calculation (UT) (in Universal Time)</td>
<td>Year and month (determines seasonal noise map for fall, winter, spring, and summer)</td>
</tr>
<tr>
<td></td>
<td>Calculation time interval</td>
<td>Hour and sunspot number</td>
</tr>
<tr>
<td></td>
<td>Year, month, and sunspot number</td>
<td></td>
</tr>
<tr>
<td>Path description</td>
<td>Receiver latitude and longitude</td>
<td>Receiver latitude and longitude</td>
</tr>
<tr>
<td></td>
<td>Transmitter latitude and longitude</td>
<td>Transmitter latitude and longitude</td>
</tr>
<tr>
<td></td>
<td>Receiver in degrees</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Altitude for aircraft</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Transmitter power in kilowatts</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Required signal-to-noise level in decibels</td>
<td></td>
</tr>
<tr>
<td>Link description</td>
<td>Frequency</td>
<td>Frequency</td>
</tr>
<tr>
<td></td>
<td>Transmitter power, noise, minimum take-off angle</td>
<td>Receiver bandwidth, baud rate</td>
</tr>
<tr>
<td></td>
<td>Required reliability signal-to-noise ratio</td>
<td>Transmitter bandwidth and transmitter EIRP</td>
</tr>
<tr>
<td></td>
<td>Transmitter and receiver antenna type and characteristics</td>
<td>Modulation type (NPSK, FSK, etc.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Antenna type (dipole, rhombic, etc.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Error correction algorithm</td>
</tr>
<tr>
<td>Nuclear degradation</td>
<td>Burst latitude, longitude, altitude</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Yield and fireball temperature</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fission fraction</td>
<td></td>
</tr>
<tr>
<td>Jammer data</td>
<td>Latitude, longitude, altitude</td>
<td>Effective radiated power</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Frequency and bandwidth</td>
</tr>
</tbody>
</table>

— Communications
— Command and control, support

### 3. Development and documentation

- The analysts' responsibilities

We conclude that all of the procedures required to establish, maintain, and use strategic C³ systems can be represented in the common format described in this report. Representing procedures, however, is the most difficult and complex part of the database task. Extracting the various procedures from communications plans, regulations, and systems descriptions usually involves paging back and forth between more than one information source. Judgement calls are necessary, and the learning curve is not as steep as with hardware, where the parameters are repetitious and purely technical. On the other hand, our limited experience indicates that, once an analyst is into the routine of creating database records, the task is relatively straightforward, though often time consuming. Furthermore, this part of the database is not as dynamic as one might think. It reflects fairly deliberate changes of the communication plans in the field. Once the plans are represented in this database, modifying them will probably be less time consuming than adapting to changes in the C³ systems themselves. Since human interpretation is not always consistent, especially for
complex procedures, the most important (and we believe valuable) part of this effort will be in setting down these procedures in a concise, unambiguous fashion.

The product of this effort has been a representation of procedures and operational data in which as many degrees of freedom as possible are removed; i.e., the procedures are as completely specified as we can make them. The intent is not only to reduce the analyst's workload but—more importantly—to require fewer choices and interpretations. The differences between models will always produce some differences in simulations, but a database constructed with more than one simulation (e.g., WAAM and STRATCAM) in mind can reduce these degrees of freedom to a minimum and simplify the task of resolving differences that inevitably arise.

As we will use the word here, procedures will represent the established methods or actions prescribed by Joint Staff and Service plans, regulations, and operations orders. Procedures and operations will entail not only the specific acts required to make the strategic communications network function, but also the time required for, and the probability of success of, carrying out these specific acts. The sources for timing and probability data are typically operational exercises and engineering tests.

Representation Categories

Table 3.2 shows the manner in which we propose to describe and specify procedures and operations.1 Each of the four major blocks—directed actions, implicit actions, timing, and probability—form separate and distinct parts of the procedures and operations database. From time to time, some data elements may be so intertwined that the data records show up as a “merger,” e.g., probability as a function of time; however, in general, we believe it will be useful to preserve the distinctions between the four classes of data.

Directed Actions

Directed actions are those assigned responsibilities that are explicitly specified in authoritative sources, such as the following:

- EAP Volume VII
- Strategic Air Command (SAC) Regulation 55-14
- SAC Regulation 55-56 (Volumes I–III)
- SAC Regulation 100-7
- SAC Regulation 100-20
- Navy Operations Order 32

---

1B. Stubbs of DISA suggests another way to organize these classes of information:

Consider instead a hierarchy of actions. At each level an action is characterized by how long it takes and what subordinate action it causes to happen. Trigger events may be explicit or implicit and the duration and triggering among subordinate events may be probabilistic.
### Table 3.2

<table>
<thead>
<tr>
<th>Procedural/Operational Information</th>
<th>Typical Information Source</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directed actions</td>
<td>EAP Vol VII</td>
<td>Stated explicitly in implementing documentation</td>
</tr>
<tr>
<td>Implicit actions</td>
<td>Ground rule, assumption</td>
<td>Not explicitly spelled out in plans, regulations, or directives, but required or logically inferred</td>
</tr>
<tr>
<td>Timing information</td>
<td>POLO HAT results</td>
<td>Sometimes directive in nature, but usually based on exercise results</td>
</tr>
<tr>
<td>Probability information</td>
<td>Exercise results, engineering data</td>
<td>Equipment reliability, human factors, i.e., probability of successful action</td>
</tr>
</tbody>
</table>

These documents for the most part spell out what communications and communications support actions must be performed at various facilities (nodes) in the network. We limit our discussion here to EAP Volume VII, which is the primary implementing document. (It was also the only one available to us.) It should be pointed out that, in actual practice, the people operating various parts of the system use the SAC regulations and Navy Operations Order 32, rather than EAP Volume VII, as their guidelines. During one of our workshops, we toured LOOKING GLASS, and staff on the aircraft confirmed they use SAC regulations rather than EAP Volume VII. This has been demonstrated in the past to produce some differences in operation. It is not clear to us how significant these differences might be.

### Implicit Actions

In many cases, implicit actions that are not specified in a communications plan (such as EAP Volume VII) are required to perform the explicitly directed actions. There may be logical functional relationships between two actions. One particular action (e.g., VLF transmission from an aircraft) cannot be performed unless a prior action (extension of a trailing wire antenna) has taken place. The methods of performing a directed action are often unspecified. Explicitly directing an action can, in some cases, be considered “assigning a responsibility,” and the implicit actions can be considered to be specifying how that assigned responsibility should be carried out. Very often, the implicit actions tend to be undocumented or cryptically annotated ground rules and assumptions made by the analyst. This is not a criticism of the analyst, who is simply working with what information is available when performing the analyses that the organization needs. The model which the analyst uses may also contain built-in implicit actions. The SSCEF Database will simply include database records for these implicit actions, filling in the information gap for a standard or baseline interpretation of the implicit actions. It has also been suggested that we should identify and document these model-specific built-in assumptions.
Timing Information

Timing represents statistical data reflecting the range of times required to perform specific actions. In our usage, it will usually consist of a distribution type plus two parameters. Some timing information is specified in EAP Volume VII, e.g., the requirement for periodic retransmission of messages. Most timing data for strategic communications comes out of tests, such as POLO HAT, and represents empirical data on how the various elements of the network have performed over the recent past. The Exercise Message Data Analysis Computer Program (EMDAC) [19], provided by The MITRE Corporation C3 Division, defines EMDAC as an automated capability-management tool that is primarily dedicated to evaluation of EAM communication paths. POLO HAT analysts use EMDAC, which is embedded in ORACLE, to evaluate strategic connectivity network performance by

- Correlating EAM dissemination data for WWABNRES, launch control centers, ballistic missile submarines, and bombers (times of message receipt and retransmission)
- Inputting data to EMDAC using the ORACLE interface
- Validating data, matching valid and correct message receipts to transmissions.

POLO HAT transmits the following data:

- Transmitter
- Message number
- Time of transmission
- Communications system
- Communications mode
- Network
- Frequency
- Remarks
- Sequence number,

and receives the following data:

- Receiver
- Message number
- Time of receipt
- Validity code
- Transmitter
- Communications system
- Communications mode
- Network
- Frequency
• Remarks
• Sequence number.

The EMDAC outputs the following information:

• Listing reports
  — Transmission records
  — Reception records
• Analytical reports
  — Connectivity diagrams
  — On-board processing times
  — Platform activity and timing
  — Message delivery summaries.

It would appear that the EMDAC outputs could be converted to an electronic data set of the form desired here, but it is not clear how difficult this would be. Not all timing information required to run strategic C³ simulations can be provided to the analyst; thus, "implicit timing data," like implicit actions, get "invented." The invention is often buried in the ground rules and assumptions and is not usually well documented.

Probability Data

Probability indicates the likeliness of an event, such as achieving a certain system reliability in a test. Probability can be expressed as a scalar in percent or can have a specified distribution. Probability data are sometimes simply equipment reliability data, but often have to do with the probability that the human operators can carry out their responsibilities (directed and implicit actions) in a timely and successful manner. Data here will usually come from exercises, such as POLO HAT, and from test and evaluation reports relating to the reliability and maintainability of hardware.

These data are also likely to be buried in the ground rules and assumptions. Well organized and clearly documented ground rules and assumptions are therefore valuable both to the consumer, by aiding evaluation of the analyst's products, and to the analyst, by providing an approved diagnostic capability in the event that discrepancies arise.

Functional Relationships Between Actions

The method of modeling procedures should incorporate specific communication actions (the verbs). The functional relationships between these actions can be looked at as "attributes" associated with each (or at least many of) the specific actions. Seven such associations or functional relationships were extracted from procedures, planning, and exercise and testing documents:

• Input logic—The input logic specifies the dependence of an action on prior events (or on the knowledge of personnel at a particular facility). A particular procedure might need to follow the successful completion of another procedure or event. The default value is simply
the absence of a constraint. If no input records exist, no earlier procedure needs to have been accomplished to allow initiation of the specified procedure.

- **Output logic**—An output logic function allows subsequent events or procedures to take place. The default value here is also absence of a constraint. If no output records exist, no subsequent procedure requires completion of the specified procedure.

- **Source**—A source is a point of transmission of a message over an individual link, which in our usage means a specified transmitter for a particular communication. If a particular communications comes in from a single source, rather than multiple points of origin, that source is specified.

- **Sink**—A sink is a point of destination for a message, again over an individual link, which in our usage means the reception (and perhaps verification and authentication) of a particular communication at a specified facility. If a particular communication is intended to go to only one facility, rather than to be broadcast, that facility is identified as a sink.

- **Priority**—The priority is the hierarchical importance and/or time sequence, a ranking, sequencing, or importance order for a series of actions. Priority defines the relative importance of parallel actions. For example, an organization monitoring more than one source with a single receiver could be instructed to monitor three sources with a rank order specified. This almost invariably leads to implicit actions in terms of how the monitoring of the three frequencies is to be modeled. [The other example was “monitoring three frequencies in a particular priority order”—which falls after “importance order of actions.” Neither is especially clear.] The default value is an absence of an entry in the records, which means that no instructions are given in relevant communication plans; this does not necessarily mean that there is no competition for resources. The analyst may still have to assume implicit actions based on assumed priorities, which should then be documented in the ground rules and assumptions.

- **Timing**—Timing specifies the time required to perform the action. The default is zero.

- **Probability**—Probability indicates the likelihood that the action will be successfully completed. The probability data are usually assumed to be scalar and, in the sample records that follow, are expressed in percent. The default is 100 percent.

**Communications Procedures**

Our examination of planning documents allows us to specify two classes of actions that are necessary for making use of a large strategic communications network:

- **System or network procedures**—These are the specific actions associated with the activation, deployment, and operations of communications systems at various nodes in the network. The system procedures essentially constitute overhead, i.e., the cost of running the network.

- **Communications, message, or link procedures**—These are the specific actions associated with the origination, transmission, reception, processing, and relay of specific communications. The communications procedures represent the communications transactions that are the main business of the network.
Table 3.3 suggests an organization or structure for these actions. Although all of these actions can be described in terms of activity at the nodes in the network, the distinction between system and communications procedures seems useful. Many of the system procedures will show up as assumptions in strategic C^3 network modeling and analysis. The assumption is often made that the system procedure (e.g., activation, deployment) simply will have taken place. Most of the communication procedures or actions, on the other hand, will show up as input data for computer runs using the network codes.

Some analysts familiar with discrete event simulations have trouble with the notion of “monitor,” since monitoring, unlike transmitting or receiving, is not usually modeled as an event per se. Monitoring can be represented as a node or equipment state, and/or one can use a “duty cycle” model to represent the monitor and receive operations. Although it is not clear that the process of modeling the monitoring procedures is carried out in the same manner with the different codes and by the different teams, the monitoring procedures can be clearly spelled out. Furthermore, monitoring is an important part of the database, in that (1) it provides an accurate match to EAP Volume VII, and, more importantly, (2) it allows a later event—receive—to take place. Monitoring is an activity with start and (potentially infinite) stop times.

<table>
<thead>
<tr>
<th>Procedure Type</th>
<th>Purpose</th>
<th>Principal Action</th>
<th>Object of Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>Establish and operate the network</td>
<td>Activate or deploy</td>
<td>Node (e.g., order worldwide airborne command post, WWABNCP takeoff)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Equipment (e.g., deploy WWABNCP trailing wire antenna)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Network (establish, restore)</td>
</tr>
<tr>
<td></td>
<td>Operate</td>
<td></td>
<td>Network (control)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Link (establish, maintain, and restore)</td>
</tr>
<tr>
<td>Communication</td>
<td>Use the network</td>
<td>Monitor (maintain a listening watch on)</td>
<td>Priority</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Source System, frequency, mode, and polarity</td>
</tr>
<tr>
<td>Receive and verify incoming messages</td>
<td></td>
<td>Equipment activation (i.e., turn-on, self-calibration)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EDAC (i.e., repeats, piecing, rules, algorithms)</td>
</tr>
<tr>
<td>Process information</td>
<td></td>
<td></td>
<td>Originate</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Record, store</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Translate Requirements (other actions)</td>
</tr>
<tr>
<td>Transmit or retransmit messages</td>
<td></td>
<td>Rules (e.g., first in, first out)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Priority System, frequency, mode, and polarity</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Requirements (other actions)</td>
</tr>
</tbody>
</table>
With members of the Strategic C³ Working Group, we discussed how the database would respond to the very complex network control logic in such systems as MILSTAR or in certain main trunking communications systems in which the channel capacity is allocated in real time, usually based on formal procedures or protocols. The problem of pointable beam antennas and time slots was raised as a particular example. In short, some systems with complicated operational procedures require very complicated control logic. It was suggested that the scope of the SSCF Database (at least as represented by the lines of database records) should be limited to the data that appear in formal plans or regulations, such as EAP Volume VII. The use of macro protocols and object-oriented protocols was discussed. We suggest that a subroutine-like macro capability would be a useful extension at some point to the C³ description language, but the Volume VII review indicates such an extension is not currently needed.

**Command and Control Actions Within a C³I Facility**

Most of the Strategic C³ Working Group teams analyze EAMs; however, the RAND task required us to investigate the problem of tactical warning and attack assessment through force management, of which EAM dissemination is only a part. Important aspects of the end-to-end problem are decisionmaking and decision support. The following procedures are typical not only of a command center, but of any type of facility that supports a staff and commander of operational forces:

1. System procedures (e.g., activation, deployment)
2. Communications procedures
3. Command/staff procedures
   - Process information
     - "Reception processing"—formatted data from communications and automated data processing
     - Store
     - Verify, tag, sort, associate, organize
     - Present/display/print
   - Provide decision support
     - Interpret, validate, discuss, coordinate
     - Analyze, assess, evaluate, estimate
     - Formulate
     - Report
     - Advise/counsel
   - Request information
   - Make decisions
   - Implement decisions
The system and communication processes at a command center can result from actions by the commander and/or his staff, who can

- Ask for a few specific items of information;
- Request advice or an estimate of the situation; and/or
- Decide to confer with other senior commanders or officials.

The options above are in roughly increasing order of complexity and will tend to impose increasing loads on the staffs and the C3I system, including supporting military and intelligence staffs.

The command center (or a subordinate) staff prepares the staff estimate requested by the commander, which includes the following:

- The mission
- The situation and courses of action
- An analysis of opposing courses of action
- A comparison of own courses of action
- A recommended decision.

The estimate could permit the commander to make a decision, or the commander could act without this input. The possibility of using or not using an input suggests the existence of dependencies of actions on prior actions within the database. Certainly they exist in EAM dissemination; i.e., a facility cannot relay an EAM that it has not received. We will explore this notion of dependency in the next section.

For our purposes, the information, decision support, and decisionmaking processes need not all be modeled in precise detail. The types of information required can be described, and the amounts of information can even be quantified, albeit over a fairly wide range. An intellectual, dynamic, interactive ("hands on") leader would probably set the upper bound requirements for information and decision support. This type of commander would—if the situation permitted—tend to ask for more information and staff estimates, confer with other leaders or commanders, and generally exercise the C3 system more heavily than another commander, who might be inclined to simply delegate authority or give a subordinate commander a broad mission without much specificity as to how that mission should be carried out. Since the latter commander would be not getting into the details of the subordinate commander's operations, he would need less information from the subordinate. The first commander would, in both his decisionmaking and command execution, tend to "load up" the system and—for the analyst—exercise the database.

Developing the Procedures and Operations Database

In converting the material from a plan to a computer file, we visualize preparing a database description and user's guide. This guide will include quotations of text from the plan or regulation, each with a narrative discussion noting the degrees of freedom or choice left open to the analyst because of ambiguities in the original plan.
The following steps are an appropriate way to extract and document data from a communications plan:

1. Quote directly from the text.
2. Provide a narrative discussion of how this text could be interpreted for analysis purposes. (There is often more than one logical interpretation.) This discussion should include both explicit and implicit actions.
3. Describe explicitly the option or options chosen.
4. Convert the explicit action, implicit action, timing, and probability information to lines of database records.

Vital to the success of this effort are (1) interaction with the plan’s OPR to make sure that the database truly captures the intent of the planner, (2) documentation of any significant interpretations or clarifications of the plan obtained in this interaction, and (3) documentation of any ground rules and assumptions made in the absence of such guidance. This will certainly be required for the expected excursions from the baseline.

The database records could be entered in a number of ways, such as

- Direct entry, using a standard editor, which allows the time-saving (but sometimes treacherous) “clone-and-modify” technique
- Use of a menu-driven tool, such as that described in Sec. 5.

Other techniques, such as using an optical character reader, could be appropriate. Better, if the full documentation in the database system were set up with character recognition in mind, information from the communications plans and system description documents could be extracted using some of the newer artificial intelligence methods. These methods would certainly help in developing the procedures portion of the standard database.

The goals we have in mind include

- Assisting the analyst who develops this part of the database in representing accurately the communications plan or operating instructions;
- Providing a human-readable format for ease of comparison and verification; and
- Ensuring compatibility “at the other end” with the model(s) for which the database is designed to be compatible.

The process must be clear to an analyst who departs from the baseline so that “excursion procedures” can be developed that are consistent with the philosophy and intent of the baseline procedures and operational data. Several important, but longer term, issues are the following:

- Can a set of database records of this type be converted to both SIMSTAR and WAAM format in such a way that both models (and all future supported models) will—to the extent that the models themselves will allow—simulate the same actions? In other words, am I closer to enforcing parity than I was before?
• Will constructing this set of database records be worth the effort? The primary goal is to make sure that it is easier to use the standard database than to pull the material out of the communications plans, but a side benefit will be to spell out, in unambiguous terms, just what the plan sets forth.

• How long will it take to convert a full communications plan?

• Can we readily separate the events from the procedures? We want a scenario-independent set of procedures, at least to the extent that the communications plans are scenario-independent.

• Since the benefit will be shared, can the effort be shared? It seems that it probably should be—in part, too, because the expertise tends to be diffused.

ENVIRONMENT

Two of the most intractable areas in the analysis of general nuclear war are

• Predicting the various effects of nuclear weapons (especially when they are used in large numbers) on our command and control systems and facilities; and

• Postulating specific, credible attacks upon the United States.

As we have seen, the task of describing the Blue C³ network is not a simple one. It is even more difficult to obtain a consensus on the likely effects of the large-scale use of nuclear weapons, what a "logical" attack might be, or what conditions would exist during that attack. We will touch upon some of these issues during the forthcoming discussion. Fortunately for this task, however, the definition and the description of the weapons effects and attack scenario databases required for analytical purposes are relatively straightforward. We can easily construct templates that represent both the effects of nuclear weapons and the specifics of an attack, even though it can be very difficult to obtain agreement on the specific entries for those templates.

The so-called benign or peacetime environment is also an important part of the inputs required to evaluate radio communications performance. Important parameters include

• The physical characteristics of the ionosphere, including diurnal variations

• The sunspot number

• The characteristics of natural and man-made noise, especially in the area of the receiver

• The conductivity of the ground or water surface over which radio communications take place.

After a brief discussion of the benign environment data, we will review two aspects of the development of the attack scenarios, both of which have database management implications.

To evaluate a C³ network under attack, we need to define that attack. For many analysts, that attack will be provided. For others, the analyst is required to develop his own attack
scenarios. For master planning and operation planning purposes, the analyst is likely to use the specified attack as a baseline, recognizing that new systems or procedures will require modifying the attack scenario to accommodate the logical changes that an attacker would make when confronted with a different C^3 network.

**Benign Environment Effects and Data**

Standardizing propagation and environmental databases and assumptions is necessary to ensure that the baseline analyses of various organizations are consistent. In Sec. 2, we discussed several standard propagation models, noise models, and environmental data files; e.g., the noise tables from the National Telecommunications and Information Administrators/ITS, CCIR, and Naval Research Laboratories. As noted also in Sec. 2, the Defense Nuclear Agency supports development of the majority of the communication analysis codes for nuclear conditions, while various agencies of the Department of the Interior develop and publish data relating to geographic distribution of radio noise and ground conductivity. These agencies independently release new information reporting the progress of their internal research programs.

The Joint Staff does not enforce standardization on all users of these databases but standards must be achieved if an objective basis for strategic C^3 modeling is to be produced. The SSCF Database manager can encourage and simplify the standardization process by serving as a clearinghouse for propagation codes and related environmental databases. The database manager should monitor the new releases of relevant propagation data and software and advise the C^3 community of the new products and either disseminate those products or the information on where to obtain them. This information should, at a minimum, be incorporated into the Database Description and Users' Guide, if not in the electronic database as well. (Obviously, credit of authorship would be directed to the originating organization.) Since much of the propagation data is either unclassified or at a low level of classification, a possible mode of dissemination would be to create and maintain an on-line, electronic bulletin board from which registered C^3 users could access the latest news on C^3 product releases and exchange notes on errors or problems found with some of the products. Further, with the consent of the producing organizations, new database products and code patches could be posted on the bulletin board for direct downloading by users desiring to obtain either the latest data or to fix errors in their link analysis software. Depending on the volume of information distributed from the C^3 archive, the archive could either be a stand-alone system or be implemented as part of the MILNET or ARPANET. With such a system, the dissemination of standardized environmental data could achieve the effect of standardizing C^3 modeling not by enforcing a rigid conformity on the community but by facilitating the distribution of the latest information to the user community.

In summary, this material would not be a part of the SSCF Database per se, but would be referenced and briefly described therein, and—with the concurrence of each originating organization—possibly disseminated with the SSCF Database.
Nuclear Weapon Laydown Attacks

Red attacks against Blue C^3 networks can produce

- Physical damage or destruction of our C^3 facilities (the laydowns); and
- Disruption or denial of radio links by jamming, blackouts, or scintillation.

The analytical development of a (necessarily hypothetical) Red attack must include

- Threats, i.e., which resources the enemy can bring to bear; and
- Effects, what the likely results of bringing these resources to bear in a specific attack would be.

An extensive database describing the Red resources is required in the development of attack scenarios. The laydown involves attacks by ICBMs, SLBMs, bombers, and cruise missiles and potentially, in the future, space weapons. In the strategic C^3 arena, virtually all jammers are known radio transmitters. Exoatmospheric bursts could come from ICBMs, SLBMs, or possibly Soviet spacecraft. Although not usually incorporated into the hypothetical Red attacks, unconventional warfare, selectively applied, could represent a significant threat.

Threat, attack, and weapons effects codes and databases include:

- Red Integrated Strategic Operations Plan: Joint Staff JAD
- JRAD: DISA
- Single Integrated Damage Analysis Capability (SIDAC): DISA
- Threat Assessment Model (TAM): AFCSA
- NUCWAVE: DISA
- CLUSTER: DISA
- WAAM Attack Generator (WAAMAG): DISA
- SINBAC: LOGICON
- STRIKE: TRW
- NWEM: AFCSA, DNA, TRW
- HEMP: HQ SAC, NRA
- Vulnerability Numbers (VNs) (Red): DIA/DB-4C
  VNs (Blue): DISA
- EVOLVING THREAT TO WWMCCS: DIA/DT-4.

The RISOP was discussed in Sec. 2. SIDAC, TAM, NUCWAVE, and CLUSTER are tools that can be used to construct a laydown and determine the effect of that laydown on a set of communication nodes. The WAAMAG is used to develop attacks on C^3 systems and is apparently being considered for “marriage” to the SIMSTAR code, since SIMSTAR lacks an attack generation facility. The JRAD is a compendium of material on Blue C^3 facilities, including estimated hardness as measured by the VN system of DIA.
Although, for some users, the RISOP would become part of the standard database addressed in this survey, the process of developing an attack and the associated database requirement require some discussion here. The reason for this is that some users will not receive the RISOP, and most users will need to develop excursions from the RISOP or whatever baseline attack they use.

It is very important to recognize that any attack developed by an analyst is a capabilities plan. The actual intentions of the Soviets are not known.

The analyst needs information on enemy force size, location, disposition, and capability, which includes warhead yield, reliabilities, ranges, speed, and accuracy. Specialized information, such as susceptibility to fratricide, dust, and the like, is also important for hardpoint targets. Force disposition can also include bomber and cruise missile routings.

A computer code, such as the SIDAC, is used to conduct physical vulnerability analysis. SIDAC is used to calculate the damage produced by two broad categories of nuclear effects: (1) prompt effects and (2) fallout. The prompt effects have usually been considered more important than fallout for most prior analysis efforts. The prompt effects considered in SIDAC are blast (overpressure and dynamic pressure), ground cratering, thermal radiation, initial nuclear radiation, and other phenomena. Fallout effects are the delayed effects associated with the residual nuclear radiation carried by dust, bomb debris, and other matter transported by winds. However, fallout effects have not been incorporated in most strategic C³ analyses to date.

AFCSA is currently developing TAM, which incorporates (1) vulnerability data for each asset, (2) weapons data via a "strike tape," and (3) calculations of damage assessment and the probability of upset and recovery. The damage (or "kill") and upset probabilities are input to TAM as lookup tables that express the damage and upset probabilities as a function of offset distance. The sources for this lookup data are not known.

The basic physical model used to predict prompt damage is based on the Physical Vulnerability system of the DIA. Input information required to calculate damage includes weapon yield, desired location of detonation relative to the target (including height of burst), accuracy of the delivery system for the weapon, and overall probability that the weapon arrives and detonates. The majority of the node vulnerability calculations in strategic C³ analyses are based on the use of the moderate damage VN, although severe damage criteria are also frequently used.

For buildings, the criteria for moderate and severe damage are as follows:

- **Severe structural damage** is the degree of structural damage to a building that precludes further use of the building for the purpose intended without essentially complete reconstruction or replacement. A building sustaining severe structural damage requires extensive repair before it can be used for any purpose.

- **Moderate structural damage** is the degree of structural damage to principal load-carrying members (columns, trusses, beams, load-bearing walls) of a building that precludes effective use of the building for the purpose intended until major repairs are made.
For military equipment, the damage criteria are as follows:

- **Severe damage** is the degree of damage that is sufficient to prevent accomplishment of any useful military function and that requires either replacement of the equipment or extensive repairs at a major repair facility.

- **Moderate damage** is the degree of damage sufficient to prevent any military use of the equipment until field repairs are effected.

The conditions that will inflict moderate damage (usually the correct criterion to use for analyses of strategic communications networks) are, unfortunately, the most difficult to determine. The light and severe damage VNs can be calculated with much greater precision than can moderate damage VNs.

For soft C³ targets, the single-shot probability of kill is unity, and the single-shot damage expectancy is simply equivalent to the attacking weapon’s system reliability.

In most analyses, aircraft that are airborne at the start of an attack are considered 100 percent survivable with respect to the physical attack. Aircraft on the ground at the start of an attack are subject to attack until the time they reach safe flyout distance. Safe flyout distance is defined as the distance from the center of the airfield at which an aircraft can survive with 95 percent probability from a nominal 1-megaton ground burst in the middle of the runway. Safe flyout time is similarly described as the time at which an aircraft reaches safe flyout distance. Both parameters are functions of aircraft type, individual alert status, and airfield location. For example, it would be expected that the safe flyout time for a particular aircraft type would be shorter in a generated posture than in a day-to-day peacetime posture.

**Link Performance in a Hostile Environment**

We will now discuss briefly the evaluation processes and required inputs in the analysis of strategic C³ link and network performance in a hostile environment. The analyst must calculate propagation statistics for both benign (signal-to-noise ratios) and disturbed (signal-to-interference, i.e., signal-to-noise-plus-jamming ratios) conditions. The SIMBAL II program computes attenuation due to blackout from high-altitude, low-altitude, and ground bursts. SIMBAL II must be run for each time of interest in the scenario; the analyst must take this into account because the different models do not always calculate link performance for times. These timing considerations are an analyst’s function, however, not a part of the database.

Considering HF as an example, inputs into IONCAP and SIMSTAR were listed in Table 3.1. Input data requirements for SIMBAL II and WESCOM are listed in Table 3.4.

If specific jammer allocations are made a part of a specified attack, such as the RISOP, they should be part of the database. If the jammer allocation has not been specified, the jammer performance must be calculated at all potentially required frequencies, and the analyst should carefully document and report the resulting jammer allocation.

Of the steps involved in developing an attack scenario, the jammer allocation usually requires the greatest degree of analyst interaction and judgement, because no economically feasible logic for determining an optimum or near-optimum allocation is presently available.
Table 3.4
Input Data Requirements for SIMBAL II and WEFCOM

<table>
<thead>
<tr>
<th></th>
<th>SIMBAL II</th>
<th>WEFCOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Epoch</td>
<td>Evaluation times</td>
<td>Year, month, day, and hour</td>
</tr>
<tr>
<td></td>
<td>Year, month, day, and hour</td>
<td>Solar flux</td>
</tr>
<tr>
<td>Ionospheric environment</td>
<td>Season</td>
<td>Receiver latitude and longitude</td>
</tr>
<tr>
<td></td>
<td>Sunspot number</td>
<td>Transmitter latitude and longitude</td>
</tr>
<tr>
<td>Path description</td>
<td>Receiver ground conductivity and dielectric constant</td>
<td>Receiver ground conductivity and dielectric constant</td>
</tr>
<tr>
<td></td>
<td>Transmitter ground conductivity and dielectric constant</td>
<td>Transmitter ground conductivity and dielectric constant</td>
</tr>
<tr>
<td></td>
<td>Average conductivity and dielectric constant along path</td>
<td>Average conductivity and dielectric constant along path</td>
</tr>
<tr>
<td>Link description</td>
<td>Frequency band (VLF/LF or HF)</td>
<td>Frequency</td>
</tr>
<tr>
<td></td>
<td>Operating frequency</td>
<td>Received bandwidth, noise factor,</td>
</tr>
<tr>
<td></td>
<td>Received bandwidth, noise factor,</td>
<td>baud rate</td>
</tr>
<tr>
<td></td>
<td>baud rate</td>
<td>VLF/LF atmospheric noise</td>
</tr>
<tr>
<td></td>
<td>HP man-made noise</td>
<td>Noise level</td>
</tr>
<tr>
<td>Nuclear degradation</td>
<td>Device type</td>
<td>Device type</td>
</tr>
<tr>
<td></td>
<td>Burst latitude, longitude, altitude</td>
<td>Burst latitude, longitude, altitude</td>
</tr>
<tr>
<td></td>
<td>Yield</td>
<td>Yield</td>
</tr>
<tr>
<td></td>
<td>Mass and mass fraction</td>
<td>Mass and mass fraction</td>
</tr>
<tr>
<td>Jammer data</td>
<td>Latitude, longitude, altitude</td>
<td>Latitude, longitude, altitude</td>
</tr>
<tr>
<td></td>
<td>Effective radiated power</td>
<td>Effective radiated power</td>
</tr>
<tr>
<td></td>
<td>Bandwidth</td>
<td>Bandwidth</td>
</tr>
<tr>
<td></td>
<td>Site conductivity and dielectric constant</td>
<td>Site conductivity and dielectric constant</td>
</tr>
</tbody>
</table>

The overall effect of VLF/LF or HF jammer allocation on network connectivity cannot be assessed on the basis of its effect on a single frequency band alone. Consequently, the allocation process requires a considerable amount of subjective judgement. The first step is prioritization of the Blue strategic C³ links according to their importance in EAM delivery, based primarily on link performance and link use data from prior analyses. Calculations are then made of simplified link probabilities of correct message reception (PCMRs) in a benign environment, using representative hypothetical jammers having locations and powers typical of the real jammers. The calculation involves the testing of each hypothetical jammer against several or all of the possible candidate Blue strategic C³ links. From an examination of these PCMRs, an initial allocation is made, recalculation of PCMRs is performed where necessary, and the overall performance of the jammed links is estimated. After an assessment of this performance, the allocation is modified, and the process is repeated until the analyst is satisfied that the most effective possible allocation of available jammers has been achieved. If a complex iterative process such as this is carried out in the development of the baseline attack, some description of this process should probably be made a part of the database so that an analyst carrying out excursions could be faithful to the philosophy and intent of the baseline jammer allocation.

It should be noted that, while reference has been made to allocation of jammers against Blue strategic C³ links, a jammer in fact functions to disrupt the operation of radio receivers.
Since the VLF/LF transmitters operate on single, fixed, assigned frequencies and cover wide broadcast areas, it is convenient to speak of the allocation of VLF/LF jammers as against specific Blue strategic C^3 transmitters. For HF communication, separate sets of frequencies are assigned to each of the Blue strategic C^3 HF subnetworks, so jamming allocation in these cases is spoken of as against specific HF subnetworks. Because of the irregular geographical dispersions of Blue strategic C^3 nodes and jamming assets, it is occasionally necessary to assign more than one jammer against a particular VLF/LF transmitter or HF subnetwork.

**PROTOTYPE DATABASE REPRESENTATION**

An analysis of the EAP Volume VII showed that the many tables of procedural data follow a pattern, and only the node-specific and link-specific data change from one instance to another. Thus, a handful of statement types is sufficient to describe the strategic communication procedures. Starting from these (English language) statements, we developed equivalent keyword formats, following the approach we will discuss later for the description of communications nodes and equipment. We developed a procedure description format that represents procedures as sequences of blank-space-separated strings and tokens. In these statements, the leading token, usually in capital letters, must be a keyword and must be followed by data tokens encoding either numeric data or alphanumeric names. An example is the procedure statement,

```
NODE neacp RECEIVE arc-96 SOURCE sacabncep &ref
```

In this example, the keyword tokens are `NODE`, `RECEIVE`, and `SOURCE`; the data tokens are `neacp`, `arc-96`, and `sacabncep`. Here, the keyword tokens control the semantics; thus, the keyword `NODE` means that the value that follows this keyword will be a name of a node in the form of a string of characters; the keyword `RECEIVE` means that the character string that follows will be interpreted as the name of a kind of equipment that is located at the node named by the string that follows the keyword `NODE`. Taken together, the first two pairs of keyword and data tokens describe the receiving end of a link. The keyword `SOURCE` means that the character string that follows will be interpreted to be the name of the node from which the message is being transmitted. Although all of the character strings or data tokens might seem ambiguous when seen by themselves, having the same appearance and format, the keyword tokens unambiguously indicate which of the strings name communication relay nodes and which one names a type of communication equipment. Thus, in interpreting the statement, the character string `arc-96` will be searched for in the data table of receivers and nowhere else; similarly `neacp` and `sacabncep` will only be sought in the data tables containing node information.

The above approach conforms to the theoretical framework described in Secs. 4 and 5—and, to a surprising degree, reflected the EAP Volume VII procedures—in a way that was natural and unforced. Project personnel found the resulting record format easy to read and could translate it from structured format to words and back to structured format with ease. This meant that the format, from the point of view of direct inspection, possessed an adequate human readability; it only remained, therefore, to verify formally that it possessed machine readability as well.
The procedures described in Sec. 3 represented the most difficult phase of the prototype database representation effort. On the other hand, our development and comparison of the input tables for each of the computer codes in Sec. 2 convinced us that the facilities and equipment data were already organized in a fashion that readily fit into the pattern:

KEYWORD value KEYWORD value KEYWORD value.

WHICH LANGUAGE TO USE FOR DEVELOPMENT

The systems orientation of this software development effort points to the use of a modern systems development language. For UNIX or POSIX systems (POSIX is a standardized version of UNIX for government use), the language of choice has for some years been C (and more recently C++). Ada is today fully capable of performing the tasks required here, so it is also suitable for this effort; in fact, the DoD mandate that it be used places it at the top of the list. Unlike C, Ada is currently available on very few hardware platforms, but it is available for the most likely workstation environment for the SSCF Database (VAXes, Suns, high-end PCs, etc.). Fewer experienced Ada programmers are currently available, but that will, of course, change.

Our development estimates (see Sec. 5) were based on C, because of our knowledge and experience with it, and because Ada had not yet matured to the point where the DoD mandate was being enforced. Code written in Ada is usually somewhat (about 20 percent) longer than in C, but there are factors that keep the development costs about the same. First, the fact that Ada is an object-oriented language seems to be a benefit in itself. Second, current Ada developers report productivity gains based on reuse of packages. We have been assured that there is no longer a significant premium for software development in Ada. We will, in Secs. 4 and 5, explore the development of the SSCF Database design in detail.
4. PROTOTYPE COMMON FORMAT DATABASE—DESIGN ISSUES

DATABASE ARCHITECTURE DEVELOPMENT

Thus far, we have examined the need for a SSCP Database and the actions of the Joint Staff, the Strategic C³ Working Group, and RAND to establish one and have taken a look at the database itself, from the standpoints of both process and product. This section treats the database architecture, that is, format and function issues. In particular, a database format is developed and its applicability examined.

Workshops and User Survey

The first step in the database architectural development was an extensive study and documentation of the data input needs and requirements of the strategic C³ modeling community and the sources of information available to meet those needs.

To this end, the Joint Staff sponsored a series of workshops involving, insofar as possible, the whole of the strategic C³ modeling community. These workshops, the resulting discussions, and a user survey implemented in conjunction played a significant role in shaping the concept of requirements and architecture.

The objectives of the requirements review were twofold. First, it was necessary to identify those models that the community was using and that the common database would need to support. Second, it was known up front that the models had widely varying input data formats. This meant that any meaningful plan would have to be based on detailed information on both the semantics (meaning) of the model input data and on the syntax (format) of the input files.

At the first Joint Staff/RAND Strategic C³ Workshop in April 1987 [20], we conducted a brief survey of the attendees [16]. Our intent was to learn more about the types of work they were performing, the scope of their efforts, analytical tools they were using, and their computing environments. Most attendees completed a 12-item questionnaire, which yielded some interesting results.

Regarding the nature of the analytical activities of the various groups,

• 30 percent focus upon "wargaming."
• 50 percent focus upon systems engineering (Planning, Programming and Budgeting System).
• 30 percent focus upon force management (operations).

That this totals 110 percent implies a limited amount of multiple activity. Wargaming, as applied here, means analytical calculations of the C³ impact on force-on-force studies, as opposed to using players and control teams to think through phases of war from the viewpoint of the commander.

Most analysts include the following C³ parameters in their analyses:
• Message handling
• Nuclear propagation effects
• Jamming and electronic warfare vulnerability
• Physical vulnerability
• Reliability

The probability of correct message reception, message delay, and equipment availability are all commonly used figures of merit. Event times are also considered important. It seems clear that emergency action message (EAM) dissemination, as opposed to tactical warning and attack assessment, conferencing, and strategic force management, is still the predominant C^3 engineering problem for these people, or for their organizations.

SIMSTAR, STRATCAM, and WAAM are the dominant network models, and WECON and SIMBAL are the dominant link codes in terms of number of users, but the use of other network and link codes is significant. Other facts gleaned from this first survey included the following:

• The majority of users (75 percent) built (or partially built) their own databases.
• Most study teams were small, i.e., two or three analysts and two or three programmers.
• Much of the work was being done in mainframe environments.
• A considerable number of different operating systems were being used.
• Numerous groups prepared their own databases, but at the time of this survey, and through the prototype development effort, only one organization used a database management system (DBMS). This was a special-purpose DBMS for the DETEC program that Los Alamos National Laboratories was just developing. No commercial DBMS was in use at that time, although there was some indication that DBMSs, such as ORACLE, would be used in the future.

We concluded from these survey results that model development had progressed sufficiently to allow even small analyst-programmer teams to make large studies.

Just prior to the preparation of this report, another, similar, questionnaire was mailed to the same group. Their responses showed that the makeup and workload of the study teams has not changed significantly, but that they are using newer tools and hardware. The dependence on mainframe computers is being replaced by high-end work stations, like the Sun Sparc Station and the Hewlett Packard 9000 series. Much wider use of IBM PC-compatible and Apple Macintosh computers was noted.

Requirements

A rather clear-cut set of requirements for the database emerged from the above preliminary studies. This section summarizes these requirements.
Self-Documentation

The documentation of previous C³ database efforts has been inadequate in the context of systems designed to support the entire diverse strategic C³ modeling community. The SSCF Database architecture must provide inherent, embedded documentation so that the meaning, units, and association of every data item are unmistakably clear. Separate documentation to achieve this objective, while workable in principle, has not worked well enough in practice, so embedded documentation is considered an architectural-level necessity.

Data-Source Traceability

The database manager must continually update the database to represent the current state of C³ system evolution. The database will be subject to the beneficial intensive scrutiny of diverse organizations. In this, it is considered essential that the source, date, and responsible organization of each data element be traceable. To enforce such unusually complete documentation and its currency, complete data item traceability data should be included in the database itself.

Data-Item Security Classification

The database will include diverse levels of classification. A minimum requirement to support the handling of such information is that each data element be capable of carrying its individual classification and/or compartmentment status. Another aspect of security is that of control. We have already indicated the need for this database to be managed and controlled by a central authority. We elaborate on the management (as opposed to the technical) aspects of database control and dissemination in App. A of this report.

Robustness

Some database approaches carry a risk of "disintegration," that is, loss of the order integrity on which the meaning of every data element depends. This can happen if the machine crashes for any reason at inappropriate times in file rebuilding processes or, in some cases, during file restructuring. The ways to avoid or minimize the impact of such catastrophes are well known and involve some trade-off of storage efficiency and retrieval speed for robustness. One example is the incorporation of redundant punctuation keywords in the data storage format. In view of the importance and, at the same time, the relatively small size of the SSCF Database, this is clearly a beneficial trade-off. Exceptional robustness is considered a practical objective.

Database Commonality and Datum Independence

The Database¹ must support a diverse set of user applications from link models to network models. With a few exceptions, these applications have largely been developed from inde-

¹If the first letters of the terms "database" and "data set" are capitalized, they refer to the SSCF Database or a part of it, which is called a Data set. Lower case will be used for generic use of these terms.
pendent backgrounds, and use different units, terms of reference, and variables to represent a largely overlapping set of fundamental parameters. Examples would be the use of noise temperature versus noise figure or the use of different but equivalent sets of satellite orbit parameters. These redundant data forms introduce another risk of database inconsistency resulting from partial database update, for instance, updating a noise figure value and failing to update a noise temperature value representing the same receiver performance.

To obviate such risks, the Database should include only a minimal, nonredundant data representation set. Any necessary derivations from it (such as noise figure from noise temperature) should be calculated by included translation software and only at the time of input data set construction. These derivations should not be stored. This database property is referred to as data-independence or orthogonality.

Machine Independence

Not surprisingly, the survey results showed a wide variety of computing machinery in use within the strategic C4I modeling community. Users should not be expected to modify their hardware ensemble for the sake of database conformity. This means that the database system and support utilities must be supported by all these machines, as well as probable future additions.

Database Organization

The method by which data is organized for storage in a computer has a significant effect on the storage efficiency, retrieval options, retrieval speed, flexibility for reorganization, and robustness of the system. Database organization schemes may (roughly) be categorized [21] as follows:

- **File** —A file consists of a number of records, or tuples, all of the same record type or structure, that is, having the same data types or fields in the same relative locations. This may be viewed as a simple two-dimensional table.

- **Database** —A database consists of one or more files of essentially different record structures.

- **Relational database** —In a relational database, files are linked by the values in certain fields common to two or more file (or record) types.

- **Hierarchical database** —In a hierarchical database, files are linked by a fixed, defined parent-child relationship between records. The full database in this case may be described in terms of a one-to-"many" downward pyramidal tree structure. In this context, many means zero, one, or several. The relationships are further constrained in that each record type or file may be involved in only one relationship as a subordinate, and only one relationship is allowed between any two record types.
Network database — In a network database, files are linked by both relational and hierarchical links as most appropriate or efficient on a link-by-link basis.

Free-form database — In a free-form database, records are not of a few prescribed, fixed, formats but rather of a finite but indefinitely large number of possible forms, and the semantic significance is determined by context or keywords.

The DBMS Approach

A survey of existing DBMS products revealed that, while the standard relational and hierarchical DBMSs have many features that would be useful for the SSCF Database, they fall far short of satisfying the requirements developed above. The physical entities, such as relay node locations and quantifiable equipment attributes, fit the hierarchical model quite well, but the objectives of self-documentation, data-item source traceability, security tags, robustness, and machine independence are largely lacking.

In the case of specifications for actions and protocol procedures, the fit is even worse. We found that the complexity involved in specifying a strategic C³ scenario was such that every modeling group had selected particular subsets of the problem to represent. Within each selected model, there were wide variations in the number and depth of detail of the specifying parameters. Further, often very distinct system features were merged into mixed representation or were represented by nonintuitive analogies. For a user who was not an expert in the particular model, the connection between the real-world system and the model representation would be difficult to follow. In this case, it appears that a language or free-form paradigm provides a more useful approach than the usual item-structured form.

Knowledge Base Systems

The overall requirements for the SSCF Database closely match those being developed currently under the name of knowledge base management systems, or sometimes intelligent databases. These systems mix conventional database concepts with artificial-intelligence concepts to provide more sophisticated features. Unfortunately, we found no commercially available products that were suitable for this application and of sufficient portability and robustness. Although these products have not yet achieved maturity in the marketplace, when available, they will offer great promise for improved usefulness over today's systems. Knowledge base systems generally include

1. Semantic explanation clauses for human- and machine-readability
2. Storage of procedures
3. Improved verification and validation support facilities
4. A knowledge base (the ability to specify rules and infer from them) usable by human or machine queries
5. Greatly improved data protection mechanisms
For example, POSTGRES [22], an extension of INGRES, being developed at University of California, Berkeley, allows one to define complex data objects together with functions (methods) for retrieving them within a standard retrieval language. This makes possible storage of awkward entities, such as network topology (trees, graphs) and structured tables, as well as procedures. Flowing from INGRES, it retains most of the flavor of that system so far as normal transactions and retrieval functions are concerned.

Vbase [23], by Ontologic, Inc., was designed from the start as an object-oriented database system. It stores and retrieves only object types and values, although it contains the full complement of standard types of objects, such as numbers (integer or floating decimal), character strings, and money. It can store a transmitter object and instances thereof. It can store procedures (methods), which can be triggered automatically on retrieval to yield outputs that are complex functions of multiple object values. Based on the C language with object extensions, Vbase has the required functionality for the SSCP Database application.

**Higher-Order Language Approach**

Consideration of the difficulty of general procedures and protocols, i.e., conditional event specifications in a conventional database, led naturally to the consideration of a higher-order language (HOL) specially developed for this purpose. Some complicated scenarios may involve tens of thousands of discrete event commands. This level of complexity is comparable to that of a sizable computer program. The computer-programming world addressed the challenge of complex software development through the invention of HOLs. In an HOL, a relatively small number of powerful instructions (commands) causes a translation program to generate a large number of the more literal machine-language instructions that computers require. By analogy, it seemed reasonable that input-scenario specification statements of greater power might be translated into the more simple-minded inputs that the C simulation models require.

Because of the many years of application experience, the creation of HOLs is supported by a highly developed, mature technology that is both powerful and efficient [24]. Powerful synthesis approaches support structured specifications of HOLs, with automatic code generators providing the necessary brute-force coding of the implementation modules. The maturity of the technology is best characterized by noting that comparability of program results is now expected across machines of widely varying types. Because of the many implementation standards, computer language users routinely are able to produce almost identical results from HOL program execution. HOL compilations, whether targeted toward a microcomputer or a super mainframe, are routinely expected, aside from run time, to yield the same result.

Powerful tools exist with which to synthesize the linguistic component of such a system. Investigation of the compiler generators indicates that these programs can readily generate a database compiler that could translate small numbers of powerful data statements into large numbers of more specific but less powerful commands. This feature will be one key to the successful specification of complicated message-handling schemes. Study and pilot experimental results obtained so far are consistent with the belief that a well-structured language that implements an adequate problem representation could efficiently represent all aspects of the SSCP Database, while satisfying the top-level requirements developed in Sec. 4.
Conceptually, in this pseudo-HOL approach, the communication network would be represented via a series of instructions in the network or scenario-specification language. In such a concept, database updating and validation would be accomplished while the data are in the form of the specification language. Here, the more compact representation and the greater readability of the information would facilitate these objectives.

Another Experience in Database Conversion

Although we were tasked to look at common format versus SIMSTAR compatibility, we also converted some WAAM data into SIMSTAR run files. Some observations on this approach that bear on the conversion to a common format are presented here. The WAAM database was very large, consisting essentially of the Minimum Essential Emergency Communications Network EAM dissemination problem. The resulting SIMSTAR file was also very large, though not all of it was needed for the study to be undertaken.

We made a modest effort to automate as much of the conversion as necessary to demonstrate feasibility and practicality. The lessons we learned in the process reemphasize earlier points about fully understanding the data meaning (semantics) and show how data can be implicitly concealed in the program instead of explicitly revealed in the database.

The WAAM data, as received, were generated from an IBM mainframe on nine-track tape. The preliminary step, therefore, was to extract the data from its tape form, convert it from EBCDIC to ASCII format, then change simple card images to straightforward lines of information in the UNIX format used at RAND.

Once the data were readable, the process of conversion began in earnest. The language Practical Extraction and Report Language (PERL), by Larry Wall of the Jet Propulsion Laboratory, appeared to be especially appropriate for the generation of SIMSTAR-like lines. We used its pattern-matching and line-splitting features extensively. PERL programs proved to be very powerful and short (perhaps a third the size of C programs that accomplish the same goal).

The object-oriented format of the WAAM datafiles made data extraction easy. Such data as were there could easily be retrieved into PERL, then, when necessary, written out in SIMSTAR format. Unfortunately, WAAM and SIMSTAR write pieces of data into different places most of the time, or they represent things to different levels of detail, or the data appear to be missing altogether. Thus, the road to conversion was very bumpy.

The following were some of the specific things we had to do with the WAAM format to permit automation of the conversion process:

- Because WAAM permits single blanks in data items (such as names), a program was written to place an underscore in place of all single blanks. Then, longer fields of blanks could be used as delimiters between data items.
- To prevent comments and blank lines from confusing the automation process, such lines were removed as part of the processing steps.
- Configuration files were scanned for default values for data items. Lack of units here and in the other files was especially noticeable and hampered the conversion process.
• SIMSTAR files carry lots of "empty" data into a run. That is, many fields have the value zero much of the time, depending in most cases on unusual or special circumstances. All such fields were defaulted to zero in the SIMSTAR output.

• WAAM takes the direct approach in assigning equipment to nodes. SIMSTAR uses the concept of transmitter and receiver "classes," which require unique names to link them to node equipment. Such names had to be "created" by the program, a somewhat artificial process that adds layers of complexity to the readability. Worse, the names are limited to six characters. The simple process of concatenating the node name with an "R" for receiver and a "T" for transmitter and numbering them 0 through 9 reached a limit immediately. Finally, the numbering was extended to include letters of the alphabet as well. Thus, node ABCD might have a receiver named ABCDRG, which helps to identify it as a receiver at node ABCD, but provides little else in the way of information for the human reader.

DATABASE ADMINISTRATION—TECHNICAL IMPLEMENTATION

Technical implementation issues include database architecture, data presentation and representation, database validation and verification, and data set distribution. As used here, data refers to the smallest unit of information needed by a model or simulation. Thus, a latitude or a power level or an antenna gain is a data element. A database is a collection of such data elements loosely related by the applications that are going to use them. The term data set is used here in a special sense to mean a selected and formatted subset of database entities. Data sets in some form will be sent from the database manager to various users. Those same, or modified, data sets are the ultimate goal of an analyst who needs data for a particular study and who will input a data set to some model. This distributed data set may be the entire SSCF Database, but this will seldom be the case. Usually it will be a subset of the Database. The data set an analyst uses for a particular study may also be a modification of the data set that the Database manager has supplied. The degree to which the database manager will prepare the data sets for computer runs by an analyst using specific models is a degree of freedom that is discussed below.

Database Utilization Processes

The SSCF Database may be considered to consist of three distinct subsystems: a representation subsystem, which is the form in which the data and all its necessary support tables are stored; a presentation subsystem, which is the man-machine interface, whereby the analyst may add, edit, or generate reports from the Database; and the production subsystem, which includes the means whereby the Database is converted into Data set formats suitable for input to the various supported simulation models. The glossary support tables give precise definitions to the keywords, even though they are selected to be close to their common engineering meaning. The overall system is depicted in Fig. 4.1. In this scheme, the presentation subsystem causes unambiguous versions of the data to be displayed on command to the Database manager's console screen. Useful presentations would be screen-oriented forms in which data items can be displayed individually or with contrasting data items. The emphasis is on data presentations in which all items appear on the screen with full definition of item units and in arrangements that reflect logical interrelations between the data items.
The role of the representation subsystem is to provide a method for symbolic storage of the SSCP Database information. At the most fundamental level, the representation system can be considered to be a language. It must be a language that is large enough to support the range of $C^3$ models that the Joint Staff decides to include within the scope of the common database. Besides spanning the information content of the supported models, the representation should be designed to support error-free implementations. Experience with systems like PostScript (picture descriptor formats) has shown that the key requirements are clearly defined documentation for defining the format and a format that is also human-readable. RAND experiments have shown that the keyword driven database format satisfies both requirements because of the syntactic structure of the keyword system—the definition of the keyword by grammar and keyword glossary is unambiguous. The statements themselves include the English-language keywords that make the information content of the statement directly readable.

Generation of the translator, or production subsystem for a particular simulation model's input data set requires the identification of items in the model data set with items or func-
tions of items in the Database. This data item translation step requires a knowledgeable analyst to match definitions of variables in the user data set with definitions in the database glossary to generate a translation table.

Screen format displays, the presentation subsystem of the database entities, coupled with efficient methods for selecting and unselecting the data, are critical to the success and use of the system. Once the appropriate keyword data are retrieved, the remainder of the production process consists of a data-conversion step and a data-formatting step. Data conversion can be accomplished by table-driven procedures. The final step of production should be table-driven as well and requires assembly of the input file data into records of the required format and output of these records in the needed order and interspersed with appropriate model-dependent control fields and records.

Top-Level Architecture Criteria

Before discussing each of the subsystems in detail, we will discuss several aspects of the requirements the projected database activities impose on the system architecture, since these architectural criteria influence the ways one can carry out those steps or processes.

Data Presentation

The SSCF Database must be organized so that its content can be unambiguously entered, browsed, or changed. This means that any particular data item in the representation format must be unambiguously accessible and translatable to user-friendly, fully self-documented screen displays.

C³ Model Support

Since the motivation for the SSCF Database is the production of input files for C³ models, the production of usable input files is the criterion for system success. This means that subsets of the database must be unambiguously translatable into the input data sets of each of the served models. In this translation process, the verifiability of the Joint Staff source data should be preserved to the maximum extent possible. This means that, at each stage of the translation process, there must be maximum visibility in the meaning and content of each data item. This in turn means that translation via "black box" utilities must be avoided.

Translatability from Model Data Sets

One method of maintaining verifiability between the common-format database and input files for the supported models could be a utility for "importing" data from the C³ models' input files back into the SSCF Database format. Insofar as possible, this translation should be performed in a way that is consistent with the presentation and support constraints. Since the input files for the various models include data that are composites or approximations of data in the SSCF Database, this data "importation" is probably not unique. This means that data importation to the SSCF Database formats will require the use of reasonable ground rules and approximations. In spite of the use of approximations, this importa-
tion capability would provide a valuable feel for the extent to which the translation process has altered the "content" and "meaning" of the Joint Staff data. Preferably, this should be done by "demonstrably correct" translation software rather than by "feel."

**Growth Compatibility**

The Database must be configured to accommodate inevitable growth and change efficiently. This implies that the database structure must support both dynamic, value-based relationships and static, hierarchical relationships. The keyword-driven format that RAND suggests meets these requirements.

**Data Presentation and Representation**

The SSSF Database will provide an interagency standard for the performance analysis of strategic communications. Figure 4.2 presents an overview of activities that are to be managed. It is assumed that there is a split in activities between the agency that prepares the database and the user agencies. The function of the Database manager is the production, maintenance, and distribution of a verified Database or Data sets in a fully documented and standardized format. The function of the user agency is to complete the interface between the common format and the input formats for the analysts' (users') models.

*Fig. 4.2—Presentation and Representation*
The Database manager will maintain and distribute the data in a standardized format to the C³ community. The maintenance of the database content is likely to be governed by documented policies on selection of input data, load-time checking of the data, data updating, and error tracing. Because of the importance of the projected application, extensive reverification of the data will be required, as well as back-tracing of any erroneous data errors that the database users detect.

Presentation

The system that supports the data administration function is the presentation subsystem. The presentation system will incorporate most of the visual benefit from features of DBMS query languages but could also benefit from augmentation with special graphics displays and cross-referencing capabilities. Map-format displays for communication networks are an example. Cross-referencing between database elements and the specification of documents of the originating agencies would be another example. The presentation process would consist of appropriate data management software and a human-friendly display system that would utilize screen formats characterized by groupings of related data items and quick transitions between summary and detailed display screens. It is anticipated that the presentation system would employ its own internal file and database structures that best supported the data management policies. As indicated in Fig. 4.2, human factors dominate the presentation process.

Representation

The representation process is the other half of the Database manager’s charter and does not have a direct counterpart in conventional database methodology. The role of the representation system is analogous to the data exchange formats that support the exchange of information between the diverse computer equipment. The page descriptor formats used in the desktop publishing industry are an example. In that field, the multiple sourcing of publication software and laser printing devices has made it essential to develop such a common exchange format. Typically, exchange formats take the form of sequential ASCII files that contain statements modeled after either English or common HOLs. This is the proposed approach for developing a standard representation system for use in strategic C³. The Database manager will maintain the standards for the common-format database and its incorporation into suitable documentation. Distribution of C³ data from the standard database might be via common-format data copied into sequential ASCII format files on magnetic tape. In support of the C³ representation format (common format), the Database manager would also supply format documentation and software advice on data importation to and from the standard format.

Production

To support multiple models that belong to multiple agencies, the final translation of the common-format data will be more under the control of the user agencies that control C³ performance models, but the Database manager still has a role to play. In Fig. 4.3, this phase is
labeled production. During this production phase, the user of the particular model supplies the required run-control information (e.g., WAAM, SIMSTAR/STRATCAM, etc.). Also, the common-format data are copied into whatever fixed format the application model uses. Since run-control information is model specific (i.e., run control flags), there is no basis for standardizing these model inputs back to the C³ system specifications.

Database Validation and Verification

To ensure the accuracy of the database, checking must be done during the formation process; the overall flow of this checking is summarized in Fig. 4.4. The objective is maintenance of a database content consistent with the most current versions of the strategic C³ documents. This requires complete traceability of the information in both directions between the source documentation and the representation of the data in the standardized C³ representation format. To support this data tracing, it is proposed that each data item carry a unique source identifier that connects the item to a documentation note. Thus, each data record (typically 1 to 20 data elements) includes an appended validation phrase (reference data, abbreviated REP_DATA), which includes complete source identification, an entry date, and security information. If the source document is changed, it will be possible to track down and modify all associated data items. Source document modifications at the replacement-page level could be accommodated by modifying the associated data. In the case of new replacement documents,
the affected database entries would be deleted in their entirety and regenerated from the new current system specification.

**Data Set Distribution Options**

Ultimately, the goal of the SSCF Database project is the provision of data to the users in the strategic C³ community. To satisfy project goals, the Joint Staff distribution must provide the data to the user in a form that is fully documented such that the recipients of the information have the means to use the data in an manner consistent with the semantic meaning of the data. In the RAND study, three main options were considered (see Fig. 4.5). These included a centralized-database option, a database management system (DBMS) distribution and database option, and a data-distribution option. The three options primarily represent different divisions of effort between the Database manager and the user.

The centralized-database option provides that the Database manager supply the various model users with ready-to-run data sets for their models. Presumably, the Database manager is tasked with ensuring that these model-specific data representations are consistent and that they produce comparable results when run through the different models. Given that comparability of results is ensured, there is little need for the user to understand the data beyond how to load the information into his model. The most obvious drawback to this technical approach is the imposition of the very heavy workload involved in producing the multiple data sets on the Database management organization. This approach also fails to provide insight to the database users.

The DBMS distribution option calls for the Database manager to develop software and matching data files, which are then distributed to the database users. In this approach, the
definition of the data is embedded into the DBMS software, which allows the user to interact with the software in a structured and unambiguous manner. Presumably, the Database manager would write procedures in the DBMS query language so that the Database information would be displayed through clearly defined screens. These displays clearly show the units and conventions used in the various data items. Further, data specifications written in the DBMS data definition language (DDL) would provide an unambiguous characterization of the database content. Overall, the goal would be to provide the model user with a system in which the Strategic C^3 model user could immediately identify the relationships between the Joint Staff data items and the input items his model uses. The user would then use the query-language features of the DBMS to convert and format the Joint Staff data into the items and form required for the user model.
The data-distribution option consists of distributing the SSCF Database as a machine-readable file on standard computer tapes. In this approach, there is a requirement to format the information in the computer files such that the information is immediately human-readable and is just as unambiguously defined as if it were supported by a DBMS system. Such a format has been studied during the course of this prototype effort and is described later in this section. In this approach, the information describing the strategic C³ systems is encoded in a special, keyword-driven computer language. Both the grammar and keywords of the language are defined in supporting documents such that the user's understanding of technical conventions and physical units is completely defined. Further, that language is structured following certain syntactic rules that allow the software that reads and implements this language to be efficiently produced via compiler-like programs. In this approach, like the distribution DBMS approach, the user develops the software that converts the Joint Staff data into the data elements and formats his model requires.

Each of the options has significant benefits and equally significant costs. The advantages of the centralized option are maximum control of data translation from the Joint Staff representation to the model representation. Through extensive test and checkout, the SSCF Database manager would have maximum opportunity to ensure that the corresponding input files for the different models produced similar results. This enforced similarity of results would require implementing the means to convert the SSCF Database elements into the items that appear in the inputs for each supported model and the means to format the information. Both of these jobs involve considerable work. This work would have to be repeated for every supported model.

With the DBMS-distribution option, the Joint Staff would have to provide initial implementation of the DBMS systems and DBMS software maintenance as the scope and application domain of the Joint Staff evolved over time. Moreover, with the DBMS-distribution approach, it is doubtful that upward compatibility could be accomplished. With each additional parameter, the record definitions in the SSCF Database would require updating. Also, the user-written query-language procedures would require updating with each succeeding DBMS version. In many ways the effort required to implement and maintain the DBMS software largely cancels out any value that this approach might enjoy over the data-distribution approach, in which the C³ data is encoded in English-language statements.

The data-distribution option is superior to the DBMS-distribution option in that it solves the problem of the compatibility of any single DBMS with the plethora of mainframes and minicomputers that are used in the strategic C³ community. In this technical approach, the distribution database takes the form of files in a standard format that is supported by most of the computers in the community, i.e., ASCII character files.

From the above considerations, distribution of the Joint Staff data encoded in a C³ description language has emerged as the preferred method in this study. We therefore recommend the data distribution option.

**KEYWORD LANGUAGE CONCEPT FOR C³**

The SSCF Database project requires a method for describing a wide range of strategic C³ items (e.g., facilities, equipment, formats, protocols, and procedures). This method must be transportable and self-documenting. DBMSs like INGRES and dBase III have powerful
features for data specification, but do not satisfy some important criteria. Because of machine support limitations, the various DBMS development systems do not have the necessary degree of machine independence. Each class of machines supports its own DBMS packages (i.e., VAX supports DATATRIEVE, UNIX supports INGRES). Further, the approach the DBMS systems use divides the information in the database into two distinct entities: the description of the data in the DDL and the separately stored, DBMS-accessible data files. Functionally, the trio of the direct-access data files, the data descriptions in DDL, and the DBMS program constitutes a unique specification of the information. A third party needs all three components to understand and intelligently use the information. This separation of the semantics (i.e., the DDL) from the data in the DBMS approach can be a problem. Besides having the data files, a third party must have his own copy of the DBMS and must be an experienced user of the DDL that the DBMS uses. Since these DDLs are intended for generic rather than C³ applications, descriptions of C³ problems tend to be complex and not very clear to a casual DBMS user. These systems are not designed to support third-party distribution.

The SSCF Database problem calls for the development of a format that describes C³ problems and that is fully comprehensible to both the users and the software implementers. Data exchange formats that meet these needs are widely used in the computer business to facilitate the transfer of data between different systems. The highly successful PostScript page description language is an example of an export format that has been a major factor in the successful interfacing of computers, software, and laser printers in the desktop publishing industry. Working by analogy, RAND has developed an input format for C³ problem description that uses English-language keywords to embed and define both the character and word inputs to the SIMSTAR/STRATCAM and WEDCOM models. We developed satisfactory keyword formats for both models. On the basis of these prototypes, we developed more formal rules for the specification and implementation of such systems. The data formats resulting from these rules are a distinct departure from those of most current models. These formats achieve their high degree of readability at a cost in file size and input processing time while the data is read into the computer. Considering the current state of the art in computer technology, the costs in storage and processing are more than offset by the simplification of the system implementation process. In fact, a key finding of the format study is that many current data-input formats reflect computer hardware constraints that no longer exist.

C³ Language Requirements

After reviewing the existing C³ data formats, RAND identified several desirable C³ language features that were considered requirements for keyword language design. These features included the following:

- Keyword format
- Table-driven grammar and glossary
- Human-readable keywords
- Use of compiler tools for automatic front-end implementation
- Commonality among models (e.g., WEDCOM)
- Compatibility with the SSCF Database requirements
• Self-documentation
• Machine-independent transportability.

To meet these requirements, RAND is developing a specification grammar and glossary format for defining a keyword-driven C³ language. Overall, the results of the prototype implementations were encouraging and suggest that the development of a keyword-driven C³ export format could contribute to strategic C³ analysis. The availability of validated strategic C³ data in such a standardized export format would provide a common point of departure for C³ studies and a method for precise description of the scope of those studies. Further, these keyword formats provide substantial software maintenance benefits, because the data-reading routines can be built around table-driven concepts that are easily modified to accommodate growth in the methodology. Further, using the keyword formats, which conform to a formal grammar, allows some of the lexical and semantic analysis software to be implemented automatically with application-generator programs.

Formal Definition for a Keyword Format

The grammar of a keyword database may be specified succinctly as follows:

```
RECORD ::= { KEYWORD_PHRASE }+ REF_DATA
KEYWORD_PHRASE ::= = KEYWORD [ VALUE ]
REF_DATA ::= = "CLASS" classification_level
                 "SOURCE" source_reference
                 "ORG" responsible_organization
                 "DATE" date
                 "EOR"
```

The first line defines a database record (row) as consisting of a series of one or more (indicated by the plus sign) keyword phrases followed, always, by reference data.

The second line defines a keyword phrase as consisting of a keyword followed by zero or more (indicated by the asterisk) data values. The number, format, and meaning of the data values following a keyword are uniquely determined by that keyword and are recorded in a keyword glossary, which serves as a parsing and interpretation control table. A sample keyword glossary will appear later.

The third line defines the reference data as providing a fixed-format, record-by-record, self-punctuating notation of the classification level, source, responsible organization, and date of data entry into the database.

Keywords have three types: primary, secondary, and terminal. Only primary keywords can start a new record. Primary and secondary keywords each have an associated "child keyword" list. A given keyword phrase can be followed by another keyword phrase if and only if the following keyword is in the child keyword list of the immediately preceding keyword. Terminal keywords have no child keyword list and therefore always terminate a record (except for the reference data). Secondary keywords may similarly terminate the record, but do not always do so.
Keyword Glossary

The keyword glossary is an integral part of the database, serving both as the DDL and the parsing and interpreting control table. Every distribution of the SSCF Database will include, on the same medium, a copy of the glossary under which it was constructed. The database structure may grow simply through addition of keyword definitions to its glossary. Since the significance of these definitions and the database records using them does not depend on position, the new definitions will inherently be compatible with the older databases.

The keyword glossary includes the following information for each keyword phrase:

- The keyword itself
- The keyword type (primary or secondary)
- The number of variables, n, and for each variable, its
  - Name
  - Index (Is the variable to be used as a relational index?)
  - Type (real, integer, string)
  - Units
  - Exact definition.
- The related child keyword list.

Note that the glossary completely controls the lexical and semantic interpretation of each record. No other information is needed. This one table completely controls the data analyzer and, at the same time, includes the semantic definitions that relate the variables to corresponding variables in other tables or code input-data sets.

There are two alternative ways to establish relationships in this structure: (1) Phrases in the same record are inherently related by the ad hoc hierarchy so formed, and this relationship overrides any other. (2) Any lengthy entity that appears repeatedly in other entities (e.g., transceiver type AN/AR-182) has a simple alias. This is indexed on certain a priori designated keywords.

In general, this basic structure is flexible enough to allow a given data entry to be expressed in a number of ways. This degree of freedom allows us to establish the necessary compromise between self-documentation and efficiency. We have established the following guidelines for this process:

1. The variables under a given keyword should be monolithically related, that is, very unlikely to be used separately. Examples are the three components of position: latitude, longitude, and altitude.

2. The number of scalar variables in a keyword phrase, n, should generally not exceed 10 except in the case of very large tables, e.g., worldwide noise, conductivity, etc., which will be treated as a single phrase.

Prototype SIMSTAR/STRATCAM Format

To create a set of prototype formats, we used the representation of the STRATCAM model to define a set of context-free, keyword-driven instruction and data formats. These formats cov-
ered relay node data, receiver and transmitter equipment specification data, and the low-
level STRATCAM message-handling commands with which STRATCAM represents EAM
procedures. Currently, the physical and link-level entities that model relay nodes and RF
equipment are implemented into a front end to the STRATCAM database manager. When
the modified STRATCAM database manager is used for data entry, the data can be saved
into and loaded from the keyword format or can be read or written as standard STRATCAM
input data files. With the keyword form, the loading is incremental by data item type, e.g.,
receivers, transmitters, nodes, etc. Further, the loading operation is an append action with a
separate file open and close for each load action. Thus, the data set can be created by merg-
ing a large number of smaller databases. Therefore, useful libraries of related equipment
classes (HF, VHF, etc.) can be defined and incorporated as needed to assemble very large
network descriptions. Figure 4.6 displays sample formats for the prototype SIMSTAR/
STRATCAM format.

To avoid the full complexity of a database-to-data-set translator, the keyword-driven front
end was implemented as a set of table-driven routines that load the input data into the
common block data structure that the existing STRATCAM database manager uses. Since
the semantic content of this data is defined by the procedural language subroutines already
in the STRATCAM database manager, this sidesteps the problem of defining the semantic
content of each record in the STRATCAM data set. After loading the keyword format data,
the user completes the generation of a STRATCAM data set by calling routines that have
already been implemented for writing out the simulator input data set.

While the existing hard-coded procedures are a maintenance headache, they are quite ade-
quate to produce the current formats. Further, since the keyword representation is pat-
terned closely after the STRATCAM internal structure, there is no requirement for conver-
sion of data units or combining of multiple items to produce merged units. To this extent,
the RAND prototype language does not model all of the features that would be involved in a full
implementation of a common format that supported many different models. The data flows
are summarized in Figure 4.7.

WEDCOM Prototype Keyword Format

Following the initial success at developing a keyword representation for the STRATCAM
model, we applied the concept to developing an input format for link-level models. Completely mapping the WEDCOM input data set to a tentative keyword format allowed us
to determine how well the proposed STRATCAM keyword format would accommodate a typi-
cal link model. With some data translation and extensions, the structure we developed for
the STRATCAM network model adequately supported the WEDCOM model. The transla-
tions were required to convert units from the STRATCAM representation to the WEDCOM
representation. The extensions in the keyword representations were required to support
such features as the more detailed descriptions of antenna parameters that the link-level
model required.

For link models, such as WEDCOM, the total input data file is so small that direct manual
entry of the whole data set may be practical. Nonetheless, this exercise provided some feel-
ing for the extent to which such a data set could be generated and retrieved automatically,
with suitable translation where necessary, from the common keyword-driven C³ language database.

**Data Item Mapping**

Table 4.1 is the detailed mapping from WEDCOM data items to the corresponding SSCF Database data (i.e., keyword-value) items. Some minor additions to the keyword database glossary were necessary. Table 4.2 lists the part of the keyword glossary used by WEDCOM.
Block 1 of the WEDCOM data set consists of run-control parameters, unique to WEDCOM and not accommodated in the SSCP Database, but remaining as required operator input.

WEDCOM block 2 includes the position origin, which is not used in this application and may be set arbitrarily to that of the transmitter; the year, month, day, and hour at the origin, which may, without loss of generality, be taken as the start of scenario time; and the solar-flux constant.

Block 3 includes the ground constants for the transmitter and receiver antennas, both available from the database and the antenna direction option parameter for both. The latter is best under analyst input control, as it involves some subtle considerations of the corresponding conventions in the counterpart program(s) under comparative study. This may involve special module programming and use of the 3TD and 3RD card inputs.

Blocks 3T and 3R are the transmit and receive node positions and antenna orientations. The positions can be readily retrieved from the SSCP Database under the control of a translation module, based on input data defining the Tx and Rx SSCP Database node names.

Block 4 provides the path ground constants. For the example in Figure 4.8, these are assumed to be provided by the default WEDCOM ground model, so Block 4 is not used.

Block 5 provides the calculation times. These are viewed as program control inputs, under direct analyst input.

Block 6 is the nuclear attack data, using the "Detail Option," the details of which would be found in the SSCP Database, as listed in Table 3.3.

Block 7 provides the transmitter and receiver details. The link codes generally work from a greater level of detail than the network codes. Nevertheless, in this case, the majority of the Tx/Rx descriptive parameters were available in the existing database glossary. Since the SSCP Database is defined as the union of data-set requirements of its various supported users, we added the remaining detailed requirements to its glossary. These requirements included noise clipper parameters, processing loss, and modulation options.
### Table 4.1

**WEDCOM Data Set and C³ Database Translation**

<table>
<thead>
<tr>
<th>WEDCOM Item</th>
<th>CSDB Item</th>
<th>Keyword(s)</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Card</td>
<td>Name</td>
<td>Keyword(s)</td>
<td>Name</td>
</tr>
<tr>
<td>1</td>
<td>Control</td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>1A</td>
<td>Header</td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>Origin Lat (Tx) NODE, FIXED</td>
<td>lat</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Origin Long (Tx) NODE, FIXED</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Year Origin (TIME)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Month Origin (TIME)</td>
<td>month</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Day Origin (TIME)</td>
<td>day</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hour (ZT @ Origin) (TIME (GMT))</td>
<td>hour*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Solar Flux (TIME)</td>
<td>sol-flux</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Tx_Conductivity (Tx) NODE, EARTH</td>
<td>conduct</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tx_Dielectric (Tx) NODE, EARTH</td>
<td>ops</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rx_Conductivity (Rx) NODE, EARTH</td>
<td>conduct</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rx_Dielectric (Rx) NODE, EARTH</td>
<td>ops</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rx_Direction Opt.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tx_Direction Opt.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3T</td>
<td>X1 (Tx) NODE, FIXED</td>
<td>lat</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X2 (Tx) NODE, FIXED</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X3 (Tx) NODE, FIXED</td>
<td>alt</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N Fixed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tx Ant Zenith Angle</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tx Ant Az Angle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3TD</td>
<td>D Not Used</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3R</td>
<td>X1 (Tx) NODE, FIXED</td>
<td>lat</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X2 (Tx) NODE, FIXED</td>
<td>lat</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X3 (Tx) NODE, FIXED</td>
<td>lat</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N Fixed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>REF OPTION Not Used</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rx Ant Zenith Angle</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rx Ant Az Angle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3RD</td>
<td>D Not Used</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Path Conductivity Not Used</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Path Dielectric Not Used</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>T1, T8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>IDX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>NB NUC</td>
<td>count*</td>
<td></td>
</tr>
<tr>
<td>6B</td>
<td>X1 NUC</td>
<td>lat</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X2 NUC</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X3 NUC</td>
<td>alt</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N Fixed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Burst Time NUC</td>
<td>burst_time</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device Index NUC</td>
<td>device_id</td>
<td></td>
</tr>
<tr>
<td>6D</td>
<td>Yield NUC</td>
<td>yield</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device Type Index 0</td>
<td>fixed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Detail Option 1</td>
<td>fixed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fission Fraction NUC_DEVICE</td>
<td>fiss_frac</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hydro Yield Fract NUC_DEVICE</td>
<td>hydro_frac</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mass NUC_DEVICE</td>
<td>mass</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alum Mass Fract NUCDEVICE</td>
<td>alum_frac</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Uran Mass Fract NUCDEVICE</td>
<td>uran_frac</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gamma Yield Fract NUC_DEVICE</td>
<td>gamma_frac</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Neut Yield Fract NUCDEVICE</td>
<td>neut_frac</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X-ray Yield Fract NUCDEVICE</td>
<td>x-ray_frac</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Rx_BW TRANSMITTER</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RX_TYPE</td>
<td>bandwidth</td>
<td></td>
</tr>
</tbody>
</table>
### Table 4.1—continued

<table>
<thead>
<tr>
<th>WEDCOM Item</th>
<th>C3DB Item</th>
<th>Name</th>
<th>Keyword(S)</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise Factor</td>
<td>RX_TYPE</td>
<td>noise_fig*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mod. Option</td>
<td>Fixed</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7M Spread</td>
<td>TRANSMITTER</td>
<td>spread</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modem</td>
<td>TRANSMITTER</td>
<td>modem</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bit_rate</td>
<td>TRANSMITTER</td>
<td>bit_rate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proc. Loss</td>
<td>RX_TYPE</td>
<td>imp_loss</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BER/CER Options</td>
<td>A</td>
<td>clip_on</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clipper Option</td>
<td>RECEIVER</td>
<td>clip_bw</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sat. Level</td>
<td>RX_TYPE</td>
<td></td>
<td>clip_level</td>
<td></td>
</tr>
<tr>
<td>8 Frequencies</td>
<td>FREQUENCY (S)</td>
<td>frequency</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NTOT</td>
<td>NOISE &amp; JAMMER</td>
<td>Count*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NNSI</td>
<td>NOISE</td>
<td>Count*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9N, J</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X1</td>
<td>NOISE OR JAMMER</td>
<td>1st</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X2</td>
<td>NOISE OR JAMMER</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X3</td>
<td>NOISE OR JAMMER</td>
<td>alt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>Coast</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rad Power</td>
<td>NOISE OR JAMMER</td>
<td></td>
<td>evrp*</td>
<td></td>
</tr>
<tr>
<td>Bandwidth</td>
<td>NOISE OR JAMMER</td>
<td>bandwidth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conductivity</td>
<td>NOISE OR JAMMER</td>
<td>conductivity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dielectric</td>
<td>NOISE OR JAMMER</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A = WEDCOM analyst-supplied  
* = simple translation required  
Use Tx Node for Origin  
Use KALLF = 0, KALCT = 0

### Table 4.2

**WEDCOM Keyword Glossary**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Primary/Secondary # Values</th>
<th>Value Name</th>
<th>Format</th>
<th>Units</th>
<th>Meaning</th>
<th>Child Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMENT</td>
<td>P</td>
<td>1</td>
<td>comment</td>
<td>string</td>
<td>Comments, arbitrary length</td>
<td>None</td>
</tr>
<tr>
<td>NODE</td>
<td>P</td>
<td>1</td>
<td>node_id</td>
<td>string</td>
<td>Standard abbreviation of node name</td>
<td>FIXED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MOBILE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LAUNCHED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SATELLITE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DAMAGE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RECEIVER</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TRANSMITTER</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PROCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MTRR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>WACID</td>
</tr>
<tr>
<td>MOBILE</td>
<td>S</td>
<td>4</td>
<td>time</td>
<td>float</td>
<td>min.</td>
<td>Time from start of scenario</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>lat</td>
<td>float</td>
<td>deg.</td>
<td>E plus</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>long</td>
<td>float</td>
<td>deg.</td>
<td>N plus</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>alt</td>
<td>float</td>
<td>km.</td>
<td>Above sea level</td>
</tr>
<tr>
<td>LAUNCHED</td>
<td>S</td>
<td>4</td>
<td>time</td>
<td>float</td>
<td>min.</td>
<td>Time from start of scenario</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>lat</td>
<td>float</td>
<td>deg.</td>
<td>E plus</td>
</tr>
<tr>
<td>Keyword</td>
<td>Primary/Secondary</td>
<td># Values</td>
<td>Value Name</td>
<td>Format</td>
<td>Units</td>
<td>Meaning</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------</td>
<td>----------</td>
<td>------------</td>
<td>--------</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>FIXED</td>
<td>S</td>
<td>3</td>
<td>lat</td>
<td>float</td>
<td>deg.</td>
<td>E plus</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>long</td>
<td>float</td>
<td>deg.</td>
<td>N plus</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>alt</td>
<td>float</td>
<td>km</td>
<td></td>
</tr>
<tr>
<td>SATELLITE</td>
<td>S</td>
<td>7</td>
<td>semi-mj-axis</td>
<td>float</td>
<td>km</td>
<td>Orbit semi-major axis, a</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>perigee</td>
<td>float</td>
<td>deg.</td>
<td>Argument of Perigee, w</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>eccentric</td>
<td>float</td>
<td>—</td>
<td>Eccentricity, e</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>incln</td>
<td>float</td>
<td>deg.</td>
<td>Inclination, i</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>node_ra</td>
<td>float</td>
<td>deg.</td>
<td>Right Ascension of</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>anomaly</td>
<td>float</td>
<td>deg.</td>
<td>Ascending Node</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>epoch</td>
<td>float</td>
<td>hours</td>
<td>True anomaly at epoch</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Epoch time</td>
</tr>
<tr>
<td>RECEIVER</td>
<td>S</td>
<td>4</td>
<td>rx_name</td>
<td>string</td>
<td>—</td>
<td>Identifies receiver within this data set (e.g., NEACP-14)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>rx_id (tr)</td>
<td>string</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ant_id (tr)</td>
<td>string</td>
<td>—</td>
<td>Identifies this receiver to Mfg. Type (e.g., AR/ARC-182)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>clipper_on</td>
<td>string</td>
<td>—</td>
<td>Identifies this antenna to mfg. type (e.g., AT-1011)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&quot;0&quot; = int, &quot;1&quot; = out</td>
</tr>
<tr>
<td>RX_TYPE</td>
<td>P, S</td>
<td>10</td>
<td>rx_id (tr)</td>
<td>string</td>
<td>—</td>
<td>e.g. (. . . , TBD)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>lo_freq</td>
<td>float</td>
<td>kHz</td>
<td>Lower frequency limit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>hi_freq</td>
<td>float</td>
<td>kHz</td>
<td>Upper frequency limit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>noise Fig</td>
<td>float</td>
<td>dB</td>
<td>System noise Fig. No external noise.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>modulation</td>
<td>string</td>
<td>—</td>
<td>Modulation type (MSK, CFSK, CSK, PSK, DPSK, NFSK, +bf)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>bandwidth</td>
<td>float</td>
<td>Hz</td>
<td>Bandwidth (?)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>imp_loss</td>
<td>float</td>
<td>dB</td>
<td>Implementation loss, dB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>clip_bw</td>
<td>float</td>
<td>kHz</td>
<td>Clipping bandwidth</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>clip_level</td>
<td>float</td>
<td>dB</td>
<td>Normalized clip level (more-TBD)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>edac</td>
<td>string</td>
<td>—</td>
<td>e.g., HAM15/11, . . .</td>
</tr>
<tr>
<td>VERT Whip</td>
<td>S</td>
<td>3</td>
<td>length</td>
<td>string</td>
<td>—</td>
<td>Vertical length of radiating element, ft</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>lo_freq</td>
<td>string</td>
<td>—</td>
<td>Lowest frequency at which used, Hz</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>hi_freq</td>
<td>string</td>
<td>—</td>
<td>Highest frequency at which used, Hz</td>
</tr>
</tbody>
</table>

NOTE: Database Record Format is:
[KEYWORD | [VALUE] *] + LICENSE PLATE

The keyword can always and only be followed by keywords in its keyword child list. The record must always terminate in the "License Plate"

Upper case words are KEYWORDS, lower case are variable names, and stand for the corresponding values in the DB.

P = Primary, starts a new record
S = Secondary, follows primary and/or secondaries.
*tr Denotes relational keyword
Block 8 provides the list of frequencies to be analyzed. This is an analyst input.

Block 9 provides the jammer parameters, all of which were available in the already defined SSCF Database data set.

**Exchange Format**

Figure 4.8 is an example data exchange format for a small but complete WEDCOM input data set, i.e., for all the data to be extracted from the SSCF Database according to the mapping assumptions detailed above. This constitutes less than a page of input even in the fully verbose, self-documenting format. This suggests that for the simplest cases, such as this example, automatic data-set generation has little to offer over manual retranscription from a hard copy printout in this form. For the link models, the small size of the keyword data sets reduces the value of automated versus manual methods for data-set preparation for single data-set jobs. It could still be argued that the procedures associated with defining the data-set elements for automated generation could result in a greater uniformity and objectivity in the automated translation-and-conversion process. Since the machine-readable model-data definitions could be reused many times to generate many different input sets for the target

```


NODE: OKA1 EARTH 0.01 15 SOURCE: FCC Maps 1983 PERSON: None DATE: 2/3/45 CLASS: Unclassified EOR.


AT:1011 Vertical 32 SOURCE: Shakespeare Catalog PERSON: None DATE: 2/18/88 CLASS: Unclassified EOR

TIME: 89 05 16 12.90 36.0 40 SOURCE: JPA PERSON: Q. Craigie DATE: 8/5/86 CLASS: Unclassified EOR

NUC: 24.2 34.5 0 3.4 KT21 2.5 NUC: 27.3 31.0 110 3.0 KT21 50 SOURCE: DNA 4-1016 12/16/82, p. 123 PERSON: L. Franke DATE: 12/54/87 CLASS: Unclassified EOR.

JAMMER: 21.2 22 3 0 51.5 50 10000 5 80 SOURCE: DDD PERSON: J. Hamper 203/304-5678 DATE: 4/5/88 CLASS: Unclassified EOR.

WORLD NOISE: (Big Array) SOURCE: ITS PERSON: Disney DATE: 9/30/76 CLASS: Unclassified EOR.

END
```

*Fig. 4.8—WEDCOM Example Exchange Format*
model, this investment in resources could eventually be recaptured even when a single data set is quite small.

For larger data sets, however, automatic data set generation could be very useful, especially if one wished to enter similar data into several different codes and still be certain that the desired commonality was not lost in the manual retranscription process.

Our WEDCOM data set mapping exercise indicated that even for link models, such as WEDCOM, approximately 70 percent of the input data set could be generated automatically from programmed retrievals from the SSCP Database data set, and the remainder appears best treated as direct analyst input, requiring some extra degree of analytic judgement.

This "possible automation" percentage is much lower than for a typical network model, because the link model, by definition, accommodates only one transmitter and one or a very few receivers at a time. Thus, the very long and repetitive lists of nodes, equipments, and procedures that make up so much of the input data set for a typical network model are not used for the link models.

Of the 70 percent or so of the input data set that appears reasonably amenable to semiautomatic retrieval and translation for WEDCOM, some 90 percent was available in the existing SSCP Database data set. The rest has now been added to it. Doing this for the rest of the class of link codes appears straightforward.

SOFTWARE TOOLS AND DATA SET PRODUCTION

Along with production and maintenance of the SSCP Database, it appears logical and advisable for the Database manager to develop and provide certain software tools or utilities that will find common use in the production, maintenance, and user applications of the Database. Among these are the presentation facilities for inputting, editing, browsing, and preparation of data sets for model users. This subsection discusses the recommended approaches to the development of these tools.

High-Level Tools

The proposed SSCP Database internal, or representation, format was developed and presented earlier in this section. This format is designed to provide:

- **Robustness**—Loss of database integrity is made extremely unlikely; any such loss is limited in domain; and catastrophic losses of the Database appear impossible.
- **Human readable format**—Data significance is largely self explanatory.
- **Included semantics**—Detailed definition and units included in the internal Database glossary.
- **Source traceability**—The entry date, source reference, and cognizant person are included for every record (typically 1 to 20 data items).
- **Classification by record**—Each record includes the security classification.
- **Flexibility**—New data types can be accommodated without programming, obsolescence, or database reorganization, through simple table addition.
• *Table-defined grammar*—This grammar makes possible general-purpose, table-driven
data access and translation utilities.

Forgoing the use of standard database management software means that an equivalent ca-
pability *must* be constructed to support data loading, data editing, report generation, and
data export file creation. The decision to opt for a procedure language implementation
should be based on the finding that the data manipulations are so extensive and ill matched
to the data manipulation language of the DBMS that sacrificing the convenience of screen
and report generators is clearly warranted.

The principal functions to be supported are as follows:

• Inputting or editing keyword glossary
• Inputting or editing data to the database
• Generating hardcopy reports
• Generating formatted input file for the user's data set.

These will be discussed in turn.

**Inputting and Editing the Keyword Glossary**

The development of the keyword glossary is the first task in producing the Database. The
glossary must be used in the production of the Database to ensure that the data records are
consistent. Similarly, once the database exists, edits to the glossary will be constrained to
preclude development of any inconsistency to the current Database.

Each Keyword glossary record consists of:

```plaintext
KEYWORD

    Type (Primary or Secondary)
    Number of values, n
    (For i=1 to n)
    Variable Name
    Index (is variable to be used as a relational index?)
    Variable Type (Real, Integer, String)
    Units
    Exact Definition of Meaning

Next 1

    KEYWORD_CHILD list
```

Recall that the child-keywords list is such that a given keyword may always and only be fol-
lowed by one of its child keywords. Thus, the child keyword structure imposes a direct
downward structural constraint and an implicit upward structural constraint, i.e., a given
keyword must be preceded by one, possibly several, of its parent keywords. The integrity of

---

2 Again in this section, the term "Database" (initial caps) refers to the the SSCP Database, proper.
this structure must be maintained throughout any additions or changes to the glossary. It will be clear that this integrity has two elements: the glossary’s own “internal integrity” and the glossary’s “external integrity” with the database.

The glossary structure will be defined as internally consistent if and only if there are no “orphan” keywords, i.e., no keywords with no path to them. To ensure this internal integrity, the glossary inputter-editor will be structured so that only a keyword that is either primary or has already been defined as a child keyword in some other keyword’s glossary record will be allowed as input. That is, each new keyword must first have one or more legitimate parent keywords. This, of course, means that the keywords must be entered initially in top-down order. Similarly, deletion of either a keyword-defining record or a child-list membership will only be allowed if it does not result in any “orphan” keywords, i.e., keywords having no path to them.

This process (and others to be discussed) will be expedited at each session by constructing, checking, and continually updating (as keywords are added or edited) a “Parent Keyword List” file, an upward dependency index that gives, for each keyword, the list of keywords of which it is a child. Each nonprimary keyword must have one or more such parents (otherwise it would be an illegitimate orphan).

Note that this multiple-parent logic allows, in general, a number of alternative, nonunique ways of expressing a given data set. The language is in this sense rich, and the operator may use this richness to maximize efficiency, readability, reference-source data groupings, security-classification groupings, or other objectives.

If the initial keyword entry for a given record input is recognized as preexisting, the program will recognize the intention as editing rather than input. Keyword editing will follow essentially the same screen-presentation and checking procedures as inputting, except that default values for all entries will be presented in the screens, with acceptance signified by “enter” (or carriage return) or a change by simply overwriting.

_External integrity_ is defined as the property that every record structure in the database is an allowed structure defined by the glossary. In general, not all of the record structures allowed by the glossary may be used in the Database. That is, the Database structures are a subset of the structures allowed by the glossary. To ensure external integrity, any proposed change to the glossary in terms of keyword name, value list, or child-keyword list will be allowed if and only if it does not violate this requirement. To expedite this constraint, a second index file, “Child Keywords Used,” will be created from the Database at each session and will be checked for compatibility with the “Child Keywords Allowed” file from the glossary at each proposed glossary editing entry. (The _same_ check will later constrain any actual data input forms during the data input session.)

These procedures will ensure the integrity of the database. Notice that only the glossary and Database are stored; all indices are recreated at each session. Even in the worst-case scenario of a crash in the midst of a database save, the worst problem in general would be that a few data records would be improperly supported, and their existence and illegitimacy would become evident the next time the Database is accessed. That this scenario is far more benign than it would be for most conventional, “efficient” databases is a direct result of the effective use of the considerable redundancy implicit in the keyword format.
Inputting and Editing Data

The data input and editing utility will be based on the same concepts of producing a database that is in exact syntactical conformance with the particular version of the glossary with which it is permanently and inextricably associated and ensuring that no additions or changes to the database are allowed that would violate these syntactical rules.

The inputting and editing programs will share the same software and presentation screens, except that for editing, the screens will display all prior parameter values as defaults to be accepted, rejected, overwritten, or edited at the operator's discretion.

Maximum utilization will be made of screens that present logically connected groups of parameters (such as properties of a given node, properties of a given equipment type, and links) together in the most comprehensible form, much as in the SIMSTAR preprocessor.

Hard-Copy Reports

Hard-copy reports will be available in several formats that will directly satisfy the majority of general-purpose Database review needs. The most basic of these, the so-called "raw" Database report, will consist simply of an ASCII dump of the Database, documented only by the internal, self-documenting features of the Database. Other forms will provide even better-formatted reports of data values, formatted by logical access bases, such as by node or equipment class.

Data-Set Production Tools

The most complex of the software utilities is that supporting translation of the SSCF Database into usable user data sets. The translation of the SSCF Database or subsets thereof into formatted data-set files for the user is necessarily an operator-interactive function. The user's operator-analyst is the best person to carry this out, making full utilization, however, of utility software and the expert consultation that the SSCF Database management organization will provide. This is so largely because of the complexity, often inadequate documentation, and sometimes transient nature of the formats and semantics of the users' data sets.

The difficulties encountered in this very project in obtaining and deciphering the semantics and formatting grammar rules for the various user data sets emphasize the necessity for making the translation process fully self-documenting. This will maintain the traceable integrity we have worked so hard to ensure in the Database itself.

In general, the translation process consists of three parts (not always in this order):

- Data item access
- Data item transformations
- Output data-set formatting.

These will be discussed in turn.
Data Item Access

Database access is specified and would be documented in the proposed architecture by the sequence of parent keyword values leading to the data item. Since this is a sometimes complex chain, it is envisioned that Database item access would be supported interactively for the operator by screens that either acquire the links in the parent chain from the corresponding links in the target data set or, when this is impossible, demand that the operator select from the valid options of the full lineage of each data item. For example, a request for noise figure would demand specification of a receiver type.

Data Item Transformation

Data item transformation, while usually trivial, is sometimes complex. A trivial example would be node latitude, which is likely to be transferred intact to the destination model. All that would be necessary in this case would be to make the identification between possibly different names for the same data item in the two different systems. More complicated transactions arise when the model requires a different parameter than is found directly in the database and is supported only indirectly as a function of one or more Database items. Examples would be the use of different units, such as kilowatts versus decibel watts; alternative sets of satellite ephemerides; or the derivation of receiver-and-antenna system figure of merit from separate knowledge of dish size, frequency, and noise figure in the Database.

One approach to this problem would be to store multiple forms of the same data items in the Database, but such an approach is demonstrably prone to the development of internal inconsistencies over time. Instead, we recommend that the Database be operated under the ground rule that the database items represent essentially independent degrees of freedom, that is, such that a change in the value of any one data item does not inherently imply a change in another.

These considerations suggest a rule-based approach to the transformation. The SSCF Database is supplemented by a rigorously specified set of translation procedures that implement the conversion of Database data item(s) to a user data-set item. The examples above suggest that a finite set of well-documented, HOL-coded procedures may be the only sufficiently general way to establish and record these transformations.

Output Data-Set Formatting

It is at least equally important that the output formatter be completely self-documenting. Often, the user's code-input readers are lengthy sequences of tests implemented in the code and prone to undocumented quirks and occasional bugs. Occasionally, the only definitive documentation of the formatting rules is that implicit in the FORTRAN source code of the input data reader itself.

In a conventional implementation, the developers would generate a specification describing the operations required to process common database-derived items into the record elements required for the target model's data set. In such a conventional approach, the procedures for formatting are coded directly into FORTRAN or C routines that represent these formatted writing steps. In this approach, the primary objective is to produce a module that works cor-
rectly and efficiently with respect to both coding and run time. Clearly, the objective of getting executable translations may be met by this approach. However, there is a significant loss in data traceability.

The above procedural-language approach corresponds to the experimental system that RAND developed around the STRATCAM/SIMSTAR preprocessor. In this quick-and-dirty approach, the parsing and semantic analysis processors were designed to extract the data from the prototype C^3 scenario language statements and load this information directly into the FORTRAN-based data structures that were defined in the STRATCAM preprocessor. Since this module contains FORTRAN routines that will directly write out the SIMSTAR/STRATCAM data sets, no further operations were required to complete this proof-of-concept system. Unfortunately, getting executable data sets is only part of what is needed for the SSCP-Database-to-Data-set translation. In the prototype system, the data starts out in a human-readable and verifiable format, and to this extent the experiment is relevant to goals of the SSCP Database project.

However, the sleight of hand involved in using the existing STRATCAM/SIMSTAR formatting routines destroys the traceability of the data once the information is written out into the STRATCAM data-set format. This direct loading process breaks the validation trail that we desire to maintain right from the information in the source documents through to the data sets that are input to the various models. In theory, the SIMSTAR/STRATCAM FORTRAN source code that does the conversion could be said to document this conversion. However, experience shows that these modules function in black-box modes, in which the users aren’t explicitly aware of the action the module is performing. This is especially true where the implementing software was developed long ago and has neither the documentation nor the necessary structure to provide adequate insight into its operation. What is needed is a self-documenting implementation in which the specification (in human-readable form) is also the program (in machine-readable form).

Table-Driven Formatter Concept

To enforce a more clearly self-documenting formatting operation, the following discussion suggests that the file-builder program should not be implemented as a set of HOL statements that match those of the model input processes. Rather, the model reader should be developed as a table-driven, general-purpose file-builder program or “engine” (i.e., a basically dumb program that obtains its “smarts” from a data table). This file-builder engine would use the finite-state machine description of the data set’s file structure as a script from which to construct the output. With such an approach, the assumptions regarding the content and sequencing of the data set can be documented far more explicitly than even the best-written FORTRAN programs can be. This explicit visibility into the data-set building process would provide a critical link in ensuring the semantic integrity of the conversion from Database to Data set.

There could be implementation savings as well. With proper design, the file-builder engine could support several models, with only the finite-state machine tables being specific to formatting a particular model’s data sets. The use of a single engine to support multiple models could also minimize the effort devoted to validation and verification, because only one program would be needed.
Format Control Table

The finite-state description of the format will probably take the form of a limited number of records. Each such record consists of

- A state identifier
- A start cue, which may include one or more number-of-items-to-follow indicators
- A sequence of zero or more variable names and the associated format identifiers
- A list of possible end cues with the identification of the next state to which they lead.

Note that the only branching control in this format is that provided by the end cue; in fact, the occurrence of any branching control is, by definition, the end of the state-definition record. The same format should be usable in the control of either a format reader or writer, except that, in the latter case, the end cues that effectively select the next format state are selected by exogenous circumstances, such as "having finished the listing of all nodes."

Many of the features that must be and, it is believed, are accommodated by this general format are illustrated in the extract from a SIMSTAR/STRATCAM database presented in Fig. 4.9. The very first marked record in this data set is a start cue that tells the reader that there are zero node data records to be read. Two flag values are provided to indicate that the following data are for the first variant on the third receiver release format. These would probably be operator input and, if they control branching read options, are to be taken as the end of one state and the start of another.

The record containing the words "RECEIVER CLASSES" may be viewed as part of a start cue.

The next labeled record contains a mixture of data and control flags. Each such branching control flag may be treated as an end cue, leading to a corresponding next read state. (Note that, in this usage, the start cue may be null.) The alphabetic data provide the name of a transmitter class; the remaining integer cues frequency-dependent read operations, while others set the number of read repetitions required to take in the time-dependent availabilities of the equipment. Each such branching control is the end cue for a state. This illustrates the treatment of a large class of data-set read mechanisms that utilizes do/for or do-while constructions to control iterative input operations to load arrays of related data items. To be effective, the database-to-data-set translator must accommodate all of these diverse format elements without losing data identity and traceability.

Data-Set Production

The proposed translation concept envisions that the operator will guide the data-set production process, which proceeds interactively, using the above-described facilities as tools under the operator's control. The operator's responsibility in this process may be described as selecting and editing from among the facts available in the Database. It is important that the operator be clearly in charge of this process at all times; in other words, the software tools should be such that the operator can compose the database entirely himself, aided only by the formatting utilities to ensure that output is syntactically correct and by the Database resources only to the extent he may request. Potentially, the data set builder should be able to
<table>
<thead>
<tr>
<th>0 2 1</th>
<th>&lt;--- (Comment record — Skip)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 2 1</td>
<td>(data &amp; cue record)</td>
</tr>
<tr>
<td></td>
<td>(pure data record)</td>
</tr>
<tr>
<td>9 2 1</td>
<td></td>
</tr>
<tr>
<td>TEHF</td>
<td>9 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.10000000e+06 0.00000000e+02 0.45000000e+11 0.00000000e+00 0.00000000e+00</td>
</tr>
<tr>
<td>TELF</td>
<td>1 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.20000000e+02 0.10000000e+02 0.45000000e+02 0.00000000e+00 0.00000000e+00</td>
</tr>
<tr>
<td>THF</td>
<td>5 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.10000000e+04 0.40000000e+02 0.15000000e+08 0.00000000e+00 0.00000000e+00</td>
</tr>
<tr>
<td>TLF</td>
<td>3 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.20000000e+03 0.30000000e+03 0.20000000e+07 0.00000000e+00 0.00000000e+00</td>
</tr>
<tr>
<td>TMF</td>
<td>4 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.20000000e+02 0.30000000e+02 0.10000000e+07 0.10000000e+06 0.00000000e+00</td>
</tr>
<tr>
<td>TSHF</td>
<td>8 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.10000000e+06 0.75000000e+02 0.50000000e+10 0.00000000e+00 0.00000000e+00</td>
</tr>
<tr>
<td>TUFH</td>
<td>7 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.50000000e+04 0.35000000e+02 0.40000000e+09 0.00000000e+00 0.00000000e+00</td>
</tr>
<tr>
<td>TVHF</td>
<td>6 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.10000000e+05 0.55000000e+02 0.10000000e+09 0.00000000e+00 0.00000000e+00</td>
</tr>
<tr>
<td>TVLF</td>
<td>2 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.18000000e+02 0.20000000e+02 0.17000000e+05 0.00000000e+00 0.00000000e+00</td>
</tr>
</tbody>
</table>

**NOTE:** These are the existing STRATCAM data-set formats. The order dependence of the records and fields and the fixed-number formats make a nightmare of preparing such inputs using standard file editor programs. For each entity type, the number of entities (e.g., nodes, receivers) must be accurately counted, and a correctly formatted cue record must be inserted.

**Fig 4.9—SIMSTAR Format (Fixed Field, Ordered-Record Format)**

Select the nodes, equipments, links, and procedures; and the threat, attack, or damage that is included in the target scenario, subject only to checks for internal consistency; and the Data-set syntax. This selection process requires both man and machine features.

Figure 4.10 depicts a possible interactive data-set-building concept. Logically, the process follows the same generic hierarchical C3 Database outline presented in Section 3. For nodes, equipment (or communication mode), networks (or links), procedures, and threat, attack, or damage in that order, the operator and Database interact in selecting data and assigning variable values. Major approximations in data assignments will be documented in the ground rules and assumptions part of the Database. The presentation screens will request categories of selection and the associated parameter values that the user data-set format requires. After each screen selection or editing process, the program will check the resulting specification for Database support; i.e., are there references to entities or values that neither the Database nor the operator has provided? Unresolved references or out-of-range values (e.g., reference to an undefined receiver type ARX-999, or a frequency out of range of a specified equipment) will be referred back to the operator for resolution on the spot or later, at his option.
Fig. 4.10—Data Set Building Concept
Although a phased implementation would initially focus on data representation and conversion, later phases of the SSCF Database support should expand the data-management software to include graphical features. Graphical aids would be invoked where possible to enhance operator comprehension of the evolving scenario. For example, graphics windows could provide wiring diagrams of the links and networks proposed for inclusion in the data set. The ability to intelligently select subsets of the information in the Database is motivated by the observation that not every model run will exercise every possible feature of the SSCF Database. Moreover, it will not always be clear to the analyst what must be included and what may be omitted from the data-set version of the strategic scenario. In the case of link models like WESCOM, a "complete" data retrieval will produce all the nuclear data required to describe the enemy laydown, the RF parameters of the transmitters and receivers, and the platform locations.

During the building process, an ensemble of data-query utilities should provide on-the-spot insight into the Database to the operator. For example, in some applications, the data-set builder may desire to specify a network-based scenario using only network-level descriptors, e.g., the retrieval of the data items relating to the Ground Wave Emergency Communication System (GWEN). In this kind of data-set request, it would be highly desirable to implement a semantic retrieval process that would collect the subset of relay facilities and radio equipment that supports the entity known as GWEN. Clearly, the relevant message-handling and network-management procedures would be desired, as well as the subset of node kills and nuclear burst that would impact GWEN operation. This means that the basic link and equipment specification keywords and statement syntax must be expanded to accommodate network and connectivity statements that establish the membership of particular equipments, relay nodes, and links in support of specified networks. Such membership statements could interact with and be referenced by the statements describing network procedures and protocols.

This relates to a key difference between the WAAM and SIMSTAR/STRATCAM methodologies, as they existed at the time of this study. In the WAAM methodology, the message handling is described by procedures that reference equipment types. This representation method is reasonably close to that of the procedural specification in the source documents and gives a quite compact machine-readable representation. The STRATCAM methodology, on the other hand, does not support any discrete-event commands that involve variables or references that must be resolved at runtime; thus, specific commands must be provided for every node, process, transmitter, or receiver through which the message must pass. In a very real sense, a STRATCAM message routing sequence can be conceptualized as the result of applying the appropriate procedure to every network entity to which the procedure can apply. In a large network, there are many possible substitutions, and message routine sequences with more than 6,000 separate commands are common. Worst of all, these sequences and programs are prepared by hand, and critical performance links are often omitted from the message routing sequence because of fatigue-induced human error. Clearly, repetitive tasks of such a magnitude should be left to machines and should be automated by suitable algorithms in the data-set translator.

The final steps in the data-set-building process are for the operator to add the run-control parameters and, finally, for the data-set formatter to build the user data-set file, making use of the table-driven format engine.
THE DATABASE LANGUAGE

Design Criteria

We used the following the design criteria to guide the development of the language.

1. Readable format—The internal storage format of the data should be in readable ASCII character and word format, comprehensible to an operator without machine interpretation or significant reference to special format tables or dictionaries.
2. Minimal vocabulary—Minimizing the number of terms will enable the operator to learn the entire vocabulary quickly.
3. Formal language definition—The grammar should be formally documented as a formal language to cultivate consistent usage in the initial assembly and continued maintenance.
4. Preclude redundancy—Insofar as possible, the language should be structured so as to discourage or prevent data redundancy and thereby preclude the possibility of internal contradictions. For example, the association of frequency with a given channel should be made at the transmitter or at a single receiver, but not both. Similarly, the nature of the data or message on a channel may be specified at transmitter or a receiver, but not both.
5. Resolve ambiguities—Common-language textual descriptions of database entities, particularly procedures, generally contain a number of nonexplicit or common-sense unwritten rules. These are potential ambiguities. Insofar as possible, such ambiguities should be resolved and made explicit during compilation of the database by using the best possible assumptions.
6. Identify assumptions—Don’t mix explicit and implicit data in same record. Assumptions should be documented in separate records within the database and should be clearly identified as such in the record REF_DATA addendum.
7. No action at a distance—Database procedural records should relate the observables and actions possible at just one node. There should be no action at a distance. For example, preclude “If the net control node is disabled, then...” Instead, “If the net control node is not heard for 15 minutes on channels, then...”

From these ground rules, we developed:

- A database structure
- A language concept
- Support tables and formats that would, in principle, fully describe the vocabulary and syntax of an engine-type, universal, table-driven language interpreter or formatters
- Target-language description tables, which would serve the same function for the various data-set formats into which the database would need to be translated to support the various user models.

Database Structure

It was early determined that the conventional database structures (hierarchical, relational, networking, etc.) did not adequately support the first criterion, that the format be easily
readable. Instead, it is clear that the largely free-form, keyword-driven format comes closest to satisfying that fundamental requirement. This is an affirmation of the basic structural concepts that have been implicitly adopted in the WAAM and SIMSTAR databases.

In the conventional database formats, the internal storage form of the data is almost completely positional; that is, a given data item is stored in a particular position within the data file. The meaning of a given data item is determined by its position, as recorded in master database structural control files.

These structures are not easily readable (other than by the database interpreter) and are subject to massive database corruption if, at any time in the editing processes and for any reason, the separate structural control tables fail to perfectly track the evolution of the database.

In the keyword format, by contrast, the data are stored in a form that, if not totally eliminating the dependence of the significance of a given data item, at least minimizes the dependence. Instead, keywords liberally used throughout the data records, according to certain prescribed syntax rules, serve both as semantic meaning keys for the analyst-operator and as structural signposts for the machine interpreter.

The keyword database provides a highly flexible database structure with a natural, readable data format; strong internal error checking; and inherent, downward-compatible growth qualities. These qualities are achieved to an unusually strong degree at some sacrifice in data storage efficiency. This, however, is not considered critical, because the database is quite small compared to typical databases.

Syntax

To provide a firm foundation for consistency in the compilation and future evolution of the database, it is important to document the language formally. A suitable mechanism for this is the Backus-Naur metalanguage format. In App. B, we first define a primitive vocabulary for this metalanguage description, then the grammar of the language itself.

The syntax describes a “record,” which will refer generally to from 1 to about 20 data items, forming a natural group by source and application. Roughly speaking, this may be considered to be a database “paragraph.”

A simple illustration of this format may be helpful:

```
NODE gep2 fixed TERMINAL vtterm RX_TYPE brll32 SUB
    TX_TYPE an133 SUB REF_DATA

SUB RECEIVER br1132 0.003 0.03 8.0 6.0 20.0 7.0 0.99 6.5 REF_DATA
```

These records describe a node named “gep2” whose form is “fixed” and which contains a “terminal” named “vtterm.” The terminal has a receiver whose characteristics are lumped into another (sub) record called “br1132.” The format of the receiver record “br1132” shows a
"positional" rather than "keyword" notation for the arguments. The keyword glossary clarifies precisely the various parameters, and this is used when displaying the values to the analyst. When printed in report format, the values appear in columns whose headings clarify their meaning. Either format, keyword or positional, is acceptable on input. The internal storage format depends on the database system used. The output may also be in either format.

Partial Glossary

A partial glossary, following the syntax rules of App. B, is given in Table 4.3. This has been particularized to the needs of WEDCOM and most, but not all, of WAAM. The initial draft of this glossary conforms to the rules insofar as possible, but its construction has highlighted the tedium and practical difficulty of doing so without errors using direct manual means. In the final system, the keyword glossary editor will ensure that all glossary entries comply fully with the rules.

Aside from the mechanical but tedious rule-following requirements, the main function in compiling the glossary is to ensure that each of the entities and properties in the target data set has a corresponding entity or property (corresponding semantically, not necessarily in name) in the glossary. This is satisfied by the initial glossary with respect to WEDCOM and with respect to the basic WAAM object vocabulary (from which there may be deviations in recent versions of the WAAM database).
### Table 4.3

**Keyword Glossary**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Level</th>
<th>Primary/Secondary</th>
<th># Values</th>
<th>Value Name</th>
<th>Format</th>
<th>Units</th>
<th>Meaning</th>
<th>Child Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMENT</td>
<td>P</td>
<td></td>
<td>1</td>
<td>Comment</td>
<td>string</td>
<td></td>
<td>Comment, arbitrary length</td>
<td>None</td>
</tr>
<tr>
<td>NODE</td>
<td>0</td>
<td>P</td>
<td>2</td>
<td>name</td>
<td>string</td>
<td>none</td>
<td>Internal name</td>
<td>TERMINAL, POSITION, ORBIT, TRAJECTORY, DAMAGE, VULN, PROCESS, AC_TRAJECT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>type</td>
<td>string</td>
<td>none</td>
<td>Fixed, ground, mobile, rocket, satellite, airborne.</td>
<td></td>
</tr>
<tr>
<td>POSITION</td>
<td>S</td>
<td>4</td>
<td>1</td>
<td>name</td>
<td>string</td>
<td>none</td>
<td>Internal name</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>time</td>
<td>float</td>
<td>hr</td>
<td>Time from start of scenario</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>lat</td>
<td>float</td>
<td>deg</td>
<td>lat</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>long</td>
<td>float</td>
<td>deg</td>
<td>N plus</td>
<td></td>
</tr>
<tr>
<td>ORBIT</td>
<td>S</td>
<td>8</td>
<td>1</td>
<td>name</td>
<td>string</td>
<td>none</td>
<td>Internal name</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>semi-major-axis</td>
<td>float</td>
<td>km.</td>
<td>Orbit semi-major axis, a</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>perigee</td>
<td>float</td>
<td>deg</td>
<td>Arg of perigee, w</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>eccentricity</td>
<td>float</td>
<td>—</td>
<td>Eccentricity, ε</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>incline</td>
<td>float</td>
<td>deg</td>
<td>Inclination, i</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ra-asc-node</td>
<td>float</td>
<td>deg</td>
<td>Right ascension of ascending node</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>anomaly</td>
<td>float</td>
<td>deg</td>
<td>True anomaly at epoch</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>epoch</td>
<td>float</td>
<td>hr</td>
<td>Epoch time</td>
<td></td>
</tr>
<tr>
<td>TRAJECTORY</td>
<td>S</td>
<td>6</td>
<td>1</td>
<td>name</td>
<td>string</td>
<td>none</td>
<td>Internal name</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td># of times</td>
<td>int</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>time</td>
<td>float</td>
<td>hr</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>lat</td>
<td>float</td>
<td>deg</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>long</td>
<td>float</td>
<td>deg</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>alt</td>
<td>float</td>
<td>ft</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keyword</td>
<td>Level</td>
<td>Primary/Secondary</td>
<td># Values</td>
<td>Value Name</td>
<td>Format</td>
<td>Units</td>
<td>Meaning</td>
<td>Child Keywords</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>-------------------</td>
<td>----------</td>
<td>------------</td>
<td>--------</td>
<td>-------</td>
<td>---------</td>
<td>----------------</td>
</tr>
<tr>
<td>A/C TRAJECT</td>
<td>S</td>
<td>4</td>
<td>name</td>
<td>string</td>
<td>none</td>
<td>Internal name</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>number</td>
<td>int</td>
<td>unit</td>
<td>Number of point pairs to follow</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>range</td>
<td>float</td>
<td>mi</td>
<td>Range from ref pt</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>height</td>
<td>float</td>
<td>ft</td>
<td>Alt above sea level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TERMINAL</td>
<td>S</td>
<td>7</td>
<td>name</td>
<td>string</td>
<td>none</td>
<td>Internal name</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>type</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>#tx</td>
<td>int</td>
<td>ea</td>
<td>Number of tx served</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>tx freq</td>
<td>float</td>
<td>kHz</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>#rx</td>
<td>int</td>
<td>ea</td>
<td>Number of rx served</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>#rx in use</td>
<td>int</td>
<td>ea</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>rx freq</td>
<td>float</td>
<td>kHz</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX_SETUP</td>
<td>S</td>
<td>6</td>
<td>name</td>
<td>string</td>
<td>none</td>
<td>Internal name</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>clipper on</td>
<td>string</td>
<td></td>
<td>&quot;0&quot; = in, &quot;1&quot; = out</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>modulation</td>
<td>string</td>
<td></td>
<td>Modulation type (MSK, CFSK, CSK, FSK, ...)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>bandwidth</td>
<td>float</td>
<td>Hz</td>
<td>Bandwidth (?)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>edac</td>
<td>string</td>
<td>—</td>
<td>a.g., HAM15/11, ...</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>frequency</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.3—continued
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Primary/Secondary</th>
<th># Values</th>
<th>Value Name</th>
<th>Format</th>
<th>Units</th>
<th>Meaning</th>
<th>Child Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>RX_TYPE</td>
<td>S</td>
<td>10</td>
<td>name</td>
<td>string</td>
<td>none</td>
<td>Internal name</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>type</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>lo_freq</td>
<td>float</td>
<td>MHz</td>
<td>Lower frequency limit</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>hi_freq</td>
<td>float</td>
<td>MHz</td>
<td>Upper frequency limit</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>noise_fig</td>
<td>float</td>
<td>dB</td>
<td>System noise fig. No external noise.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>impl_loss</td>
<td>float</td>
<td>dB</td>
<td>Implementation loss, dB</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>clip_bw</td>
<td>float</td>
<td>Hz</td>
<td>Clipping bandwidth</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>clip_level</td>
<td>float</td>
<td>dB</td>
<td>Normalized clip level (more-TBD)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>reliability</td>
<td>float</td>
<td>unit</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>mtr</td>
<td>float</td>
<td>hr</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX_SETUP</td>
<td>S</td>
<td>6</td>
<td>name</td>
<td>string</td>
<td>none</td>
<td>Internal name</td>
<td>FREQUENCY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>modulation</td>
<td>string</td>
<td></td>
<td>Modulation type (MSK, CPSK, CSK, PSK, ...</td>
<td>ANTEENA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>bandwidth</td>
<td>float</td>
<td>Hz</td>
<td>Bandwidth (?)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>code</td>
<td>string</td>
<td></td>
<td>e.g., HAM15/11, ...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>main</td>
<td>float</td>
<td>dBw</td>
<td>Forward power output</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>frequency</td>
<td>float</td>
<td>kHz</td>
<td>Operating frequency</td>
<td></td>
</tr>
<tr>
<td>FREQUENCY</td>
<td>S</td>
<td>2</td>
<td>name</td>
<td>string</td>
<td>none</td>
<td>Internal name</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>frequency</td>
<td>float</td>
<td>kHz</td>
<td>Frequency monitored by parent Tx or Rx</td>
<td></td>
</tr>
<tr>
<td>TX_TYPE</td>
<td>S</td>
<td>7</td>
<td>name</td>
<td>string</td>
<td>none</td>
<td>Internal name</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>type</td>
<td>string</td>
<td></td>
<td>External name, e.g. AN/ARC-182</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>lo_freq</td>
<td>float</td>
<td>MHz</td>
<td>Lower frequency limit</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>hi_freq</td>
<td>float</td>
<td>kHz</td>
<td>Upper frequency limit</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>reliability</td>
<td>float</td>
<td>unit</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>mtr</td>
<td>float</td>
<td>hr</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>max_point</td>
<td>float</td>
<td>dBw</td>
<td>Max available power output</td>
<td></td>
</tr>
<tr>
<td>Keyword</td>
<td>Level</td>
<td>Primary/Secondary</td>
<td># Values</td>
<td>Value Name</td>
<td>Format</td>
<td>Units</td>
<td>Meaning</td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
<td>-------------------</td>
<td>----------</td>
<td>------------</td>
<td>--------</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>ANTENNA</td>
<td>S</td>
<td>5</td>
<td></td>
<td>name</td>
<td>string</td>
<td>none</td>
<td>Internal name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>type</td>
<td>string</td>
<td>none</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>length</td>
<td>string</td>
<td>ft</td>
<td>Length of radiator element</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ft</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>lo_freq</td>
<td>string</td>
<td>MHz</td>
<td>Lowest frequency at which used</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>hi_freq</td>
<td>string</td>
<td>MHz</td>
<td>Highest frequency at which used</td>
</tr>
<tr>
<td>NUC</td>
<td>P</td>
<td>5</td>
<td></td>
<td>name</td>
<td>string</td>
<td>none</td>
<td>Internal name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>lat</td>
<td>float</td>
<td>deg</td>
<td>Burst position in</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>long</td>
<td>float</td>
<td>deg</td>
<td>Standard</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>alt</td>
<td>float</td>
<td>km</td>
<td>Coordinates</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>burst_time</td>
<td>float</td>
<td>hour</td>
<td>Time of burst</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>yield</td>
<td>float</td>
<td>MT</td>
<td>Yield</td>
</tr>
<tr>
<td>NUC_DEVICE</td>
<td>S</td>
<td>10</td>
<td></td>
<td>name</td>
<td>string</td>
<td>none</td>
<td>Internal name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>device_id</td>
<td>string</td>
<td>—</td>
<td>Standard ID outside dB.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>fiss frac</td>
<td>float</td>
<td>—</td>
<td>Fission fraction</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>hydro_frac</td>
<td>float</td>
<td>—</td>
<td>Hydro yield fraction</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>mass</td>
<td>float</td>
<td>G</td>
<td>Mass</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>sum frac</td>
<td>float</td>
<td>—</td>
<td>Aluminum mass fraction</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>uran_frac</td>
<td>float</td>
<td>—</td>
<td>Uranium mass fraction</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>gamma_frac</td>
<td>float</td>
<td>—</td>
<td>Gamma yield fraction</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>neut_frac</td>
<td>float</td>
<td>—</td>
<td>Neutron yield fraction</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x_ray_frac</td>
<td>float</td>
<td>—</td>
<td>X-Ray yield fraction</td>
</tr>
</tbody>
</table>

**Threat**
### Table 4.3—continued

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Primary/Secondary</th>
<th># Values</th>
<th>Value Name</th>
<th>Format</th>
<th>Units</th>
<th>Meaning</th>
<th>Child Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAMMER</td>
<td>P</td>
<td>9</td>
<td>name</td>
<td>string</td>
<td>none</td>
<td>Internal name</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>lat</td>
<td>float</td>
<td>deg</td>
<td>Latitude</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>long</td>
<td>float</td>
<td>deg</td>
<td>Longitude</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>alt</td>
<td>float</td>
<td>km</td>
<td>Altitude</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>frequency</td>
<td>float</td>
<td>KHz</td>
<td>Center frequency</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>eirrp</td>
<td>float</td>
<td>dBW</td>
<td>Effective lossless vertical radiated pow</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>bandwidth</td>
<td>float</td>
<td>Hz</td>
<td>Bandwidth</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>conductivity</td>
<td>float</td>
<td>mho/m</td>
<td>Surface cond at Tx</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>epar</td>
<td>float</td>
<td></td>
<td>Relative dielectric const at Tx</td>
<td></td>
</tr>
</tbody>
</table>

**Attack**

| MISSILE       | P                 | 9        | name       | string | none  | Internal name                  |                                |
|---------------|-------------------|----------|------------|--------|-------|--------------------------------|                                |
|               |                   |          | type       | string | —     | external reference name        |                                |
|               |                   |          | lat        | float  | deg, N | lat of launch site             |                                |
|               |                   |          | long       | float  | deg, W | long of launch site            |                                |
|               |                   |          | mirmv      | int    | ft    | number of MIRVs per missile    |                                |
|               |                   |          | cep        | float  |       | CEP                            |                                |
|               |                   |          | misk       | int    | —     | number of missiles of type at site |                                |
|               |                   |          | alloc      | int    | —     | number already allocated       |                                |
|               |                   |          | sitename   | string |       | launch site external reference name |                                |

**TARGET**

| P             | 5                 | name       | string | none  | Internal name                  |                                |
|---------------|-------------------|------------|--------|-------|--------------------------------|                                |
|               |                   | type       | string | —     | Name for external reference    |                                |
|               |                   | pvna       | int    | —     | Physical vulnerability number  |                                |
|               |                   | location   | string | —     | «Buried», «surface», <airborne>, etc. |                                |
|               |                   | #aimpoints | int    | —     | Number of aimpoints             |                                |
Table 4.3—continued

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Level</th>
<th>Primary/Secondary</th>
<th># Values</th>
<th>Value</th>
<th>Name</th>
<th>Format</th>
<th>Units</th>
<th>Meaning</th>
<th>Child Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>BURST</td>
<td>P</td>
<td>8</td>
<td></td>
<td>name</td>
<td>string</td>
<td>none</td>
<td>Internal name</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>launch_time</td>
<td>float</td>
<td>hrs</td>
<td>Launch time, hrs after To</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>impact_time</td>
<td>float</td>
<td>hrs</td>
<td>Impact time, hrs after To</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>hbr</td>
<td>float</td>
<td>ft</td>
<td>Height of burst, above ground</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>launcher_id</td>
<td>string</td>
<td>—</td>
<td>Launcher iid</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>trajectory_id</td>
<td>string</td>
<td>—</td>
<td>Trajectory iid</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>target_id</td>
<td>string</td>
<td>—</td>
<td>Target iid</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>weapon_id</td>
<td>string</td>
<td>—</td>
<td>Weapon type iid</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Damage

| DAMAGE | S     | 3                 |          | name  | string | none   | Internal name |                   |
|        |       |                   |          | time  | float  | min.    | Time from start of scenario |
|        |       |                   |          | pk    | float  | —      | Probability of kill |

Scenario

| TIME   | P     | 7                 |          | name  | string | none   | Internal name |                   |
|        |       |                   |          | year  | int yy | years  | Scenario start year, 19yy, all GMT |
|        |       |                   |          | month | int mon | month  | month |
|        |       |                   |          | day   | int dd  | days   | day |
|        |       |                   |          | hour  | float  | hours  | hour |
|        |       |                   |          | duration | float  | hours  | Duration |
|        |       |                   |          | sol_flux | float  | —      | B-22 W/sq. 10.7 cm solar flux |

MESSAGE

| MESSAGE | P     | 4                 |          | name  | string | none   | Internal name |                   |
|         |       |                   |          | type  | string | —      | External name |                   |
|         |       |                   |          | source_desig | string | —      | designation of source node |
|         |       |                   |          | time  | float  | hrs    | Time of msg, orig. at source node |
5. SUPPORT SOFTWARE FOR PROTOTYPE
COMMON-FORMAT DATABASE

This section discusses some implementation tools suggested by the common-format database
design discussed in Sec. 4. Each subsection that follows describes a tool, where it fits in the
processing, and who would use it and includes an order-of-magnitude estimate of the size of
the software tool.\footnote{Estimates of code size are order-of-magnitude judgements. The C
language is typical of high-level languages, but Ada is also considered here.} The final subsection summarize these software tool sizes and provides
rough cost estimates.

KEYWORD GLOSSARY MANAGER

General Description

The keyword glossary manager (KWGM) is a specialized, full-screen editor software package,
providing the usual add, delete, and modify functions, specialized to the requirements for
creation and maintenance of the keyword glossary (KWG) described in Sec. 4. The presenta-
tion will make maximum effective use of separate windows displaying status, selection op-
tions, and input-output devices.

The KWGM will also extensively check every offered entry to ensure the initial and contin-
ued integrity of the glossary and the Database. Initial entries are prompted for each
required object in sequence according to the KWG format. Where there are a limited number
of valid options, those options are presented for selection. Every offered edit entry is checked
before acceptance, and any entry that would result in a violation of format, keyword ground
rules, or glossary-Database congruence is rejected with complete error diagnostics and a sug-
gested correction. It should be impossible to affect the KWG via the KWGM in such a way as
ever destroy the glossary-Database congruence or to violate the keyword glossary format
and ground rules.

The KWGM will also provide utility functions for such services as “borrowing” any part of the
structure of another keyword glossary as a starting point, “reporting” the structure of the
glossary in useful formats, and the usual file-handling save, load, copy, rename, delete, and
directory-change functions.

Functional Description

Internal Integrity Enforcement. To ensure initial and continued compliance with the
ground rules and internal consistency within the glossary, the KWGM must fully check every
offered entry, deletion, or modification to the glossary for adherence to the following rules
before accepting it into the glossary:

- Each keyword within the glossary must be unique. The KWGM will report any keywords
  that are “close” in the sense of a spelling checker.

---

104
• If a keyword does not already appear in another's child list, it is intended to be a primary keyword. If a keyword is not designated as a primary keyword, its parent must be entered first.
• Each keyword is followed immediately by a "name" value.
• All values must be defined in terms of name, format, units, and definition.
• Every child or secondary keyword must appear in one child list.

External Integrity Enforcement. The KWCM must fully check every offered deletion or modification of a glossary entry for the following points of consistency with the associated database before accepting it:

• No child can be deleted from a child list if there is an instance of its occurrence in the database.
• No keyword can be deleted from the glossary if there is an instance of its occurrence in the database.
• No keyword can be added to a child list if there is an instance of its occurrence in the database as a primary keyword.

Adding or Deleting Values. Adding or deleting values to or from the value list must be accomplished in a block of simultaneous changes to both glossary and database such that no values are ever left in the database without a corresponding definition in the glossary and such that no instance of a keyword in the database lacks any required value per the glossary value list.

Reports. The following reports will be available:

• Keyword genealogy for any secondary keyword
• Complete genealogy tree printout
• Complete genealogy chain(s) (not necessarily unique) for any specified value name.

Design Features

Indexes. To expedite rapid execution of the principal editing functions, there will be an index system, recreated from the glossary and Database on each invocation and updated with each acceptable edit entry.

The following indexes will be created and maintained throughout each session:

• The desginator list is an index of all "name" values appearing under each keyword and the number of the Database record in which they occur. This list is sorted alphabetically by keyword and is derived from the Database.
• The genealogy list is an index of all keyword ancestors of each keyword. This list is sorted alphabetically by keyword and is derived from the glossary.
• The ancestor list is an index of all keyword-pair ancestors of each keyword pair. This list is sorted alphabetically by the complete keyword-pair ancestor list, is derived from the Database, and is checked for consistency with the genealogy list.

• The values list is an index of all value names appearing in the glossary and the keywords under which they may appear, the Database record number, and the designator of all instances in the database. This list is sorted alphabetically by value name and is derived from the glossary and the Database.

Modular Structure

The KWGM will be highly modular to maximize module sharing between the various KWGM functions and between the KWGM and the Database manager and the report generator. The principal high-level modules include:

• Editor initiation
• Main program options
• Glossary record parser
• Database record parser (developed under Database manager task)
• Genealogy list generator
• Ancestry list generator (developed under Database manager task)
• Values list generator (developed under Database manager task)
• Designator list generator (developed under Database manager task)
• Phonetic similarity list matcher
• Glossary record builder
  — Add and check keyword
  — Update genealogy list
  — Add value line
  — Add and check child
• Glossary record editor
  — Glossary record editor screen generator
  — Glossary record inquiry generator
  — Full-screen record editor
  — Glossary record integrity check
  — Update indexes
  — Update glossary
• Report generator.

Development Program

The possibility of building upon an existing editor has been considered. The problem addressed here has some elements common to other editors (the basic add, delete, modify and file handling functions) and a number of factors that are unique to the KWGM. On balance,
it appears that the advantage in the use of an existing editor would generally be outweighed by the disadvantage of the added difficulty of customizing it to the KWG editor requirements unless the full editor source code were available in a widely portable, high-level language. Overall, it appears most realistic to assume that the entire editor would be developed from scratch, probably making maximum use of widely available screen-generator utilities.

Size and Availability

The KWGM software support tool will require about 10,000 lines of C code of about medium complexity. This utility is not commercially available.

DATABASE MANAGER

Database Manager Functional Description

The Database manager (see Figure 5.1) is the main interface to the Database. It is used to enter all data into the Database and can also be used for data retrieval and for entering data corrections. It is a glossary-driven, data-entry forms-and-menu system. The menu screens would display the Database-manager-created screens for each keyword data type and set up fields with correct units. The menu screens would display keyword types and definitions from the glossary in a human-readable form. On selecting a valid data type from the menu, the user can put the system either into a data-retrieval or a data-input mode. In the data-retrieval mode, the system would build an appropriate search string, which would be matched against the database records to retrieve the desired information. Incomplete screens serve as templates to retrieve matching information from the database. Template matching is the only browsing capability provided in the first version. Alternatively, the system could be put into the add-data mode, which would bring up the appropriate data entry form to ensure that

![Database Manager Diagram]

Fig. 5.1—Database Manager
the data entered are consistent with the glossary and with the existing data. Entered values are checked for conformity with limits entered in the glossary. Before including the information in the database, the Database manager would search the data statements to ensure that there was no inconsistency with existing data items. The system must also ensure that there are no duplicate entries in the database. New classes of data could be entered only after modifying the keyword glossary to accept the new class of data.

Design Description

The key features of the proposed design are the use of a glossary-driven menu-and-form driver, and a glossary-driven Database executive. The menu-and-form driver uses the entries in the keyword glossary to determine which menu options to offer to the user and to determine the location and labeling of the data-entry fields in the forms. The simplest version of the menu-and-form driver would consist of a menu for selecting the data types to be edited (allowed keywords) and a generic data display and entry form.

The other main module would be the Database executive. At the heart of the Database executive are efficient algorithms for conducting pattern matching searches and statement unification over the database. The statement unification operation would require efficient implementation, because this feature would be applied frequently to resolve relationships between interacting entities. For example, given a procedure, it would be important to determine all of the transmitter and receiver equipment that the procedure referred to. Data checking is also important. During data entry, the menu-and-form driver would pass incomplete keyword records or search strings that would need to be pattern matched into the Database to determine whether inconsistent information was being entered. The conflicting database records would have to be passed to the menu-and-form driver for presentation to the user.

Note the strong similarity of this tool to the KWCM. Careful design should allow the two to share significant portions of software. The manager here is significantly expanded to interface to the Database. The display formatter has increased capabilities for screen generation. The Database manager is probably two to four times the size of the KWCM. The user interface is especially critical here, as it is in the data-set translators with which it can share these interactive features.

The principal feature of the Database manager will be a screen-oriented interface that both presents and accepts data in a tightly controlled manner. A screen generator should drive the display directly from the syntax and semantic definitions that are contained in the keyword glossary file. This approach would preclude disintegrations between versions of the keyword glossary and the current Database manager configurations. Figure 5.2 presents a suggested screen layout using multiple windows that would accommodate the special needs of the SSCP Database management.

The various sections of the screen have the following functions:

- The error message area is used to output errors that the Database manager has detected, such as data-to-keyword disintegrity or a computer system error (e.g., disk full). On a monochromatic system, these errors should blink. On a color system, they should appear
in a suitable highlighting color, such as red. These error messages are transient and would be erased from the screen after the pressing of any key.

- The **operator communication area** is used for prompts from the system and for the corresponding user responses, such as input and output file specifications.

- The **page area** indicates how many pages of data there are, so that the user can switch between pages for reviewing large quantities of information.

- The **date and time area** displays the current date and time.

- The **general display area** contains the forms for data entry. The coding of the information form the various entry locations would be transparent to the user. Under some conditions, the general display area would be divided into transient windows supporting more complex transactions. Fig. 5.2 shows three such transient windows within the general display area:
  - Text display
  - Data entry
  - Entity selection.

The entity selection window allows the user to designate existing entities by pointing and clicking, rather than by retyping the entities’ names. This avoids typographical errors.

- The **enabled key area** shows which cursor and control keys are currently active (up arrow, down arrow, page up, page down, etc.).

- The **function key area** displays the current assignments of each of the function keys, directly below its number. Multiple layers of key codes can support large numbers of special functions.
• The **path area** displays the path and file specifications for the file currently being reviewed or revised.

• The **REF-DATA area** shows the source of the data. This facilitates the maintenance of database consistency when a given source document is changed.

**Prototypes or Experience**

There is currently no full example of the proposed Database manager. However, significant parts of the concept have been implemented and studied separately. The RAND SIMSTAR 85-A experiment demonstrated that information could be encoded into and retrieved from a keyword-driven format. In the current STRATCAM database manager, the extensive use of menus and forms has demonstrated this technique's ability to reduce operator errors and improve the quality of the resulting database.

**Development Program**

The proposed development program for the Database manager should be conducted in two phases. In the first phase, the basic menu-and-form driver and the Database executive would be developed. In a later phase, both of the basic modules would be expanded, and a graphics presentation package would be developed for the display of communications networks and procedures analysis. A key requirement for the Database executive would be an efficient algorithm for matching search patterns against the contents of the database.

In the second phase, graphics and intelligent analysis capabilities would be built onto the basic capabilities to manage the C3 information. A key algorithm would be a goal-directed planning algorithm that could systematically identify nodes and equipment references in message-handling procedures and that could thereby construct the message paths and networks implicit in the procedural descriptions. Such analytic capabilities would require the development of a rule-directed production system for the analysis.

**Size and Availability**

The Database manager software support tool will require about 20,000 lines of C code of medium complexity. This utility is not commercially available.

**DATA-SET TRANSLATORS**

**Functional Description**

The data-set translators are those programs which, interacting with an analyst, read the Database, select data items, and translate them into input sentences (records) of the specific model to run a simulation. As with any language translator, the program must know both languages fluently. This knowledge is embedded in two structures. The keyword glossary knows how Database items are structured. The model-specific data-set grammar (or format script) knows how to construct the required data sets from database items. To change from one model to another should in principle require only changing the data-set grammar. In
fact, it may be more complicated than that, but careful design and considerable familiarity with the many different models supported should minimize code specialization.

A data set is generated in close cooperation with the analyst. Thus, a well-constructed user interface is required. With this interface, the analyst chooses the model, selects sets of data to be translated, enters model run data, and tailors data to his special needs. All of these capabilities must therefore be provided. The model-specific data-set grammar is represented as a script that has been written specifically for the model and that knows how to retrieve required data from the database representation. This script is written by a programmer who must know the model's data-set requirements and must specify precisely how the required items are retrieved and how the analyst will be allowed to change them. Selection, sequencing, and formatting are steps in this process.

Database contents can be matched to model inputs in two different ways:

1. The translator can scan the database (every item), determine if the model data set requires the item, and let the analyst determine whether to include the item or not.
2. The translator can scan the data set input requirements, look for matches in the database, and offer them to the analyst for inclusion.

The first way has the advantage that new database materials might be located for which the model script was not planned. The analyst is made aware of this omission and can institute measures to improve the model script.

The second way is quicker, because the database is searched only for "needed" items, this need being determined by the model format script. This has a decided advantage for models (such as WEDCOM) that require only a small fraction of the database's contents.

In either case, the no-match cases should be called to the attention of the analyst, to make him aware of possible deficiencies in his model in the first case and of possible deficiencies in the database in the second.

Ideally, both methods would be implemented, with the choice left to the analyst. The impact is that dual-format scripts would be needed for each model. They would, however, be similar to each other and thus easier to generate.

**Design Description**

Once the user has selected the model (Figure 5.3), its script is read and leads the user step by step through the data-set preparation process. When data from the database are needed, the user may be prompted to make a selection. Once selected, data are set aside until the sequencing can begin. Sequencing refers to the ordering of data required in a data set to ensure that its semantics match those of the model's data reader, which expects its data in a very specific order. Likewise, formatting is very model-specific and is driven by the model's grammar format script.
Prototypes or Examples

The key to translation is embodied in the model-translator script, which is written by a qualified programmer and links specific data-set quantities to their location, retrieval, and functional modification, as necessary. Interaction with the user is necessary, because not all model inputs are found in the Database. For example, number of Monte Carlo runs, specific report generator flags, and data-set filename all have to be entered. For such items, the script must collect answers from the user. Even for data-set entities contained in the database, the analyst may wish to limit the direct retrieval or perhaps to edit the result of a general retrieval command from the script selectively. The model-translator script is therefore a computer program.

For example, the model may require an entry indicating how many nodes are about to be listed. The number of nodes would not be known until the nodes had been retrieved and edited, selectively deleted, or supplemented. Thus, when all nodes are complete, the keyword (data set) line "NODES 27" (for example) can be written, followed by the 27 formatted node records.

A more serious matter arises in translating procedures from the Database to a data set. There is no such thing as a "standard" representation of procedures in the current models (e.g., WAAM, STRATCAM). This is an important database issue. The procedural "data" may
be collected from the database, but then they must be formulated into more complex structures for input to the model. The procedure representation system used in the SSCF Database provides a more general system for representing procedures than is contained in any of the models. Therefore, producing model inputs entails forming a collection of model input entities acceptable to the model. For instance, in the STRATCAM model, the system must generate events, message handling programs, and node delay processes as a consequence of data entities in the SSCF Database that have an entirely different representation. In essence, the mapping process must first identify those physical assets that are referenced by a given operational procedure. Then the mapper must apply the procedure to those assets so that the input entities required by the target model are produced.

The process of identifying procedurally referenced entities is equivalent to the "unification" problem. In other words, one must identify those transmitters, receivers, and nodes that can have the procedure consistently applied to them. For instance, if a particular type of HF equipment is associated with a given network, one must build a list which contains the actual names (i.e., valid substitutions and/or unifications) of the designated equipment in the database. If a message is to traverse certain nodes, a list of those nodes must be constructed, and any required process delays must also be listed.

The final input production process entails applying the procedure mapping rules to each item in the entity lists, one at a time. If a STRATCAM target is being produced, message-start events must be added to the exogenous event records, and routing command records must be generated for each transmitter, receiver, and node that appears on the procedure reference list. The use of an expert system shell known as a production system is one possible approach to generating the required records. In this approach, a relatively simple program reads and applies a series of rules in the if-then form. If data and conditions as specified in the if part are true, then the actions described in the then part are executed. A node being a message injection node would be an example of an if clause and the operations for generating the STRATCAM message-event record would be specified in the then clause of the production rule.

Development Program

Phase I of this development will be to organize and categorize the supported model data sets, determining, for each item required, where the item can be retrieved from the Database, or, if it is not already in the Database, how to solicit it from the user. It is also necessary to document the formats required of the output, the exact nature of flow control in the data set for cases of skipped or missing data (meaning that defaults may be accepted in the model or that certain processing steps may be omitted). The details of procedure translation must be studied here also.

During Phase II, it will be decided how the results of Phase I are to be represented in the program. In an interpreted language, statements express the desired steps and functions. But there is no harm in using a computer language directly to describe the processing steps. If an ad hoc language is created, there is some chance that later improvements to a model might not have been anticipated and could not be accommodated; a computer language already has the requisite generality.
Phase III will consist of writing the code (for each model) that will carry out the requirements of Phase I using the method of Phase II. The first part of this phase is to interface the user with the model script (whichever form it takes). Then, most of the Database editor features should be usable from the interface. In the second part of Phase III, the model translator scripts must be written for each of the supported models. This second part includes the generation of appropriate data set code for procedures, an item which is very model dependent.

Size and Availability

The Data-set translator shares many modules with earlier programs, especially the Database manager. The translator should require about 30,000 lines of C code of medium complexity and is not commercially available. This total is composed of 10,000 lines that are common plus 5,000 to 10,000 lines per supported model.

REPORT GENERATORS

Functional Description

The administrator and users of the Database need a variety of reports in the course of using the system. At least the following should be provided:

- The keyword glossary (prettyprint)
- All or selected sections of the Database (prettyprint)
- The translator scripts (for the administrator) (prettyprint)
- Special reports in special formats: nodes, equipments, topology, etc.
- Printed versions of data-set output with interpretation.

As used here, prettyprint means a fixed, standard format meant to clarify the data and its hierarchical relationships.

In addition, of course, graphic and map-format outputs would be useful but would require additional programming and hardware availability. A minimum printer capability should include a boldface font for printing keywords. Likewise, the display console should have highlighting or reverse video capability. Finally, all reports should be directed to the display console, the printer, or both, at the user’s option.

Design Description

At the topmost level, the user is offered a menu of available reports (Figure 5.4). Either the selected one is automatically produced immediately or the user is led into further selection processing. This unit shares many modules with other units, and the prettyprinting components may already have been built for display generation in the KGM and the Database manager. The specialty reports and such user-definable formatting as is provided must be separately programmed.
Fig. 6.4—Report Generator Structure

It is appropriate to consider using commercial report generators, especially those accompanying DBMSs in which the Database may be embedded. Such systems usually contain a flexible report-generator capability. Using them would serve many of the needs of this section. They are not well suited to prettyprinting, however.

The report generator is essentially menu driven, offering a selection of reports of interest to the Database administrator and to the analyst-user of the Database. The functions of administrator and user are quite different, so the first menu branching should be based on this function. The administrator is probably more interested in individual pieces of data and the REF_DATA information (source, installer identifier, date, security classification, etc.) and less concerned with the big picture, selection processing and the like.

The user must have control over the selection of data displayed, which may consist of editing a default selection (retrieval) command:

All NODES with Latitude > 20 and Latitude < 50.

Development Program

The bulk of the development work consists of writing modules that implement parts of the many reports. While individual programs could be written for each report, there would be
considerable duplication in the functions provided. It is probably more difficult to interact with the user if many fixed report writers are supplied (mainly due to lack of uniformity).

Generally, each report involves

- Finding the data in the Database,
- Formatting data lines,
- Formatting header lines, and
- Orderly output to display or printer.

Only the first of these has interactive components with the user specifying or selecting the data to be displayed or printed. The last steps are quite routine programming.

**Size and Availability**

The report generator modules should require about 20,000 lines of C code of medium complexity, though the total is highly dependent on the number and types of reports. We assumed the minimum five requirements. None of these are commercially available.

**REFERENCE DATA—THE REF_DATA**

Each record is terminated with the expression &ref. This &ref is simply an abbreviation for the record's administrative support data. REF_DATA will tend to consist of an unintelligible (machine readable but not particularly human readable) set of letters and numbers. The reference data would be "code" for important information that must accompany each database record.

In real records, the reference data entry is a significant part of and may, in fact, exceed in size the model-data portion of the record. Four examples of a REF_DATA follow:

1. & S V7 R29 D89 D571 EOR
2. & S D89 D571 EOR
3. & D571 EOR
4. &571

Number 1 would tell the analyst that this database record is secret; the primary source is EAP Vol. VII; and RAND employee number 29 made the entry in April 1989. D571 tells the analyst specifically where to look in the Database Description and Users' Guide for detailed information concerning this database record. EOR designates the end of each record. This long REF_DATA entry gives the analyst a fair amount of information (classification, major source, organization producing the information, and vintage, by inspection). This REF_DATA entry is somewhat long and unwieldy, however, even if the spaces are removed. Number 2 gives the analyst the classification and vintage of the item, as well as where to go to look it up. Number 3 simply tells the analyst where to look it up, and number 4 is a more concise version of the same thing.
Items the REF_DATA must contain, or point to, include the following:

1. Security classification of the record
2. Source document or documents for the record's data
3. Person and organization entering the record into the database
4. Date of entry
5. Pointer to detail description of the database record.

In brief, this is the Database administrator's part of the record, but it is also important to the analyst-user. The keyword for the reference data (e.g., &ref) also serves to terminate the model-data part of the record and begin the administration part.

DEVELOPMENT COST ESTIMATES

We estimated the cost to proceed with the full development and documentation of the SSCF Database and the associated support software (keyword glossary, KGM, Database manager, data-set translators, and report generator). Our purpose for developing these rough-order-of-magnitude estimates was to allow one to weigh the cost against the potential benefits of the concept.

Our Method of Estimating the Software Development Effort

In performing this task, we followed the recommended practice [25] of making multiple estimates, using a combination of analogy, judgement, and bottom-up analyses. We also compared results with another organization: the Defense Communications Agency (DCA) (now the Defense Information Systems Agency [DISA]). These multiple estimates were reasonably consistent with one another.

Our estimates were constructed from individual analysis plus a series of about ten working sessions, which can be lumped to four more or less discrete steps:

- Step 1 involved defining the package in general terms, estimating the effort by analogy, making top-down estimates, and then refining them in a group discussion.
- Step 2 involved joint working meetings with personnel from the DCA and Science Applications International Corporation (SAIC), the SETA contractor.
- Step 3 involved discussions with DCA, SAIC, and the Air Force Center for Studies and Analysis.
- In Step 4, the RAND team went through the most detailed iteration. We refined the functional descriptions and bottom-up estimates of each of the software packages and conducted a final group discussion and negotiation to arrive at the final estimates.

Support Software Development Effort

These estimates should be viewed as first-cut, order-of-magnitude figures and are subject to the following caveats:
1. We assumed that the effort would be performed by small teams of senior programmers with good programmer skills, rather than larger teams with junior skills.

2. We did not include long-term or continuing tasks, such as software modification and documentation or configuration management.

3. We assumed that the separate tasks would be coordinated so that modules in one task would support other tasks.

It has been suggested that the first assumption above may have been ill-advised, in that, when the software packages are competed, the bid would probably be made and the task performed using larger teams with junior skills. This would mean a somewhat higher cost than we project. On the other hand, the DoD has the means to influence or control the outcome, specifying quality workmanship in the Commerce Business Daily announcement and in the Request for Proposals. Even more importantly, DoD can fix the contract price at a level that would require an efficient effort.

Development of the four major support software tools should consist of the items listed in Table 5.1. The code, which we recommend be written in Ada or C, is of medium complexity and is not commercially available. These estimates assume good system design in advance of coding. Approximately one third of the time should be devoted to design issues; one third to coding; and one third to checkout, debugging, and documentation.

### Database Development and Conversion

In the process of defining and scoping the effort required to develop the keyword glossary editor, we estimated the effort required to develop the keyword glossary itself (Table 5.2).

The keyword glossary, which would be part of the Database, should require from four to ten staff months to develop. Note that the scope of this effort is a direct function of the number of the codes supported.

We did not formally estimate the cost of converting various databases to the common format but our prototype effort did require us to convert some SIMSTAR, WAAM, and EAP Vol. VII data. The process of performing a part of each task allowed us to form an opinion concerning

<table>
<thead>
<tr>
<th>Module</th>
<th>Approximate Lines of C Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyword glossary manager</td>
<td>10,000</td>
</tr>
<tr>
<td>Database manager</td>
<td>20,000</td>
</tr>
<tr>
<td>Data set translators</td>
<td>30,000</td>
</tr>
<tr>
<td>Report generator</td>
<td>20,000</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>80,000</strong></td>
</tr>
</tbody>
</table>
the effort required to accomplish the full task. Admittedly, however, we had only to satisfy ourselves and not a set of needs and standards specified by others. On that basis, our estimates could be expected to be low.

The conversion of the facilities, hardware, and attack scenario data from being model specific to a common format was seen to be straightforward. Based on our prototype effort, it appeared to us that these conversions should require no more than two or three staff months and could possibly be accomplished in half that time. Upon reflection, however, it would probably be prudent to use the high end of our estimates, or even double them as a worst case scenario; so, for our estimate, we assumed a range of from three to six staff months.

The most difficult (but probably the most valuable) conversion would be representation, preferably in a common format, of the procedures and operational data (described in Sec. 3). Scaling up from our prototype effort, we estimate that this would require several staff-months (possibly as much as a staff-year) to accomplish and should only be attempted by someone experienced in C2 operations, modeling, and analysis. Again, we will be conservative and assume a range from nine to twelve staff months.

Summing these three parts of the database, we estimate that from 16 to 28 staff months, with a best estimate of 22 staff months, should be sufficient to convert the entire database to a standard, common format. (Based on having spent a few weeks at the task, we would predict that the "glitches" or problems with that database would more likely be in the substance than in the format.) It is not clear, however, that this effort should be "charged" to the SSCF Database approach; much of the work would actually be required for the ongoing database updates performed by the various analysis organizations.

The Cost of the Enterprise

At a current cost/productivity rate of about $11.50 per line of documented and tested code, the 80,000 lines of code translate to a cost estimate for the full package of approximately $920K. Because this study is preliminary, the estimates shown in our summary and conclusions are expressed to the nearest $100K. Our estimate of the efforts for the analysis and coding tasks as we saw them suggested a range from $800K to $1200K.

This wide range reflects uncertainties that exist not only because the specifications at the interface-drawing level have not been developed, but also because the level of desired sophisti-
cation and performance has not been decided upon. The uncertainty range should shrink substantially and does not suggest the likelihood of a large overrun on the software development effort. In fact, we should also point out that a quality effort at the lower end of this range is also quite possible, especially in today's lean and competitive market.

The conversion of the strategic communications database itself could probably be accomplished in about 16 to 28 staff months, for roughly $240K to $420K. We acknowledge at this point, however, that this database development estimate is definitely cruder than the software development estimates. Again, we will round off our estimates to $300K to $400K.

Although the program development could cost less, we must project a likely cost of over $1M for the full program. In today's funding environment, we would not expect to see initiation of the complete program. Thus, we considered a less expensive alternative.

An Austerel Development Program

Software development would not only be possible, but more efficient, if DoD were to proceed at a deliberate pace. An austerel, but useful effort could involve development of the keyword glossary editor, data editor, and translators to support the WAAM and—to a limited degree—STRATCAM. We estimate that this austerel effort could be accomplished for $300K to $400K. If this effort were merged with the database conversion, the total cost should be less than the sum of the two; for example, a savings of $100K on analysis, management, and reporting is possible and would result in a cost for the package of from $500K to $700K.

Much of this cost could be offset by savings in the strategic C3 analysis community's database development efforts. We also visualize incremental and diffused funding over several years and from different organizations.
6. PROTOTYPE COMMON-FORMAT DATABASE RECORDS FOR PROCEDURES

Now that we have described the SSCF Database and the prototype common-format database design issues, we will develop about 150 example database records that will include the various procedures and operational data that we have identified. These database records are intended to demonstrate that all these classes of data can be readily translated into easily comprehended common-format database records. In this example, we will include equipment and procedures that would be typical of those

- At a fixed command center at the top of the hierarchy
- At an airborne command post
- At a fixed relay station
- For command-center succession.

We will develop examples of actions and functional elements converted to the format of the SSCF Database. These example database records are intended to demonstrate only the feasibility of representing this information in this format. The specific values represent only functional data and are intentionally not representative of "real world" data.

Finally, we made up a test case based on a similar example, which we ran as a demonstration in SIMSTAR 85A. This demonstration will be described in Sec. 7.

NETWORK AND SCENARIO

In the example case that follows, we will assume the following basic sequence of events:

1. Warning number 1
2. Change of defense condition (DEFCON)
3. Warning number 2
4. Decision
5. Emergency action message (EAM)

This simple sequence of events is intended to simulate a preliminary warning that causes a change of DEFCON, followed by a second warning that results in a decision to respond and the dissemination of an EAM. This sequence of events is constructed to require a wide range of database elements in all four categories (explicit actions, implicit actions, timing, and probability) that can be represented by common-format database records. We would not call this sequence of events a true scenario, in that we are not specifying a laydown, postulating timing, calculating probabilities of node damage, etc. The general sequence of events is a logical one that could lead to the actions and operational data that are converted to common-format database records.
A similar test case, involving some classified data, was run using the SIMSTAR-85A discrete-event simulation.

Figure 6.1 shows a network involving two sensor sites, a fixed command center, an airborne command post (which acts as an alternate command center), three simple fixed relay nodes, and a single force element. Underneath the name of each node is the mnemonic (e.g., “s1” for sensor 1, “cc1” for Command Center 1). These mnemonics are used as node identifiers in the database record tables. Figures 6.1b, c, and d indicate the processes that would go on in Command Center 1, Command Center 2, and the relay stations, respectively.

Table 6.1 includes some of the more important definitions that are required to interpret the tables that follow. These definitions are preliminary, but it is important to include them because they are an important part of the prototype database product.

**Table 6.1**

<table>
<thead>
<tr>
<th><strong>Keyword</strong></th>
<th><strong>Definition</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>MONITOR</td>
<td>To maintain a listening watch and/or to establish communications on a particular frequency and usually with a particular source.</td>
</tr>
<tr>
<td>SOURCE</td>
<td>The node in the network from which direct communication is expected. A source is not the same as an originator, which is the “ultimate source.”</td>
</tr>
<tr>
<td>PRIORITY</td>
<td>A hierarchical sequence in order of decreasing importance for a series of actions, i.e., monitoring and transmitting. There can be a “tie”; e.g., one could have several priority-three message sources.</td>
</tr>
<tr>
<td>FREQUENCY</td>
<td>The radio frequency on which a particular communication is to take place.</td>
</tr>
<tr>
<td>TIMING</td>
<td>The mathematical expression of the likely time required to perform a specific action or function. Timing data would include the timing distribution (normal, log normal, uniform) and the specific parameters (expected value, standard deviation, minimum, maximum) that quantify this timing distribution.</td>
</tr>
<tr>
<td>PROBABILITY</td>
<td>The mathematical or engineering data required to specify the likely probability of occurrence of an action or event. Like timing, this could include a distribution and the parameters required to quantify that distribution, but it will normally be a scalar, i.e., percent.</td>
</tr>
<tr>
<td>RECEIVE AND VERIFY</td>
<td>A series of actions required to ensure that a correct message has been received at a particular location.</td>
</tr>
<tr>
<td>IN</td>
<td>The input logic function, i.e., the identification of prior actions that require successful completion to allow the action specified at the beginning of a particular record.</td>
</tr>
<tr>
<td>OUT</td>
<td>The output logic function, i.e., the identification of the actions that require this particular action as an input logic function.</td>
</tr>
<tr>
<td>INHIBIT</td>
<td>Essentially a negative output logic function, i.e., it prevents particular subsequent actions.</td>
</tr>
<tr>
<td>AUTHENTICATE</td>
<td>To determine that a correct message passes a specific check, which is in turn designed to ascertain that the message comes from a valid and authorized source.</td>
</tr>
</tbody>
</table>
a. Network

From Sensors 1, 2
Landline ("wire") → rcvmodem

From Sensors 1&2, Command Center 1
222.4, 218.7 MHz → sat rcv 75
216.2 MHz

to Sensor 1 → sat xmt 75

&sat xmt 75 → 213.8, 212.6 MHz

hfxmt 29 → 29.51, 25.47 MHz

b. Command Center 1 (Fixed)

222.4, 218.7 MHz
From Sensor 1, 2 → sat rcv 77

213.8, 216.2 MHz
From Command Center #1 → sat rcv 77

hfxmt 29 → To Relay 1, 2, 3;
Force 1
27.49, 23.45 MHz
c. Command Center 2 (Airborne)

From Command Center 1
Landline ("wire") → rcvmodem
txmodem → Landline ("wire")

From Command Centers 1 and 2
213.8, 212.6, 212.0 MHz → satrcv80

satxmt75 → 209.4, 208.7 MHz

hfxmt29 → 29.92 MHz

From Command Centers 1 and 2 → hfrcvr47
29.51, 27.49, 25.47, 23.45 MHz
d. Relay Station 1 (Fixed)

Fig 6.1—Example Network and Key Nodes
FIXED COMMAND CENTER

Figure 6.2 is a listing of about seventy Database records that represent the explicit actions, implicit actions, timing, and probability data relevant to a command center. Even someone who is already intimately familiar with EAM dissemination and the WWMCCS would have difficulty interpreting Fig. 6.2. Therefore, we will walk through these database records to explain, on a line-by-line basis, what the Database records mean.

The first 13 records in Fig. 6.2 deal with monitoring and receiving—the procedural elements associated with the initial warning of an attack on the United States, which is Event Number 1 in our miniscenario. Monitoring is an action whereby a message can be received; communications personnel maintain a listening watch on specified frequencies. In Record 2, cmdctr1 is a mnemonic for Command Center 1, which is the node name of the command center at the top of the hierarchy of our small network. Personnel at cmdctr1 monitor 222.4 MHz using receiver class satrvu75. They maintain a listening watch for communications from Sensor 1. Monitoring this frequency provides the output allow_receive_satrvu75, which is input to Record 8.

In Record 3, personnel at cmdctr1, using a different receiver (but one of the same class) monitor 218.7 MHz with similar results. These communications personnel also monitor landline communications from Sensors 1 and 2 (Records 4 and 5). Record 6 indicates that it is assumed that there are no timing delays or reliability degradations to be modeled. Record 7 takes us to the receive function. The monitoring function in Record 2 produced two outputs—allow_receive_satrvu75 and allow_receive_rcvmodem—which then show up as inputs to Records 8 through 11.

Records 8 through 12 parallel Records 2 through 6. The outputs of Records 8 through 11 represent actions associated with the second warning of the particular attack, a warning that produces or allows a change in DEFCON. Record 12 indicates that there are no procedural delays or reliability problems. This comment record is actually superfluous, since a comment is not normally provided for standard defaults, but is shown here for clarity. It should also be pointed out that the time and probability information that would be entered here do not come from the link or network performance calculations. Rather, they represent strictly procedural delays and reliabilities.

As indicated in Records 14 through 18, it is assumed in this example that the various functions, decisions, and actions associated with changing alert status (e.g., deploying aircraft) are not modeled in most simulations. Most analyses assume that the resources in the network all automatically function at the same, appropriate, DEFCON level. The data records for Command Center 2 (Fig. 6) will show database records for deployment, however.

Records 19 through 37 treat the procedural elements (monitor, receive, information processing, and staffing and decision support) associated with the reception of a second warning message. Records 20 through 29 are virtually identical to Records 2 through 11. Record 30 indicates that this second warning causes decisions and actions at Command Center 1 that lead to a response requiring EAM dissemination through the network. These EAM dissemination tasks will be represented in data Records 64 through 67 below.

The first part of this process, however, is the information processing by the staff at Command Center 1 (Record 37). Record 32 represents the process incorporate_warn2, which can be
manual and/or automatic, of incorporating the output from Sensor 1 and/or Sensor 2–warn2
in Records 26 through 29. This information will be put into a format, formatwarn2, that can
be used by display and printing devices at Command Center 1. Thus, the output
formatwarn2 of Record 32 becomes the input to Records 33 and 34, where information is
displayed and printed for use by the Command Center 1 staff. In Procedural Element 3.2,
staffing/decision support (Record 35), the staff interprets and assesses this information
(Record 36). The staff uses electronic and written displays ctwarn2 and printwarn2 in
determining whether the warning information warrants a formal report to the commander.
The output of Record 36 is a report, staffwarn2, which is input to Record 37, the process of
reporting events and the situation to the commander on duty at Command Center 1.

Note that Records 26 through 29 each show timing and probability values as hypothetical
examples. Records 26 through 29 are assumed to be relatively mechanical processes involving
essentially uniform delays of a minimum of 1 minute and a maximum of 2 minutes with a
high (99 percent) probability of success.

Records 32 through 36 involve much more in the way of variability in human performance, so
the hypothetical distribution is assumed to be log normal. The 0.7 represents the natural
logarithm of approximately 2 minutes and the 0.3 represents the natural logarithm of the
standard deviation. This would represent a log normal distribution ranging from approxi-
mately 0.8 minutes (minus 3 σ) to 4.9 minutes (plus 3 σ). It should be emphasized here,
however, that none of these times should be considered real. Again, the frequencies, timing,
and probability data, as well as the transmitter and receiver classes, are purely and com-
pletely fictional. Our sole purpose is to demonstrate a human- and machine-readable format
for representing these kinds of information.

Records 38 through 63 represent the decision process. Records 39 and 40 represent requests
by the commander for information of two types:

• An estimate of the situation by his own staff (Record 39)
• A request for more information from the Sensor 1 facility.

In Record 39, we see a new keyword, REQUEST. In this case, the commander has requested
a staff input, a staff estimate. Input to both Record 39 and Record 40 is reportwarn2 from
Record 37. The outputs are req_est and req_info.

Records 26 through 37 also apply to Records 39 through 63.

Records 42 and 43 represent the preparation of the information request, queriesensor1, and
the transmission of that request, which will be sent out via ultrahigh frequency (UHF)
satellite communications (SATCOM) and landline. Note the keyword PREPARE in Record
42. Record 43 is the first of the Joint Staff common-format database records to require
several lines of data. The first two lines represent a satellite communications transmission,
and the second two lines represent a landline communication of the same query to Sensor 1
(msg_req_info_uhfsat and msg_req_info_wire SATCOM and landline formats of the same
message). If both were clear voice, the formats would be identical, and there would only be
one output, qsl. The first line of Record 42 specifies that the message is transmitted form
Command Center 1’s UHF satx75 to satrcv89 receivers at Relay stations 1, 2, and 3.
next two lines show transmission via landline from the txmodem to each rxmodem at Relay Stations 1, 2, and 3.

Records 44 through 55 represent monitor, receive, information processing, and staffing and decision support procedural elements very similar to those represented by Records 19 through 37. The only significant difference is that, in Record 55, one of the outputs is a decision to request a teleconference with Command Center 2 (req_conf_cmdctr2). This shows up as the input to Record 57.

Record 57 simply specifies that Command Center 1 has a teleconference with Command Center 2. This teleconference is partially formatted (e.g., for exchange of data) and partially free form (i.e., voice conversation). The input is the request for the conference; the output is advice to the commander at Command Center 1 from the commander (and presumably his staff) at Command Center 2. Records 58 and 59 represent the staffing and decision support based on the reply from Sensor 1, the staff’s situation estimate, and the advice from Command Center 2. The outputs of Record 59—estimate_w2 and allow_decision_w2—are merged with the outputs of Record 57—advice_cmdr2 and allow_decision_w2—to become inputs to the decision (Records 60 and 61).

The major effect of this decision to respond in Record 61 is the forced execution implementation (Records 62 and 63), which produces a planning option selection.

Records 64 through 67 represent EAM preparation and transmission. The planning option selection option_w2 in Record 63 becomes input to Record 65, which produces an EAM text, eam_w2_text. This text is then transmitted via SATCOM (eam1_sat) and landline (eam1_land) to Relay Stations 1, 2, and 3 (Record 66). The procedure is repeated 30 minutes later (Record 67). Note also that, in Records 66 and 67, a backup transmission at 243.0 MHz is contemplated in the event that contact is not made on 213.8 MHz. These data records imply that simultaneous transmissions on 213.8 and 243.0 MHz are not allowed, which is the reason for the inhibit 243.0 and inhibit 213.8 entries.

We have noted that it is easy, when reviewing these database records, to get into the details of how the real network works and what the real procedures are. Please remember that these prototype database records are intended to show how the common-format database approach would work and to demonstrate the feasibility of this approach. The entries are deliberately fictional and, in many cases, do not include the richness of detail that would be required for the actual database. More detailed information on most of these items would be a part of the real database, such as the following:

- Error detection and correction, single-source and multisource piecing, and the authentication procedure at the time of EAM reception should be spelled out.
- Technical details for conferencing procedures (Records 56 and 57) would be required.
- Technical descriptions of timing, distributions, and perhaps probability data would be contained in the Database Description and User’s Guide.
AIRBORNE COMMAND POST

This section describes the database representations for the unique actions of Command Center 2, which is an airborne command post—an alternate to the fixed, vulnerable Command Center 1. As indicated in Fig. 6.2, Command Center 1 survives long enough to carry out the command function, including the decision to respond and the injection of the EAM into the network. Command Center 2 does survive and deploy but, with the survival of Command Center 1, acts only as a standby backup command center and as a communications relay.

Figure 6.3 shows database records that represent the backup command-center role and deployment before survival. Communications relay database records are not shown, since that function is described in detail in the next subsection.

Records 1 through 8 of Fig. 6.3 are virtually identical to the first eight database records in Fig. 6.2, since Command Center 2 is performing the same command function, albeit in a backup role.

For the changing of DEFCON and alert status, however, the database records for Command Center 2 are different. Command Center 2, which is an airborne command post, would take off, deploy a very low frequency/low frequency (VLF/LF) antenna and would activate a communications network (here modeled as a high frequency (HF) network). Record 11 indicates that Command Center 2 would take off on the basis of warning information WARN1. The timing statistics are represented by a log normal distribution with the parameters shown. The probability of accomplishing this function is shown as 99 percent. Once the aircraft has taken off, this would allow, after some time interval, the deployment of a VLF/LF trailing-wire antenna. After Command Center 2 arrives at altitude, it activates an HF network. Records 12, 13 and 14 could be expanded if more detailed modeling of the takeoff, antenna deployment, and network activation were to be specified. For example, for network activation, one could show various calls and responses. These database records would be in the same format as other transmit, monitor, and receive database records, so no new or unique database record types would be required.

Although Command Center 2, operating as a backup command center, would probably go through many of the same processes as Command Center 1 (short of decisionmaking and EAM injection into the communications network), the database records in Fig. 6.3 indicate only the support to Command Center 1 implied by the actions previously shown in Fig. 6.2. Records 15 through 29 indicate the monitoring, receiving, information processing, and internal staffing efforts that the backup command center would normally be expected to perform in preparation to assume command, should the need arise.

In Records 30 through 37, Command Center 2 continues to perform these backup functions, in this case, receiving and assimilating the message traffic associated with Command Center 1's request to Sensor 1 for more information, and Sensor 1's response.

In Records 38 through 50, the only activity required at Command Center 2 is participation in the conferencing activity with Command Center 1 (Record 44). No additional directly related activity for decision support, decisionmaking, or implementation is relevant for Command Center 2, since Command Center 1 survives to carry out these functions.
Fig. 6.3—Database Records for Command Center 2
As indicated in Records 51 through 55, Command Center 2 performs communications relay functions. Note that the comments "no directly related activity" represented by Record 65 and others are not strictly required. The default rule of "no database records—no activity" would obviously apply, but comment records are shown here for tutorial purposes.

**FIXED RELAY STATION**

This task for a simple fixed relay station (one of three) are reception via UHF and VLF and transmission via HF. No other functions take place at these nodes. Figure 6.4 lists the data records for Relay 1 ("r1").

Although Relay 1 would almost certainly be aware of prior events, the EAM dissemination is the first that would directly involve Relay 1 in an operational sense. Thus, this is the first database entry in Fig. 6.4.

In Records 1 through 7, Relay 1 monitors SATCOM and HF communications from both Command Center 1 and Command Center 2 using a single SATCOM receiver and a single HF receiver. For this example, Relay 1 monitors the UHF satellite frequencies in the priorities shown. Relay 1 also monitors four HF frequencies, but with only three priority classes. The monitoring operation allows Relay 1 to receive over the specified links. Records 8 through 14 are the "receive" records. The differences in timing and probability for priorities 3A and 3B, as indicated in Records 13 and 14, are based on an assumption that Relay 1 would implicitly and of necessity effectively assign one frequency a higher priority than the other, simply because the two frequencies could not be tuned in at the same time.

In Record 15, when EAM signal received is received at Relay 1, error detection and correction (EDAC) procedures are followed. The REF_DATA at the end of the record would point to algorithm and timing information, since the time required for EDAC would be a function of the EDAC technique. The information can show up as database records, comment cards, and/or detailed information in the Database Description and User's Guide.

In Record 17, the correct signal is then authenticated. The same process is carried out for the LF receptions.

If Relay 1 required a fairly complex translation process from one format to another, the entries would reflect this additional process with another entry.

The final record shows transmission of the authentic and correct message with the timing and probability shown.

**COMMAND CENTER SUCCESSION**

Figure 6.5 lists procedures and functional elements associated with command center succession. This case falls in the system procedures category. It primarily involves what is often an unstated set of assumptions. The processes described here are for the most part not covered in the standard documentation. If one does not model intra-command center communications, or if one assumes perfect connectivity there, no data records would be required. Many analysts do not model intra-National Military Command System communications for
Fig. 6.4—Data Records for Relay 1
**Activation/Deployment**

- Node (e.g., WWABNCP takeoff)
- Timing data (class, parameter #1, parameter #2)
- Probability data (percent)
- Input logic function
- Output logic function

**NODE cc2 DEPLOY takeoff TIMING lognormal 2.9 5.1 &lp**
**NODE cc2 DEPLOY takeoff PROB 99 &lp**
**NODE cc2 OUT takeoff IN warning ALLOW antenna &lp**

**Equipment (e.g., WWABNCP TW Antenna deployment)**

- Timing data (class, parameter #1, parameter #2)
- Probability data (percent)
- Input logic function
- Output logic function

**NODE cc2 DEPLOY antenna IN takeoff &lp**
**NODE cc2 DEPLOY antenna TIMING lognormal 2.3 0.2 &lp**
**NODE cc2 DEPLOY antenna PROB 96 &lp**

**Network (Establishment, restoration)**

- Timing data (class, parameter #1, parameter #2)
- Probability data (percent)

**NODE cc2 NETWORK uhfpax IN takeoff OUT allow monitor**
  **TIMING normal 6.6 2.2 PROB 99 &lp**

**Monitor (for each potential link):**

- Priority
- Source
- System/frequency/mode/polarity
- Timing data (class, parameter #1, parameter #2)
- Probability data (percent)
- Input logic function
- Output logic function

**NODE cc2 MONITOR uhfsat SOURCE neacp PRIORITY 1 FREQ 2430 &lp**
**NODE cc2 MONITOR uhfsat SOURCE neacp PRIORITY 2 FREQ 2299 &lp**
**NODE cc2 MONITOR uhfsat SOURCE neacp PRIORITY 3 FREQ 2516 &lp**
**NODE cc2 MONITOR uhfsat PRIORITY 3 TIMING uniform 6 12 &lp**
**NODE cc2 MONITOR uhfsat PRIORITY 3 PROB 90 &lp**

**NODE cc2 MONITOR vif1f SOURCE neacp PRIORITY one &lp**
**NODE cc2 MONITOR vif1f SOURCE neacp FREQ 29.51 &lp**
**NODE cc2 MONITOR fpax SOURCE cc2 FREQ 2430 &lp**

Fig. 6.5—Database Records for Command Center Succession
originator succession. Transfer of command would be assumed by having a successor node originate communications, but the process of transference of command would not itself be simulated. Typically, then, there would be no data records except for activation and deployment. To specify the delays involved, a typical record might appear as follows:

```
NODE CMDCTR2 ASSUMPTION_FROM cc1 TIMING uniform 1 2 PROB 100 & lp
```
The Command Center 2 decisionmaking process or communications with higher authority would then be invoked.

Ground rules and assumptions for command-center succession typically might include the following:

- Each command center is assumed to have perfect connectivity with any higher authority.
- Only one originator can exist at any specific time, and the originators have a specified precedence.
- The entire EAM transmission procedure must be completed by a designated EAM originator, or else the EAM is not considered to have been sent.
- A probability draw determines the survivability as a function of time of each potential originator or command center.
- If the designated originator is destroyed prior to completion of the EAM dissemination procedure, the next potential originator will, if it has survived, become the designated EAM originator. It will begin EAM dissemination with some delay.

**OTHER STRATEGIC DATABASE ELEMENTS**

A number of miscellaneous operational and technical elements of the SSCF Database can also be converted to common format. Fig. 6.6 lists a few of the more important types of data that are characteristics of nodes or facilities. Records 1 through 3 provide a full node identifier and categorization in terms of name, “identity tags,” function, and type.

Since the example in Fig. 6.6 is a mobile platform, Records 4 through 11 specify the location (including altitude) of the platform as a function of time. For a fixed facility, obviously, only one record would be required.

Records 12, 13, and 14 specify the hardness of a particular facility, which can be expressed in terms of (1) the static overpressure or dynamic pressure in pounds per square inch (psi) at which damage is expected or (2) the vulnerability number (vntk). The database also identifies whether a severe, moderate, or light damage criterion prevails for the facility.

Records 15 through 18 specify the fragility of the node or equipment at a node to an electromagnetic pulse (EMP). The fragility is expressed in probability of damage in percent followed by the field strength that produces that probability of damage. More than one damage–and–field strength pair can be presented. This is followed by a series of times at which the EMP field strength reaches particular peak values.

Records 19 through 21 provide information concerning the reliability, availability, and maintainability of nodes and equipment.
* Fig. 6.6—Miscellaneous Operational and Technical Node Data *
7. DEMONSTRATING THE COMMON-FORMAT DATABASE

To demonstrate the keyword representation format for command and control communications (C³) systems, we created a small-scale analysis problem and corresponding database. The sections below describe the objectives the demonstration was intended to satisfy, define the test C³ system, describe the representation of the test C³ problem in the common format, and describe the process of translating the common-format C³ system description into an input file for the SIMSTAR model.

DEMONSTRATION OBJECTIVES

The demonstration task was intended to explore the degree to which the proposed format encoded equipment and procedural aspects of C³ and the ease with which the format could be read by both programs and operators. It was especially important to demonstrate the adequacy of the representation with respect to procedures. Procedural description of C³ systems is currently the least standardized and exhibits the great diversity of both model formats and concepts. With most of the models, restrictions on procedure representation have limited the C³ system features that could be treated without introducing large approximations.

To create a universal, strategic C³ database, it is essential that the new format completely encode the C³ system features that the C³ analysis models reference. Hence, demonstrating that communication procedures were completely representable in the common format was an important goal. Here, the definition of complete is operational. A complete procedure representation is one whose statements can be provably derived from the specification documents and that can serve as the sole information base for the synthesis of model inputs for the supported models. In this sense, a complete representation constitutes a superset of the data the supported models use. Of course, completeness of representation cannot be achieved at the expense of either machine or human readability. For instance, the creation of separate data files in the formats each of the supported models use would satisfy the "completeness" requirement but would not be very human readable and would only be partially machine readable. Such a union of databases is unsatisfactory, because a single system description fact is represented in multiple, dependent ways. Such dependence between data entities introduces ambiguities that require complex interpretation rules and algorithms. The keyword representation format is intended to provide a system description method that is complete but that facilitates human and machine readability through data independence and natural-language-like semantics and syntax.

It was important to show that the demonstration database exhibited both human intelligibility and machine readability. The requirement for human and machine readability derives from the two-fold need to facilitate verification of database correctness and translation of the database content into the formats used by the supported models. A database that was neither verifiable nor translatable could not serve as a basis for a description of a standardized strategic C³ system. Here, the definition of translatable is operational as well. A translat-

---

¹These demonstrations were conducted at RAND on Sun workstations using the SUN UNIX Operating System.
able representation is one that allows translation under a consistent set of software implementable rules. Such machine translatability was an important feature to demonstrate, since a source of discrepancies in the earlier analytic studies had come directly from the differing interpretations that entered at every phase of setting up the C³ problems for simulation using different models. Not only could different groups use different assumptions in their C³ modeling, but different analysts within the same group could introduce additional variations. Under these circumstances, interpretations and approximations could be pyramided, one on top of another, until a point could be reached where even very large discrepancies in the results could not be resolved with certainty. Also, under the prevailing standards for configuration control, it was often difficult to trace the reasoning behind assumptions that later could prove to have a crucial impact on simulated system performance. Translation under software control would eliminate a significant source of analytic variance. Thus, to be of value, the SSCF Database concept needed to meet the triple requirements of completeness, human readability, and machine translatability as well. Here, machine translatability was assumed to require a statement grammar and syntax that were uniquely parsable. Maintainability has been taken to imply the ability of operators to read representation statements directly and compare their contents with the equivalent natural language statements directly in the specification documents. The demonstration experiment was structured to meet these needs.

To demonstrate completeness of representation, a small but highly structured C³ system was first defined and then encoded into the common format. Although the number of nodes was small, a wide variety of equipment and procedures provided ample opportunity to illustrate and try out the new formats. This reflected the philosophy that the number of records would easily scale up in going from a small to a large system, while the same claim would be false when going from representing a simple system to representing a complex one. The selected test was designed to be small but operationally complex. The variety of C³ features in the test case, in terms of equipment and operations, provided ample scope to demonstrate the descriptive powers of the proposed common format and also allowed us to test the degree to which a human reader could understand what was being described in the formatted statements.

**TEST CASE C³ SYSTEM**

The Joint Staff, DCA, AFCSA, and RAND representatives agreed that the new format should be tested against the full strategic C³ network. Although this plan was later modified for reasons of data unavailability, the goal of modeling the full range of strategic C³ activities was retained. Focusing on the representation of procedures led to an emphasis on command centers, which have the greatest variety of procedures taking place inside them. The following elements span the range of systems and telecommunications procedures in the WWMCCS that needed to be represented in the common format.

1. National Emergency Airborne Command Post (NEACP)
2. LOOKING GLASS
3. High frequency (HF) network control station (LOOKING GLASS)
4. Relay node—JIM CREEK
5. Relay node—TACAMO LANT
6. A hypothetical command center

For security reasons, it was agreed not to use the actual classified system descriptions of current facilities and strategic command and control procedures. Instead, we based a representative demonstration C³ system on material in the system-specification documents. In satisfying the security requirements, we used the system specifications only to identify the types of actions that went on inside the large command centers. Although it contains no classified information, the model's built-in monitoring and message-handling requirements cover the range of activities contained in the actual system. With these caveats, the simple network in Fig. 7.1 represents the first five elements above. This network comprises nine nodes and eighteen links. All elements of this simple network—the equipment; the communication links (including frequencies); and the procedures, timing, and reliability data are deliberately fictitious. While familiar mnemonics connote command functions, we avoided representing the “real” WWMCCS elements in this test case. (The original intent was to use the actual classified data that describe these facilities in detail, but this information could not be made available for this prototype effort.)

![Diagram](image-url)
To define the test case's connectivity, we developed a wiring diagram and a standardized nomenclature with which to describe the records. The network wiring diagram is depicted in Fig. 7.1. The large communication nodes are labeled neacp, sac, and tacamo. In operation, the neacp node serves as the injection point for the message, which is then relayed first to the ancillary message-handling platforms, labeled sac and tacamo. These platforms then relay the message to the terminal nodes represented by the ICBM Wing and the SSBN nodes or to further relay nodes (HF RELAY nodes 1, 2, and 3). In the common format Database, the sending nodes perform transmit procedures, while the destination nodes execute monitor and receive procedures. The C³ procedure documents use a double-ended link description, describing message transport between nodes both from the transmitting and the receiving ends of the link. Some simulations use single-ended procedures focusing on either the transmitting or the receiving end. We therefore developed the formatted procedure descriptors TRANSMIT, MONITOR and RECEIVE to support such double-ended descriptions. The numbers 1 through 18 of Fig. 7.1 represent individual links or, if more than one frequency is used, a set of links. Each set, however, is limited to one part of the frequency spectrum. Thus "1HF" means, in this case, a pair of HF links, and "12LF" means a single LF link. This terminology is used in the discussion below.

While no classified node procedure data were used, the test-case command-center nodes were configured to have a variety of message handling, monitoring, receiving, and transmitting activities. Since the purpose of the test was to demonstrate the adequacy of the representation as to type, the test-case node would show one or two monitoring operations, while a real node would have an order of magnitude more channels in operation. This simplification was both desirable from a security point of view, to protect the operational flow in the real nodes, and allowable from the point of view of the test, which was to demonstrate that operations were generically representable rather than to represent an actual system. To make this fictitious case immediately comprehensible to experienced C³ personnel, the test-case nodes were given names that would suggest their role in the test case. Thus, the message-origination node functions were associated with the NEACP, even though the operations and equipment on the real NEACP are completely different from those postulated for the simulated "neacp" of the test case. The only similarity is that the real NEACP can serve as a message-injection node, and that is the role of the neacp of the test case. The main strategic message relay was associated with the Strategic Air Command Airborne Command Post (here, sac), and the airborne communication relay to one of the mobile destination nodes was called tacamo. These "functional" associations between the test case and the strategic C³ system are summarized in Table 7.1. The message handling steps and the process delay times are fictitious. Despite

### Table 7.1

<table>
<thead>
<tr>
<th>Name and Description</th>
<th>Mnemonic</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEACP</td>
<td>neacp</td>
</tr>
<tr>
<td>LOOKING CLASS, SAC airborne</td>
<td>sac</td>
</tr>
<tr>
<td>JIM CREEK VLF/LF relay station</td>
<td>jicork</td>
</tr>
<tr>
<td>Three HF relay stations</td>
<td>hfrly1,2,3</td>
</tr>
<tr>
<td>TACAMO VLF/LF relay station</td>
<td>tacamo</td>
</tr>
<tr>
<td>ICBM Wing at Grand Forks</td>
<td>godiks</td>
</tr>
<tr>
<td>SSBN location number 1</td>
<td>ssbn1</td>
</tr>
</tbody>
</table>
the fictional nature of the data, the representation of the small-scale C³ system differs from that of the full strategic C³ system only in the numbers of records required and the precise data values that appear in the records.

Several types of equipment were postulated as supporting the links in the test-case system. Three frequency bands (VLF, LF, and HF) were included in the network. Within each band, multiple frequencies were used. This is an important representation area, because link performance under degraded propagation conditions is a critical issue for the strategic C³ analyst. It was important to demonstrate that the link budget parameters of both transmitters and receivers were represented adequately to compute the effective signal-to-noise ratios and, hence, error rates that would apply over the links. Once again, the dozens of equipment types found in the real strategic C³ system were pared down to a small number, because the demonstration issues dealt with quality of representation, rather than quantity of representation. The test case included four generic transmitters, one generic transceiver, and four generic receivers, as summarized in Table 7.2. The table lists the equipment name, the frequency bands, the frequencies monitored by the equipment in the test case, and the installation nodes. The transmitter names are referred to in the following discussion. The names given the equipment classes stand for the real types of equipment that would be installed in the various facilities, which would bear more concrete names in practice, such as ARC/72.

To complete the description of the demonstration C³ system, representative internal process flows were developed for each of the communication nodes. The intranode processes were described using flow network diagrams and specifications of distribution type and parame-

<table>
<thead>
<tr>
<th>Equipment</th>
<th>Frequency Band</th>
<th>Frequencies Used (MHz)</th>
<th>User Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmitters</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>txhf1a</td>
<td>HF</td>
<td>3.1 MHz</td>
<td>necsp, snc</td>
</tr>
<tr>
<td>txhf2a</td>
<td>HF</td>
<td>3.2 MHz</td>
<td>necsp, snc</td>
</tr>
<tr>
<td>txhf3a</td>
<td>HF</td>
<td>3.3 MHz</td>
<td>necsp, snc</td>
</tr>
<tr>
<td>txlf1b</td>
<td>LF</td>
<td>41.9 kHz</td>
<td>snc</td>
</tr>
<tr>
<td>txvlf1d</td>
<td>VLF</td>
<td>23.0 kHz</td>
<td>tacoma</td>
</tr>
<tr>
<td>txlv1m</td>
<td>LF</td>
<td>41.1 kHz</td>
<td>jmcrc</td>
</tr>
<tr>
<td>Transceivers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>trh</td>
<td>HF</td>
<td>Transmit: 3.4 MHz, 3.5 MHz, 3.6 MHz</td>
<td></td>
</tr>
<tr>
<td>Receivers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rxhf1a</td>
<td>HF</td>
<td>3.1 MHz</td>
<td>necsp, snc, tacoma, asbn1</td>
</tr>
<tr>
<td>rxhf2a</td>
<td>HF</td>
<td>3.2 MHz</td>
<td>necsp, snc, tacoma, asbn1</td>
</tr>
<tr>
<td>rxhf3a</td>
<td>HF</td>
<td>3.3 MHz</td>
<td>necsp, snc, tacoma, asbn1</td>
</tr>
<tr>
<td>rxlp1b</td>
<td>LF</td>
<td>41.1 kHz</td>
<td>tacoma, gnis</td>
</tr>
<tr>
<td>rxvlf1d</td>
<td>LF</td>
<td>41.1 kHz</td>
<td>asbn1</td>
</tr>
<tr>
<td>rxlv1m</td>
<td>VLF</td>
<td>23.0 kHz</td>
<td>asbn1</td>
</tr>
</tbody>
</table>
ters for the time delays caused by each operation. The flows and distribution parameters were fictitious in the test case, but, in a real system, this information would encode the operation sequences described in the operational orders and the delays measured during test and exercise of the system. Operational functions include reception of communications, processing of information, and transmission of communications. Operational data also include timing and probability information. Table 7.3 lists a generic set of processes with the associated timing and probability data for two receive, two reformat, and two transmit processes. Intranode descriptions in this format appear in the following sequence for each node in the test case.

In Table 7.3, timing data are presented in the format X(n,n), where if X is constant (K), (n) is the time in minutes; if X is uniform (U), (n,n) represents the maximum and minimum times in minutes; and if X is log normal (L), (n,n) represents the alpha (mean of the natural log of the delay time) and the beta (standard deviation of the natural log of the delay time). The default is zero delay (dashes). The probability of success is a scalar in percent, with a default of 100 percent.

Thus, receiving process R1 is assumed to take a minimum of one minute and a maximum of two minutes, with a uniform distribution and a probability of successful occurrence of 99.9 percent. The second receive process takes exactly twice as long and has a probability of successful completion of 99 percent. The first reformatting operation is assumed to have a log-normal timing distribution. Log normal distributions are typical of operations involving human decision. The sample figures shown for reformatting process P1 represent a situation in which the process would usually take on the order of 1.65 minutes, but there is about a 5 percent probability that the operation could take as little as 1.35 minutes or as much as 2.0 minutes. Reformatting process P2 is assumed to take zero time and will occur with 100 percent probability since the entries are defaults.

Figure 7.2 shows the equipment processes, frequencies, timing, and probability data for the NEACP node. Since NEACP is the originator, there are no monitor or receive processes. The two processes that take place, ncpp1 and ncpp2, are assumed to occur with a 100 percent probability with the log-normal and uniform timing distributions shown. The log-normal distribution here implies a 5 percent probability of accomplishment in less than 1 minute, an average delay of about 2 minutes, and a 5 percent probability of taking as long as 3.7 minutes. The message goes out over two transmitters of the same class. In the common-format database, this would be equipment class txhfx. Since SIMSTAR 85-A requires separate equipment classes for each frequency, the equipment classes are shown here as TXHFA1 and

<table>
<thead>
<tr>
<th>Process Class</th>
<th>Process Name</th>
<th>Timing</th>
<th>Probability of Success (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receive</td>
<td>R1</td>
<td>U(1, 2)</td>
<td>99.9</td>
</tr>
<tr>
<td>Receive</td>
<td>R2</td>
<td>U(2, 4)</td>
<td>99</td>
</tr>
<tr>
<td>Reformat</td>
<td>P1</td>
<td>L(0.5, 0.1)</td>
<td>90</td>
</tr>
<tr>
<td>Reformat</td>
<td>P2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transmit</td>
<td>T1</td>
<td>U(1, 2)</td>
<td></td>
</tr>
<tr>
<td>Transmit</td>
<td>T2</td>
<td>K(0, 3)</td>
<td>95</td>
</tr>
</tbody>
</table>
The receiver names, which are completely internal to the model and are chosen by the modeler, are `ncpt1` and `ncpt2`. Note that Links 1 through 4 each include both 3.1 MHz and 3.2 MHz frequencies; thus, each could be considered two separate links in parallel.

Figures 7.3 through 7.8 show the processes, equipment, timing, and probability data for the remaining nodes in the example network.

TEST-CASE REPRESENTATION IN COMMON FORMAT

Following the specification of the test-case C³ system in connectivity charts, equipment specifications, intranode process flows, and delay distribution tables, we developed the equivalent representation in common-format records. This translation was accomplished in three parts. The first part was the physical description, which describes the locations of the communication relay nodes. In the common format, the communication equipment that is physically present at the node is specified as part of these database records. The second part was the RF characterization of the link equipment using keyword formats that are adequate to specify the link budget parameters that control link availability and throughput. The third part was encoding of the message handling procedures, as well as the “delay” data for the message handling processes. The Database’s organization is like the seven-part protocol model used in the International Standards Organization standards for describing Open System Interconnect (OSI) standards for computer communications. The common format, in essence, has three parts: part 1 is the physical description of network assets (node locations, transmitters, and receivers); Part 2 describes the link RF parameters for transmitters and receivers; and Part 3 describes the network procedures (transmit, monitor, and receive actions and message delay processes). The network description in Part 3 corresponds to the upper five parts of the OSI protocols, i.e., the transmission, network, session, presentation, and application parts.
Figure 7.3—Test-Case Data for the SACHF (HF Net Control) Node

Part 1 records of the common-format database describe the locations of relay nodes and the transmitters and receivers that are physically present at the nodes. In essence, Part 1 describes the physical layout of the communication system. From the node data, the location of the communication centers is fixed by geographic coordinates for fixed nodes, by trajectories for moving nodes, and by orbital elements for satellites. The node data describe the equipment present at each site or platform. In addition, communication node availability parameters describe the ability of the node to support communications as a function of time. This means that the node data must include reliability and survivability information. In the test case, efforts were made to insure that all required node information types appeared. The test case node description records are described and appear below in Fig. 7.9.

In Part 1 of the database for the test case, the record format for the node locations includes five information fields, each preceded by a keyword that determines both the semantic meaning and the physical units. This system promotes readability. For example, the node location data records for the gdks node are as follows:
Fig 7.4—Test Case Data for HF Relays 1, 2, and 3

Fig. 7.5—Test-Case Data for the Jim Creek Node
Fig 7.6—Test-Case Data for the TACAMO Node
Fig 7.7—Test-Case Data for the Grand Forks ICBM Wing Node

Fig 7.8—Test-Case Data for the SSBN Atlantic Ocean Node
Fig. 7.9—Database Records for Nodes: Part 1 of the Test-Case Database
<table>
<thead>
<tr>
<th>NODE</th>
<th>TYPE</th>
<th>PLATFORM</th>
<th>FORM: LAT(deg), LON(deg), ALT(km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>h/relay3</td>
<td>fixed facility</td>
<td>relay</td>
<td>COMMENT: TIME_MIN = 35.00, LON = -120.00, ALT = 0.00</td>
</tr>
<tr>
<td>h/relay3</td>
<td>WACID is UNKNOWN</td>
<td>CATEGORY CODE is UNKNOWN</td>
<td>COMMENT: PROB OF DAMAGE</td>
</tr>
<tr>
<td>h/relay3</td>
<td>500 psi</td>
<td>HARDNESS</td>
<td>COMMENT: PROB OF DAMAGE</td>
</tr>
<tr>
<td>h/relay3</td>
<td>SEVERE = 32Q10</td>
<td>MODERATE = 30Q8</td>
<td>LIGHT = 26Q10</td>
</tr>
<tr>
<td>h/relay3</td>
<td>DAMAGE = 22Q7</td>
<td>COMMENT: DAMAGE = 22Q7</td>
<td></td>
</tr>
<tr>
<td>h/relay3</td>
<td>WILL NOT BE RESTORED IF DESTROYED</td>
<td>COMMENT: DAMAGE = 22Q7</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NODE</th>
<th>TYPE</th>
<th>PLATFORM</th>
<th>FORM: LAT(deg), LON(deg), ALT(km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>j/relay3</td>
<td>fixed facility</td>
<td>relay</td>
<td>COMMENT: TIME_MIN = 47.00, LON = -122.00, ALT = 0.00</td>
</tr>
<tr>
<td>j/relay3</td>
<td>WACID is UNKNOWN</td>
<td>CATEGORY CODE is UNKNOWN</td>
<td>COMMENT: PROB OF DAMAGE</td>
</tr>
<tr>
<td>j/relay3</td>
<td>300 psi</td>
<td>HARDNESS</td>
<td>COMMENT: PROB OF DAMAGE</td>
</tr>
<tr>
<td>j/relay3</td>
<td>SEVERE = 32Q10</td>
<td>MODERATE = 30Q8</td>
<td>LIGHT = 26Q10</td>
</tr>
<tr>
<td>j/relay3</td>
<td>DAMAGE = 22Q7</td>
<td>COMMENT: DAMAGE = 22Q7</td>
<td></td>
</tr>
<tr>
<td>j/relay3</td>
<td>WILL NOT BE RESTORED IF DESTROYED</td>
<td>COMMENT: DAMAGE = 22Q7</td>
<td></td>
</tr>
<tr>
<td>j/relay3</td>
<td>RECEIVER DATA</td>
<td>DATA</td>
<td>COMMENT: DATA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NODE</th>
<th>TYPE</th>
<th>PLATFORM</th>
<th>FORM: LAT(deg), LON(deg), ALT(km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>n/relay3</td>
<td>aircraft</td>
<td>relay</td>
<td>COMMENT: TIME_MIN = 50.00, LON = -70.00, ALT = 10.00</td>
</tr>
<tr>
<td>n/relay3</td>
<td>WACID is UNKNOWN</td>
<td>CATEGORY CODE is UNKNOWN</td>
<td>COMMENT: WACID is UNKNOWN</td>
</tr>
<tr>
<td>n/relay3</td>
<td>300 psi</td>
<td>HARDNESS</td>
<td>COMMENT: HARDNESS = 300 psi</td>
</tr>
<tr>
<td>n/relay3</td>
<td>SEVERE = 32Q10</td>
<td>MODERATE = 30Q8</td>
<td>LIGHT = 26Q10</td>
</tr>
<tr>
<td>n/relay3</td>
<td>DAMAGE = 22Q7</td>
<td>COMMENT: DAMAGE = 22Q7</td>
<td></td>
</tr>
<tr>
<td>n/relay3</td>
<td>WILL NOT BE RESTORED IF DESTROYED</td>
<td>COMMENT: DAMAGE = 22Q7</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NODE</th>
<th>TYPE</th>
<th>PLATFORM</th>
<th>FORM: LAT(deg), LON(deg), ALT(km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>n/relay4</td>
<td>aircraft</td>
<td>relay</td>
<td>COMMENT: TIME_MIN = 48.00, LON = -82.00, ALT = 10.00</td>
</tr>
<tr>
<td>n/relay4</td>
<td>WACID is UNKNOWN</td>
<td>CATEGORY CODE is UNKNOWN</td>
<td>COMMENT: WACID is UNKNOWN</td>
</tr>
</tbody>
</table>

**Fig. 7.9—continued**
NODE sac CATEGORY_CODE_IS UNKNOWN
NODE sac HARDNESS 300 psi
NODE sac SEVERE 32Q10 MODERATE 30Q8 LIGHT 20Q10
NODE sac DAMAGE 22Q7
NODE sac WILL_NOT_BE_RESTORED_IF_DESTROYED
COMMENT: NODE sac Receiver DATA
NODE sac RECEIVER SCRR1 RXHFA1 3.100
NODE sac RECEIVER SCRR2 RXHFA2 3.200
COMMENT: FORMAT NODE_NAME RECEIVER_NAME RCV_CLASS_NAME
COMMENT: FORMAT NODE_NAME XMTR_NAME XMTR_CLASS_NAME
NODE sac TRANSMITTER SHFT1 TXHFA3 3.300
NODE sac TRANSMITTER SLFT1 TXLF3 0.041

*------------------------------------------------------------------------

NODE sshn NODE_TYPE relay PLATFORM_TYPE submarinepositional_position
COMMENT: FORMAT IS LAT(deg), LON(deg), ALT(km)
NODE sshn TIME_MIN na LAT_DEC 38.00 LON_DEC -25.00 ALT_KM 0.00
NODE sshn WACID IS UNKNOWN
NODE sshn CATEGORY_CODE_IS UNKNOWN
COMMENT: NODE DAMAGE FORMAT sshn TIME (MIN) PROB OF DAMAGE
NODE sshn SSDP_MOD 0 TIME 0m 0s
NODE sshn HARDNESS 300 psi
NODE sshn SEVERE 32Q10 MODERATE 30Q8 LIGHT 20Q10
NODE sshn DAMAGE 22Q7
NODE sshn WILL_NOT_BE_RESTORED_IF_DESTROYED
COMMENT: NODE sshn Receiver DATA
COMMENT: FORMAT NODE_NAME RECEIVER_NAME RCV_CLASS_NAME
NODE sshn RECEIVER SL1R1 RXHFA1 3.100
NODE sshn RECEIVER SL1R2 RXHFA2 3.200
NODE sshn RECEIVER SL1R3 RXHFA4 3.600
NODE sshn RECEIVER SL1R4 RXLF1 0.041
NODE sshn RECEIVER SL1R5 RXLF3 0.041
NODE sshn RECEIVER SL1R6 RXVS 0.023

*------------------------------------------------------------------------

NODE tac NODE_TYPE relay PLATFORM_TYPE aircraft
COMMENT: FORMAT IS LAT(deg), LON(deg), ALT(km)
NODE tac TIME_MIN na LAT_DEC 30.00 LON_DEC -70.00 ALT_KM 6.20
NODE tac WACID IS UNKNOWN
NODE tac CATEGORY_CODE_IS UNKNOWN
COMMENT: NODE DAMAGE FORMAT tac TIME (MIN) PROB OF DAMAGE
NODE tac SSDP_MOD 0 TIME 0m 0s
NODE tac HARDNESS 300 psi
NODE tac SEVERE 32Q10 MODERATE 30Q8 LIGHT 20Q10
NODE tac DAMAGE 22Q7
NODE tac WILL_NOT_BE_RESTORED_IF_Destroyed
COMMENT: NODE tac Receiver DATA
COMMENT: FORMAT NODE_NAME RECEIVER_NAME RCV_CLASS_NAME
NODE tac RECEIVER tacr1 RXHFA1 3.100
NODE tac RECEIVER tacr2 RXHFA2 3.200
NODE tac RECEIVER tacr3 RXHFA3 3.500
NODE tac RECEIVER tacr4 RXLC1 0.041
NODE tac RECEIVER tacr5 RXLC3 0.041
COMMENT: NODE tac XMTR DATA
NODE tac TRANSMITTER tac(TJ) TXVAL 0.023
COMMENT: FORMAT NODE_NAME XMTR_NAME XMTR_CLASS_NAME

*------------------------------------------------------------------------

Fig. 7.9—continued
NODE gdfks TIME_MIN na LAT_DEG 47.10 LONG_DEG -97.90 ALT_KM 0.0

This record provides the node name, time in minutes (here, not applicable because the node is fixed), latitude in degrees, longitude in degrees, and altitude in kilometers. The TIME_MIN field is used to indicate when a mobile platform would be at the specified position. Multiple records would supply positions for various times, and positions at intermediate times would be linearly interpolated. Augmenting the records providing geographic coordinates are references to such well-known catalogs as the World Aeronautical Chart identification numbers (for which the keyword is WACID IS).

Node vulnerability data records contain the keywords HARDNESS, SEVERE, MODERATE, LIGHT, and DAMAGE. The field following HARDNESS gives the hardness of the facility in pounds of overpressure per square inch (psi). The other keywords are tied to the DNA vulnerability characterization system, which characterizes damage according to mechanism. If dynamic overpressure is dominant, it is a Q-type target (e.g., 20Q10), and if direct overpressure is dominant, it is a P-type (e.g., 99P0). The input data format can also specify assessed survivability. The keywords SSPD_MOD and TIME give the probability that a single weapon will create moderate damage to a target as a function of the weapon yield and delivery accuracy versus the hardness of the target as specified by the vulnerability data. Most models assume that moderate damage "kills" C3 targets in the operational sense.

The equipment data at the node are divided into transmitters, receivers, and processors. The keywords TRANSMITTER and RECEIVER each require three parameters (e.g., JCLFR, RXLFDI, 0.041). The first parameter uniquely identifies the specific transmitter or receiver installation; the second identifies the class of the equipment; and the third gives the frequency in megahertz. The second parameter leads to the equipment description data records to set such RF parameters as bandwidth, data rate, modulation type, and noise characteristics. Both data records enter into analysis of the communication system and its performance. The receiver and transmitter location data are inherited from the description records of the node where they are located, and the RF parameters are selected from the equipment descriptions in the second part of the description database.

The transmitter and receiver equipment parameters in Part 2 of the common-format database support the assessment of the connectivity and performance of the C3 system links. The supported parameters are exactly those needed to compute the link budget, the error rate, and the data rate and bandwidth parameters that control throughput. Figure 7.10 presents the common-format records describing the transmitters, while Fig. 7.11 presents the equivalent records for the receivers. For transmitters, it is essential to know the data rate to determine the message transmission delay and to know the radiated power, frequency and modulation to determine signal detectability. The receiver noise bandwidth, frequency, modulation, sensitivity, and antenna gain are needed to determine the effective error rate. Numerous types of transmitters and receivers were included in the test case to prove out the feasibility of the keyword representation for equipment types. The frequencies included representative VLF, LF, and HF systems. The keywords include the measurement units: BANDWIDTH(HZ), FREQUENCY(HZ), ANTENNA_GAIN(DATABASE), TIME(MIN), and PROB_DAMAGE. Modulation types were specified by keyword values that closely follow familiar abbreviations, such as NFSK, PLSO-1, PSK, and PHASES. Experience with this equipment description format indicated that the statements were highly readable, which
<table>
<thead>
<tr>
<th>Comment</th>
<th>Transmitter Class Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXHF1</td>
<td>TXHF1 Frequency (Hz): 0.310000e+07</td>
</tr>
<tr>
<td>TXHF1</td>
<td>TXHF1 Bandwidth (Hz): 0.150000e+03</td>
</tr>
<tr>
<td>TXHF1</td>
<td>TXHF1 Data Rate (bps): 0.750000e+02</td>
</tr>
<tr>
<td>TXHF1</td>
<td>TXHF1 Channel Utilization (%): 0.00</td>
</tr>
<tr>
<td>TXHF1</td>
<td>TXHF1 ERP in DBM: 0.400000e+02</td>
</tr>
<tr>
<td>TXHF1</td>
<td>TXHF1 Antenna Type Const GAIN: 0.000000e+00</td>
</tr>
<tr>
<td>TXHF1</td>
<td>TXHF1 Antenna Gain (dBi): 0.300000e+01</td>
</tr>
<tr>
<td>TXHF1</td>
<td>TXHF1 Meteor Bursts Are Not Used</td>
</tr>
</tbody>
</table>

**Fig. 7.10—Database Records for Transmitters:**

Part 2 of Database Equipment Description
Fig. 7.11—Database Records for Receivers
Part 2 of Database Equipment Description

RCV_CLASS RXFNC2 FREQUENCY_CLASS LF
RCV_CLASS RXFNC2 FREQUENCY(THZ) 0.411000e+05
RCV_CLASS RXFNC2 BAND_WIDTH(THZ) 0.200000e+02
RCV_CLASS RXFNC2 ANTENA_TYPE LOOP
RCV_CLASS RXFNC2 MODEN_MECH_MODE_9
RCV_CLASS RXFNC2 PHASES 8
RCV_CLASS RXFNC2 ERROR_CORRECT PLSO_1
RCV_CLASS RXFNC2 TM_MODE_ANALYSIS_ONLY
RCV_CLASS RXFNC2 NO_NARROW_BAND_SUPPRESSION
COMMENT: FORMAT TIME (MIN) DAMAGE PROB.
RCV_CLASS RXFNC2 DAMAGE 0. 0.100
RCV_CLASS RXFNC2 UNIT_WILL_NOT_BE_RESTORED

RCV_CLASS RXFD1 FREQUENCY_CLASS LF
RCV_CLASS RXFD1 FREQUENCY(THZ) 0.411000e+05
RCV_CLASS RXFD1 BAND_WIDTH(THZ) 0.200000e+02
RCV_CLASS RXFD1 ANTENA_TYPE LOOP
RCV_CLASS RXFD1 MODEN_MECH_MODE_9
RCV_CLASS RXFD1 PHASES 8
RCV_CLASS RXFD1 ERROR_CORRECT PLSO_1
RCV_CLASS RXFD1 TM_MODE_ANALYSIS_ONLY
RCV_CLASS RXFD1 NO_NARROW_BAND_SUPPRESSION
COMMENT: FORMAT TIME (MIN) DAMAGE PROB.
RCV_CLASS RXFD1 DAMAGE 0. 0.100
RCV_CLASS RXFD1 UNIT_WILL_NOT_BE_RESTORED

RCV_CLASS RXFD2 FREQUENCY_CLASS LF
RCV_CLASS RXFD2 FREQUENCY(THZ) 0.411000e+05
RCV_CLASS RXFD2 BAND_WIDTH(THZ) 0.200000e+02
RCV_CLASS RXFD2 ANTENA_TYPE LOOP
RCV_CLASS RXFD2 MODEN_MECH_MODE_9
RCV_CLASS RXFD2 PHASES 8
RCV_CLASS RXFD2 ERROR_CORRECT PLSO_1
RCV_CLASS RXFD2 TM_MODE_ANALYSIS_ONLY
RCV_CLASS RXFD2 NO_NARROW_BAND_SUPPRESSION
COMMENT: FORMAT TIME (MIN) DAMAGE PROB.
RCV_CLASS RXFD2 DAMAGE 0. 0.100
RCV_CLASS RXFD2 UNIT_WILL_NOT_BE_RESTORED

RCV_CLASS RXVLX FREQUENCY_CLASS VLF
RCV_CLASS RXVLX FREQUENCY(THZ) 0.320000e+03
RCV_CLASS RXVLX BAND_WIDTH(THZ) 0.120000e+03
RCV_CLASS RXVLX MODEN_MECH_MODE_13
RCV_CLASS RXVLX PHASES 2
RCV_CLASS RXVLX ERROR_CORRECT PLSO_1
RCV_CLASS RXVLX TM_MODE_ANALYSIS_ONLY
RCV_CLASS RXVLX NO_NARROW_BAND_SUPPRESSION
COMMENT: FORMAT TIME (MIN) DAMAGE PROB.
RCV_CLASS RXVLX DAMAGE 0. 0.300
RCV_CLASS RXVLX UNIT_WILL_NOT_BE_RESTORED

RCV_CLASS TRKFR1 FREQUENCY_CLASS HF
RCV_CLASS TRKFR1 FREQUENCY(THZ) 0.310000e+07
RCV_CLASS TRKFR1 BAND_WIDTH(THZ) 0.150000e+03
RCV_CLASS TRKFR1 ANTENA_TYPE CONST_GAIN
RCV_CLASS TRKFR1 MODEN_MECH
RCV_CLASS TRKFR1 ANTENA_GAIN 0.080000e+00
RCV_CLASS TRKFR1 PHASES 2
RCV_CLASS TRKFR1 ERROR_CORRECT PLSO_1
RCV_CLASS TRKFR1 TM_MODE_ANALYSIS_ONLY
RCV_CLASS TRKFR1 NO_NARROW_BAND_SUPPRESSION
COMMENT: FORMAT TIME (MIN) DAMAGE PROB.
RCV_CLASS TRKFR1 DAMAGE 0. 0.020
RCV_CLASS TRKFR1 UNIT_WILL_NOT_BE_RESTORED
suggests that the database should be maintainable using ordinary screen-editor programs. If necessary, the keyword definitions can be expanded to describe new equipment features without needing to modify the records describing existing equipment types.

Part 3 of the test-case database describes the handling of messages through the physical network and communication assets defined in Parts 1 and 2 of the database. The keywords used in procedure definition are MONITOR, RECEIVE, TRANSMIT, FREQUENCY, SOURCE, INPUT, OUTPUT, INHIBIT, TIMING, PRIORITY, FREQUENCY, PROCESS, and DESTIN. In combination, these keywords support a double-ended description of link transmission of messages and the sequencing of operations within the nodes. The sequence of operations is controlled by the logical variables that follow INPUT, OUTPUT, and INHIBIT. A procedural action cannot proceed until all input variables are true. At the conclusion of a procedural element, the variables following OUTPUT are set to false. Connectivity in the network is controlled by the node names that follow SOURCE and DESTIN. The list following SOURCE names the nodes from which reception is specified in the plan; the names following DESTIN are the nodes to which transmissions are directed. For a link to be specified, the source and destination nodes must appear in the appropriate lists. TIMING specifies time delays. The parameters following the keyword are the distribution type for the time delay and the parameters of the distribution. The processing and delay of messages are described by queuing processes that cause stochastically distributed time delays to simulate message processing and equipment posturing. An operational description of the delays that these processes cause is supported. In this operational picture, the process appears as a time-random variable that is described as one of five types of parametric distributions: constant delay, uniformly distributed delay, normally distributed delay, exponentially distributed delay, and log-normally distributed delay. The records that describe the node process use the keywords NODE, PROCESS, UNIFORM, EXPONENTIAL, LOG-NORMAL, and NORMAL. NODE is followed by the name of the node, while PROCESS is followed by the name of the process. The distribution type specifications (i.e., UNIFORM, EXPONENTIAL, LOG-NORMAL, and NORMAL) are followed by three parameters whose semantics depend on the distribution type. The first parameter specifies a constant delay in minutes and has the same semantics for all distribution types. For the UNIFORM distribution, the last two parameters specify the minimum and maximum delays, while for the other distribution types, those same parameters specify the mean and standard deviation of the process delay.

The SSCF Database records for NEACP are shown in Fig. 7.12. (Lower case “neacp” is used in the records.) Each database record in Fig 7.12 could begin “NODE neacp,” but the node is suppressed in this printout for clarity of presentation. Some analysts prefer not to suppress redundant keyword labels. Thus, such a presentation option should be available in an implemented system. Likewise, in Lines 6, 7, and 8, “TRANSMIT txhfa” is suppressed for the same reasons, and the same arguments apply. In Line 7, “TRANSMIT txhfa” is repeated, since the frequency has changed. In Line 11, it is repeated again because both the frequency and timing information have changed, and likewise in Line 15, where the frequency information has changed. The 30-minute constant timing delay in the last eight lines represents a repeat of the transmission procedure (which was not actually simulated). Figure 7.13 shows data records for the Looking Glass-like node (sacbn) in the database. Note in Line 4 that the entry “INHIBIT allow_rtv_3.1” means that when the operator at Looking Glass
EVENT #5: EMER ACTION MESSAGE  PROCEDURAL ELEMENT 2.4 MESSAGE TRANSMISSION

NODE neacp PREPARE cem INPUT transmit.cam1 OUTPUT cem1_text
TIMING lognorm 0.7 0.3 PROB 99
TRANSMIT txhfa RCVR rxhfa DESTIN sacabn tacawl sabnl
   RCVR thr DESTIN gnd/fks
   FREQ 3.12MHz INPUT cem1_text
   TIMING constant 0.0 uniform 12 PROB 99
TRANSMIT txhfa RCVR rxhfa DESTIN sacabn tacawl sabnl
   RCVR thr DESTIN gnd/fks
   FREQ 3.2MHz
   TIMING constant 0.0 uniform 12 PROB 99
TRANSMIT txhfa RCVR rxhfa DESTIN sacabn tacawl sabnl
   RCVR thr DESTIN gnd/fks
   FREQ 3.12MHz INPUT cem1_text
   TIMING constant 30.0 uniform 12 PROB 99
TRANSMIT txhfa RCVR rxhfa DESTIN sacabn tacawl sabnl
   RCVR thr DESTIN gnd/fks
   FREQ 3.2MHz &ref
   TIMING constant 30.0 uniform 12 PROB 99

Fig. 7.12—Database Records for NEACP

EVENT #5: EMER ACTION MESSAGE  PROCEDURAL ELEMENT 2.1 MONITOR

NODE sacabn MONITOR rxhfa SOURCE neacp FREQ 3.1MHz PRIORITY 1
   OUTPUT allow_rec 3.1
   MONITOR rxhfa SOURCE neacp FREQ 3.2MHz PRIORITY 2
   INHIBIT allow_rec 3.1 OUTPUT allow_rec 3.2 &ref

EVENT #5: EMER ACTION MESSAGE  PROCEDURAL ELEMENT 2.2 RECEIVE

NODE sacabn RECEIVE rxhfa SOURCE neacp FREQ 3.1MHz INPUT allow_rec 3.1
   OUTPUT cem1_text TIMING uniform 0.3 PROB 70
   RECEIVE rxhfa SOURCE neacp FREQ 3.2MHz INPUT allow_rec 3.2
   OUTPUT cem1_text TIMING uniform 0.7 PROB 30 &ref

EVENT #5: EMER ACTION MESSAGE  PROCEDURAL ELEMENT 3.1: INFO PROCESSING

NODE sacabn PROCESS reformat INPUT cem1_text OUTPUT cemout_text
   TIMING lognorm 0.1 0.1 PROB 99
   PROCESS reformat INPUT cem1_text OUTPUT cemout_text
   TIMING lognorm 0.7 0.3 PROB 99 &ref

EVENT #5: EMER ACTION MESSAGE  PROCEDURAL ELEMENT 2.3 MESSAGE TRANSMISSION

NODE sacabn TRANSMIT txhfa RCVR rxhfa DESTIN hfrly1 hfrly2 hfrly3
   FREQ 3.32MHz INPUT cemout_text
   TIMING uniform 0.5 1.0 PROB 99
TRANSMIT txhfa RCVR rxhfa DESTIN hfrly1 hfrly2 hfrly3
   FREQ 3.2MHz INPUT cemout_text
   TIMING uniform 1.2 PROB 99 &ref

Fig. 7.13—Database Records for SACABN

is using this receiver to monitor 3.2 MHz, transmissions on 3.1 MHz cannot be monitored or received.

Figures 7.14 through 7.20 present the common-format data procedure records for the remaining nodes in the network.
### Fig. 7.14—Database Records for JIMCRK

<table>
<thead>
<tr>
<th>Event #8: EMER ACTION MESSAGE</th>
<th>PROCEDURAL ELEMENT</th>
<th>2.1 MONITOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE jimcrk MONITOR rx/id SOURCE nceep FREQ 41.0KHz PRIORITY 1</td>
<td>OUTPUT allow_rev_4.1.0 TIMING uniform 1.0 2.0 PROB 99 &amp;ref</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event #6: EMER ACTION MESSAGE</th>
<th>PROCEDURAL ELEMENT</th>
<th>2.2 RECEIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE jimcrk RECEIVE rx/id SOURCE nceep FREQ 41.0KHz INPUT allow_rev_4.1.0</td>
<td>OUTPUT casmi_hf PROB 99 &amp;ref</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event #5: EMER ACTION MESSAGE</th>
<th>PROCEDURAL ELEMENT</th>
<th>3.1: INFO PROCESSING</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE jimcrk PROCESS reformat INPUT camu_tlf OUTPUT camu_tlf</td>
<td>TIMING lognorm 0.1 0.1 PROB 99 &amp;ref</td>
<td></td>
</tr>
</tbody>
</table>

### Fig. 7.15—Database Records for HPRLY1

<table>
<thead>
<tr>
<th>Event #5: EMER ACTION MESSAGE</th>
<th>PROCEDURAL ELEMENT</th>
<th>2.1 MONITOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE hprly1 MONITOR trh SOURCE ssabn FREQ 3.3MHz PRIORITY 1</td>
<td>OUTPUT allow_receive_3.3 &amp;ref</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event #6: EMER ACTION MESSAGE</th>
<th>PROCEDURAL ELEMENT</th>
<th>2.2 RECEIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE hprly1 RECEIVE trh SOURCE ssabn FREQ 3.3MHz INPUT allow_rev_3.3</td>
<td>OUTPUT camin_hf PROB 99 &amp;ref</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event #5: EMER ACTION MESSAGE</th>
<th>PROCEDURAL ELEMENT</th>
<th>3.1: INFO PROCESSING</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE hprly1 PROCESS reformat INPUT camin_hf OUTPUT camout_hf</td>
<td>TIMING lognorm 0.1 0.1 PROB 99 &amp;ref</td>
<td></td>
</tr>
</tbody>
</table>

### Fig. 7.16—Database Records for HPRLY2

<table>
<thead>
<tr>
<th>Event #6: EMER ACTION MESSAGE</th>
<th>PROCEDURAL ELEMENT</th>
<th>2.3 MESSAGE TRANSMISSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE hprly1 TRANSMIT trh RCVR trh DESTIN gadiks FREQ 3.4MHz INPUT camout_hf OUTPUT signal_hf</td>
<td>TIMING UNIFORM 1 2 PROB 99 &amp;ref</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event #5: EMER ACTION MESSAGE</th>
<th>PROCEDURAL ELEMENT</th>
<th>2.3 MESSAGE TRANSMISSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE hprly2 MONITOR trh SOURCE ssabn FREQ 3.3MHz</td>
<td>OUTPUT allow_receive_3.3 &amp;ref</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event #6: EMER ACTION MESSAGE</th>
<th>PROCEDURAL ELEMENT</th>
<th>2.2 RECEIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE hprly2 RECEIVE trh SOURCE ssabn FREQ 3.3MHz INPUT allow_rev_3.3</td>
<td>OUTPUT camin_hf PROB 99 &amp;ref</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event #5: EMER ACTION MESSAGE</th>
<th>PROCEDURAL ELEMENT</th>
<th>3.1: INFO PROCESSING</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE hprly2 PROCESS reformat INPUT camin_hf OUTPUT camout_hf</td>
<td>TIMING lognorm 0.1 0.1 PROB 99 &amp;ref</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event #6: EMER ACTION MESSAGE</th>
<th>PROCEDURAL ELEMENT</th>
<th>2.3 MESSAGE TRANSMISSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE hprly2 TRANSMIT trh RCVR rxhfs DESTIN tacwl FREQ 3.5MHz</td>
<td>INPUT casmi_hf OUTPUT signal_hf</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event #5: EMER ACTION MESSAGE</th>
<th>PROCEDURAL ELEMENT</th>
<th>2.3 MESSAGE TRANSMISSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE hprly2 TRANSMIT trh RCVR rxhfs DESTIN tacwl FREQ 3.5MHz</td>
<td>INPUT casmi_hf OUTPUT signal_hf</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event #6: EMER ACTION MESSAGE</th>
<th>PROCEDURAL ELEMENT</th>
<th>2.3 MESSAGE TRANSMISSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE hprly2 TRANSMIT trh RCVR rxhfs DESTIN tacwl FREQ 3.5MHz</td>
<td>INPUT casmi_hf OUTPUT signal_hf</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event #5: EMER ACTION MESSAGE</th>
<th>PROCEDURAL ELEMENT</th>
<th>2.3 MESSAGE TRANSMISSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE hprly2 TRANSMIT trh RCVR rxhfs DESTIN tacwl FREQ 3.5MHz</td>
<td>INPUT casmi_hf OUTPUT signal_hf</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event #6: EMER ACTION MESSAGE</th>
<th>PROCEDURAL ELEMENT</th>
<th>2.3 MESSAGE TRANSMISSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE hprly2 TRANSMIT trh RCVR rxhfs DESTIN tacwl FREQ 3.5MHz</td>
<td>INPUT casmi_hf OUTPUT signal_hf</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event #5: EMER ACTION MESSAGE</th>
<th>PROCEDURAL ELEMENT</th>
<th>2.3 MESSAGE TRANSMISSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE hprly2 TRANSMIT trh RCVR rxhfs DESTIN tacwl FREQ 3.5MHz</td>
<td>INPUT casmi_hf OUTPUT signal_hf</td>
<td></td>
</tr>
</tbody>
</table>
EVENT #5: EMER ACTION MESSAGE  PROCEDURAL ELEMENT 2.1 MONITOR

NODE hfrly3 MONITOR trh SOURCE sacaba FREQ 3.4MHz PRIORITY 1
OUTPUT allow_recv.3_3 & ref

EVENT #5: EMER ACTION MESSAGE  PROCEDURAL ELEMENT 2.2 RECEIVE

NODE hfrly3 RECEIVE trh SOURCE sacaba FREQ 3.3MHz INPUT allow_recv.3_3
OUTPUT camin_hf PROB 99 & ref

EVENT #5: EMER ACTION MESSAGE  PROCEDURAL ELEMENT 3.1: INFO PROCESSING

NODE hfrly3 PROCESS reformat INPUT camin_hf OUTPUT camout_hf
TIMING lognorm 0.1 0.1 PROB 99 & ref

EVENT #5: EMER ACTION MESSAGE  PROCEDURAL ELEMENT 2.3 MESSAGE TRANSMISSION

NODE hfrly3 TRANSMIT trh RCVR rchfa DESTIN sbnh1
FREQ 3.6MHz INPUT camout_hf OUTPUT signal_hf
TIMING UNIFORM 1 2 PROB 98 & ref

Fig. 7.17—Database Records for HFRLY3

EVENT #5: EMER ACTION MESSAGE  PROCEDURAL ELEMENT 2.1 MONITOR

NODE tacwl MONITOR rchfa SOURCE nceep FREQ 3.1MHz PRIORITY 1
OUTPUT allow_recv.3_1
MONITOR rchfa SOURCE nceep FREQ 3.2MHz PRIORITY 2
INHIBIT allow_recv.3_1 PROB 30
MONITOR rchfa SOURCE hfrly1 FREQ 3.4MHz PRIORITY 3
INHIBIT allow_recv.3_3 PROB 30
MONITOR rchfa SOURCE sacaba FREQ 41.6KHz PRIORITY 1
OUTPUT allow_recv.4_0
MONITOR rchfa SOURCE jmcwh FREQ 41.1KHz PRIORITY 2
INHIBIT allow_recv.4_0 PROB 98 & ref

EVENT #5: EMER ACTION MESSAGE  PROCEDURAL ELEMENT 2.2 RECEIVE

NODE tacwl RECEIVE rchfa SOURCE sacaba FREQ 3.1MHz INPUT allow_recv.3_1
OUTPUT camin_hf TIMING uniform 0 4 PROB 60
RECEIVE rchfa SOURCE sacaba FREQ 3.2MHz INPUT allow_recv.3_2
OUTPUT camin_hf TIMING uniform 0 7 PROB 30
RECEIVE rchfa SOURCE sacaba FREQ 4.4MHz INPUT allow_recv.3_4
OUTPUT camin_hf TIMING uniform 0 9 PROB 10
RECEIVE rchfa SOURCE jmcwh FREQ 41.6KHz INPUT allow_recv.4_0
OUTPUT camin_hf TIMING uniform 0 3 PROB 70
RECEIVE rchfa SOURCE jmcwh FREQ 41.1KHz INPUT allow_recv.4_1
OUTPUT camin_hf TIMING uniform 0 7 PROB 30 & ref

EVENT #5: EMER ACTION MESSAGE  PROCEDURAL ELEMENT 3.1: INFO PROCESSING

NODE tacwl PROCESS reformat INPUT camin_hf OUTPUT camout_vlf
TIMING lognorm 0.1 0.1 PROB 99 & ref

EVENT #5: EMER ACTION MESSAGE  PROCEDURAL ELEMENT 2.3 MESSAGE TRANSMISSION

NODE tacwl TRANSMIT trh RCVR rchfa DESTIN sbnh1
FREQ 23.0KHz INPUT cam1_text
TIMING UNIFORM 2 4 PROB 95 & ref

Fig. 7.18—Database Records for TACWL (TACWL + TWILF)
CONVERSION FROM COMMON FORMAT TO SIMSTAR FORMAT

With the test-case information expressed in the common-format database, it remained to demonstrate that the encoded information could be used to construct an input file that would run on an operational C³ simulation model. Since the record formats were constructed to correspond to unique parsing trees, the information was known to be machine readable. Such machine readability was demonstrated on several occasions when information encoded for the SIMSTAR 85A models was first converted to keyword format and then was reconverted back into SIMSTAR format. These machine-translation efforts had covered both Part-1 node description data and Part-2 equipment data. Because of differences in the SIMSTAR 85A message-routing scheme and the double-ended message procedure description of the common format, it was clear that significant approximations were involved in converting from the common format to SIMSTAR. It was understood that these approximations would closely correspond to those normally encountered in approximating a C³ system description in the entities SIMSTAR 85A supports. To gain better understanding of these procedural translation problems, a parallel SIMSTAR 85A representation of the test case was prepared. The accommodations and approximations necessary to effect this parallel representation are described below.

Simulating a C³ system described in the common-format database requires translating the common-format statements into entities and functions supported by the target C³ simulation. Despite the machine readability of the common-format database, the semantic differences between the common-format representation and the model representations make translation
EVENT #5: EMER ACTION MESSAGE  PROCEDURAL ELEMENT 2.1 MONITOR

NODE ssn1 MONITOR rxf1 SOURCE nacop FREQ 3.1MHz PRIORITY 1
    OUTPUT allow_rev_3.1
MONITOR rxf1 SOURCE nacop FREQ 3.2MHz PRIORITY 2
    INHIBIT allow_rev_3.1 3.4 OUTPUT allow_rev_3.2
MONITOR rxf1 SOURCE hiryl FREQ 3.4MHz PRIORITY 3
    INHIBIT allow_rev_3.1 3.2 OUTPUT allow_rev_3.4
MONITOR rxf1 SOURCE xacaba FREQ 41.0MHz PRIORITY 1
    OUTPUT allow_rev_41.0
MONITOR rxf1 SOURCE jimerk FREQ 41.1MHz PRIORITY 2
    INHIBIT allow_rev_41.0 OUTPUT allow_rev_41.1
MONITOR rxf1 SOURCE tacaw FREQ 23.0MHz PRIORITY 1
    OUTPUT allow_rev_cws & ref

EVENT #5: EMER ACTION MESSAGE  PROCEDURAL ELEMENT 2.2 RECEIVE

NODE ssn1 RECEIVE rxf1 SOURCE nacop FREQ 3.1MHz INPUT allow_rev_3.1
    OUTPUT canminhf TIMING uniform 0.4 PROB 60
RECEIVE rxf1 SOURCE nacop FREQ 3.2MHz INPUT allow_rev_3.2
    OUTPUT canminhf TIMING uniform 0.7 PROB 30
RECEIVE rxf1 SOURCE nacop FREQ 3.4MHz INPUT allow_rev_3.4
    OUTPUT canminhf TIMING uniform 0.9 PROB 10
RECEIVE rxf1 SOURCE jimerk FREQ 41.0MHz INPUT allow_rev_41.0
    OUTPUT canminhf TIMING uniform 0.3 PROB 70
RECEIVE rxf1 SOURCE jimerk FREQ 41.1MHz INPUT allow_rev_43.1
    OUTPUT canminhf TIMING uniform 0.7 PROB 30
RECEIVE rxf1 SOURCE tacaw FREQ 23.0MHz INPUT allow_rev_23.0
    OUTPUT canminhf TIMING uniform 0.0 PROB 99 & ref

EVENT #5: EMER ACTION MESSAGE  PROCEDURAL ELEMENT 3.1: INFO PROCESSING

NODE ssn1 PROCESS emer_inch_cmd INPUT canminhf OUTPUT launch_weapons
    TIMING lognorm 0.1 0.1 PROB 99
PROCESS emer_inch_cmd INPUT canminhf OUTPUT launch_weapons
    TIMING lognorm 0.1 0.1 PROB 99 & ref

Fig. 7.20—Database Records for SSBNL

difficult, approximate, and dependent on the problem the simulation experiment is intended to resolve. The translation problem requires producing a set of model inputs that are equivalent to a real C³ system in at least the operational sense that equivalent inputs will produce statistically equivalent outputs in both the real world and in the simulation. To achieve statistical equivalence, both the real-world experiment (field tests) and the simulation-world experiment must produce nearly isomorphic event types, event reliabilities, and delay distributions over event times. Without such equivalence, the simulated events of the model cannot give useful knowledge of the real C³ system.

The difficulty of achieving a translation with statistical fidelity is equivalent to directly inputting the C³ system for simulation. The common format intentionally corresponds in semantics to the natural-language statements that appear in C³ system specifications. The common format accomplishes only a syntactic transformation of the source documents, but it is a transformation that makes the data machine readable. The semantics of the data are left largely unchanged. The semantics of the prototype test case are almost identical when expressed in either natural language or in the common format. The fact that a human reader gains the same information from either format demonstrates semantic equivalence between document and database. Logically, the next step was to show that the semantic con-
tent of the database was sufficient to construct input for a currently running model, i.e., SIMSTAR.

By demonstrating adequacy of semantic support, the translation of the prototype test case
from the common-format database to SIMSTAR input format explored problem areas and
approximation issues between the two representations. During the task, it became clear that
there were trade-offs between developing translations that accurately reproduced system-
output statistics and those that merely "looked" like pseudomorphs of the system being
modeled. Where the supported elements of the simulation were very different from those of
the system to be modeled, large numbers of the simulated elements would have to be
combined to replicate the real-world statistics. The resulting complex of model entities often
had little surface resemblance to the system which was being studied; the connection
between reality and simulation was no longer immediate or obvious. Large diversities of
form between the system and its simulation meant that only lengthy inductive and deductive
argument could connect the two—making verification difficult. We found that the semantics
of system and model are sufficiently different that a one-size-fits-all translation approach is
inadequate. Whether working from the common-format database or directly from
specification documents, judgement and experience are needed in deciding what to include,
exclude, or approximate.

Interpretation issues were encountered both in the representation of procedures and to a
lesser extent in the representation of the communication relay nodes. The procedural ele-
ments in the prototype common-format database included channel monitoring and message-
handling procedures. Many options were available for translating the monitoring actions
implied in the common-format source statements. The translation of the monitoring actions
could have involved creating multiple event entities: virtual kill events, virtual restoral
events, and causally connected contingent events. Event sequencing did require adding de-
lay processes described by probability distributions to enforce the time delays specified in the
source statements. There were differences in the specification of message-handling delays
between the common-format database statements and the SIMSTAR representation.

The representation of delay processes required adapting the corresponding SIMSTAR entity
to include the reliability weight that is associated with the delay penalty in the common for-
mat; this was done by creating new distributions, such as the extended UNIFORM or com-
bining virtual SHUNT processes whose parallel connection approximated an arbitrary con-
tinuous-discrete delay distribution. For the most complex communication nodes, i.e., the
command centers, the number of receivers, transmitters, and message handling processors
became too large for the simulation's data structures. Since only a few very important nodes
violated those constraints, those nodes were modeled as multiple connected nodes in the
simulation. What was in reality a single supernode became in the model a cluster of more
modest entities connected together by delayless, virtual communication links, with both the
node multiplicity and the virtual entities being artifices of the interpretation.

The prototype common-format database was developed to demonstrate a variety of C³ seman-
tics involving both equipment and procedure descriptions. Among the procedures described
were message handling, transmitter operations, and channel monitoring. Although there are
more complex procedures, the above selection was felt to be adequate to demonstrate proce-
dure translation. The complex procedures that start up the network were omitted because
dynamic channel-monitoring involves the key discrete-event features—just fewer of them.
In the prototype common-format database, the message-handling delays were modeled as random variables drawn from continuous-discrete distributions. For instance, a system may cause no message delay if ready or a random delay for setup if it is not; this example has finite probability of exactly zero delay and the complement of that probability, 1.0, that a random setup delay will occur.

A variety of store-and-forward operations can be modeled by such continuous-discrete distributions: preparation delays, message distribution, message decoding, format conversion, or release authorization. These events proceed independently and therefore can be modeled by independently distributed random variables. Many transmitter posturing event delays can be modeled as independent random variables: antenna deployment, antenna retuning, transmitter power-up, or transmitter self-test. Although many operations proceed independently, coordinated, interacting actions like monitoring and network initiation require detailed treatment of interactions that cannot be modeled by independent random variables. While the common-format database readily represents such dependence between events, such dependence can be difficult to represent with the models.

In modeling monitoring, the differentiating feature is a responsiveness to environment and the taking of contingent actions, such as retuning to compensate for loss of signal or the presence of unacceptable interference. This means that high-fidelity modeling requires that certain events sharing common causes must occur together in a correlated fashion, while events whose causes are mutually exclusive must not be allowed to occur unless the other event is precluded. For instance, under some monitoring rules, a receiver will not be retuned unless the carrier signal on the base channel has been lost for a specified period of time.

The common-format database provides directly for event synchronization with the keywords \texttt{INPUT}, \texttt{OUTPUT}, and \texttt{INHIBIT}—these three keywords operate to structure events into effects of causes (positioned after \texttt{INPUT}) or into causes of effects (positioned after \texttt{OUTPUT}). \texttt{INHIBIT} negates a cause. The SIMSTAR event semantics are very different. The key SIMSTAR entities are the events \texttt{kill}, \texttt{restore}, \texttt{cause}, \texttt{prevent}, \texttt{enable}, and \texttt{disable}. To create a given truth table, the SIMSTAR model may need to add fictitious nodes that can be "killed" and "restored" to replicate the necessary coordinations between events. Most of these complexities were not encountered in translation of the common-format database prototype for SIMSTAR. In the prototype common-format database, the fact that only a single message was involved allowed most of the monitoring activities to be modeled as independent delay distributions.

Since SIMSTAR doesn't support continuous-discrete delay distributions, two methods were developed for approximating this feature of the C³ common format. These approximations resolved a semantic deficiency of that model; this deficiency arose because the continuous-discrete delay process is a useful characterization of a wide variety of real-world situations. For instance, in channel monitoring, the operator may have standing orders to scan through a cycle of frequencies. Under these conditions, the receiver has two states: Either the receiver is tuned to the transmitter frequency, or it is not. If the receiver is appropriately tuned, message reception can begin immediately (zero delay); otherwise, reception must wait until both transmitter and receiver are on the same frequency (uniformly distributed delay). The delay statistics for transmissions attempted during the receiver nonlistening state will follow a uniform distribution (U) of times between zero and the total time that the receiver stays tuned to other frequencies. The probability (P) of zero message delay is the ratio of the
on-frequency time to the total monitor-cycle time. For this type of monitoring and with single, widely spaced transmissions and single nodes, the continuous-discrete delay distribution is a natural model.

Figure 7.21, a process flow graph that models this compound delay distribution, consists of a switched link that is closed with probability (P) in parallel with a uniform delay process (U). When the switch is closed, the message is delivered straight through; otherwise, it is delayed by a random draw from distribution U.

In view of the complex, multistate process involved in changing between many different receive frequencies, it is surprising that a two-state stochastic process can adequately model the average message delay. This simplification is possible because the proposed experiment does not look at correlated events, such as the conjoin message-arrival statistics for two separate messages offered on separated frequencies of the receiver’s frequency time plan. In this example, the joint statistics would be wrong because the approximating delay distribution model would allow both messages to be received with either maximum delay or with no delay at all; in the real system, this could not happen. Simultaneous reception would imply that the receiver was tuned to two frequencies at once; maximum delays for both messages would violate the fact that the receiver must be tuned to at least one of the frequencies. When synchronization of actions is important, the continuous-discrete delay distribution is inadequate, and complex interacting events must be modeled so that mutually exclusive alternatives become active and available only after their alternatives are rendered inactive and unavailable. The operations of the INPUT and OUTPUT keywords on the logic variables of the C 3 common format provide an easy approach to describing such dependent events unambiguously. Although multiple, dependent messages were not included in the test case, they could have been included easily.

Despite the need for a continuous-discrete delay distribution, the SIMSTAR model only allows delay distributions with a 100 percent chance of occurrence. Equivalently, a SIMSTAR delay can only take one of two forms: It can be a fixed (discrete) delay, or it can be drawn from a distribution (continuous). A SIMSTAR random draw cannot be selected from a distribution with both discrete and continuous parts, which is what is needed for modeling the common-format database statements. It is not possible to have a finite probability of a

![Fig. 7.21—Flow Diagram of a Continuous-Discrete Delay Process](image-url)
single, discrete value of delay, e.g., zero, and then draw random delays between zero and a maximum value according to the complement of that probability.

Two methods for working around this were developed. The first method, extended uniform distribution, exploited the fact that the simulation does not allow time delays to point to times prior to when the delay is computed. Thus, if a uniform distribution, \( U(-a,b) \), were specified with a negative value for the lower boundary, all variates that fell below zero would be mapped onto zero—thus creating a probability mass for zero delay that could be set equal to \( P \) by requiring that

\[
a = P \frac{b}{1-P}.
\]

The equivalent flow diagram for this approximation appears in Figure 7.22. Although the negative, lower value of delay caused the model's database manager to issue error diagnostics, the simulator executed the random delays so as to model a continuous-discrete uniform distribution with a probability of nonzero delay \( P \), \( U(0,b;P) \). The corresponding delay distribution is given by the following nonswitched random process flow. Of course, this method was not entirely satisfactory, because it can accommodate only discrete-continuous distributions of a certain type; the process must have a finite probability of zero delay with the remaining probability distributed uniformly between zero and a maximum value. The problem arises because the continuous part of the continuous-discrete distribution need not be uniform. Nonuniform delay distributions, such as the exponential, normal, and lognormal, can occur and need to be modeled. Uniform distributions with a minimum value larger than zero also cannot be modeled. This motivated the development of a second approach, which modeled the switch-off process with a shunt distribution.

The shunt process approach models the zero-delay bypass switch with a suitable parallel-connected uniform-delay process, \( U(-a,b) \), with the desired continuous-delay process. The corresponding flow diagram appears in Fig. 7.23. The value of \( b \) is selected to be much larger than either the mean or standard deviation of the arbitrary distribution which is to be switched. The value of \( a \) is selected according to the formula displayed above—to force the shunt process to have the probability \( P \) of producing a zero delay. Since SIMSTAR acts on the smaller of two parallel processes, the primary effect of the shunt process is to set the delay equal to zero with probability \( P \). Similarly, by setting maximum value, \( b \), of the uniform shunt process \( U(-a,b) \) equal to a large number, it is highly probable that any draw from the arbitrary distribution \( D(\mu, \sigma) \) will be smaller than any non-zero draw from the

![Truncated uniform distribution](image)

**Fig. 7.22—Flow Diagram of the Extended Uniform Distribution Approximation**
shunt distribution and this will produce a distribution of form \( D(\mu, \sigma) \), which will be bypassed with probability \( P \).

In translating the delay processes present in the common-format test case, we did not use the shunt process approach, either because the weighted delays could be modeled using the extended uniform distribution or because the bypass probabilities were so small that they could be ignored. The error involved in setting a few 98-percent and 99-percent probable events to 100 percent was taken to be tolerable in a test-case run with 100 Monte Carlo repetitions or less. The switched processes involving arbitrary distributions were modeled with the continuous distributions, which are supported in SIMSTAR. This permitted the delays in the SIMSTAR system representation to preserve a one-to-one relationship with the entities in the common-format database representation. Had the shunt technique been used, the one-to-many feature of that approximation method would have led to fuzzy traceability. This fuzziness results from adding increasing numbers of artificial, virtual features to a simulated, finite-state machine to obtain greater statistical fidelity. In a limited way, this demonstrated the increasing complexity that is required to reach increasing levels of stochastic accuracy, the most extreme examples of which are the virtual nodes, virtual equipments, and conditional events that are required to model synchronized processes in SIMSTAR.

Modeling the communication links for the common-format database required developing equivalent representations of the transmitters and receivers referenced in the source statements using the class entities of the SIMSTAR model. Here, a small semantic difference between the common format and the SIMSTAR model semantics was encountered. In SIMSTAR, a transmitter at a node must be assigned a frequency class, and that frequency class has only a single frequency: unlike a physical transmitter or receiver, it cannot be retuned. In the common format, the transmitter and receiver frequencies are not considered to be part of the equipment description, and the assigned frequency or frequencies monitored.
can be taken to be any of those that fall in the tuning band of the equipment. This meant that an item of equipment which monitored multiple channels gave rise to an array of receivers and transmitters such that an assignment of a “simulated” unit could be made to each channel.

Thus, in preparing the SIMSTAR representation, some of the single pieces of equipment in the common format quickly became two or three pieces of virtual SIMSTAR equipment—and this was required just to provide connectivity over the specified frequencies. For example, where a transmitter class (e.g., TXHFA) is used to transmit over more than one frequency (no matter at how many nodes), a SIMSTAR “pseudo-class” (e.g., TXHFA1) must be used for each equipment class—and-frequency pair, since SIMSTAR does not allow a transmitter class to use more than one frequency. The approximations are summarized in Table 7.4 for both transmitters and receivers. Comparison with Table 7.2 demonstrates this artifice and its role in adding complexity to the SIMSTAR 85A representation. Receiver and transmitter

<table>
<thead>
<tr>
<th>Equipment Classes (database, SIMSTAR)</th>
<th>Freq Band</th>
<th>Frequencies Used</th>
<th>User Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANSMITTERS TXHFA</td>
<td>HF</td>
<td>3.1, 3.2, 3.3</td>
<td>MHz</td>
</tr>
<tr>
<td>TXHFA1</td>
<td>HF</td>
<td>3.1</td>
<td>MHz</td>
</tr>
<tr>
<td>TXHFA2</td>
<td>HF</td>
<td>3.2</td>
<td>MHz</td>
</tr>
<tr>
<td>TXHFA3</td>
<td>HF</td>
<td>3.3</td>
<td>MHz</td>
</tr>
<tr>
<td>TRANSCIEVERS TRHT</td>
<td>HF</td>
<td>3.4, 3.5, 3.6</td>
<td>MHz</td>
</tr>
<tr>
<td>TRHT1</td>
<td>HF</td>
<td>3.4</td>
<td>MHz</td>
</tr>
<tr>
<td>TRHT2</td>
<td>HF</td>
<td>3.5</td>
<td>MHz</td>
</tr>
<tr>
<td>TRHT3</td>
<td>HF</td>
<td>3.6</td>
<td>MHz</td>
</tr>
<tr>
<td>trhr</td>
<td>HF</td>
<td>3.1, 3.2, 3.3, 3.4</td>
<td>MHz</td>
</tr>
<tr>
<td>TRHR1</td>
<td>HF</td>
<td>3.1</td>
<td>MHz</td>
</tr>
<tr>
<td>TRHR2</td>
<td>HF</td>
<td>3.2</td>
<td>MHz</td>
</tr>
<tr>
<td>TRHR3</td>
<td>HF</td>
<td>3.3</td>
<td>MHz</td>
</tr>
<tr>
<td>TRHR4</td>
<td>HF</td>
<td>3.4</td>
<td>MHz</td>
</tr>
<tr>
<td>RECEIVERS RXHFA</td>
<td>HF</td>
<td>3.1, 3.2, 3.5, 3.6</td>
<td>MHz</td>
</tr>
<tr>
<td>RXHFA1</td>
<td>HF</td>
<td>3.1</td>
<td>MHz</td>
</tr>
<tr>
<td>RXHFA2</td>
<td>HF</td>
<td>3.2</td>
<td>MHz</td>
</tr>
<tr>
<td>RXHFA3</td>
<td>HF</td>
<td>3.5</td>
<td>MHz</td>
</tr>
<tr>
<td>RXHFA4</td>
<td>HF</td>
<td>3.6</td>
<td>MHz</td>
</tr>
<tr>
<td>RXIF</td>
<td>LF</td>
<td>41.0, 41.1</td>
<td>kHz</td>
</tr>
<tr>
<td>RXLF C1</td>
<td>LF</td>
<td>41.0</td>
<td>kHz</td>
</tr>
<tr>
<td>RXLF C2</td>
<td>LF</td>
<td>41.1</td>
<td>kHz</td>
</tr>
<tr>
<td>RXIFD</td>
<td>LF</td>
<td>41.0, 41.1</td>
<td>kHz</td>
</tr>
<tr>
<td>RXLF D1</td>
<td>LF</td>
<td>41.0</td>
<td>kHz</td>
</tr>
<tr>
<td>RXLF D2</td>
<td>LF</td>
<td>41.1</td>
<td>kHz</td>
</tr>
<tr>
<td>HARD-WIRE PSEUDO-CONNECTIONS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RXVRY</td>
<td>(N/A)</td>
<td>0</td>
<td>(n/a)</td>
</tr>
<tr>
<td>TXVRY</td>
<td>(N/A)</td>
<td>0</td>
<td>(n/a)</td>
</tr>
</tbody>
</table>
pseudo-classes are instances of the class with a frequency attribute added. We tracked the nomenclature from the common-format database equipment classes, through the SIMSTAR equipment classes and pseudo-classes, to the SIMSTAR equipment names (e.g., nept1). An analyst using a specific model and the common-format database should do the same. Besides the proliferated transmitters and receivers, the model required special fictitious hard-wired classes when coupling virtual nodes that corresponded to a single physical node. The fictitious hard-wired link transmitters and receivers are classes TXV7Y and RXVRY.

With many of the approximations, alternative approaches could have been adopted with varying degrees of complexity. For instance, synchronizing the frequency coverage would have required even more complexity. One approach to simulating the tuning over several frequencies is to create a series of fictitious jamming stations, one for each receiver frequency. Using conditional-kill events, the jammers can be enabled or disabled such that only one of the frequency channels appears to be open at one time; the others are jammed out. In modeling synchronized events, seemingly trivial common-format database statements become vast collections of interacting events and entities, no single one of which has an identifiable counterpart in the source database. In the semantics of natural languages, this occurs when one language either cannot express a concept from the other at all or can express it only by a clumsy and lengthy circumlocution. Along these lines, one of the reassuring observations of the translation of the common format to SIMSTAR was that no single instance was found where data in the common format were more compactly or accurately represented in the SIMSTAR description. This indicates that the common-format semantics have the feature of greater primacy, which was a key motivation for the project.

Although the semantics between the SIMSTAR representation for communication stations (nodes) and the platform description of the common-format database are quite close, a second-order approximation was needed to produce an efficient interpretation. The need for this approximation arose because of the model's static data model, which is implemented in FORTRAN 77. For the communication relay nodes, a set of arrays provides storage for the pointers to the transmitters, receivers, and processes that are present at each node. Assuming that each node possesses the maximum number of transmitters, receivers and message processors results in a very large and unwieldy data structure. If the array dimensions are set smaller for the equipment at an average node, there will not be enough space for that of a major, complex node. To get around such limitations, SIMSTAR users must resort to dividing complex nodes into clusters of component nodes that connect together with delayless virtual links to "simulate" the operations of a single, large supernode. The example SIMSTAR database contains two such nodes: a node similar to a communication relay aircraft, represented by three nodes (TACWL, TWLHF, and TWLIF), and a command aircraft node, also represented by three nodes (SACR, SACT, and SACLIF). These pseudo-node approximations are diagrammed in Figures 7.24 and 7.25. The three boxes labeled sacr, saclf, and sachf are three model nodes that, when coupled together, form a single supernode representing Looking Glass.

Our efforts in translating the prototype common-format database indicate that, for any reasonably complex C3 system, there is no one, correct SIMSTAR interpretation. Depending on the scope of the planned experiment, the interpretation scope in SIMSTAR entities depends critically on the fineness of detail that the analyst seeks. For instance, where complex
Fig. 7.24—SIMSTAR 85A Representation of a SACABN-Like Node via Three Pseudo-Nodes (sacr, sachf, and saclf)

Fig. 7.25—SIMSTAR 85A Representation of a TACAMO-Like Node via Two Pseudo-Nodes (tacr and tact)
details of a channel hand-over protocol are required, a complex finite-state machine must be synthesized in which cause-events and effect-events interact with hardware readiness states to model the availability of first one channel and then the other. Where only simple end-to-end delays are involved, the overall message delay caused by sharing message traffic between the two channels can be modeled as a distribution function over message-delay time.

The implication of this freedom in producing model interpretations is that any fixed rules and conventions for making these translations must be a compromise between translation fidelity and simulation complexity. In particular, in the translation production-rule approach we examined in the course of the project, the rule base controlling the common-format database translations would have to support such a compromise. The rule base would need to be written to support dependence on the application for which the translation was intended. The translations produced from a fixed rule base would not be able to match the complexity of the translation to the objectives of the analyst. This would make it essential to document the translation's limits and the scope of the simulation experiments it could support.

In summary, the experience with the prototype database suggests that C³ systems described in the common format can also be adequately interpreted for simulation in SIMSTAR, and that this format can also be used as a starting point to create simulation experiments for other models. Furthermore, the common-format description appears to be more fundamental than those used in the SIMSTAR model, in the senses that fewer semantic entities are needed to describe a C³ system and that the entities used are more independent. This is significant, in that independent semantics mean both fewer statements and fewer interactions between statements to verify for correctness. Another conclusion is that models should be designed to handle these primitive concepts to more accurately model “real” systems.
8. CONCLUSIONS AND RECOMMENDATIONS

This report has described the results of our prototype database development effort. We now offer seven major conclusions and a set of recommendations.

CONCLUSION 1: CONCEPT

The basic concept provides two things that are new:

1. A standard strategic command and control communications (C³) database, an approved set of information that is designated and centrally controlled by the Joint Staff to serve as a baseline for analyses by the commanders in chief, Services, and Agencies.

2. A common-format version of this standard strategic database, including support software and documentation.

The combination of these new items is the Standard Strategic Common Format (SSCF) Database. Note that there is a distinction between a standard database and a common-format database. The former implies a formal status, i.e., review and approval of the content by the Joint Staff; it can also imply directed usage or application. The latter is a database design choice made to make the database itself easier and less costly to use. It does, however, require the development of support software and documentation.

An interim standard strategic C³ database exists. It has been defined to be the database used in Strategic Connectivity Master Plan computations by the Defense Information Systems Agency (DISA) for the Joint Staff. This database is not yet a formal standard, as defined above, nor is it in a format easily read by other codes. Nevertheless, a standard database package can be developed and promulgated prior to and independently of the development of a common-format version of the database. This approved standard should be a mandatory baseline from which each organization would conduct excursions appropriate to its needs.

A common format would facilitate the use of the standard database. An analyst would use the support software to extract a data set from the SSCF Database and apply it to a network simulation or a communications link's engineering code. One could also use the support software to extract a data set from a particular model's input files and make it a part of the SSCF Database.

CONCLUSION 2: IMPLEMENTATION AND CONTROL

The Database would be controlled by the Joint Staff/J-6, but would be maintained and distributed by the Defense Communications Agency. Central control is vital to maintain the integrity, quality, security, and ultimately the confidence of the user community in the SSCF Database.
The database “product” to be disseminated would consist of

- Selected parts of the database
- A glossary
- Support software (data translation and utility files)
- User reference documentation, the Database Description and Users' Guide.

The Database Description and Users' Guide will tell the user what is in the Database (or the data sets provided), how to use the Database and its supporting facilities, and how to produce a data-input set appropriate to selected models. This package would be sent out by the database manager (DISA) at the request of the user upon receiving approval by the database executive (J-6).

CONCLUSION 3: ORGANIZATION AND CONTENT

We conclude that, for detailed analyses, comparisons, and diagnoses, a baseline database encompassing the following categories is required:

- Nodes (locations; equipment and hardware parameters)
- Equipment (receiver, transmitter, modem, power, and other engineering parameters)
- Network topology (interconnection among nodes)
- Procedures (messages, routing, priorities, delays, and conferences)
- Environment (benign, threat, attack, or damage)
- Ground rules and assumptions.

The first three items represent the “hard core” of the database and are also the most straightforward to implement. Items 4 and 5 must also be included in the standardization process, because they are often the root causes of unintended differences and discrepancies.

Item 6, ground rules and assumptions, although not considered a part of the database in the past, should be articulated, documented, and disseminated with the database. They, too, are frequently the causes of major differences between otherwise similar assessments.

Some parameters specified by the analyst (e.g., report generation and format requests) are model-specific or related to a specific analytical operation. They should not be included in the database. Widely used items, such as the CCIR noise tables, could be referenced in the documentation accompanying the database, even though they are not formally an integral part of the database.

CONCLUSION 4: FORMAT

The common-format database should be compatible with

- DISA's WWMCCS Allocation and Assessment Model (WAAM)
• The Air Force's Strategic C³ Analysis Model (STRATC³AM or STRATCAM), including the earlier version—SIMSTAR
• The Navy's Strategic Communications Simulator/Strategic Communications Continuing Assessment Program (NSCS/SCAP)
• The Defense Nuclear Agency's Weapons Effects on Communication Systems (WECOM) Codes
• The Defense Nuclear Agency’s Simulation of Multiple Bursts and Links (SIMBAL II).

This list is not intended to be constraining; that is, the concept must be sufficiently flexible to be able to support other models in the future. Newer models will continue to be developed and will need database support.

Our recommended common-format database file consists of a sequence of keyword records, each consisting of a sequence of keyword phrases followed by a validation phrase. Each keyword phrase consists of a keyword followed by a list of values. Each validation phrase consists of a specifically defined sequence of keywords and their corresponding values. The semantic content and structural constraints of the database are defined by a glossary. Each keyword definition also includes the list of those child keywords, and only those, that may immediately follow keyword (the parent) within the same keyword record.

The application of the common-format approach to characterizing procedures and operational data is feasible and practical. We found that it was possible to translate all of the different types of procedures in the Emergency Action Plan Volume VII to the common KEYWORD—and—VALIDATION PHRASE format.

CONCLUSION 5: DEVELOPMENT EFFORT

We estimate that the full development and documentation of the support software (keyword glossary manager, database manager, data set translators, and report generator) would cost from $800K to over $1200K.

The conversion of the strategic communications database itself could probably be accomplished for about $300K to $400K, although we acknowledge that the database development estimate is cruder than the software development estimates.

An austere effort involving the development of the keyword glossary editor, data editor, and translators to support only the WAAM—and to a degree, STRATCAM—could be accomplished for about $300K to $400K.

CONCLUSION 6: APPLICATIONS

Over a dozen government agencies and their supporting analytic teams could be users of the Standard SSCP Database.

These organizations typically use one network discrete-event simulation, several communications link engineering codes, and—rarely, and only if required—the “physical principle” computer models (which perform the calculations of link performance in even greater detail and with even higher fidelity than the link engineering codes). From our survey and contacts
with the strategic C³ technical community, we observed that adequate tools—network simulations and communications-link codes—are available and that, for the most part, the analysis teams are using the latest versions available. With regard to databases and information sharing, there was, at the outset of this effort, a dearth of communication between members of the community. A series of Joint Staff workshops, however, has improved this situation markedly.

From this survey, we also conclude that a standard database should be made available to the strategic C³ analysis community and that it should be in common format. The potential users of this database voiced repeatedly in our workshops that the SSCF Database would be useful to them. (It should be acknowledged, however, that—to no one’s surprise—this enthusiasm was not translated into a corresponding willingness to pay for the enterprise.)

The approach recommended here also has application in the theater and tactical nuclear weapon C³ arena and in other C³ and non-C³ databases involving multiple users.

RECOMMENDATIONS

Although the strategic C³ analysis community would clearly benefit from the development, maintenance, and use of a standard SSCF Database, today’s funding environment precludes immediate implementation of the full common-format approach described here. We recommend initiating a deliberate building block approach:

- **Block 1**: Complete the development and formal specification of the SSCF Database. This does not depend on the development of a common format database or the support software.
- **Block 2**: Develop specifications for the procurement of the common-format database and support software.
- **Block 3**: Initiate an austere version of the common-format database and support-software development, as described in Sec. 5.
- **Block 4**: Develop the common-format database and support software package along the lines described in Sec. 4 and 5.

Each block will build upon its predecessor, and downstream blocks can be implemented, redirected, delayed, or dropped based on the best information available at the time.

The **Block 1** development of a standard (but not “automated”) database can be carried out (or not) regardless of what is done with the common-format database or the support-software development. This effort could consist of as little as a J-6 letter designating a document, such as the Strategic System Description, as the standard reference and a particular WAAM data tape as the current approved standard database. This need entail no new funding.

A draft specification effort (Block 2) could be developed *ad hoc* by several members of the Strategic C³ Working Group. Then, if the decision were made for full development of the support software for the common-format database, this draft specification could be adapted by the appropriate government agency (presumably DISA) at the time of the initiation of a request for proposal. This, too entails no new funding.

**Block 3** offers a way to keep the effort moving forward at a pace and level of effort considered appropriate at the time (e.g., 1992 or 1993). **Block 3** could be the final stage, a first cut at
Block 4, or skipped entirely, initiating Block 4 without first developing the Block 3 austere version. Implementation of Block 4 should include the development of the SSCF Database.

The benefits of this action will accrue to all of the users; therefore, the burden should be shared. If the Joint Staff proceeds deliberately, and through the Working Group, enthusiasm for this sharing can probably be developed. What is more, unlike a hardware production, proceeding at a deliberate pace will actually be more economical than conducting a faster-paced effort. Developing the preliminary support-software specifications as a Strategic C³ Working Group ad hoc effort would be a case in point, and we recommend that this action be initiated.
Appendix A
PROJECT ADMINISTRATION

In Sec. 4, we discussed database administration from a "computer science" viewpoint, dealing with a number of technical implementation aspects of the SSCF Database problem. Sections 4 and 5 elaborated on the recommended concepts—a keyword language format for the database; software tools for production, maintenance, and application; and data set production processes. In this appendix, we will look at the administration of the SSCF Database from the Joint Staff Action Officer's viewpoint.

We begin by addressing some fundamental aspects of the program manager's responsibilities—those associated with resource allocation. Will the utility of the SSCF Database justify the resources required to create, maintain, and distribute it? What are the benefits? What are the costs and risks? Can the cost be distributed equitably over the user community? Although some of the answers to these questions are still tentative, we will address all of them. The following subsection treats the important oversight, security, and control aspects of the SSCF Database once it is in use. We next address the database distribution process from the standpoint of support requirements or capabilities of the various organizations involved. The final subsection contains our recommendations concerning the Joint Staff/J-6 management of this important enterprise.

SOME COST-VERSUS-BENEFIT CONSIDERATIONS

At the First Joint Staff/RAND Strategic C^3 Workshop, it was made clear that—at some point—the management at each member organization would have to make resource allocation decisions involving the database. For many organizations, receiving this database will simply save time and staffing resources and will improve the accuracy, precision, and credibility of their study outputs. Other organizations, however, would also be involved in contributing to the database. Two concerns—the sensitivity of the database and the effort required to contribute to the database—must be considered. First, the classification of the SSCF Database will mandate careful dissemination and strict security procedures. Second, a database that is perfectly adequate for internal purposes often contains legitimate shortcuts and valid analytic workarounds. Before sending such data to the SSCF Database manager, therefore, a donor organization would probably want to clean up and verify the database, which is not always a trivial activity.

The pros and cons of the database were mentioned periodically throughout the first Workshop, but not in an organized or carefully articulated way. The attendees of the second Workshop reviewed the needs and benefits, as well as the costs and risks associated with creating and distributing the database. In this section, we will briefly recount both sides of the ledger with as little bias as we can, acknowledging that we have come to believe that the value of the standardized database—as a baseline or point of departure for in-depth analyses of Strategic C^3 issues—makes its creation and maintenance a valid, useful, and economical concept.
POTENTIAL BENEFITS

First, let us recount the benefits. As noted in the 1984 Workshop and reiterated in the first 1987 Workshop:

- Many organizations are conducting sophisticated, vital analyses, but on important occasions their results have differed markedly for reasons that have not been readily apparent or explained. We need to eliminate this confusion and increase our understanding of these problems when they arise.¹
- We need to maintain, and even increase, our credibility with decisionmakers.
- We would also like to decrease community workload.

In addition, the modern, rapidly changing computer environment makes possible an immense benefit in terms of improved accuracy, fidelity, and efficiency. Higher quality studies really can be produced in less time and at lower cost than was possible just a few years ago. Faster computers with much larger storage, better codes, and better support software allow small analytical teams to perform sophisticated studies with less likelihood of the software development horror stories of 10 to 15 years ago. Now it is also easier—at least from the computer science point of view—to assimilate, update, verify, and modify a database. The data access, acquisition, and conversion processes are still required, but modern database-management systems (DBMSs) and approaches can make the manipulation of databases much easier for an analyst. The logic of taking advantage of technological advances and existing electronically stored databases is as inescapable as that of never going back to large-scale typing once you have learned word processing.

The Working Group activity that the Joint Staff initiated as part of this effort has already benefited several CINC and defense agency efforts. In addition, the exchange of ideas contributes to both the corporate memory of the organizations involved and the professional growth of the participating individuals.

POTENTIAL COSTS AND RISKS

The above benefits are not without certain costs, however. Developing and maintaining the SSCF Database, even in a modern computer environment, is not trivial. We scoped the development effort in Sec. 5; we discuss the maintenance and distribution activities below.

In addition, this changing computer environment makes it easier for serious security problems to develop and can exacerbate the seriousness of security problems. A great deal of highly classified data can be compromised if a large electronic database is lost or exposed.

Others have noted the "black-box syndrome," wherein an analyst or analyst team could take a user-friendly computer program, plug a large externally generated database in it, and conduct a lot of "number crunching," without really understanding the significance of their computer inputs and outputs. Neophyte organizations could cause problems if they used an

¹As pointed out by the CINCSAC representative at the Fourth Workshop, it makes sense to assume that, in the future, different studies will produce apparently or actually conflicting results. The existence of the database and the efforts of the Joint Staff Strategic C5 Working Group can cut down the number of conflicts and allow us to more quickly and better resolve these conflicts when they occur.
"approved model" and an "approved database" to come up with unrealistic results, interpretations, and/or claims because they did not understand either the model or the inputs. Some feel that the answer to the black-box syndrome is simply to force each group to build its own database, presumably learning it in the process. Others recognize the syndrome, but regard the duplications of effort and extra expenditure of government funds on those resources as being wasteful, if not prohibitive. Furthermore, each of these "home grown" databases will almost certainly differ from the others—in significant ways and to a significant degree, leading to the very kind of problem that was the genesis of this effort some seven or eight years ago. The latter group sees the "home grown" cure as being worse than the disease, that there are better ways to combat the black-box syndrome.

Several additional concerns were noted:

- Pressure for the release of this database, once its existence was known in the community, to organizations with only a marginal need to know
- Increased odds of security compromises, simply based on the proliferation problem
- "Fairness" issues, when an incumbent contractor would have this database and competitors for an upcoming source selection would not.

Although these concerns have historical validity, we felt that they are amenable to control. The Working Group approach described below offers a way for individual organizations to be relieved of the pressure to release their own databases. If the donor organizations choose, they can simply refer "marginal" requesters to the Joint Staff Strategic C³ Working Group. With respect to security, it is our belief that the methods recommended here will reduce—not increase—the likelihood of security compromises. Finally, fairness of the distribution of databases would probably be enhanced by the Working Group process, wherein a range of viewpoints would be represented in the distribution decisions.

These problems are not new. They all exist today. The value of the database simply exacerbates them. The upshot of these security and management concerns is a desire for rigid security and control procedures, which are not themselves without cost. We will address database security and control in the next section.

DATABASE ASSESSMENT

The SSCF Database is intended to provide a valid baseline description of the current Strategic C³ network for use by specific government organizations. These organizations must, individually and collectively, benefit from the development, maintenance, and use of the SSCF Database.

Utility criteria include operating efficiency, fidelity of analysis, level of development effort, top-level decisionmaking, and capital investment. Let us consider each criterion in turn:

Operating efficiency: Most of the respondents to our survey indicated that they were the sole builders of their databases and all but one indicated that they at least partially build theirs. About 18 percent of the study resources of the Working Group members was devoted to database acquisition and assimilation. The efficiency of these analysis processes will improve substantially if they can be provided with a correct, validated baseline database.
Based on the experiences of the principal investigator of this study and on conversations with other analysts, getting a debugged, valid baseline is usually the biggest hurdle; the excursion cases fall more or less into place after that.

Fidelity of the analysis: The results of these organizations will likewise be improved by the verification and validation (V&V) aspects of the standard database. The different excursions made by the various research organizations will not benefit from this V&V, but most of the ground truth is represented by that baseline—not by the variation represented by, say, a new transmitter and receiver combination possession by 15 percent of the network.

Development effort: The database management task by the designated organization should not require excessive effort. The designated database manager will, in all likelihood, be a DISA organization that is already responsible for a large part—if not all—of this database. The additional responsibilities will be significant, but the effort required to carry them out need not be great. One problem that comes to mind, however, is that when a number of the member organizations begin to rely on this product, the impact of major technical errors or program delays will be multiplied accordingly.

The top-level decisionmaking efforts concerning the use and distribution of the database should only become significant for the managers involved when serious problems arise. The problem can easily be rendered self-regulating by management, i.e., by limiting the number of times the "executive committee" will meet. This might result in long waits for the database and/or frustrated clients, but that very fact would be testimony to the recognized value of the database.

Capital investment: The capital costs should be low relative to the value of the efficiency and fidelity improvements realized. This criterion will be difficult to satisfy and more difficult to prove a priori. The large cost for the standard, common-format database ($1M) is not likely to be recovered. The austere program ($250K) could be amortized over three or four years if ten study teams each cut their data acquisition, assimilation, and debugging efforts by some 20 percent. The latest survey responses concerning efforts by the research organizations suggest that our earlier predictions may have been optimistic.

Uniformity restrictions of hardware, software, and database management tools would not be imposed on the various users.

We see that most, but not all, of these criteria can be met.

DATABASE OVERSIGHT, SECURITY, AND CONTROL

The database oversight, security, and control issues are interwoven with both the technical implementation issues and project management issues, such as cost. Thus, the approach recommended here is as much an administration and management tool for Joint Staff/J6 coordination of the database effort as it is a technical implementation or security plan.

We recommend that the oversight, security, and control of the SSCP Database activities be managed through an Joint Staff–chaired Strategic C3 Working Group. The seven Joint Staff/RAND Workshops that have taken place in the last few years have created a de facto Working Group, which can be used as a management tool for further efforts. We recommend
that Joint Staff/J-6 develop implementing directives for authorization purposes, but use the Working Group as a management tool.

The Working Group

The Working Group should deliberate policy, delegating to technical panels or member organizations appropriate actions and responsibilities. Some subset, or "steering group," could be used if the Working Group proper turned out to be too large for effective decision making. A postulated membership of this Working Group is shown in Table A.1. All of these organizations have been represented in Strategic C^3 Working Group meetings in the last year. Some efforts should be assigned or delegated to specific organizations, whereas a "technical panel" approach might be more appropriate for others. Implementation directives from the Joint Staff could be desirable in some cases, but would not be necessary in many others. The resulting three-part management approach—Working Group, technical panels, delegation to or volunteer action by the CINCs, Services, or defense agencies—would meet the needs of all the organizations involved. Joint Staff leadership would be combined with cooperative working relationships. This machinery will be useful for more (and larger) problems than management of the database per se.

The following discussion addresses both security and efficiency issues. The potential character of the Working Group would be to

- Identify and examine issues
- Set Working Group policy and agenda
- Review and compare results.

Table A.1

<table>
<thead>
<tr>
<th>Member</th>
<th>Strategic C^3 Working Group</th>
<th>Technical Panels</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Database</td>
<td>Codes</td>
</tr>
<tr>
<td>Joint Staff/J6</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Joint Staff/J8</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Joint Staff/J36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OSD/PA&amp;E</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>ASD/C^3I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JSTPS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPACE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AFCSA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOF-941D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAND</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>Mitre</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NOTES: 1=CHAIR, 2=CO-CHAIR, *=MEMBER, S=SUPPORT
Some suggested operational guidelines are as follows:

- The group should consist of small technical panels (6 to 8 people).
- The group should not become a time-consuming task for any one person.
- Meetings should be held only as required (2 to 3 per year).
- Informal communications should be relatively frequent.
- J-6 should spell out the group's responsibilities.

The Working Group is already carrying out much of this charter without, we believe, imposing a significant workload on any individual member or organization. We believe that the Strategic C³ Working Group activity should not be a major portion of any of the member's working schedules. The Working Group efforts should normally be a logical part or a minor extension of normal activities. The Working Group effort should not appear to be a burdensome additional duty unrelated to the mission of the user or the participating organizations. We see much of the Working Group effort as being similar to the Joint Staff/J8 Red Planning Board development, control, and dissemination of the Red Integrated Strategic Operations Plan (RISOP). All of these organizations are currently participating in working group activities. The supporting organizations—the Mitre Corporation and RAND—will pull together and co-host the workshops, publish and distribute the proceedings, and provide a continuity of corporate memory for the Working Group (and perhaps some of the panels).

**Working Group Actions**

We suggest that several panels could be formed to carry out some of the Working Group activities. We recommend that a single panel be responsible for determining database content and for control of that database. The panel's responsibilities would include

- Database development
  - Sources
  - Database compatibility
- Database management and control
  - DBM procedures
  - DBMS development
- Release approval

As the list indicates, the members, who would be responsible for control procedures and decisions involving the release of the Strategic C³ database to other entities, must have intimate knowledge of the sources and content of the database. This is why we recommend a single panel.

First, this panel should identify the appropriate data sources for ensuring database compatibility and should identify database requirements involving newly selected codes. This panel should also be responsible for reviewing database management procedures and DBMS development. The panel should process requests and obtain a release approval in accordance with procedures developed or approved by the Working Group proper. Like the Working Group it-
self, the database control panel should be in a position to delegate to its members, who can—
where appropriate—use contractor support to get a job done.

The role of the codes-and-methodology panel would consist of

- Code selection and evaluation
- Database compatibility
  - Requirements to database panel
- Methodology evaluation
  - Applications
  - MOE
  - Ground rules and assumptions
  - Alternative approaches
- Analysis products evaluation
  - Validation and verification, comparisons, and tests
  - Black-box syndrome.

This panel should assist the Strategic C³ Working Group in the evaluation of various
Strategic C³ codes and the selection of those to be monitored, and perhaps, supported by the
standard database. The panel should review the various V&V efforts, code comparisons, and
tests. They should address database compatibility issues, levying requirements on the
database panel as appropriate.

The suggested functions of the security panel are as follows:

- SSCP Database security and control procedures document development
- Database access approval
- Computer and facility security evaluation
- Review of outputs and results of receivers of the SSCP Database
- User security organization coordination.

This panel should be made up of professional security personnel from the Working Group or-
ganizations. This panel should establish communications with the security organization
within each of the government and contractor entities involved. The philosophy of the secu-
ritv measures should be to remove incentives to cut corners and to make violations less
likely, less rewarding, more trouble, and more risky. The panel should also review the out-
puts and results of the analysis conducted by the receivers of the C³ database. This review
should be conducted for the purpose of ascertaining whether appropriate use of and restric-
tions on data have been implemented.

The threat and scenario panel effort involves, in addition to the Working Group, the intelli-
gence community. The effort should encompass threat definition, as well as threat evalu-
atations and threat scenarios, such as the RISOP. The panel might also address, if required, the
development of new (non-RISOP) scenarios. For example, a "plain vanilla" attack, less sensi-
tive than a RISOP, might be developed as a diagnostic tool to accompany a standard
database.
The security panel should document C^3 database security and control procedures. It should either approve database access or—more likely—support the approval process of the Working Group proper, depending on procedures decided on by the Working Group. The security panel should see that each requesting and user organization’s computer and facility security is adequate and appropriate for the storage and use of the database at the classification level prescribed for that facility.

The C^3 database security and control procedures document should set objectives, define terms, contain or refer to a classification guide, and characterize the various classes of facilities that the analysis groups might use and that might therefore form repositories for the database. Most importantly, the document should specify procedures for generation, modification, documentation, and transmission of the database. The following is a suggested outline:

1. Introduction and Purpose
2. Objectives of this Policy and Their Rationale
3. Database Definition and Description
   - Definition
   - Description
   - Sources
4. Terms of Reference
   - Joint Staff C^3 Working Group
5. Responsibilities
   - Joint Staff C^3 Working Group
   - DISA
   - DNA
   - CINCs
   - Services
   - Database Users
6. Database Procedures
   - Generation
   - Transmission
   - Use
   - Modification
   - Documentation
7. Classification Guide

APP. A References
   - "DoD Trusted Computer System Evaluation Criteria"
• Security Classification Guide(s)
• Other appropriate DoD regulations

APP. B  User Team and Facility Characterization; e.g.,
• Central Repository/SCIF (e.g., DISA)
• Analysis/SCIF
• Analysis/screen room
• Analysis/secret facility
• Encryption capability
• Shared versus stand-alone computation

DATABASE DISTRIBUTION SUPPORT

The resources required to distribute strategic C³ data to the modeling community largely depend on how the Database manager and Database users divide up the Database distribution functions. These functions consist of database maintenance, data selection, semantic interpretation of the data, and formatting the information for particular models. The goal of database maintenance is the creation of a collection of checked and verified information that accurately characterizes either the current or planned configurations of the strategic communication systems. The data selection process involves selecting a subset of the total database that corresponds to the information needs of a particular user and his approved need to know with respect to the information. The semantic interpretation step provides for converting the units and concepts of the Joint Staff C³ representation into those used by the user model. The formatting step consists of mapping the semantic information into the field and record formats that the model input reader can process.

There is substantial leeway in dividing up the support functions between the user and the database manager. At a minimum, the Database manager must maintain the SSCF Database to ensure the uniformity of analytic assumptions; this is the chief objective of the database project. Also, the Database manager should be in charge of preparing and selecting the data provided to various users to guarantee that the need-to-know requirements for data access are scrupulously enforced. With the other options, there are substantial advantages in transferring the formatting and interpretation functions to the user; these advantages, of course, must be traded off against a substantial loss of control over the applications to which the user puts the information.

As pointed out in Sec. 4, three database distribution options find sensible software and hardware support. The options also correspond to differing levels of support. The first—the centralized option—simply imposes an unreasonable workload on the Database manager and is therefore not recommended. The second option—DBMS distribution—would require imposition of a database system and is therefore not recommended. We recommend the third option—data distribution—which would be best for efficient transactions with experienced users.
JOINT STAFF MANAGEMENT OF THE SSCF DATABASE

In summary, it is our view (which, we believe, is shared by most of the Strategic C³ Working Group) that the utility of the approach recommended here has been established. The SSCF Database should be developed, maintained, and distributed as recommended here. DISA is a logical central organization for implementation, although DNA OPNAV, AFCSA, CINCSAC and CINCSpace should all be actively involved. Finally, we believe that the Working Group, which has become a real, albeit informal, organization, is the key to the effective management of this effort.

First, the Working Group is made up of people already working in this area. Their organizations will each benefit not only from the existence of the SSCF Database but also from the Working Group activities themselves. Much of this activity will simply be a different way of doing something they already have to do. Thus, the shared interests and the mutual benefits of the enterprise suggest that the efforts and costs involved can likewise be shared. In addition, the Joint Staff should delegate responsibility to one organization (we suggest DISA) to be the Database manager. We suggest also that DISA be assigned responsibility for developing the common-format database software capabilities identified and initiated in this RAND prototype effort. Finally, the Joint Staff should solicit DNA, CINC, and Service cooperation concerning compatibility with user codes, such as DNA's WECOM and SIMBAL codes, the Navy's NSCS/SCAP, and the Air Force's STRATCAM.

The keys, then, to Joint Staff management of the SSCF Database are that

- This is not a new responsibility, whether it is performed in the manner recommended here or in some other fashion; federal laws [26] and DoD directives [27–29] mandate such an effort.

- Activities should be assigned or delegated to member organizations (e.g., DISA, NOP-941D, AFCSA) who will share in the benefits and who will also usually need to do something like the delegated activity as part of their normal functions.

- The Working Group/Panel effort must not be very time-consuming, and the individuals involved should perceive the Working Group activities as something contributing to their individual professional growth and opportunities to establish contacts with fellow C³ specialists and organizations.

- The organizations involved should perceive the Working Group activities as something contributing to their organizational missions and charters. OSD/Joint Staff tasking may also accompany specific activities.

The Working Group is in place, motivated, and currently addressing a number of aspects of this problem. The enthusiasm of the Working Group members and organization has suggested willing, active participation and a high probability of success if J-6 chooses to use the Group as a management tool.

The Working Group organization and efforts described here appear to be imposing and to represent an additional burden. This is not the case. We are simply suggesting a way of organizing, coordinating, and managing efforts that—for the most part—already take place.
Appendix B

A FORMAL SYNTAX FOR THE SSCF DATABASE

This appendix describes precisely the structure of database records. The reason for such a description is that while computers require such precision, human readability should not suffer in consequence. This description can be given to a formal tool, such as "lex," to generate code that automatically recognizes keywords, tokens, and other grammatical structures. This simplifies the parsing problem for the Database and helps render the translation process more orderly.

GRAMMAR.

Using the primitive vocabulary in Table B.1, the grammar of the keyword database may be specified succinctly in the Backus-Naur notation. A term in braces is repeated 0 or more times, a term in brackets is an optional term. Words in all capital letters denote literal text within the Database. For example, see Table B.2.

The number, format, units, and meaning of the data values (if any) following a keyword are uniquely determined by that KEYWORD and defined in a keyword glossary which serves as a parsing and interpretation control table. The keyword glossary also defines a child list of each keyword, which implicitly defines an ancestral hierarchy for each keyword. An example of such a glossary appears in Table 4.4.

<p>| Table B.1  |
| Primitive Vocabulary |
| primary_keyword | The highest category and lead keyword in a record, roughly a one-word description of what the record is about. |
| secondary_keyword | A second or lower-order category keyword, roughly what the following part of the record is about. |
| name | A key identifier by which the particular instance of a keyword class of object is identified within the database. This will generally be mnemonically suggestive of the external-world name of the particular object but may, if desired, be supplemented by a more extensive full name in the following values section of the record. |
| value | A predefined property of the particular instance of keyword. |
| classification_level | The security classification of a record. |
| source_reference | The source of the value data within a record. |
| responsible_person | The knowledgeable person to contact regarding exact meaning or derivation of a value datum within the record. |
| date | The date the record was entered in the database. |</p>
<table>
<thead>
<tr>
<th>Backus-Naur Notation</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;primary_key_pair&gt;</code> := <code>&lt;primary_keyword&gt; &lt;name&gt;</code></td>
<td>A primary/secondary key_pair consists of a primary/secondary “keyword” plus a unique (within the database) “name” string, which is the local name for reference within the database.</td>
</tr>
<tr>
<td><code>&lt;secondary_key_pair&gt;</code> := <code>&lt;secondary_keyword&gt; &lt;name&gt;</code></td>
<td>A value_list consists of zero or more values.</td>
</tr>
<tr>
<td><code>&lt;value_list&gt;</code> := <code>{&lt;values&gt;}</code></td>
<td>A primary keyword_phrase consists of a primary key_pair followed by a value_list.</td>
</tr>
<tr>
<td><code>&lt;primary_keyword_phrase&gt;</code> := <code>&lt;primary_key_pair&gt; &lt;value_list&gt;</code></td>
<td>A secondary keyword_phrase consists of either a secondary key_pair followed by a value_list or a secondary key_pair followed by the string SUB.</td>
</tr>
<tr>
<td><code>&lt;secondary_keyword_phrase&gt;</code> := <code>&lt;secondary_key_pair&gt; &lt;value_list&gt;</code></td>
<td>Reference_data consists of a fixed-format, record-by-record, self-punctuating notation of the classification level, source, responsible person, and date on which the data were entered in the data base, and ending in the string EOR.</td>
</tr>
<tr>
<td><code>&lt;reference_data&gt;</code> := <code>CLASS &lt;classification_level&gt; SOURCE &lt;source_reference&gt; PERSON &lt;responsible_person&gt; DATE &lt;date&gt; EOR</code></td>
<td>A record consists of one or more keyword phrases followed by a reference_data.</td>
</tr>
<tr>
<td><code>&lt;record&gt;</code> := <code>&lt;primary_keyword_phrase&gt;</code> [<code>&lt;secondary_keyword_phrase&gt;</code>] <code>&lt;reference_data&gt;</code></td>
<td>A sub_record consists of the string SUB followed by a key_pair followed by a value_list followed by a reference_data.</td>
</tr>
<tr>
<td><code>&lt;sub_record&gt;</code> := <code>SUB &lt;secondary_key_pair&gt; &lt;value_list&gt;</code></td>
<td></td>
</tr>
</tbody>
</table>
The keywords and their associated value lists and child lists must obey certain ground rules, as follows:

1. Keywords will only be used to specify generic concepts, of which there are generally several instances in the database.

2. Designation of "which" particular instance or what "type" of instance of a keyword will always be denoted by the name string (not by keyword). The first value following each keyword must always be the name string that is used within the database to distinguish the particular instance of KEYWORD and to establish interrecord relationships. This name value will generally be mnemonically suggestive of the external-world name of the object, but may be supplemented by longer or synonym name values, as necessary, within the subsequent value list.

3. Keywords will be of primary and secondary type. Each record must start with a primary keyword, and primary keywords can only start a record. Each secondary keyword shall appear in the child list of one or more parent keywords, as defined in the associated keyword glossary.

4. Under this arrangement a keyword phrase may only be followed by one of its child keywords or sibling keywords, i.e., other children of the same parent, or by reference data. Similarly, a KEYWORD may only be preceded by a keyword phrase of its parent or of a sibling or the term SUB.

5. This means that, while there is not necessarily a unique keyword chain, there is at least a definable, reasonably small number of possible ancestral chains of "keypairs" leading to the location of each data item. These ancestral chains can be extracted by the keyword interpreter and are the key to both synthesis and analysis of keyword-pair sequences to be searched for, leading to each data item in the database.

6. A special type of database record is defined, which will be called a subrecord, after the concept of the subroutine, to which it has some similarities. A subrecord starts with the term SUB followed by a secondary keypair. Thereafter, it follows all the glossary-defined syntax for the keyword, including values list, and reference data.

7. A subrecord is in effect "called" from a regular record by the special keyword phrase consisting of the keyword pair sequence, followed by the indicator term SUB, which is interpreted as a directive to get the subrecord starting with the same keyword pair and substitute its properties for the present instance.

These rules would indeed be cumbersome if the user analyst had to apply them directly. However, the keyword glossary manager will apply them transparently, forcing the resulting glossary to comply. Since this will not require interaction with the analyst, he or she does not even need to be aware of the rules or their operation.
REFERENCES


