

**FINAL**  
**Road Map To Conversion**  
  
for the  
**OSD CALS IWSDB Project**  
  
An MVP Joint Venture

**January 18, 1995**

Submitted by  
ManTech International Corporation  
Technology Applications Operations Center  
1313 Locust Avenue  
Fairmont, West Virginia

In support of  
Contract DAAB10-89-D-0503  
And in compliance with  
CDRL Sequence Number A021

SOW Numbers 3.0, 3.5



---

Robert S. Kidwell

---

Jack G. Richman

Technical Director

OSD CALS

ManTech International  
Corp.  
OSD CALS Project  
Manager

## **PREAMBLE**

The enclosed report covers the very dynamic and complex topic of the Interactive Electronic Technical Manual (IETM). The IETM concept supports the entire life cycle of that fundamental and key Department of Defense (DoD) document, the Technical Manual (TM). Known as a compound document due to the high integration of textual, tabular, and graphical illustration content; the Technical Manual (Technical Order) represents one of the highest Return On Investment (ROI) elements of the CALS initiative.

Often, the U.S. Navy for example, highlights the fact that an AEGIS Cruiser requires a storage facility of approximately 23.5 tons for Technical Manuals and Engineering Drawing Documentation. By converting this paper warehouse to a form of electronic media under the CALS initiative; the Navy would recover precious ship spaces, provide integrated and rapid access to the contents of Technical Manuals plus facilitate current changes and updates now not possible with paper. All, at highly reduced life cycle cost factors.

Although the report primarily focuses on issues and direction supporting a "Roadmap For Conversion to IETM(s)," the report does indeed provide an excellent current view of the DoD use, commercial products available, and critical issues regarding standards and technology of this highly important discipline known as "IETM(s)."

## EXECUTIVE SUMMARY

Converting a paper or electronic technical manual of any kind to a format which complies with the IETM specifications is a problem with many possible solutions. The difficulty is that any meaningful procedure or methodology for conversion of legacy data has a baseline requirement that both a source (conversion from) and a destination (conversion to) must be specified in some manner. The problem with conversion, Technical Manuals (TMs), Electronic Technical Manuals (ETMs), and Interactive Electronic Technical Manuals (IETMs), is that neither the source nor the destination can be specified in a way that is both precise and universally applicable.

The task that this report is the subject of focused on the investigation of the basic conversion problem just mentioned. The investigation involved a wide range of research, analysis, technical evaluation(s), project assessments, and review of the current status of the IETM industry. One of the objectives of this task was to identify efficient methodologies and corresponding Commercial Off The Shelf (COTS) products that facilitate the rapid conversion of older weapons systems manuals (whether they are paper or electronic) to the ETM and/or IETM environment.

In short, performing the above items required gathering a great deal of information on how ETMs and IETMs are being made and how they are being used in both DoD and Industry. In summary, the following work was performed:

- Developed and defined a precise set of definitions for ETM/IETM data types.
- Researched and assessed the data handling methodology of the JCALS Program.
- Evaluated COTS software relevant to the conversion, creation, and presentation of technical manual content.
- Researched and assessed conversion methodologies for the conversion of TM information of varying formats to a format that is consistent with the IETM specifications.
- Evaluated the IETM specifications (MIL-M-87268, MIL-D-87269, MIL-Q-87270) with the current set of CALS standards for compatibility factors for creating and sustaining IETMs.

In regards to the first item in the above list, one of the principal considerations of this task was to look at the conversion of Type B and B+ data to Type C data. These data types are defined in Section 2, but in very general terms they can be referred to as follows:

- Type "B" Electronic Page Turner,
- Type "B+" HyperText Electronic Navigation, and
- Type "C" Interactive Electronic Page Assimilation.

The intent was to insure that the converted Type C data would be fully compliant with the IETM specifications. In the course of this project, however, it became apparent that the Type B and B+ data types are not particularly well defined for the purposes of legacy data conversion because they are based on desirable electronic technical manual characteristics rather than on any

particular technical manual preparation specification or standard. Furthermore, there are no restrictions on the authoring and presentation systems that may be used in conjunction with the Type B or B+ data. Type C data is also problematic because it is based on the IETM specifications which are themselves not particularly well defined for establishing a general legacy data conversion methodology.

The problems surrounding the definitions of Types B, B+, and C affected this task in two important ways. First of all, it became necessary to redefine these data types. The redefined data types are termed "TM data types" (rather than ETM data types). To avoid ambiguities with existing programs they were also renamed to Types MA, MA+, MB, MB+, MC, and MC+. Secondly, it became necessary to concentrate on the information and information structures within technical manuals in a general fashion which would be broadly applicable (rather than on just the information in paper or electronic technical manuals prepared according to one particular set of standards or specifications).

From a conversion standpoint, it must be the case that technical manuals of any form (paper or electronic) convey the same information. The only real difference is the way in which the data (technical manual information) is presented to the user (maintenance technician). Currently, both paper and electronic technical manuals exist in a wide variety of formats.

Paper manuals may be in the form of highly structured system/task-oriented work packages or they may be in a somewhat less structured chapter-oriented format. Some paper manuals may contain part information, maintenance tasks, and troubleshooting instructions all within a single manual whereas others may contain only maintenance tasks, with separate manuals for both part information and troubleshooting instructions. Furthermore, some paper manuals are constructed for single maintenance levels (i.e. Organizational) whereas others contain instructions for multiple maintenance levels (i.e. Organizational and Intermediate, etc.).

Electronic manuals also occur in a wide variety of formats. In terms of the underlying data, those formats are generally not compatible among individual implementations. This is true even among those implementations that are said to comply with the IETM specifications. Furthermore, this is true even when all of the underlying data exists in CALS-compliant formats.

Regardless of the form of the source data, conversion of a technical manual from one format to another requires a comprehensive understanding of the information structures that will be encountered. The information structures of particular interest are given in Section 4 and consist of the overall structure of the manual, procedural (task) data, non-procedural data, troubleshooting data, and graphical data. Important issues and methodologies associated with conversion of these information structures are also given in Section 4.

One of the important considerations during this task was to study the data handling methodology of the JCALS system and how JCALS could be used with the converted ETM/IETM data. The capabilities of the JCALS system are analyzed in Section 3. The analysis revealed that due to the current state of ETMs and the IETM specifications, the issue of interfacing converted ETM data with the JCALS system can really only be resolved on a case by case basis. Ultimately, JCALS

will be treated as a management system and it is envisioned that ETMs will be managed like other types of TMs. With this, the results given in this report show that this will require the development of independent interfaces between the JCALS system and each ETM authoring/presentation system that it is to be used with.

It is noted in Section 6 and repeated here that, currently, embarking on any project which involves creation of an ETM or creation of ETM data can be difficult. One of the reasons it is difficult is due to the fact that there is not a single authority or office within DoD with a clear mandate to provide the guidance which is needed. Guidance is needed particularly on:

- ETM Type or Class selection (with BCA or CBA data to back it up).
- ETM authoring system selection.
- ETM presentation system selection.
- DTD selection (when applicable).
- Authoring, conversion, and presentation of troubleshooting information (appropriate for selected Type/Class and system).
- Authoring, conversion, and presentation of procedural maintenance tasks (appropriate for selected Type/Class and system).
- Authoring, conversion, and presentation of part information (appropriate for selected Type/Class and system).
- Authoring, conversion, and presentation of graphics (appropriate for selected Type/Class and system).
- Validation guidelines and procedures for both authored and converted ETM data of all approved Types or Classes.

The findings given in this report show that a lot of important issues must be considered before converting legacy paper or electronic TM data to ETM data of any type beyond that of an electronic page-turner. The only guidelines which currently exist for creation of ETM data are the IETM specifications (MIL-M-87268, MIL-D-87269, and MIL-Q-87270). This current set of IETM specifications is not sufficient for assuring interoperability among IETM implementations. Also, creating an IETM database or a DTD by following the Content Data Model in MIL-D-87269 is not sufficient for assuring that IETM systems will be capable of exchanging their data without an independently developed intermediate translation step. Furthermore, following the IETM specifications does not even guarantee that two IETM systems which can both import and export their data using the same generic layer and content-specific layer DTDs will be capable of exchanging their data without an independently developed intermediate translation step. Moreover, these intermediate translation steps may or may not be entirely automatable.

## Table of Contents

<b>EXECUTIVE SUMMARY</b> .....	<b>iii</b>
<b>LIST OF TABLES</b> .....	<b>vii</b>
<b>LIST OF FIGURES</b> .....	<b>viii</b>
<b>1.0 INTRODUCTION</b> .....	<b>1</b>
1.1 Define Data Types and Assess JCALS .....	1
1.2 Sub-Task Assignments.....	1
1.3 Format of Report .....	1
<b>2.0 DEFINING AND CLASSIFYING ELECTRONIC TECHNICAL MANUALS</b> .....	<b>3</b>
2.1 TM(s) ETM(s) and IETM(s) Defined.....	3
2.2 Baseline Requirement.....	4
2.3 Precise Data Types Defined .....	4
2.4 Multiple Data Type Definitions .....	9
2.4.1 Army and Air Force Data Types .....	10
2.4.2 Navy Data Types.....	12
2.4.3 JCALS Data Types.....	13
2.5 Defining CALS-Compliant ETM/IETM .....	14
<b>3.0 JCALS ASSESSMENT</b> .....	<b>16</b>
3.1 JCALS Overview .....	16
3.1.1 History Of JCALS .....	16
3.1.2 Current JCALS System Description.....	17
3.1.3 Current JCALS System Functional Overview .....	17
3.1.3.1 Application Process .....	17
3.1.3.1.1 Product Generation Tools.....	18
3.1.3.1.1.1 Authoring/Editing .....	18
3.1.3.1.1.2 Viewing/Browsing .....	19
3.1.3.1.2 Design Tools .....	19
3.1.3.1.3 Analysis Tools.....	19
3.1.3.1.4 Decision Support Tools .....	19
3.1.3.2 Implementation .....	20
3.1.3.3 Work Management .....	20
3.1.3.4 Information Management .....	20
3.1.3.5 System Administration .....	21
3.1.3.6 User Support .....	21
3.1.3.7 System Infrastructure Support.....	21
3.2 JCALS Data Handling Methodology .....	21
3.2.1 Author/Editor.....	22
3.2.1.1 SGML Text .....	22
3.2.1.2 Graphics .....	23
3.2.1.3 LSAR Data.....	23
3.2.2 Data Management .....	23
3.2.2.1 Cataloging .....	23
3.2.2.2 GDMS.....	23

3.2.2.3 Official DBMS.....	24
3.2.3 Data Base.....	24
3.2.4 Display.....	24
3.2.4.1 Document Publishing.....	24
3.2.4.2 Viewer/Browser.....	24
3.3 JCALS Interface Requirements.....	25
3.4 ETM Interface Assessment.....	26
<b>4.0 ROADMAP TO CONVERSION.....</b>	<b>27</b>
4.1 A Look at the Composition of Technical Manuals.....	28
4.2 Conversion Issues with Types MC and MC+ Data.....	31
4.3 Technical Manual Information and Type MC Data.....	33
4.4 Information Structures within Technical Manuals.....	36
4.5 Mapping TM Information Structures into a Type MC Format.....	38
4.5.1 Establishing the Initial Structure.....	39
4.5.2 Conversion of Non-Procedural Data.....	40
4.5.3 Conversion of Procedural Data.....	43
4.5.4 Conversion of Troubleshooting Information.....	45
4.5.5 Conversion of Graphics.....	47
4.5.6 Conversion Automation.....	51
4.6 Functional Capabilities of Authoring and Presentation Systems.....	53
4.6.1 Lockheed Fort Worth Presentation and Authoring System.....	53
4.6.2 The World Wide Web.....	54
4.6.3 Interactive Authoring and Display System (IADS).....	55
4.7 Multiple Levels of Maintenance within Single Paper Manuals.....	56
4.8 Planning an ETM Conversion Project.....	57
4.9 Methodologies for Conversion.....	60
4.9.1 Conversion for the F/A-18.....	61
4.9.2 Lockheed Fort Worth Conversion for the F-16.....	63
4.9.3 LM2500 Gas Turbine Conversion Effort.....	65
4.9.4 AEGIS Fire Control System Conversion Project.....	67
4.9.5 Newport New Shipbuilding Conversions.....	69
4.9.6 Conversion and the Joint Integrated Maintenance Information System.....	69
4.9.7 A General Formula for Conversion.....	71
4.10 COTS Products for Conversion.....	73
4.10.1 Requirements.....	74
4.10.2 Evaluation Matrix.....	74
4.10.3 Overview of COTS Products Considered.....	79
4.11 The Cost of Conversion.....	85
4.11.1 Cost Estimation.....	85
4.11.2 Cost Justification.....	92
<b>5.0 VALIDATION, VERIFICATION, AND ACCEPTANCE TESTING.....</b>	<b>94</b>
5.1 Definitions.....	94
5.2 Conversion and the Validation Process.....	95
5.3 The Validation Plan.....	96
5.4 Validation of Technical Manual Information and Personnel Issues.....	98

5.5	Validation and ETM System Software .....	99
5.6	Validation and ETM Data Sets.....	100
5.6.1	Type MC+ ETM Data Sets.....	101
5.6.2	ETM Data Sets in General.....	102
5.6.3	Software and Types MC, MC+ Data .....	103
5.7	CALS and Converted ETM Data .....	104
5.8	Verification.....	104
5.9	Acceptance Testing.....	104
<b>6.0</b>	<b>FINDINGS AND RECOMMENDATIONS .....</b>	<b>106</b>
	<b>REFERENCES.....</b>	<b>110</b>
	<b>APPENDIX A: VENDOR CONTACT REPORT.....</b>	<b>A-1</b>
	<b>APPENDIX B: ELECTRONIC TECHNICAL MANUAL CONVERSION</b>	
	<b>PACKAGE SURVEY MATRIX .....</b>	<b>B-1</b>
	<b>APPENDIX C: SAMPLING OF COMPLETED SURVEYS.....</b>	<b>C-1</b>
	<b>APPENDIX D: IETM RESEARCH LIBRARY DOCUMENTATION LISTING .....</b>	<b>D-1</b>
	<b>APPENDIX E: DoD ETM SYSTEMS AND PROGRAMS.....</b>	<b>E-1</b>
	<b>APPENDIX F: ABBREVIATIONS AND ACRONYMS.....</b>	<b>F-1</b>

## List of Tables

<b>Table</b>	<b>Title</b>	<b>Page</b>
Table 2.3-1	Type MA+	5
Table 2.3-2	Type MB	5
Table 2.3-3	Type MB+	6,7
Table 2.3-4	Type MC	7,8
Table 2.3-5	Type MC+	9
Table 2.4.1-1	Type B	10
Table 2.4.1-2	Type C	11
Table 3.1.3.1.1.1-1	JCALs Author/Editor Vendors	18
Table 3.1.3.1.1.2-1	JCALs Viewing/Browsing Vendors	19
Table 3.1.3.6-1	JCALs User Support Components	21
Table 3.1.3.7-1	JCALs User Support Components	21
Table 4.3-1	Classifying Technical Manual Information	34
Table 4.3-2	Maintenance Functions	35
Table 4.5.5-1	Data File Formats	48
Table 4.6.3-1	IADS Reserved Tag Names	55
Table 4.8-1	Mapping Complexity	59
Table 4.9.6-1	JIMIS High Level Libraries	70
Table 4.10.1-1	Military Specifications	75
Table 4.10.1-2	Military Standards and Handbooks	76
Table 4.10.1-3	Other Government Documents	76
Table 4.10.1-4	Non-Government Documents	77
Table 4.10.1-5	Documents Under Development by the Tri-Service IETM Working Group	77
Table 4.10.1-6	Requirements for COTS Identification	78
Table 4.10.3-1	COTS Products and Services	80
Table 4.10.3-2	Electronic Presentation Systems	80
Table 4.10.3-3	General Electronic Authoring Systems	81
Table 4.10.3-4	SGML/Text Editors	81
Table 4.10.3-5	Graphics Conversion and Viewing	81
Table 4.10.3-6	SGML, Tag-based, and Text-based Conversion Software	82
Table 4.10.3-7	Electronic Document Analysis/Design Tools	82
Table 4.10.3-8	Electronic Document Managers/Databases	83
Table 4.10.3-9	HyTime Software	83
Table 4.10.3-10	Parsers	83
Table 4.10.3-11	Page Layout Tools	84
Table 4.10.3-12	Conversion Service Bureaus	84
Table 4.10.3-13	Others	84
Table 5.3-1	Elements to be Included in a Validation Plan for Converted ETM Data	97

## List of Figures

<b>Figure</b>	<b>Title</b>	<b>Page</b>
Figure 3.2-1	Graphical Depiction of JCALS Data Handling Methodology	22
Figure 4.11.1-1	Technical Manual Conversion Costs As Estimated In [19]	88
Figure 4.11.1-2	Technical Manual Information vs. Tags in the IADS Type MB+ Data Conversion Experiment	90
Figure 4.11.1-3	Technical Manual Information vs. Tags in the Type MC+ Data Conversion	90
Figure 4.11.1-4	The Number of Generic vs. Content-Specific Elements used in the Type MC+ Data Conversion	91

## **1.0 INTRODUCTION**

This report is a result of the performance of OSD-CALS IWSDB tasking for Interactive Electronic Technical Manuals (IETM) titled: CALS And The Interactive Electronic Technical Manual (IETM). The task involved a wide range of research, analysis, technical evaluation(s), project assessments, and review of the current status of the IETM industry. Our work involved the identification of Commercial Off The Shelf (COTS) products which provide measurable levels of conformance to the CALS standards and the IETM specifications (i.e., MIL-M-87268, MIL-D-87269, and MIL-Q-87270). Once candidate products were identified we then assessed and evaluated rapid and economical methodologies which could provide conversion support for existing Weapon System Technical Manuals to the IETM environment. As the basis for our conversion methodology, we were required by the Statement Of Work (SOW) to assess the relationships of the following three data types defined in general terms as:

- Type "B" Electronic Page Turner,
- Type "B+" HyperText Electronic Navigation, and
- Type "C" Interactive Electronic Page Assimilation.

### **1.1 Define Data Types and Assess JCALS**

As a result of our research and analysis we were required to provide a precise set of definitions for the above data types and perform an evaluation assessment of the data handling methodology of the Joint Continuous Acquisition and Life-cycle Support (JCALS) Program. This assessment was to provide a defined interface between the Type C data type (i.e., Interactive Electronic Page Assimilation) and JCALS data handling methodology.

### **1.2 Sub-Task Assignments**

For ease of task organization and management we first divided the IETM Tasking herein after referred to as Task Five into the following sub-task assignments:

- Develop and define precise set of definitions for IETM data types.
- Research and assess the data handling methodology of the JCALS Program.
- Evaluate COTS software relevant to the conversion, creation, and presentation of Technical Manual (TM) content which conforms to IETM Specifications.
- Research and assess conversion methodologies of TM information of varying formats which are consistent with and conform to the IETM specifications and provide migration strategy from Type B and Type B+ to Type C.
- Evaluate IETM specifications with the current set of CALS standards for compatibility factors for creating and sustaining IETM(s).

### **1.3 Format of Report**

In order to provide the reader with a comprehensive, straightforward presentation of an extremely complex topic, we have divided our report into the following key sections:

- Section 1: Introduction.
- Section 2: Defining and Classifying Electronic Technical Manuals (ETM).
- Section 3: JCALS Assessment.
- Section 4: Roadmap to Conversion.
- Section 5: Validation, Verification, and Acceptance Testing.
- Section 6: Findings and Recommendations.

Each section is intended to present a specific body of work performed by our Task Five Team and also serve as a building block for subsequent section presentations. Taken in total, this report represents perhaps the most comprehensive body of work to date in offering definitions, conversion methodologies, issues for further investigation, potential strategies, and resource estimates leading to a standard approach for successful conversion to IETM(s).

## **2.0 DEFINING AND CLASSIFYING ELECTRONIC TECHNICAL MANUALS**

This section presents our work in defining multiple data types and provides in a summary fashion the current status of the IETM definitions which exist today within the DoD.

### **2.1 TM(s) ETM(s) and IETM(s) Defined**

To begin our work we first needed to put the range of TMs, ETMs and IETMs into perspective.

**TM:** a paper based compound document which can contain a variety of information formats of textual, tabular, and graphical data. As part of Section 4 (4.1, A Look At The Composition Of Technical Manuals), we offer a look at the composition of a TM. The final make up of any particular TM is governed by one or more appropriate military standards and/or specifications which from a conversion point of view makes defining a single methodology difficult.

**ETM:** an electronic digital representation stored on either magnetic or optical media of a full implementation of a TM representing the instructions for the installation, operation, maintenance, training, and support of a weapon system and its support equipment. An ETM can range in capabilities from a simple Raster Scanned Electronic Page Turner to a range of HyperLink characteristics based on subject matter content.

**IETM:** combines the capabilities of an ETM with an Expert System (Artificial Intelligence (AI)). The Expert System will at a minimum incorporate preventive and diagnostic maintenance capability. Where applicable, it will also link to the Built-In-Test (BIT) (via a Standard Bus), prognostic routines, and the Automated Logistics System. The IETM includes all the information necessary for a system's maintenance, diagnostics, and repair. The IETM will guide the user through the process of identifying/confirming the specific system's fault(s).

**IETM:** a TM, prepared (authored) by a contractor and delivered to the Government, or prepared by a Government activity, in digital form on a suitable medium, by means of an automated authoring system; designed for electronic screen display to an end user, and possessing the following three characteristics:

- a. The format and style of the presented information are optimized for screen presentation to assure maximum comprehension; that is, the presentation format is optimally designed and formatted for display as on a monitor screen, and not as on a paper page.
- b. The elements of Technical Information (TI) constituting the TM are so interrelated that a user's access to the information he requires is facilitated to the greatest extent possible, and is achievable by a variety of paths.

- c. The computer-controlled TM display device can function interactively (as a result of user requests and information input) in providing procedural guidance, navigational directions, and supplemental information; and also in providing assistance in carrying out logistic-support functions supplemental to maintenance.

## **2.2 Baseline Requirement**

In order to develop a Roadmap To Conversion methodology we first made the following determination. Any meaningful procedure or methodology for conversion of Legacy Data has a baseline requirement that both a source (conversion from) and destination (conversion to) must be specified in some manner. As stated above, the sources and destination that we were to evaluate consists of data types B, B+, and C. To take these somewhat generalized data types and develop precise definitions; we considered how these data type definitions could be modified to assist with developing our Roadmap To Conversion methodology. This led us to develop a set of data type definitions which cover a full range of TM(s), ETM(s) and IETM(s) and are referred to as (ManTech) "MA, MA+, MB, MB+, MC, and MC+."

## **2.3 Precise Data Types Defined**

The following (ManTech) precise data type definitions are offered as the foundation for our work and will be utilized throughout this report:

### **Type MA Data**

Paper Only (a TM data type, not an ETM data type)

### **Type MA+ Data**

This data type is a non-Standard Generalized Markup Language (SGML) electronic page turner that consists of pure raster. This is the next logical step above a paper document. It is the cheapest method to convert legacy data to data that can be stored and accessed electronically. The raster scan must be at a resolution appropriate for human readability and comprehension. With this data type, it is simple to assign page numbers to each page image and provide links from the table of contents and index. This data type would be desirable either for older systems that will soon be discontinued or for systems that rarely need to be serviced. Black and white raster page images can be stored using CCITT 4 raster (MIL-R-28002). Gray-level or color raster page images must be stored using raster primitives in Computer Graphics Metafile (CGM) files; they cannot be stored using MIL-R-28002. Table 2.3-1 lists the requirements for Type MA+ data.

**Table 2.3-1 Type MA+**

<b>Type MA+ Data</b>	<b>Fixed Length Electronic Page Image Technical Manual</b>
Display Media	Electronic.
Data Primitives	Graphics (CCITT 4 or CGM).
Data Organization	Electronic replica of paper manual. Recognition of page numbers, index entries, and table of contents.
Dynamics	Access pages via intelligent index. Limited use of hotspots.
Data Format	Raster scan of paper pages.

**Type MB Data**

This data type is simply an electronic page turner that was constructed with SGML. This is the next logical step above a raster page turner. The important difference between this and the raster page turner is that because the text is now in SGML, things like word searches, automatic indexing, and other text processing algorithms can be executed on the data. After completed, formatting information can be applied with a Formatting Output Specification Instance (FOSI) for either electronic or paper presentation as listed in Table 2.3-2.

**Table 2.3-2 Type MB**

<b>Type MB Data</b>	<b>Fixed Length Electronic Page Technical Manual</b>
Display Media	Electronic.
Data Primitives	Text. Tables. Graphics (IGES, CCITT 4, CGM).
Data Elements	SGML tagged ASCII text. Limited format tagging. Structure tagging (chapters, sections). No content tagging required. No HyTime required.
Data Organization	Redundant elements. Integrated and/or non-integrated files. Format information may be applied with a FOSI.
Dynamics	Limited hyperlinking via the presentation system (not authored in). Non-interactively authored; viewer may provide text processing capabilities. Displayed pages can be printed. Static graphics with or without hotspots.
Data Format	Possibly SGML files. Possibly Compiled SGML/FOSI files (i.e., Postscript).

### Type MB+ Data

Type MB+ data is a hypermedia document that may be either page-oriented or frame-oriented. From a conversion standpoint, Type MB+ can be created by simply taking a regular Type MB SGML file and augmenting it with hyperlinks/hotspots. The hyperlinks may be implemented through the use of SGML's IDREF/CONREF to ID pointing facility, or through Hypermedia/Time-based Structuring Language (HyTime) hypermedia links. This is the next logical step to increasing the utility and information-richness of a document. The display of Type MB+ data does not have to remain page-based. If all page breaks and references to pages are removed from the Type MB data, then an SGML/HyTime compiler could be constructed such that each chapter or section could be represented in one window (frame) with scrolling capabilities as listed in Table 2.3-3. Implementing the hyperlinks would require some comprehension of the TM content, but domain-specific experts would not be required. Note that HyTime coordinate and quantum-based addressing is not recommended for hyperlinking of text here, because of the possibility of updates and changes to the original SGML file.

**Table 2.3-3 Type MB+**

<b>Type MB+ Data</b>	<b>Electronic Hypermedia Technical Manual</b> Possibly fixed-length pages Possibly frame-based Possibly an electronic scrolling document
Display Media	Electronic.
Data Primitives	Text. Tables. Graphics (IGES, CCITT 4, CGM). Audio. Video. Processing.
Data Elements	SGML tagged ASCII text. Limited format tagging. Structure tagging allowed. No content tagging required. HyTime-compliant hypermedia links preferred but not required.
Data Organization	Redundant elements. Integrated and/or non-integrated files. Format information may be specified with a FOSI. Traversal through troubleshooting trees and diagnostic data with hyperlinks.

**Table 2.3-3 Type MB+(continued)**

<b>Type MB+ Data</b>	<b>Electronic Hypermedia Technical Manual</b> Possibly fixed-length pages Possibly frame-based Possibly an electronic scrolling document
Dynamics	Extensive use of hotspots and hyperlinks which are authored in: Displayed windows can be printed. Non-interactively authored with an interactive hypermedia presentation system. Text, Tables, and Graphics may appear in the same or in separate windows. Static graphics with or without hotspots.
Data Format	SGML files.

**Type MC Data**

This is very advanced and significantly different from any of the previous types. This is the first data type that is strictly system and task-oriented. It will require organization of technical manual data into a system, sub-system, sub-assembly hierarchy with the appropriate tasks and all associated information for each. This data type is basically an SGML instantiation of MIL-D-87269 that is valid for all maintenance levels and also allows for less content-intensive tagging. One of the key features that separates this type from the previous types is the logical NEXT function. Logical NEXT allows for efficient authoring of alternative information (i.e., unit level tasks versus intermediate level tasks) and rule-based diagnostics. Depending on the format of the original, correct placement of Type MC tags may require full comprehension of the TM content as well as domain-specific experts. Table 2.3-4 lists Type MC requirements.

**Table 2.3-4 Type MC**

<b>Type MC Data</b>	<b>Frame-Based Electronic Hypermedia Work Package</b>
Display Media	Electronic.
Data Primitives	Text. Tables. Graphics (IGES, CCITT 4, CGM). Audio. Video. Processing.

**Table 2.3-4 Type MC (continued)**

Type MC Data	Frame-Based Electronic Hypermedia Work Package
Data Elements	<p>SGML tagged ASCII text.</p> <p>Frame tags and embedded scripting tags (i.e., IF-THEN) used for presentation.</p> <p>Minimum requirements are: basic set of system and task-oriented content tags as in MIL-D-87269.</p> <p>Tagging of descriptive, procedural, part, and troubleshooting information as in MIL-D-87269.</p> <p>Limited format tagging.</p> <p>Content tagging as in MIL-D-87269 to the extent desired.</p> <p>HyTime-compliant hypermedia links.</p>
Data Organization	<p>Redundancy reduced to the extent possible.</p> <p>Traversal through troubleshooting trees and diagnostic data with context-dependent filtering capabilities rather than hyperlinks.</p> <p>Alternative information authored in using context-dependent filtering capabilities.</p>
Dynamics	<p>Context dependent filtering.</p> <p>Dialog driven interaction and branching.</p> <p>Interaction functions per MIL-M-87268 to the extent possible.</p> <p>Extensive use of hotspots and hyperlinks that are authored in.</p> <p>Displayed windows can be printed.</p> <p>Interactively authored and displayed with an interactive hypermedia presentation system.</p> <p>Text, Tables, and Graphics may appear in the same or in separate windows.</p> <p>Logical NEXT and BACK function.</p> <p>Runtime variables maintained in a state table at presentation time.</p> <p>Dynamic and/or static graphics with or without overlays and hotspots.</p>
Data Format	SGML files.

**Type MC+ Data**

This is specification-compliant IETM data that does not necessarily consist of SGML files. The ETM data objects are authored to and are accessed through either a relational or an object-oriented data base. Although this data type may not necessarily consist of SGML-tagged objects, the database information should be named, attributed, and hierarchically structured as described in MIL-D-87269. This data type is appropriate for creating and maintaining data that will be required by multiple, interrelated ETMs and IETMs. As in the Navy Class 4 IETM definition, the data elements that will constitute a particular IETM are extracted, compiled, and formatted to create a "View Package" run-time version of the data base that can be processed for display (in accordance with MIL-M-87268) by suitable presentation system software. An important issue in

terms of CALS-compliance here is that a system utilizing Type MC+ data should be capable of importing and exporting its ETM/IETM data as tagged SGML files (similar to Type C) so that different authoring/editing/viewing systems with unique databases can easily share/exchange data. Table 2.3-5 lists the Type MC+ requirements.

**Table 2.3-5 Type MC+**

<b>Type MC+ Data</b>	<b>Specification-Compliant IETM</b>
Display Media	Electronic.
Data Primitives	Text. Tables. Graphics (IGES, CCITT 4, CGM). Audio. Video. Processing.
Data Elements	Fully attributed database structure per MIL-D-87269. Content-specific layer developed using the generic layer in MIL-D-87269. Authored to a favorite Database Management System (DBMS).
Data Organization	Redundancy reduced to the extent possible. Traversement through troubleshooting trees and diagnostic data with context-dependent filtering capabilities rather than hyperlinks. Alternative information authored in using context-dependent filtering capabilities.
Dynamics	Context dependent filtering. Dialog driven interaction and branching. Interaction functions per MIL-M-87268. Interactively authored and displayed with an interactive hypermedia presentation system. Logical NEXT and BACK function. Dynamic graphics with overlays and hotspots.
Data Format	No restriction on database format. May or may not have SGML import or export capability.

## 2.4 Multiple Data Type Definitions

During our research phase we discovered that many opinions exist as to how data types for both ETMs and IETMs can and should be defined. So for purposes of completeness, we include a summary set of ETM data type definitions that currently exist within the DoD. The Army and Air Force data types appear to be created from the end user perspective. On the other hand the U.S. Navy Classes which are expected to become the DoD standard data types appear to be created from an authoring/functionality perspective. And finally, the JCALS data types appear to be

created from a somewhat practical perspective in terms of providing tools for authoring, editing, and viewing. Note that at the DoD Tri-Service IETM Working Group meeting in September 1994, it was agreed that the Navy data types would supersede the Army and Air Force data types and be used for definitions purposes only, not for acquisition purposes. Also, note that the JCALS data types were proposed by Computer Sciences Corporation but were later rejected at the DoD Tri-Service IETM Working Group meeting in April 1994.

Again, to reiterate our position, the precise data types defined above with the prefix of "M" are the foundation of our work and are to be utilized throughout this report. The following multiple data type definitions are offered for purposes of completeness and are also covered later in this report under findings as part of Section 6. The multiple data types are as follows.

### 2.4.1 Army and Air Force Data Types

The U.S. Army and Air Force generally refer to and support the following data types:

#### Type A

Conventional hard copy paper manuals.

#### Type B (Electronic Pages)

This data exists primarily for the printing of paper technical manuals from electronic files. The electronic files consist of tagged SGML files compiled with an appropriate FOSI as listed in Table 2.4.1-1.

**Table 2.4.1-1 Type B**

<b>Display Media</b>	<b>Paper</b>
Data Primitives	Text. Tables. Graphics.
Tagging	Limited format tagging. Structure tagging.
Data Organization	Redundant Elements. Non-integrated files. Limited cross referencing.
Dynamics	Versioning. No user interaction. Static presentation.

**Type B+ (Between Electronic Pages and Interactive Database)**

This data is Type B data which has been enhanced by any, but not all of the Type C characteristics. It is expected that this would be created primarily for electronic hypertext presentation rather than paper output.

**Type C (Interactive Database)**

This Type C data is referred to as fully interactive. It is database oriented in accordance with MIL-D-87269. Retrieval is associated by frame orientation rather than page orientation. A frame is composed of variably packaged text and graphics based on user-interaction and/or automated diagnostics based prompting. Table 2.4.1-2 list the formatting requirements for Type C.

**Table 2.4.1-2 Type C**

<b>Display Media</b>	<b>Electronic</b>
Data Primitives	Text. Tables. Graphics. Audio. Video. Processes.
Tagging	No format tagging. Content tagging.
Data Organization	Non-redundant referencing. Integrated database. Relational links.
Dynamics	Context-dependent filtering. User interaction and branching. Dynamic presentation.

This data type is further defined in the DoD IETM specifications as:

- The format and style of the presented information are optimized for screen presentation to assure maximum comprehension; that is, the presentation format is optimally designed and formatted for display as on a TV screen, and not as on a paper page.
- The elements of TI constituting the TM are so interrelated that a user's access to the information he requires is facilitated to the greatest extent possible and is achievable by a variety of paths.
- The computer-controlled TM display device can function interactively (as a result of user requests and information input) in providing procedural guidance, navigational

directions, and supplemental information; and also in providing assistance in carrying out logistic-support functions supplemental to maintenance.

## **2.4.2 Navy Data Types**

The Navy Carderock Division of the Naval Surface Warfare Center (CDNSWC) has developed a set of Classes for defining ETMs and IETMs. These Classes are more restrictive than the Army and Air Force data types. Highlights of each Class are given below:

### **Class 0**

- Non-electronically indexed page images (not an ETM Class).
- Display: Printed pages or full page electronic viewing.
- Data Format: Word processor, SGML, or graphics files.
- Functionality: To print or view pages.

### **Class 1**

- Electronically indexed page-images.
- Display: Full page electronic viewing.
- Data Format: MIL-R-28002 (Raster Graphics Representation in Binary Format, Requirements for) or Postscript.
- Functionality: Access pages via intelligent index and/or header information, limited use of hotspots.

### **Class 2**

- Electronic scrolling documents.
- Display: Scrolling text window. Could contain hot-spots on text and graphics. Text is minimally formatted for display window.
- Data Format: Text is ASCII. May be SGML tagged with no page breaks. Graphics may be Bitmap (BMP) or CALS.
- Functionality: Browse through scrolling information with user selectable hot-spots on text or graphics.

### **Class 3**

- Linearly structured IETMs.
- Display: Text and graphics simultaneously displayed in separate windows when keyed together. Interaction functions per MIL-M-87268 (Manuals, Interactive Electronic Technical: General Content, Style, Format, and User-Interaction Requirements) to the extent possible. Interaction through dialog boxes. View smaller logical blocks of text, less use of scrolling text compared to Class 2.
- Data Format: ASCII text with SGML tags. Uses content tags rather than format tags. MIL-D-87269 used to the maximum extent possible.
- Functionality: Dialog-driven interaction. Logical display of information in accordance with content with user-selectable cross-references and indices.

### **Class 4**

- Hierarchically structured IETMs.

- Display: Similar to Class 3 except very limited use of scrolling text, much less than Class 3. Interaction functions per MIL-M-87268. Interaction through dialog boxes with user prompts.
- Data Format: Fully attributed database elements described in MIL-D-87269. MIL-D-87269 content tags with full conformance to generic layer architectural formats. Authored directly to a database specifically for interactive electronic output. Interactive features are authored in, not added later.
- Functionality: Same as Class 3.

### **Class 5**

- Integrated Data-Base IETM.
- Display: Multi-function display session. Same as Class 4 when in an IETM mode. Interactive electronic display per MIL-M-87268. Expert system to allow some display session and viewing system to assist in simultaneous access to many differing functions such as troubleshooting, training, and parts inventory.
- Data Format: Identical to Class 4. Does not use separate databases for other applications data (i.e., training, parts inventory). Includes coding for Expert System rules and AI software.
- Functionality: A single viewing system for simultaneous access to multiple information sources. Class 4 standards for the IETM application. Expert System to assist in NEXT function based on information gathered during session.

### **2.4.3 JCALS Data Types**

The JCALS data types consist of the following:

#### **1000 Class**

- Paper-based documents.
- 1300 Class: One that has been raster-scanned and is available to the user as a page for page view with no ability to search the text.
- 1500 Class: One that exists as clear ASCII text with only ASCII control codes. No other markup is available. Searches can be made on the text for retrieval purposes and indices can be built to facilitate searching and locating information in the document.
- 1700 Class: Page Description Language (PDL) (i.e., Postscript) representation of the document that provides the text and formatting information to allow the page to be presented as it was in the printed form. It provides for searching and indexing.

#### **2000 Class**

- Electronic documents.
- 2300 Class: SGML-tagged document where all internal references are resolved and no "cross-reference" tags are used to generate the cross-references automatically. No hyperlinking capability other than possibly from a table of contents or figure/table list.
- 2500 Class: SGML-tagged document that uses the SGML ID and IDREF (or similar functionality) to generate internal references and thus supports the generation of

hypertext links within the document. Links may also exist to other external documents and/or databases.

- 2700 Class: SGML-tagged document that includes all the functionality of the 2500 Class document but also includes conditional logic capabilities to handle hypertext linking dependencies. The document needs to be changed to include the proprietary code required for the conditional linking to occur.

### **3000 Class**

- Interactive, non-linear, intelligent documents.
- 3500 Class: Based on the model presented by the David Taylor Labs (but not necessarily the same architecture). It is a nonlinear "document" that consists of four functions:
  1. A set of frames and/or other media to be presented as required.
  2. A scripting function that reads from and writes to the frames and presents the frames and/or other media to the user in order specified by.
  3. An inference engine that is either an Artificial Intelligence engine or an Expert System.
  4. A HyTime engine that orchestrates the entire process.

## **2.5 Defining CALS-Compliant ETM/IETM**

To close out this Section 2, we offer a summary analysis of the multiple data types definitions above and their ability to be CALS-compliant.

If the U.S. Navy Classes become the DoD standard digital data types for ETM(s), some consideration must be given as to what constitutes not just an ETM, but a CALS-compliant ETM. Although the current Navy Classes are sufficient for delineating the differences between both the format and functionality of ETMs, their use does not imply CALS compliance. One difficulty, from a CALS perspective, is that the Navy Classes do not force a standardized CALS-compliant format for all of the underlying data. Although there are fundamental differences between CALS-compliant data and a CALS-compliant ETM, definition of a CALS-compliant ETM necessarily begins with the format of the underlying data.

Defining a CALS-compliant ETM also requires consideration to the implementation and operation of the presentation system. The operation of the presentation system, of course, depends on the inherent functional capabilities of the data itself. Defining a CALS-compliant presentation system for any Class of ETMs or IETMs implies that all compliant presentation systems will display the same data to the user. Because the user would see the same data, a compliant presentation system should also provide a certain common set of standard functions or capabilities (i.e., scroll bars, etc.).

In considering the presentation system issue, one must keep in mind that when data is authored electronically with today's technology, that same data must remain accessible far into the future (using tomorrow's technology). If special hardware or software is required in presenting or

viewing of a given electronic document, there is no guarantee that those same systems will exist in the future. If all the data, however, is in a neutral, standardized format that is intended to be maintained, then there is a greater possibility of accessing that data in the future.

This issue of the presentation system is addressed to some extent, in defining CALS-compliant ETMs, for Classes 1, 3, 4, and 5. In the case of Class 1, the format of the data restricts its functionality and hence the implementation of the presentation system. This presentation/functionality problem is also handled to a (small) degree for Classes 3, 4, and 5 through the recommended use of MIL-M-87268.

Although establishing the CALS-compliance of an ETM by taking into consideration its presentation system is important, it must be regarded in accordance with the CALS philosophy. In other words, the files that represent ETM data (regardless of Class) should be completely decoupled from their corresponding presentation systems. In an ideal world, a Class 3 presentation system, for instance, would be capable of presenting all existing validated Class 3 ETMs that were authored for the purpose of becoming valid Class 3, CALS-compliant ETMs. Whether this is possible within each Class, given the variety of authoring and presentation systems already in use throughout DoD, requires further in-depth investigation. Also, some claim that the data and functional capabilities that constitute Class 2, 3, 4, and 5 ETMs simply cannot be stored in a neutral data format.

Given the preceding discussion concerning the presentation system, the formidable difficulty that must be addressed is the final format of the raw ETM data. The graphical domain is limited by the fact that there are only three CALS graphics standard formats with little variation in their implementation. The textual domain, on the other hand, is the primary concern because with it comes an infinite number of implementation possibilities because, Document Type Definitions (DTDs) for underlying SGML files can be created at will.

Proper presentation of the textual elements of the ETM requires knowledge of the fundamental capabilities that are inherent to the given DTD. Some difficulty in the question of CALS-compliance would be alleviated if the Classes that consist of SGML files are required to use a specific DTD or set of DTDs in conjunction with a standardized "reference" presentation system. That standardized presentation system would be designed with respect to the appropriate DTDs without allowing for functional variations. A "reference" presentation system is needed for the purposes of validation, because currently, even Classes 3 and 4, which are well defined in recommending the use of the DTD in MIL-D-87269, are subject to variations (which the presentation system must account for) on implementation. These issues seriously impact any requirements concerning the establishment of standard ETM interfaces to the JCALS system.

Many opinions exist on the issue of CALS-compliant ETMs. Some feel that to create a CALS-compliant ETM/IETM, they must, among other things, follow the Content Data Model (CDM). Some claim that it is simply too expensive to create CDM-compliant ETMs/IETMs. Although having CDM-compliant data is highly desirable, it should not be considered critical for establishing the CALS-compliance of all ETMs/IETMs. Each DoD ETM Class should highlight the format and functionality of those elements deemed necessary for CALS-compliance.

### **3.0 JCALS ASSESSMENT**

This section highlights our team's JCALS Assessment. The section begins with a brief summary overview of the JCALS Program, highlights the data handling methodology as required by our SOW and offers our assessment for potential ETM interface support.

#### **3.1 JCALS Overview**

To understand the JCALS data handling methodology and interface requirements, we first researched both the history and the current state of JCALS. Below is a brief historical view of JCALS and an overview of the entire JCALS system. The overview will go more in-depth into the area that pertains to the scope of this task, i.e., the authoring, editing, and viewing of electronic technical manuals. Also, where possible, important COTS software packages that were identified are given with their functionality stated.

##### **3.1.1 History Of JCALS**

In the early 1980s, the Army Material Command (AMC) conducted a study to resolve a major problem held by the Program Manager (PM) of the Army's M1 tank. An excess of paper was crippling the procurement process. This study recommended reducing paper by using interactive video disk technology.

At the conclusion of the study by the AMC, initiatives were started to address the problems associated with paper-based weapons systems. The Technical Information Management System (TIMS) program was specifically started to address these problems. This program became the initial draft of the Army CALS Request For Proposal (RFP). In mid 1988, the final RFP was released, and in mid 1989, Phase 1 contracts were awarded to four contractors. The contractors were competing for Phase 2 contracts by developing a conceptual design.

At the beginning of 1991, Phase 2 started with the awarding of a contract option to Xerox and Computer Sciences Corporation (CSC), two of the original four contractors in Phase 1, to demonstrate selected portions of their conceptual designs. PM Army CALS was directed to incorporate minimum essential requirements of the Air Force, Navy, Marine Corps, and Defense Logistics Agency (DLA) within the Army CALS program. As a result, the Assistant Secretary of Defense for Production and Logistics established the JCALS program as a joint defense program.

Phase 3 started with the selection of CSC as the development contractor for the System Operational and Support Capability (SOSC). Currently, JCALS is operating as a prototype technical manual development system at the following DoD sites:

- SOSC, Marlton, NJ,
- Robbins AFB, Warner Robins, GA,
- Tinker AFB, Oklahoma City, OK,
- US Army Missile Command, Huntsville, AL,
- Marine Corps Logistics Base, Albany, GA,
- Port Hueneme Division Naval Surface Warfare, Port Hueneme, CA, and

- US Army Publishing and Printing Command, Alexandria, VA.

JCALs has expanded beyond the initial installations to include additional prototype sites. The performance and evaluation of capabilities provided at the prototype sites could lead to a contract option to supply the approximately 400 remaining JCALs sites in Phase 4.

### **3.1.2 Current JCALs System Description**

The JCALs technical manual system is primarily based on COTS software, but some non-COTS software is used to integrate individual software packages. This system is heavily customized to improve on automated technical manual work flow. In addition to the numerous prototype sites, the JCALs program will include an SOSC facility. The SOSC will provide the overall network management between the various prototype sites. Each prototype site consists of the following four subsystems:

- Network Management,
- Data Management,
- Workstation, and
- Network Distribution.

The Network Management Subsystem controls communication to other sites and to the SOSC. The Network Distribution Subsystem performs the subsystem interfacing at a site. The Data Management Subsystem not only maintains the overall system status, but also administers, updates, and controls access to the entire system. The Human/Computer Interface (HCI) occurs at the Workstation Subsystem.

### **3.1.3 Current JCALs System Functional Overview**

The JCALs system may be divided into seven functional areas that may be tailored to the requirements of each user, site, and organization. These functions are:

- Application Process,
- Implementation,
- Work Management,
- Information Management,
- System Administration,
- User Support, and
- System Infrastructure Support.

#### **3.1.3.1 Application Process**

This function provides the tools for the generation of a variety of system products, as well as design, analysis, and decision support tools. These tools include graphical and textual editors and viewers, Computer Aided Design (CAD) tools, and Logistics Support Analysis (LSA) tools. This function supplies tools needed for comprehensive engineering design. These tools are grouped into the following four functional areas:

- Product Generation,
- Design,
- Analysis, and
- Decision Support.

### 3.1.3.1.1 Product Generation Tools

This tool set provides the user with an array of functionality that may be applied to any authoring task. Product generation tools simplify the authoring process and are used as viewers. These tools consist of the suite of tools that support the following two functions:

- Authoring/Editing, and
- Viewing/Browsing.

#### 3.1.3.1.1.1 Authoring/Editing

The Authoring/Editing function provides a set of flexible authoring, editing, composition, and generic tools used to create any type of technical documentation. Examples include text, tables, equations, illustrations, and the composition of integrated, formatted publications with camera-ready output.

The Document Composer tool is a general-purpose composition system that formats and paginates a source SGML document entity and produces Postscript output files. CALS-standard graphics may also be converted to obtain Postscript versions of the graphics entities.

A Technical Illustrator tool is provided to create and modify graphic images that are linked to various documents, especially technical manuals.

Table 3.1.3.1.1.1-1 lists vendors/products used in JCALS for authoring and editing technical manuals.

**Table 3.1.3.1.1.1-1 JCALS Author/Editor Vendors**

<b>Process Function</b>	<b>Vendor/Product</b>
SGML Editor	ArborText SGML Editor.
DTD Editor	ArborText Document Architect.
Graphics Editor	InterCap, AutoCad, Asterx.
SGML Auto-Tagger	Avalanche FastTag.

### 3.1.3.1.1.2 Viewing/Browsing

The Viewing function provides SGML browsers and viewers, and graphical viewers to view both textual and graphical data stored in a variety of formats. Other viewers and browsers provided for the system are the Forms SGML Browser and Forms Viewers. The browser enables the user to view completed forms in a "read only" mode, while the viewer can invoke the document composer to prepare the form for editing.

The JCALS system will provide a Type MB SGML Browser and a Type MB Document Viewer. The Type MB SGML Browser provides a semi-formatted view of the Type MB document without incurring the overhead associated with an SGML editing session. No hyperlinking capabilities are provided, and elements such as tables, figures, and footnotes may be presented in separate windows when formatting requirements are not complex.

Table 3.1.3.1.1.2-1 lists vendors/products used in JCALS for viewing and browsing electronic technical manual data.

**Table 3.1.3.1.1.2-1 JCALS Viewing/Browsing Vendors**

<b>Viewing/Browsing</b>	<b>Vendor/Product</b>
IETM Browser	Electronic Book Technology (EBT) Dynatext
IETM Screen Presentation	EBT Makebook
IETM Graphics Presentation	Rosetta Preview

### 3.1.3.1.2 Design Tools

The graphic design tools provide a generic tool set for mechanical and electrical part design and review. These tools include electrical CAD and mechanical CAD. The electrical CAD tools include analog analysis, logic synthesis, fault simulation, and Application Specific Integrated Circuit (ASIC) design. The mechanical CAD tools include solid modeling, complex geometric design, and engineering drawing detailing. The design tools are available on all authorized workstations.

### 3.1.3.1.3 Analysis Tools

These tools are used to perform LSA, Integrated Logistics Support (ILS) analysis, and Reliability, Availability, and Maintainability (RAM) analyses. These tools reduce costs through efficient data handling and the elimination of manual methods for recording data. These tools allow for a timely analysis of designs to identify supportability shortfalls and cost drivers early in the design process.

### 3.1.3.1.4 Decision Support Tools

These tools provide specific guidance to aid the user in the development of a product. These tools ascertain the analytical procedures and/or expert knowledge necessary to solve the problem

and then create an integrated hybrid expert system that can assist the user in deriving solutions to the problem.

### **3.1.3.2 Implementation**

This function provides the tools and applications for each organization to tailor the basic system to accommodate its needs. DTDs are implemented in the JCALS system by using InfoDesign Corporation's software.

### **3.1.3.3 Work Management**

This function provides users with the capability to define and organize work, plan projects, and electronically review, provide comments, and approve work under development. The JCALS desktop utilizes an application called Controlled Access Folder Executive (CAFE) to launch the proper application when an action is performed on an object in the folder. The CAFE is integrated with two other work management applications called work flow manager and task manager. The work flow manager assigns tasks and information to individuals according to the task manager. These assignments and information are tracked in Sybase. The task manager acts as a subset to the work flow manager. The task manager tracks the work flow for a given job and provides individuals a "to-do" list.

### **3.1.3.4 Information Management**

This function provides users with access to data stored within the system as well as data stored on existing systems. It provides tools for importing and exporting data in both paper and electronic formats and capabilities to convert data to CALS-compliant formats.

Three types of data are in the JCALS data management system: private, public, and official. Private data pertains to information used by the individual. Public data is described as work in progress. Official data represents published work. CSC developed special search/retrieval capabilities for the official data.

JCALs data is stored in three parts: record data, graphical data, and SGML-encoded documents. Record data is stored in a Sybase relational database. Graphical data is stored in the UNIX file system (DEC ULTRIX). SGML-encoded documents are stored using Information Dimensions Basisplus. Basisplus provides full-text and structured-text searching and retrieval down to the object level.

The official data, referred to as the Reference Library, is accessed through individual applications via the Reference Library Browser. Ad-hoc queries can be made against the official database through a graphic query interface provided by General Research Corporation (GRC). This single graphical query retrieves information from SGML-encoded documentation, Logistic Support Analysis Record (LSAR) data and graphics. Additionally, MIL-STD-1840B (Automated Interchange of Technical Information) read/write utilities are provided by InfoDesign Corporation's MIL-STD-1840B utilities.

### 3.1.3.5 System Administration

This function provides the capabilities to monitor and maintain the system. This includes tools supporting: security administration, network monitoring, data and database administration, and problem reporting and follow-up capabilities.

### 3.1.3.6 User Support

This function utilizes a graphical desktop that provides an environment for user activation of workbench applications. The workbench applications include: an office automation package, context sensitive hypertext user help, teleconferencing, and file transfer functions. Table 3.1.3.6-1 lists user support components in the JCALS system.

**Table 3.1.3.6-1 JCALS User Support Components**

User Support Function	Vendor/Product
Office Automation	Asterix Word Processor Asterix Graphics Asterix Spreadsheet Asterix Electronic Mail
Hypertext Help	X DeskhelP
Windowing Manager	X Window.
Graphical User Interface	Motif
Graphical File Manager	X Desktop

### 3.1.3.7 System Infrastructure Support

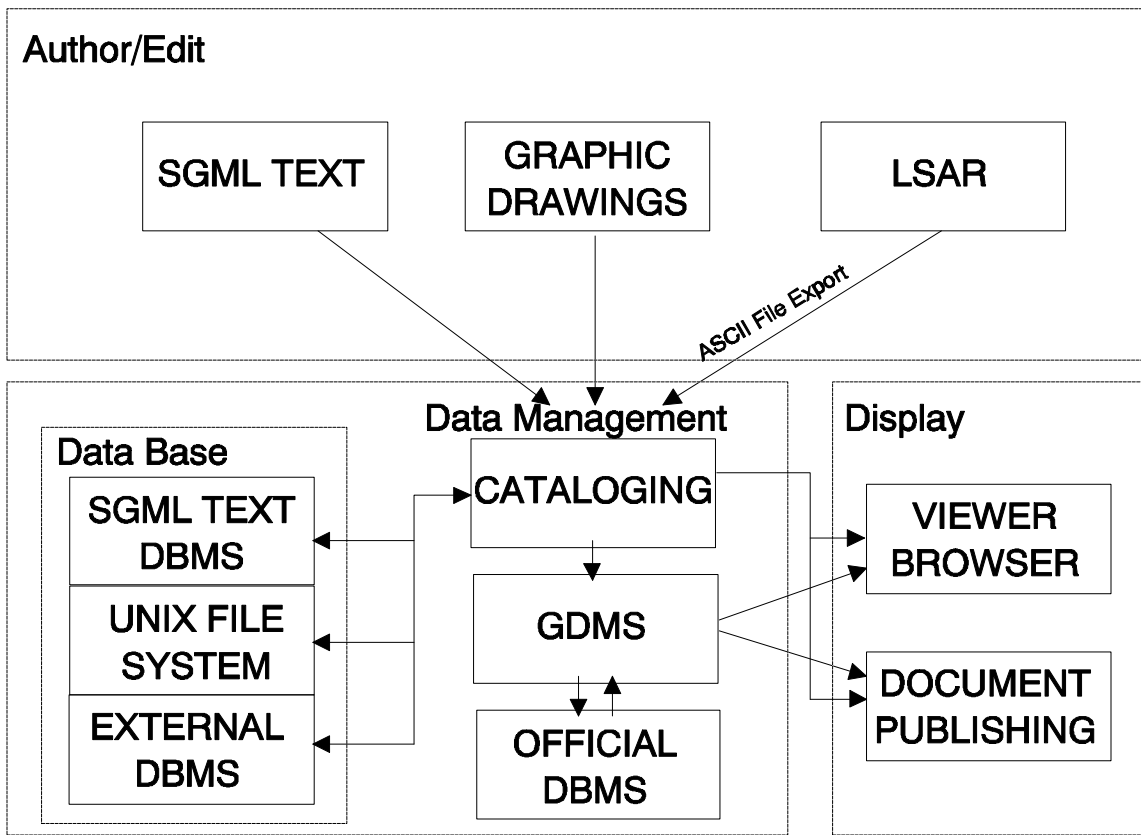
This functional component includes the operating system, the DBMS, a Global Data Management System (GDMS), communications functions, and system security features. Table 3.1.3.7-1 lists system support components in the JCALS system.

**Table 3.1.3.7-1 JCALS User Support Components**

System Support Function	Vendor/Product
Operating System	DEC ULTRIX MLS+
Database Management System	Informix Online Secure Server
Text Database Management System	Information Dimensions Basis Plus
Global Database Management System	Informix Net, SQL Host, SQL Client
System Security	DEC Network Manager, and CAFE

## 3.2 JCALS Data Handling Methodology

The following details the structure and purpose of the various functional areas of the JCALS system. Figure 3.2-1 depicts the JCALS data handling methodology.



**Figure 3.2-1 Graphical Depiction of JCALS Data Handling Methodology**

### 3.2.1 Author/Editor

To be utilized by the JCALS system, data must be imported in a predetermined format. The following describes the creation of SGML-encoded document instances and graphical illustrations in the appropriate format for use by the JCALS system. Examples of COTS products used for these purposes are ArborText, InterCap, AutoCad, and Asterix.

#### 3.2.1.1 SGML Text

For the creation of text, tables, and equations, a SGML editor is provided. SGML is defined by the ISO Standard 8879 and provides a structured editing environment in which the author is guided by a DTD. The DTD defines the type of elements such as paragraphs, steps, and lists that are logical at each order of context in the document. A DTD can be modified easily to meet evolving requirements of the user. The editor also provides basic editing features such as cut, copy, paste, "undo," search, replace, and spell checking. Examples of COTS products used for these purposes are ArborText and DLComposer.

The SGML editor also provides markup features using document templates as a guide. The tags are already defined by the DTD for the template. The author has the ability to fill in text corresponding to the tags.

A Forms Editor is also provided in this tool set. The Forms Editor facilitates the completion of electronic forms within the system. The Forms Editor performs pre-population of form fields based on information already entered into associated fields. The forms are stored in an SGML Form Template.

### **3.2.1.2 Graphics**

To create and modify graphic images, graphics tools are provided. These graphics files may then be linked to various documents, including technical manuals. Examples of COTS products used for these purposes are InterCap, AutoCad, and Asterx.

### **3.2.1.3 LSAR Data**

From the information provided, it has not been determined whether relevant LSAR data will reside locally in the LSAR Relational Database Management System (RDBMS), or whether LSA reports in ASCII format could be referenced in the UNIX file system. This area was not researched further because at this time, the handling of LSA/LSAR data is not a contract requirement with CSC.

## **3.2.2 Data Management**

The Data Management Subsystem maintains the overall system status and controls access to the entire system. Configuration control data is maintained through the cataloging system and stored in the DBMS.

### **3.2.2.1 Cataloging**

The JCALS system includes a tool that can catalog existing documents and drawings, and abstract data types into the official database. An authorized user registers these items via the cataloging software. The user is prompted to specify the type of object, along with its Class, name, revision level, and other attributes. This information is then stored in the official database. An example of a COTS product used for these purposes is CATALOGManager, by GRC Inc.

### **3.2.2.2 GDMS**

The CSC product GQDS is synonymous with GDMS. This system provides access to, integration with, and management of data across distributed, heterogeneous systems. This system provides capabilities for multiple users, multiple databases, cataloging of abstract data types into the dictionary, and existing documents and drawings, and graphical interactive queries against the conceptual database model.

### **3.2.2.3 Official DBMS**

The JCALS system employs the Oracle database from the Oracle Corp. This database is used to store data primitives such as bibliographic information pertaining to SGML text, pointers to graphics files, and LSAR data. Utilizing IDI's revision tracking, GRC developed the capability to track revisions to documents and graphics stored in the official database. This data is then officially "checked-in" and "checked-out" based on security levels and task ids, using the work flow/task manager developed by InfoDesign Inc.

### **3.2.3 Data Base**

The SGML textual database management system is imbedded within ArborText. Each database may consist of several different documents, set up by the user. The database(s) may be searched by structure or content. Searching may be ranked by relevancy and/or weighted by relevancy and other types of ranking. Annotations and other auxiliary information may be attached to the document for review by other authors. The other authors see this auxiliary information when they login to use that particular library document.

### **3.2.4 Display**

The JCALS system requires a method for display of the data it manages. The following describes how the JCALS system furnishes the user with the capability to view electronic or paper output.

#### **3.2.4.1 Document Publishing**

Each JCALS site will support batch publishing capabilities provided by Datalogics Inc.'s JCALS Composer. This software (DL Composer) automatically compiles DTDs and SGML instances into PostScript printable pages according to the FOSI.

#### **3.2.4.2 Viewer/Browser**

JCALs is storing information already compiled in DynaText and retrieving it from the file system. DynaText and Makebook have been heavily customized using EBT's software integrator's toolkit. As described further in Section 3.1.3.1.1.2, this customized version of DynaText meets some of the Type MB and MB+ electronic technical manual presentation system requirements.

Also, a Type B Document Viewer presents formatted, pageless text, graphics, figures, and tables in separate windows. Although it is not a general solution to presentation of ETM data, it does provide access to SGML encoded information. Hyperlinking capabilities are also provided.

In addition to their Type B capabilities, JCALS also includes some limited Type C capabilities. According to their Functional Description, they will provide a Type C SGML browser that will provide the capability to browse an IETM database file. The browser is expected to simulate the responses the user would see based on user input to the information requested on the displayed frames, and depending on the responses, branch to the next frame. And, among other things, it is stated in the Functional Description that it will also allow the user to begin at an arbitrary point in the IETM and preset the user-input variables that would have had to be set to have arrived at that

point. Although they expect to provide a Type C browser, they will not provide a Type C viewer and will have no Type C (and therefore Type MC, MC+) composition or compilation capabilities.

### **3.3 JCALS Interface Requirements**

JCALs uses the military standard MIL-STD-1840B for the requirements of data handling into and out of the system. The purpose of this standard is to standardize the formats for the exchange of digital information between organizations or systems exchanging digital forms of technical information necessary for the development and logistic support of defense systems throughout their lifecycle. This standard addresses technical information that is part of the traditional technical data package used for item acquisition, technical information used for design, manufacturing, fielding, and disposal of an item, and the technical documentation used for item support. This includes information such as product data, product acquisition and implementation information, and product support data.

The JCALS system requires the MIL-STD-1840B format; any use of JCALS data will require transformation from its MIL-STD-1840B formatted data to the target format. On the other side, any existing non-JCALs data will have to be reformulated into the MIL-STD-1840B-compliant format in order to be accessed by the JCALS system. Below is a high-level description of the header requirements. For more information in this area, refer to the MIL-STD-1840B document.

Under MIL-STD-1840B, data is exchanged in transfer packages, which are in turn composed of transfer sets. Each transfer set is composed of a number of transfer units. Transfer units are the basic component of data transfer. A transfer unit is always composed of a transfer unit declaration file and transfer unit data files. This is to say that there are two different headers involved: one at the beginning of the entire transfer unit and one at the beginning of the data files themselves.

The transfer unit declaration file provides all information necessary to uniquely identify the transfer unit. Informally, this means that the data using an MIL-STD-1840B header format will describe facts about the data including where it came from, where it goes, what type of data is included, and how much data is contained in the specific transfer unit data files. Additionally, the transfer unit declaration file contains the count of each transfer unit data file type contained in the transfer unit.

The header at the beginning of the data files themselves specifies the transfer unit data file types, the transfer unit file name, and identifying header records. The record identifiers can take many predetermined values such as the specification version, the source system document identifier, and others as allowed by MIL-STD-1840B. Certain records are mandatory for every data file type and are also listed in MIL-STD-1840B.

### **3.4 ETM Interface Assessment**

Under the initial JCALS contract, the only reference to TMs was the utilization of the Joint Services Requirements Document titled Technical Manual/Functional Description (TMFD). Unfortunately the concept and transition to ETMs was not part of DoD's requirements.

During the course of our meetings with PM/JCALS and review of the data handling methodology component, it was determined that the JCALS system does not include or facilitate at this time a direct interface support for ETMs. Although they intend to provide Type B authoring/browsing/viewing and Type C browsing capabilities, these will not be sufficient for fully supporting the variety of Types or Classes of ETMs that already exist in DoD. (See Appendix E for a listing of DoD ETM systems/programs that have been identified during this task.)

Due to the current state of ETMs and the IETM specifications, the issue of interfacing converted ETM data with the JCALS system can really only be resolved on a case by case basis. JCALS will be treated as a management system and it is envisioned that ETMs will be managed like other types of TMs. Given the current state of ETMs in DoD this will require that independent interfaces be developed between the JCALS system and each ETM authoring/presentation system that it is to be used with.

Developing the independent interfaces is really the only way to provide each service with the capability of modifying, maintaining, and updating their ETMs. This is true even when all ETM data is maintained in CALS-compliant formats (i.e., SGML, etc.). It will become apparent after reading Section 4.0 of this report that modifying and updating ETMs requires much more than access to an SGML editor. Reliably updating Type MC and MC+ SGML files, in particular, is likely to be too complex to be handled with an SGML editor and will require access to the authoring system that generated that SGML file. Also, the only way to guarantee the validity of the modified ETM data is to have the capability of previewing it with the presentation system that was designed to present it. Note that if the Metafile for Interactive Documents (MID) project, initiated by CDNSWC, is completed, it may eventually be possible for the JCALS system to have interfaces to many of the different ETM authoring systems while only requiring one standard presentation system for previewing ETM data (so long as each authoring system has the capability of mapping its data into the MID file format).

Steps towards the establishment of interfaces to ETM systems have already been taken. To give one example, in recent weeks there have been important discussions between PM/JCALS and the U.S. Air Force F-16 Program Office. The discussions focus on an interface between the F-16 IETM Integrated Maintenance Information System (IMIS) software and the JCALS system. We view this as a very positive event.

## 4.0 ROADMAP TO CONVERSION

Conversion from Type B or B+ data to Type C data that complies with both the CALS standards and the IETM specifications is neither a simple nor an exactly specifiable process. The primary difficulty stems from the fact that this process essentially requires the extraction of more detail from less. Another difficulty is that because of the high rate of change and technical innovation in the computer hardware and software industry, most of the different Types or Classes of electronic technical manuals are now and are likely to remain somewhat broadly defined. These broad definitions, in fact, make it very difficult to specify a single set of conversion steps that could be used to go from one data type to another, and at the same time be applicable for all instances of those particular data types.

Some classification scheme is necessary to remove some of the ambiguity associated with the functional capabilities of any given ETM/IETM. It is important to recognize that one Class or Type is not necessarily better than another; they just have different functional capabilities. Although we have encountered widely differing opinions in both DoD and industry on this matter, it is our opinion that all the different Types or Classes are necessary and all will coexist for the foreseeable future.

When considering converting between Types or Classes of paper or electronic technical manuals, the conversion process, and hence a conversion plan, is much more than simply a set of procedures for data translation. The conversion process must begin with an in-depth evaluation of the Type or Class of technical manual that is most suitable, both functionally and economically, for the system that it is to be created for. The economical considerations, discussed in more detail in Section 4.11, are very important because, depending on the desired target data type, there can be significant differences in the level of expertise, the amount of time, and hence the cost required to carry out the conversion process. It is important to note that when converting to non-page based electronic formats, even if the desired target data type consists of SGML files, we must consider how that data will be both authored and presented.

During this task, it became apparent that a precise and universally applicable mapping from either Type B or Type B+ data to Type C data would be impossible. The only way that a precise set of conversion procedures could be established would be to have precisely defined data types and presentation methodologies coupled with consistent implementations. In other words, this problem would be much easier if:

1. All ETM data was in SGML.
2. And there was a single DTD for all Type B manuals, and that DTD would be appropriate for maintenance at any level, and there was a single system or method for presenting these Type B manuals.
3. And there was a single DTD for all Type B+ manuals, and that DTD would be appropriate for maintenance at any level, and there was a single system or method for presenting Type B+ manuals.
4. And there was a single DTD for all Type C manuals, and that DTD would be appropriate for maintenance at any level, and there was a single system or method for presenting Type C manuals.

Currently, even if ETM data is maintained in a standard neutral format such as SGML, that format will be tied to a particular presentation system. With the above problems, one way to write a conversion plan for ETMs consisting of CALS-compliant data would be to select individual DTDs, operating with certain presentation systems, and list detailed steps and diagrams to map between them. These procedures, however, would be relevant only for those particular implementations and would be useless for others. There is also no guarantee that any particular implementation will not soon be obsolete because of technological advances. In short, there are many interpretations and implementations of Type B, B+, and C data, and it would take an enormous amount of time and money to correctly determine an optimal set of conversion procedures for each.

The alternative to establishing precise conversion procedures for all different implementations is to define a general approach that deals with those aspects of Type B, B+, and C data that do not depend on the format of the data or the electronic presentation system that is used. This approach was taken during this task. In this section we will follow this approach by first analyzing technical manual information.

Before continuing, we should note that during this task we were to recommend conversion methodologies for converting Type B and B+ to C. In order to clearly establish these methodologies, we began by defining Types B, B+, and C. In the course of this project, we found it useful to begin with the commonly accepted definitions for B, B+, and C and then found it necessary to modify these definitions. After discussing this matter with others in the IETM community, it was suggested that in order to avoid confusion, we should rename our newly defined data types. These definitions are given in Section 2 above and consist of MA, MA+, MB, MB+, MC, and MC+. In this report, MB and MB+ are equivalent to, but not exactly the same as, B and B+. Similarly, Type MC and MC+ are equivalent to Type C. Type MC+ is equivalent to an electronic technical manual that fully complies with the IETM specifications. Type MC is an intermediate format that is more appropriate for conversion of legacy data to get IETM-like functionality. Type MC is equivalent to an electronic technical manual that fully complies with the CALS standards while having the option of either partially or fully complying with the IETM specifications.

#### **4.1 A Look at the Composition of Technical Manuals**

Due to the large possible set of variant interpretations for any given ETM Type or Class, a comprehensive look at the problem of conversion should begin with a complete understanding of both the data that is contained in a technical manual and also how that data is used in the performance of maintenance duties. As will be seen in detail later, this latter point is particularly important when selecting Type MB or MB+ as the target of a conversion process. In this section, we will take an in-depth look at two approaches for creating paper technical manuals. The reason for looking at paper technical manuals is two-fold: first, paper technical manuals have been around a long time and their structure has matured a great deal, and second, since we are interested in conversion, paper manuals must contain all the same information that would be required to make an ETM or IETM.

In terms of composition, a technical manual contains a wide variety of information that can be presented in an even wider variety of ways. In other words, the information that is given in technical manuals, although governed by one or more appropriate military standards and/or specifications, can be formatted and organized in a variety of ways (thus making it more difficult to specify an exact set of conversion procedures).

In this section we will pay particular attention to MIL-STD-361 (Manuals, Technical: Writing Style and Format Requirements). As noted in [1] the U.S. Army Publications and Printing Command (USAPPC) has been operating a three phase Digital Publications Standards Development (DPSD) program with the goal of streamlining and modernizing the acquisition, preparation, production, and distribution of Army publications. During this program they evaluated and selected the best approach for changing the Army's publications development process from paper to digital. One very important objective of the first phase of this program was to consolidate 41 Army, Navy, and Air Force technical manual preparation specifications into a single tri-service standard.

To see the difficulty that multiple formats can create in the context of conversion, we will ignore the results of the latter phases of the DPSD program for the moment and consider MIL-STD-361 as it was first issued. In MIL-STD-361, the content and organization requirements are given for all levels (U, I, D) of Army technical manuals. As an example, a direct support maintenance manual prepared according to MIL-STD-361 has the following overall format:

1. Front Matter.
2. Chapter 1 - Introduction.
  - Section I - General Information, and
  - Section II - Equipment Description and Data.
3. Inspection and Test of Conventional and Chemical Ammunition or Components Containing Radioactive Materials.
  - Section I - General,
  - Section II - Classification of Materiel Defects,
  - Section III - Visual Inspection,
  - Section IV - Function Testing,
  - Section V - Evaluation of Defects,
  - Section VI - Classification of Defects, and
  - Section VII - Disposition of Lots.
4. Maintenance Instructions Chapter(s).
5. Appendixes.

The most challenging portion of this technical manual structure, in terms of conversion, is the maintenance instructions chapters. These maintenance instructions chapters follow the same basic structure for the following maintenance levels:

- Unit (-20).
- Direct support (-30).
- Aviation intermediate (-30).

- General support (-40).
- Unit and direct support (-23).
- Unit, direct support, and general support (-24).
- Direct support and general support (-34).

Depending on the type and complexity of the equipment, these maintenance instructions chapters may themselves contain the following chapters:

- End item maintenance chapter(s) (the highest level system).
- Component maintenance chapter(s) (significant systems and subsystems).
- Assembly maintenance chapter(s).
- Subassembly maintenance chapter(s).
- Auxiliary equipment maintenance chapter(s).
- Software maintenance chapter(s).
- Ammunition maintenance chapter(s).

Furthermore, each of these maintenance instructions chapters may then contain any or all of the following sections:

Section I:	Repair Parts; tools; special tools; Test, Measurement and Diagnostic Equipment (TMDE); and support equipment.
Section II:	Service upon receipt procedures.
Section III:	Equipment check procedures or equipment/user fitting instructions.
Section IV:	Preventive maintenance checks and services information, lubrication instructions, and mandatory replacement parts.
Section V:	Troubleshooting procedures.
Section VI:	Maintenance procedures.
Section VII:	Preparation for storage or shipment procedures.

These sections may or may not actually exist in the technical manual. For complex equipment, these sections may actually be paragraphs within some subsystem level that is already delineated as a section. Also, when a technical manual contains multiple maintenance chapters, Sections I, II, and IV may be included in a single chapter, while Sections V and VI may each be a single chapter with its own title, and Section III may be included in the chapter titled Maintenance Instructions. Note that this organization is actually somewhat common for organizational-level maintenance manuals in moderately complex equipment such as military vehicles (i.e., trucks, heavy equipment, etc.).

Although not all maintenance manuals are prepared in accordance with MIL-STD-361, the above breakdown is given because it gives a good representation of the variety of information that may appear, in some form, in any particular single or group of technical manuals. Regardless of whether this information is in Type MA, MA+, MB, MB+, B, or B+ format, the ultimate goal is to convert all of the technical manual information to a Type MC or MC+ data format.

To go further with specification of appropriate conversion methodologies, we now require an in-depth look at Type MC and MC+ data, how they are made, and how they can be used to represent arbitrary and imprecisely defined technical manual data formats. With this in mind, we will now describe Type MC and MC+ data relative to the data found in lower-level technical manuals. After that, we will further decompose technical manual information in a way that facilitates the establishment of conversion methodologies, and then show how these methodologies can be used for conversion into the principal components of a Type MC+ data format that is consistent with Type MC.

## **4.2 Conversion Issues with Types MC and MC+ Data**

Type MC+ or IETM data is often referred to as being information-rich or content-oriented. The content-oriented nature of Type MC+ is only one aspect of creating IETM data. Following the IETM specifications, Type MC+ data must be structured according to the CDM. This means that the data must be structured in a multilayered fashion with a generic layer and a content-specific layer [29]. The generic layer gives general characteristics and semantic rules that are supposed to be common across all IETM database implementations. The content-specific layer contains data elements that refer to the actual content of the technical manual. Regardless of how this structure is ultimately implemented, any IETM database which is developed to comply with MIL-D-87269 must follow this two-layer CDM structure.

This CDM structure is modeled, in some sense, after the HyTime standard. The HyTime standard contains architectural forms that are specified in SGML and, among other things, allow for the hyperlinking and scheduling of information. The SGML tags that are used to implement HyTime compliant hyperlinking can have any desired name so long as the element, its corresponding attributes, and the interpretation of its use are in accordance with the HyTime standard. This architectural form methodology makes it possible for HyTime engines to be created to resolve all the addressing information associated with hyperlinking, while at the same time leaving the implementor free to assign memorable names to the hyperlinking related elements that he or she wishes to create. Note the similarity between the way HyTime is implemented and the way content-specific layers are supposed to be modeled after generic layer templates. Clearly, given the wide variety of information associated with technical manuals, this methodology is ideal for dealing with the standardization problems inherent in representing IETM data. In the case of IETMs, one can imagine having an IETM engine that would be able to process any IETM with any correctly implemented content-specific layer (by processing NODEs, NODE-ALTs, etc.). Although the current version of MIL-D-87269 is modeled after and even references the HyTime standard, it is not currently compliant with that standard. The Tri-Service IETM Working Group is expected to correct this problem in the next revision of MIL-D-87269.

In converting arbitrary data formats to Type MC+, we are faced with the problem that, aside from following the CDM structure, there are no real restrictions on the database design or implementation methodology. This point is important, because IETM databases may be built using any database management system with not only no restrictions as to whether it is relational, object-oriented, or otherwise, but also no restrictions on the query language used. Given this fact, the only alternative for the specification of conversion procedures is to attempt to convert the

data to some intermediate format that would be accessible in a popular, non-proprietary, and standardized format. To this end, the most suitable intermediate format for the converted Type MC+ data is SGML. The reasons for this selection are two-fold: first, it gives a starting point for establishing an interface with JCALS, and second, the use of SGML for IETM data is provided for, in some sense, in paragraph 3.3.1 of MIL-D-87269 which reads:

3.1.1 Data Base interchange requirements. When specified, IETMDBs which are to be delivered to the Government under this specification shall be structured and tagged in accordance with the DTDs and the tag set descriptions included as Appendices A through D of this specification.

Paragraph 3.1.1 actually raises two very important issues concerning both conversion and the present interoperability of IETM presentation and authoring systems. The first issue, which has a serious impact on conversion, is that there is no guarantee that a system which complies with the IETM specifications will be capable of importing or exporting its data as SGML files. The second issue is that although the government may specify that an entire IETM database of information be delivered using the DTDs in MIL-D-87269, our research has shown that this SGML IETM database instance is not likely to be directly compatible, for the purpose of presentation, with any other commercial or government-owned IETM system. There are many reasons why this is true, and the most fundamental reason is due to variant interpretations as to how the SGML elements and attributes are to be used for encoding IETM data.

As stated in paragraph 3.1.1 above, Appendices A through D of the IETM database specification contain DTDs and tag set descriptions. These appendices, in effect, give a generic-layer DTD that is the same for any content-specific layer and a content-specific layer DTD that was developed for organizational-level maintenance. Using these DTDs, an entire IETM database instance for organizational-level maintenance manuals can be created in SGML. Also, it should be the case that any two IETM systems that use these DTDs for importing and exporting their data should be capable of directly exchanging their data without any intermediate cross-translation steps. Although this should be the case, because of the lack of a standard authoring guide, it is not currently true among existing IETM implementations.

One very simple example of this compatibility problem concerns the simple referencing of an IETM graphic. IETM graphic implementations are given in the generic layer DTD and therefore should be constant, at least structurally, with respect to any content-specific layer that allows for the inclusion of graphics. Referencing a graphic in SGML requires that the name of the graphic file be given somewhere within the SGML file. Because the graphic file is itself not in SGML and is therefore external to the document, the common SGML convention is to put the name of the graphic file in an ENTITY and then reference that ENTITY. The DTD given in MIL-D-87269 gives no guidelines as to where the graphic filename will reside in an SGML IETM database instance. This has resulted in three interpretations that we are familiar with:

- The AIMSS system appears to place the graphic filename in the name attribute of an element that functions as a graphic primitive (bitmap).
- The Lockheed Fort Worth IETM system puts the graphic filename in as the content of a graphic primitive (#PCDATA) rather than in an attribute.

- The JCALS IETM prototype created a new attribute for the graphic element, called BOARDNO, and the graphic filename becomes the value for that attribute.

Note that although each of these constructs provides a workable solution, they are all different, and none corresponds to the common SGML practice mentioned above. Whether they conform to common SGML usage is immaterial; the important thing to note here is that even if the above systems all used the same content-specific layer, with all the same elements and attributes, their graphic data would be incompatible and would therefore require translation in order to be exchanged. This problem is not the fault of the implementors; it is a defect in MIL-D-87269.

The issues just given concerning MIL-D-87269 have a significant impact on this entire task. Even if Types B and B+ were precisely defined, these issues make it impossible to establish a comprehensive methodology for conversion to arbitrary, specification-compliant, IETM databases. In this task we attempt to overcome these fundamental problems by concentrating on the conversion to Type MC instead of Type MC+. The only real difference between Type MC and MC+ is that Type MC requires that the ETM data be in SGML and also allows for less content-intensive implementations.

### **4.3 Technical Manual Information and Type MC Data**

The format and structure of Type MC data defined in Section 2 were decided upon after looking at currently available COTS products, ETM implementations, technical manual standards and specifications, the IETM specifications, and technical manual conversion efforts. Out of all of these, Type MC data got its principle characteristics from the following:

- Lockheed Fort Worth Presentation and Authoring system,
- LM2500 Gas Turbine ETM,
- World Wide Web (WWW),
- CDNSWC Classes 3 and 4,
- Interactive Authoring and Display System (IADS),
- Aquidneck Data Company IETM Presentation system,
- Digital Publications Standards Development program,
- IETM Specifications,
- CALS Standards, and
- MIL-M-81927B.

In this section we will look at how the structure of Type MC data was established. According to the definitions given in Section 2, Type MC data should be organized according to a system, subsystem, sub-subsystem, subassembly hierarchy with four basic types of information at each level in the hierarchy. This structure is modeled after the content-specific layer given in MIL-D-87269. In particular, the content-specific layer for organizational level maintenance is structured with respect to the system, subsystem, sub-subsystem, subassembly hierarchy of a weapon system. Every maintenance significant level in this hierarchy is termed a system, and each is delineated using an SGML system tag. Moreover, each system in the hierarchy has four basic types of information: procedural, descriptive, parts, and fault information.

These four types of information are also applicable for describing the data contained in all chapters of the direct support I-level maintenance manuals given in Section 4.1, including the maintenance instructions chapter(s) which are representative of maintenance at many levels. It is also worthy to note that during the second phase of the DPSD program, it was discovered that the majority of maintenance manuals, regardless of the level of maintenance, contain similar functional requirements. Furthermore, it was given in [1] that the essential information of a technical manual at any maintenance level, including the one in Section 4.1, can be divided as:

- Introductory information with theory of operation,
- Operator’s instructions,
- Maintenance instructions,
- Troubleshooting procedures,
- Repair parts and special tools list, and
- Supporting information.

Table 4.3-1 below shows how this information fits within the framework outlined by the content-specific layer in MIL-D-87269.

**Table 4.3-1 Classifying Technical Manual Information**

Procedural	Operator’s instructions. Maintenance instructions. Troubleshooting procedures (tests and rectifications).
Descriptive	Introductory information with theory of operation. Supporting information.
Parts	Repair parts and special tools list.
Fault	Troubleshooting procedures (faults).

Recall that one objective of the DPSD program was to consolidate 41 Army, Navy, and Air Force technical manual preparation specifications into a single tri-service standard. In essence, what we have here is a perfect high-level data structure for converting arbitrary technical manual formats, as described in the section on technical manual composition, into Type MC data.

Given that technical manuals contain four principal types of information, we will now discuss the preferred organization of that information. The preferred organization of technical manual information appears to be the work package format. The work package concept is used in [3] and was also adopted for [2]. According to [3], the work package concept is a logical combination of requirements analysis and improved presentation techniques. A work package has three primary elements:

- A logical organization and arrangement,
- An easily understood comprehensive style, and
- A highly visible eye-attracting format.

These three elements are used to improve access to technical information and simplify presentation methods. The work package is further defined in [1] as consisting of all data required for a certain function, and each work package is made up of one or more tasks.

According to [3], a work package technical manual is organized by system or equipment and is arranged by functional task in logical task order sequence. The system/equipment organization corresponds to the system, subsystem, sub-subsystem, subassembly hierarchy that is preferred for Type MC data.

This system, subsystem, sub-subsystem, subassembly hierarchy is also appropriate for [2]. One of the baselines for preparation of a technical manual in accordance with [4] is the Maintenance Allocation Chart (MAC). According to MIL-STD-361-1, both the operators and maintenance procedures of these technical manuals shall be organized in MAC functional group order. To be specific, the MAC is prepared from the equipment top-down breakdown, and a functional group code sequence is used to consolidate and identify those groups on the list that involve maintenance functions. In [4], maintenance functions are limited to those in Table 4.3-2.

**Table 4.3-2 Maintenance Functions**

Inspect	Remove/Install
Test	Replace
Service	Repair
Adjust	Overhaul
Align	Rebuild
Calibrate	

Aside from the setup information, these maintenance functions all include procedural information and some also include troubleshooting procedures (i.e., fault information) and references to part information. All other information could be considered descriptive information for the appropriate system or subsystem. Because technical manuals prepared according to [4] use the MAC as a baseline, we see that by using the four basic types of information as outlined by the content-specific layer in MIL-D-87269, we can reproduce the desired TM structure as well as the information that it contains.

This section has focused on how the overall structure of Type MC data was arrived at. The basic structure was derived from MIL-D-87269, the Army's DPSD program, and a technical manual preparation specification approved for use by the Naval Air Systems Command. By looking at these, we see a preferred technical manual organization and four basic types of technical manual information. The preferred organization is the work package (task-oriented) format. The work packages themselves are arranged according to a system, subsystem, sub-subsystem, subassembly hierarchy similar to a MAC or [5]. Each system then has procedural, descriptive, parts, and fault information.

The preferred structure just discussed is consistent with MIL-D-87269. It is also important to note that Armstrong Laboratory's Logistics Research Division (AL/HRG), as well as others in the IETM community, believe that the DTD for the content-specific layer in MIL-D-87269 is appropriate for maintenance at any level. The findings in this report are in complete agreement with this.

#### **4.4 Information Structures within Technical Manuals**

In the previous section we identified an overall structure and basic information content for the converted Type MC data. Now we shall take a more detailed look at how this information can be decomposed for the establishment of efficient conversion procedures. Recall that this type of analysis would not be necessary if the original Types B, B+, and C had precise definitions with consistent implementations.

In the development of a general approach for the conversion to Type MC, we will set aside pure content, for the moment, and look at technical manual information from a structural standpoint. Looking at the structure only, we see that technical manual information consists of two even more basic types of information: procedural data and non-procedural data. More precisely, procedural data is information that is given as a series of steps that are to be followed, i.e., troubleshooting, assembly, removal, installation, etc. Non-procedural data would then consist of everything else, i.e., descriptive information, part information, etc. Given the nature of maintenance, technical manual information can be divided into two such basic categories. The purpose of a technical manual is to guide and instruct the technician in the performance of maintenance. Putting this information in a manual as a series of clearly defined steps is the most efficient way to provide unambiguous guidance to the technician. Regardless of whether the manual is in paper or digital form, procedural information is, the quintessence of the technical manual.

The idea that technical manual information can be divided into procedural and non-procedural information was obtained by correlating the results in [6] with the apparent information content of technical manuals from various stand-alone as well as combined maintenance manuals. This idea was further reinforced by the conversion effort that was carried out by Lockheed Fort Worth which will be discussed later. The results in [6] show a breakdown of the information typically found in various types of NAVAIR work packages. An example, taken directly from [6] of a procedural vs. non-procedural breakdown for work packages which conform to MIL-M-81927B, is as follows:

- A.1 Procedural Constructs
  - A.1.1 Graphical
    - With embedded text
  - A.1.2 Tabular
    - Fault / probable cause table
  - A.1.3 Textual
    - With embedded graphics
    - Without embedded graphics
  - A.1.4 Other

- A.2 Non-Procedural Constructs
  - A.2.1 Graphical
    - Illustrated Parts Breakdown
      - Self-contained
      - Schematics
  - A.2.2 Tabular
  - A.2.3 Other
    - Introductory
    - Support/Equipment

Integrating all the various breakdowns given in [6], we can see an all-encompassing breakdown of the information contained in NAVAIR work packages:

- A.1 Procedural Constructs
  - A.1.1 Graphical
    - With embedded text
    - Without embedded text
  - A.1.2 Tabular
    - Fault / probable cause table
    - Other
  - A.1.3 Textual
    - With embedded graphics and with embedded tables
    - With embedded graphics and without embedded tables
    - Without embedded graphics and with embedded tables
    - Without embedded graphics and without embedded tables
  - A.1.4 Other
- A.2 Non-Procedural Constructs
  - A.2.1 Graphical
    - Illustrated Parts Breakdown
      - Self-contained
      - Not self-contained
    - Schematics
    - Diagram
  - A.2.2 Tabular
  - A.2.3 Other
    - Introductory / Theory of Operations
    - Support/Equipment

Although this breakdown is an integrated look at NAVAIR work packages, it is also applicable for technical manuals in general. Examination of these procedural versus non-procedural breakdowns reveals that they are composed of three principal components, i.e., text, tables, and graphics. These three components, regardless of the Type or Class of manual that is to be created, are the basic data primitives that are most important to the execution of the conversion process.

The important point of looking at the above breakdown is to become aware of the variety of unique cases that may occur during a conversion effort. It is important that these issues are dealt with, to some extent, at the outset, so that they do not impede the entire conversion effort. Aside from knowing the kind of data to expect, it is also necessary to know if both the authoring and presentation systems can handle these cases in accordance with the given requirements. We will now give two examples of these difficult issues.

The first issue is the graphical procedural constructs. An example of a graphical procedural construct would be a simple troubleshooting diagram, known as a logic tree, that can look very similar to a regular flowchart. The difficulty here, with respect to conversion, is making the decision as to whether this sort of diagram should be converted to a raster graphic, procedural text (a sequence of steps), vector graphic, or some intermediate mixture of these. Treating it as a raster graphic would be the least expensive possibility in terms of conversion. The raster image, however, has the disadvantage that it may be more difficult for the technician to use because of the likelihood of having a small computer screen (relative to the minimum human-readable size of the image). A small computer screen would force the technicians to scroll around the image when traversing steps, and therefore may cause them to lose their place or go to the wrong step by mistake (similar graphic conversion issues were encountered and are discussed in [40] and are further discussed in Section 4.5.5). Treating the diagram as pure text would be equivalent to converting it to fault logic text, and the format of this text would depend on the Type or Class that one wishes to convert to. If converting to Type MC+, then each step in the diagram would fall within a step sequence associated with the troubleshooting information for the appropriate system. In the interest of then becoming compliant with MIL-M-87268, the converted steps would each appear individually as separate frames. Traversing to the next step or frame would then depend on input from the user via a dialog prompt or the evaluation of a precondition.

The next example issue is the tabular procedural constructs. The problems here are similar to those indicated in the previous discussion. The size of the computer screen on a Portable Maintenance Aid (PMA) may not allow the technician to see an entire procedural table at once and consequently may lead to mistakes that would not have been made with a paper manual. One solution to this problem, as above, would be to convert the entire table to procedural text (a sequence of steps). Doing this sort of reauthoring would, in some cases, be more difficult than converting a flow chart. Some tabular procedural constructs are specially structured in the form of tabular logic trees and proper conversion may, in fact, require consultation with subject matter experts. Note that conversion of either this or the above construct to pure text should require revalidation of the troubleshooting sequence itself (which would not be necessary if raster scans were feasible).

#### **4.5 Mapping TM Information Structures into a Type MC Format**

Now we will look at how the basic information structures that make up a TM, with their associated text, tables, and graphics, can be efficiently mapped into a Type MC format. Along the way, we will identify various controversial issues associated with these information structures. These issues are likely to be encountered in any TM conversion project, and so we will provide recommendations as to how they can be resolved. This section does not attempt to define an

optimal set of steps to be taken during conversion; it will instead focus on the steps that must take place, at some point, during the conversion of arbitrary TM information into a Type MC format. Recommended sequences of steps for conversion will be examined in a later section.

In this section, the conversion steps will make specific references to the use of SGML elements and attributes and thus assumes that a DTD consistent with these elements will be used. Although the DTDs given in MIL-D-87269 are imperfect at this time, they provide some elements which are fitting for the level of detail that will be given in this section.

It is important to understand that the actual Type MC data format DTD that is used will determine the ultimate complexity of the mapping. This complexity is directly proportional to the number of SGML elements and attributes required to convey the desired information. An illustrative example of this can be found in [7]. In [7], several thousands of pages of paper technical manual data were converted for the F-16 IMIS. Ignoring the graphic file sizes, the Type MC+ IETM data that was delivered consisted of pure SGML files. Out of these SGML files, only 8% of their content was actual TM text. The other 92% consisted of the SGML markup (tags and attributes). It is also stated in [7] that non-IETM SGML encoded change pages (Type MB data), typically average approximately 88% actual text and only 12% markup. From this, it is clear that creation of IETM specification-compliant Type MC or MC+ data is very complex indeed.

#### **4.5.1 Establishing the Initial Structure**

Assuming that the authoring and presentation systems have been selected appropriately, a suitable first step to mapping TM information into a Type MC format is to ascertain the appropriate system/subsystem hierarchy of all the information that is to be converted. Carrying out this step is very useful for identifying the highest-level conversion tasks that can be done simultaneously (in parallel). Obtaining this information depends on how the manual was originally constructed.

If, for instance, the TM is a Type MA manual that was constructed according to MIL-STD-361, the MAC could be consulted and can usually be found in the TM that contains unit maintenance instructions for the highest level system in question. Obtaining the system hierarchy for NAVAIR work packages is also not difficult, because every maintenance significant system or subsystem should be an individual work package with an identifying number. The system hierarchy can also be obtained from LSA data, if it is available. To remove any ambiguity here, in the converted SGML document instance, a system element will be created for any component (i.e., vehicle, system, subsystem, subassembly) that has any associated technical information (i.e., descriptive, procedural, fault, or part information). This view is also consistent with one of the requirements of the content-specific layer in MIL-D-87269.

The system hierarchy may be different for manuals at different maintenance levels. If the goal of the conversion process is to create only a unit-level IETM, then only elements for the systems or subsystems that have associated technical information at this maintenance level need to be created. If the goal is to combine information from all maintenance levels (i.e., unit, intermediate, and depot), then an all-encompassing system hierarchy should be created. This initial step in the

conversion process leads to a number of issues that ultimately separate Type MB+ functionality from Type MC functionality. These issues will be highlighted in a later section when we discuss the conversion of combined multi-level maintenance manuals.

For now, we will assume that the system hierarchy has been correctly established. Following this, a logical next step would be to insert task elements for each maintenance task for the system in question. This step is necessary because individual tasks associated with the same system usually have their own unique sets of input/setup requirements and procedural information. Like the system hierarchy, the tasks are easily identified, and elements for these can be created using an ASCII text editor or an SGML editor. Note also that, if the content-specific layer in MIL-D-87269 is followed, then a task element must be created for each maintenance procedure. For instance, it is not uncommon for a paper manual to place two independent maintenance procedures within the same task, i.e., a removal procedure and an installation procedure. In many cases, these are put in the same task because they have the same setup requirements. The result of following the O-level DTD in MIL-D-87269 would be the division of these types of combined tasks into individual tasks which reference the same setup information.

When converting to Type MC, it is worthwhile at the outset to have top-level elements not only for tasks but also for each of the four basic types of technical manual information. The top-level elements defined in Appendix B of MIL-D-87269 for encoding the principle O-level technical manual information are as follows:

- Procedural information  
    <task>
- Descriptive information  
    <descinfo>
- Part information  
    <partinfo>
- Fault information  
    <faultinf>

Now, in accordance with the analysis of the information structures associated with TM information, we will look at conversion of the procedural and non-procedural data associated with each system. Differentiation between these two data types is important, because it will become clear that the expertise and hence the cost required to convert the non-procedural data is likely to be less than it would be to convert the procedural data.

#### **4.5.2 Conversion of Non-Procedural Data**

Converting the non-procedural information is the least complex and begins by selecting those portions of the manual that serve as introductory information, theory of operation, equipment descriptions, and task setup requirements. This information is likely to consist of text along with possibly some tables and static graphics. Without going too deeply into the details of the format of the source data, a first pass at converting the data would be to either convert the text to or verify that it is in ASCII, and either convert the static graphic to or verify that the graphic is in an

appropriate CALS-compliant format. The appropriate graphic formats will be discussed in a later section. At this point, the data can then be placed with the appropriate systems and tasks. The introductory, theory, and equipment data would get <descinfo> tags in the SGML, and the task setup requirements would initially be tagged with an <input> tag and then, if the content-specific layer in MIL-D-87269 is followed more closely, additional tags would further divide this information into seven other categories, namely:

- <alert> for warnings, cautions or notes.
- <reqcond> for any required conditions which must be met.
- <person> to identify personnel requirements.
- <refmat> for any reference materials required for the task.
- <equip> for all support equipment that is required.
- <expend> for any expendables.
- <consum> for any consumables.

Care must be exercised when using the O-level DTD in MIL-D-87269, because it is not the case that any of these tags can be simply placed around the ASCII text. Some of the ASCII information may become the values of various attributes whereas others require <text> tags. Also this DTD allows for alternatives to be specified for most pieces of information (i.e., reqcond-alts, person-alts, etc.). As we will see later, these alternatives become especially useful when converting combined, multi-level manuals.

Converting input conditions for a task can be either very simple or very complicated depending on the Type MC functionality that is desired. The main problem here that must be resolved before undertaking a conversion effort is deciding on how to handle the cross-referencing that can take place within the non-procedural setup requirements for each task. A cross-reference is generally defined as a reference from one part of a document to another. In Type MC and MC+ data, some cross-references must be converted such that the information that is being referenced is dynamically formatted and presented on the screen. In other cases, cross-references are not traversed and are merely used so that the presentation system can dynamically assign figure numbers, table numbers, and anything else in the ETM that may be numerically sequenced. This is commonly done for creating paper-based documents that have numerically sequenced chapters, sections, paragraphs, and pages. These simpler kinds of cross-references can be implemented using SGML's ID to IDREF pointing facility, making the creation of Type MA manuals from Type MB files much more efficient.

Examples of cross-references within setup requirements include other, previously defined, tasks that must be performed before starting the current task. According to an analysis done by Lockheed Fort Worth, after converting several thousand manual pages for the F-16, each task, on average, referenced 3.99 other tasks. Cross references may also include part information that must be obtained from a parts list and also references to entities that may require a locator graphic to identify, etc. With an electronic technical manual, there are various ways to handle these different types of cross-references.

The simplest and least expensive solution is to treat the cross-references as they are treated in paper manuals (as regular text primitives). In this case, each of the 7 tags given above could be used within an input tag, and then each item within these tags could be tagged as simply paragraphs or steps. Doing this, of course, would be the least attractive solution. The next step above this would be to take those same text primitives and simply create hyperlinks to link the page, frame, or task containing the referenced data to the setup condition which references it.

There are various approaches to implementing hyperlink capability. Some systems tag the hyperlinked cross-referenced information with either <hotspot>, <xref>, <a>, or <link> tags. Although HyTime is not currently in widespread use, if the system supports it, it is preferable to implement the hyperlinking in a HyTime-compliant format for Type MB+, MC, and MC+ ETM's. HyTime provides many options for hyperlinking information. One of the more interesting capabilities of HyTime is its coordinate space addressing capabilities. These make it possible to create hyperlinks between any two objects (text, table, and graphic) without modifying the original SGML file or its DTD. All hyperlinking information is instead stored in a separate (Web) file that has a unique DTD. The Web file then accesses the original document by referencing it as an ENTITY. This is a very useful capability, but care must be exercised when using it. Changes or updates to the original SGML file may, in some cases, invalidate the Web file. The system should therefore have the capability to automatically update the Web file when the original SGML document is modified.

The most expensive solution, in terms of conversion, would be to make the cross references completely transparent to the user at presentation time. This could be done by either copying or linking to the referenced data. Copying the data is undesirable here, not only because of the data redundancy, but also because it makes it more difficult to incorporate manual updates. Using links to make cross-references transparent requires either a highly structured authoring technique or a sophisticated hyperlinking technique such as that provided by the HyTime standard. Both of these cases would require a very sophisticated presentation system. During this project, no tools were found which would allow cost effective conversion of cross-referenced task setup data to a transparent state (through linking).

Non-procedural data with embedded tables may also be encountered. It should be noted that a table created according to MIL-M-28001 is somewhat different from one that is created according to MIL-D-87269. Therefore, it is not likely that it would be possible to simply copy the MB or MB+ tables. Conversion of these tables is highly automatable, however. Conversion of simple portrait or landscape tables, from scanned paper pages, is not significantly difficult and has been completely automated by some conversion service bureaus. Conversion of more complex tables such as troubleshooting tables or tables with embedded graphics and/or equations is not nearly as easy however.

Another major set of non-procedural information to convert would be that in the Repair Parts and Special Tools List (or Illustrated Parts Breakdown (IPB)). These consist of text, tables, and graphics, and these can all fall within a <partinfo> element. The O-level DTD also uses a <partbase> element whose attributes contain the actual part numbers. If the part information from the source contains static graphics, then conversion of these graphics for a Type MC manual

raises an issue with respect to MIL-M-87268. The difficulty here involves controlling the level of detail when the graphic is displayed. These problems are discussed in depth in a later section that deals with graphic conversion problems.

### **4.5.3 Conversion of Procedural Data**

At this juncture, the conversion process has reached a significant milestone and has dealt with, in a high-level fashion, all of the top-level non-procedural constructs which constitute TM information. Now we will shift our focus to the procedural constructs and their associated conversion issues. Procedural constructs are likely to consist of either maintenance procedures or troubleshooting procedures. Of these two, the troubleshooting procedures are usually the most difficult to convert and are therefore discussed in the next subsection.

Looking at conversion of the maintenance procedures, we find that in a typical manual, the majority of the maintenance procedures are textual with embedded graphics (and without embedded tables). There are basically two types of maintenance procedures or tasks: preventive maintenance tasks and corrective maintenance tasks. These maintenance procedures usually consist of one or more of the functions given in the maintenance function table above. Each of these maintenance functions may involve at least one sequence of steps. It should be pointed out that the content-specific layer given in MIL-D-87269 does not differentiate between preventive maintenance tasks and corrective tasks. Both are simply tasks.

In a simplistic sense, conversion of tasks involves the recognition of steps and step-sequences. Cross-references also appear within steps and, as with cross-references in the setup requirements, decisions on how to deal with these should be made before undertaking conversion. Additionally, tasks, setup requirements, steps, and even graphics may contain effectivities that must be considered before starting the conversion project. As stated in the previous paragraph, the majority of the maintenance procedures are textual with embedded graphics (and without embedded tables). In a Type MC or MC+ ETM, at presentation time, steps should be displayed one at a time. Each step that references a graphic should be displayed together with that graphic. Ideally, only those portions of the graphic that are relative to the current step should be displayed.

It is not always clear how cross-references should be handled within procedural data. For instance, it is not uncommon for a step in one task to reference an entire other task. In a Type MC+ system, these cross-references could, in many cases, be transparent to the technician. For example, one task may have only 3 steps, but one of those steps may reference a task with 20 additional steps. In this case, when the circumstances are appropriate, the referenced task would automatically appear when the step is reached. When the referenced task is completed, the user would then automatically be returned to the original task. Additionally, if the setup information required for the reference task were the same as that for the calling task, it would not be displayed again; the user would simply follow the steps that are presented.

Another example of a dynamically traversed (transparent) cross reference would be the linking of a procedural step that refers to Figure 1 to that figure. In a Type MC or MC+ system, this sort of cross reference should result in both the step and the figure being displayed on the screen together

or in separate windows. Likewise, a step that refers to three different figures will be displayed along with all three figures, each in a separate window, or arranged in accordance with MIL-M-87268.

It is difficult to speculate on a general solution to the problem of converting cross-references. As in non-procedural data, the inexpensive alternative is to simply link the source to the target with a hyperlink tag (or through a separate Web file). When encountering the hottext, the technician could select it and be presented with the information at the target. When converting the Type MC or MC+, care must be taken when implementing hyperlinks. Type MC and MC+ systems function using a state table. If the values in the state table become unexpectedly modified after traversing a hyperlink, this may adversely affect the validity of all subsequent data displayed to the technician. Note that the values in the state table may be rendered invalid whenever the technician performs any acts other than what the Type MC or MC+ system specifically asks for (e.g., following paper instructions, using another ETM with either no state table or a different state table, uncoordinated maintenance of two or more personnel on the same end item, etc.).

Following the identification of steps and step-sequences, the conversion of procedural task data will involve a great deal of conversion of text with either embedded or referenced graphics. As mentioned above, it is necessary to display both the step and its associated graphics. In a Type MC+ manual and optionally in a Type MC manual, the graphic must display only the detail required by that particular step. This function is required by MIL-M-87268. Members in the IETM community have noted that although this capability is difficult to implement, from a conversion standpoint, it is preferred by technicians. During conversion to Type MC or MC+, this function is implementable with the authoring system's context dependent filtering capabilities. This is explained in more detail in a later section when discussing conversion of graphics.

Another issue affecting the conversion of procedural data is the handling of effectivities. The treatment of effectivities can be significantly different, depending on the capabilities of the authoring and presentation system employed. In a Type MC or Type MC+ implementation, effectivities can be completely transparent to the user without significant difficulty. The key to handling an effectivity transparently is using context-dependent filtering capabilities. An example of an effectivity would be a step having 3 graphics associated with it where each is effective for a particular system. A Type MC+ system would have the system identified in the state table, and the step would be linked to 3 alternative graphics specified within a <graphic-alts> tag. When reaching that step, the presentation system would consult the state table and display only the one appropriate graphic to the user.

Obviously, there are many issues involving conversion of procedural data. It can be much more complex than simply tagging each set of steps as a step sequence. Ultimately, the complexity of this conversion depends on both the source and the target data. All issues, including those concerning effectivities, cross-references, and embedded graphics, must be considered before beginning the conversion project.

#### 4.5.4 Conversion of Troubleshooting Information

The conversion of troubleshooting information, depending on how it is rendered in the source TM or ETM, is likely to be the most difficult aspect of the entire conversion process. Proper presentation of diagnostic, fault isolation, and troubleshooting procedures is where most of the user interaction takes place and therefore puts the I in IETM. As noted in [8], troubleshooting procedures which are carefully structured and correctly authored can lead to significantly lower false removal rates and can also greatly increase the ability of a novice technician to successfully troubleshoot a system.

According to Lockheed Fort Worth, lowering the false removal rate will be one of the more significant benefits of the F-16 IETM. Other tests mentioned in [8] show that inexperienced technicians using an IETM were able to correctly isolate faults 100% of the time, whereas experienced technicians using traditional technical documentation correctly isolated faults only 80% of the time. Although converted troubleshooting information is essentially the same whether it is on paper or electronically presented, these tests show that the electronic versions have apparently been presented in a superior fashion.

Results given in [9] further accentuate the advantage that properly converted and presented troubleshooting information can have. Tests of using Type MC+ troubleshooting capabilities with a PMA during the F/A-18 diagnostic demonstration showed that:

- Technicians using the PMA made 54% fewer maintenance errors and 52% fewer total errors than technicians using a paper Work Package (WP).
- Technicians using the PMA completed the troubleshooting tasks in 37% less time than the technicians using the WP.
- Technicians using the PMA for fault isolation took 18% less time than the technicians using the WP.
- Technicians using the PMA to prepare for troubleshooting tasks took 50% less time than the technicians using the WP.

As weapon systems become increasingly complex, there is really no substitute for properly prepared troubleshooting information. The rise in complexity requires the development of more effective troubleshooting techniques. The advantage of using Type MC and MC+ data is their system and content-oriented structure. The more modular and information-rich a document is, the easier it is to create processing algorithms that can use that information in diagnostic routines.

One of the advantages to having a system-oriented structure as in Type MC and MC+ is the ease of incorporating sophisticated diagnostics in the form of a model-based expert system. This capability is very difficult to include in a paper manual. ICAT made by Automated Technology Systems Corporation is one example of a model-based expert system. ICAT relies on the organization of components (sub-assemblies) into functional groups (systems and subsystems), and identification of dependencies and interconnections between each group. Given this, probabilities are assigned to indicate likely fault locations. Specifically, every system and interconnection has a certain probability of being Good and a certain probability of being Bad (these two probabilities sum to one). As the troubleshooting process continues, the probabilities

are automatically modified, the next test is automatically recommended by the software, and the search area is continuously narrowed down. This is an extremely efficient way to troubleshoot highly complex systems. The inputs required by ICAT are not that much different than the data contained in a file that is compliant with the content-specific layer for O-level maintenance in MIL-D-87269. In addition to this, the Mean Time Between Failure (MTBF) data in the LSARs could possibly be used to initialize the probabilities. Following initialization, the probabilities could be refined every time the IETM is used, allowing for faster isolation of faults during future maintenance sessions.

In general, troubleshooting information in a technical manual is usually contained in either logic trees or troubleshooting diagrams. Logic trees can occur in many forms. They sometimes appear in tabular form. Other times they appear as procedural steps that are to be navigated in a non-linear fashion. The branching instructions are sometimes given in embedded tables that accompany the procedural steps. In MIL-STD-361, troubleshooting information may be specified as either logic trees or troubleshooting diagrams. These two categories are further broken down as follows:

#### Logic Trees

- Tabular,
- Branching, and
- Narrative.

#### Troubleshooting Diagrams

- System functional diagrams,
- Test-loop functional diagrams, and
- Fault-isolation diagrams.

Troubleshooting information, although similar in appearance among manuals, may be defined differently in other technical manual preparation specifications or standards. In MIL-M-81927B, troubleshooting information is referred to as troubleshooting diagrams that may consist of the following three formats:

- Logic tree (flow-chart-like boxes),
- Fault logic text (steps with a two-column embedded table), or
- Fault-probable cause (table).

In most procedural troubleshooting situations, there are only two possible outcomes, i.e., Yes/No, True/False, Pass/Fail, etc. In any case, in the content-specific layer in MIL-D-87269, all troubleshooting information is coded up as fault information for the appropriate system. The fault information then becomes a series of tests, where each test is performed via a task and all possible outcomes are then given for that test. Usually, outcomes will either result in the identification of a fault or result in the performance of another test. Identification of a fault will lead to a rectification task.

Conversion of legacy troubleshooting information can be easy or difficult and depends on the authoring methodology originally employed. One need only consult MIL-STD-361-9 for a number of examples of the difficulties that may be associated with converting different types of troubleshooting information. One of the disadvantages of converting either narrative or tabular logic trees, for instance, is that the procedural flow is not always easily discerned. This is in contrast to branching logic trees that often have readily apparent procedural flows.

Depending on how they appear in the Type MA, MA+, MB, or MB+ manual, tabular and branching procedural troubleshooting information (logic text and tables) may require significant reauthoring when converting to Type MC or MC+. In many cases, this will require the conversion of graphical and tabular procedural constructs, that are highlighted in Section 4.4 as being difficult. Ideally, the ETM should ask simple questions and navigate according to the user input, thus obviating the need for interpretation of complex troubleshooting diagrams or tables. This functionality can be implemented in Type MC and MC+ systems with variables and conditional branching. In Type MB+ systems, this functionality would be implemented through htext and hyperlinks.

In MIL-STD-361, troubleshooting diagrams such as system functional and test-loop diagrams are prepared as backup information in support of the troubleshooting procedures. References are usually provided such that when troubleshooting procedures are not successful, the appropriate troubleshooting diagrams can be consulted for further testing. These particular diagrams must show the operation of the equipment, systems, and subsystems as an operational unit. They also show relationships from one system to another, including the wiring from one component to another at the system and the subsystem level. In addition to this, there may be input and output signals between subsystems, controls, and indicators. Test points, waveforms, and instrument hook ups are also shown. In addition, all control indicators that are referred to in troubleshooting procedures must also be shown. Conversion of these particular kinds of troubleshooting diagrams to Type MC or MC+ can be taken to arbitrary levels of complexity. The most cost-effective solution, however, may be simply to insert them as graphics in either a raster or CGM format with hotlinks to the system, task, or any procedural step that references it. The presentation system should then have the capability of panning, zooming, and printing the graphic. The limited size of computer screens may make it difficult to use the larger diagrams effectively.

#### **4.5.5 Conversion of Graphics**

Conversion of graphics can be either extremely easy or exceedingly difficult, depending on the degree of compliance with the IETM specifications. The problem with graphics is that converting to a specification-compliant IETM requires the conversion of static raster, vector, and mixed graphics that were not originally authored as dynamic graphics to dynamically configurable, IETM graphics. A graphic in an IETM is composed of graphic primitives, and these primitives must be in a CALS-compliant format. These requirements are taken directly from paragraph 3.2.3.3 of MIL-D-87269 which reads:

3.2.3.3 Graphics. Graphics (drawings, illustrations) information shall be structured in a hierarchical manner and consist of logically related groups. Graphics shall be composed of a series of illustrations which can be overlaid on each other to build a complete graphic. These graphic "building blocks" are

called graphic primitives. Graphic primitives may be combined to produce composite information which can be referenced and selected. Graphics shall be composed of information represented in accordance with the graphic standards included in MIL-STD-1840.

Even when graphics are static and consist of only one primitive, numerous issues can arise here, especially when converting Type MA manuals to Type MA+, MB, MB+, MC or MC+. When Type MA graphics are converted into an electronic format, they are usually electronically scanned and stored in a raster format. On conversion, it is likely that we will require not only a binary raster format, but also a gray-level raster format and a color raster format. Consulting MIL-STD-1840B (Automated Interchange of Technical Information) Table 4.5.5-1 reveals three relevant data file formats for this:

**Table 4.5.5-1 Data File Formats**

Data File Type	Code Letter
CGM file	C
Raster file	R
Gray scale / color data file	Z

The raster associated with code letter R refers to that which is defined in MIL-R-28002 and is therefore only useful for binary (black and white) raster graphics. CGM files, on the other hand, are preferred for technical manual illustrations but are usually employed for their vector graphic capabilities. Other than this, it is not entirely clear in MIL-STD-1840B which CALS-compliant graphic format is most appropriate for gray scale and color raster files. This is a problem when creating ETMs of any Class or Type. It is an important issue, because the use of gray-scale and especially color photographs is likely to be much more prevalent in ETMs compared to paper manuals. This opinion is reinforced by the AEGIS IETM project that is being managed out of Port Hueneme. In this project, they are creating an IETM for the Fire Control System using the Hughes AIMSS system. Their ETM has lots of photographs compared to a paper manual, because it is very easy to put photographs in a computer compared to putting them in a paper manual. They estimate the cost at one dollar per photo. Callouts are also included on these photos by using the AIMSS built in graphics editor.

The lack of a clearly specified gray scale and color raster format has resulted in the use of various popular de facto and industry standard formats such as TIFF, PCX, BMP, GIF, etc., which are not governed by either CALS or international standards. CGM has the capability of representing gray scale and color raster graphics. The problem here is the fact that the older and more widely used versions of CGM provided no real compression capabilities for any of its raster or raster-like primitives. There is, however, a newer version of CGM, sometimes referred to as CGM Version 3, which supports binary, gray-level, and color raster graphics and also various lossless and lossy compression techniques (this is all in addition to its vector graphic capabilities). The three versions of the newest CGM standard (ISO/IEC 8632:1992) are discussed in [47] and are also described in [10] as:

- Version 1 is a basic drawing and picture interchange format with a simple structure.
- Version 2 includes graphical segments where a segment is a group of graphical primitives that is saved once and named and then repeatedly instanced into the picture with different scaling, attributes, etc.
- Version 3 is a major increase in power over Versions 1 and 2, good for exchanging graphics in any field including: engineering, graphic arts, technical illustrating, and architectural drawing, etc. It can also represent compressed tiled images, define external symbol libraries, and have fully integrated vector and raster data.

In addition to this, Amendment 1 to the CGM ISO standard, among other things, will include a complete and testable specification for certifying generators and interpreters. CGM Version 3 currently has the drawback that it does not appear to be in wide use, but it is capable of handling all the raster and vector graphic capabilities that may be required by any conversion project.

Deciding on an appropriate format is the easiest problem encountered during a conversion because this choice is usually fixed by the capabilities of the authoring and presentation systems which are selected. Much more difficult issues are encountered when creating Type MC or MC+ manuals. Problems arise when attempting to comply with MIL-M-87268. According to paragraphs 3.3.5.3 and 3.3.5.5 of this specification:

3.3.5.3 Graphic density. Graphics shall show only that detail which is needed to support the action being described.

3.3.5.5 Level of detail in graphics. Graphics shall present only the equipment items to which the action statements refer, plus sufficient surroundings to permit the user to locate and isolate the hardware item without error.

Obviously, these can present problems for conversion of static technical manual graphics. The fact is that controlling the level of detail of a converted graphic may require a significant amount of manual reauthoring. Rather than attempting to remove details from a graphic, a less expensive approach that does not specifically comply with the above paragraphs would be to highlight or draw attention (in some way) to the details that are needed to support the action being described. There is also a similar issue with paragraph 3.3.5.15 concerning callouts:

3.3.5.15 Callouts. Callouts shall be provided to identify specific features of interest on graphics. Callouts shall consist of leader lines (with arrow heads) drawn from the index number to the equipment item. The following provisions for callouts shall be followed:

- a. Callouts on graphics shall have a leader line connecting the number to the correct point on the graphic.
- b. A callout shall only be displayed if it corresponds to the step currently highlighted in the text area of the display.
- c. Callouts shall be keyed to the steps of a procedure or individual descriptive statements.
- d. Callouts and identifier numbers shall not be crowded into spaces between graphic elements, and shall be clearly distinguishable from the graphic elements and each other.
- e. Leader lines shall be straight rather than angled wherever possible.
- f. Leader lines should not cross graphic lines if an alternative is possible.

- g. When straight lines are not possible, the leader lines shall have one bend at an angle of 45 degrees. If the leader line has a 45 degree angle, it shall be drawn with the angled end pointing to the graphic. Leader line width shall be differentiated from graphic line width.
- h. Use of part names or other graphic element identifiers may be used as a labels instead of index numbers but the labels shall be identical to that used in the associated text.
- i. Leader lines shall not cross or come in contact with other leader lines nor shall they obscure essential details on the subject matter graphic.

Provision b and c in 3.3.5.15 will, in some instances, be difficult to comply with in the context of a conversion project. The reasons are similar to those given for controlling the level of detail. An additional difficulty with provisions b and c is that many legacy technical manual graphics contain drawings and photographs with a large number of callouts on them. In many cases, portions of the callout overlap or intersect portions of the graphic. If these callouts are removed, the appearance of the base graphic may be modified. Correcting the modified graphic may require both manual editing and consultation with subject matter experts. As mentioned with the level of detail, an alternative to provisions b and c would be to highlight or draw attention to the one or more callouts of interest (which is easily done on computer screens) rather than removing all other callouts from the graphic. Doing this would be less expensive while having the same effect.

Provision g in 3.3.5.15 may also provide difficulties for converting graphics. Paper manual graphics that are to be converted to interactive graphics do not always have leader-line widths that are different from the lines which make up the graphic. Also, the single bend and 45 degree angle restrictions are not always obeyed in paper manual graphics.

It is important to note that technicians prefer simple graphics which contain only necessary details. According to a technical manual consultant for CDNSWC, technicians prefer graphics that not only are simple but have a minimum of "stair-stepping." The "stair-stepping" phenomenon often occurs when paper drawings are scanned at lower resolutions. Lower resolutions are often preferred by ETM system designers, however, because they require less storage space. During conversion, some tradeoff must be made between the issue of resolution versus storage space.

Making a converted graphic suitable for electronic presentation, whether it is static or dynamic, may prove to be a very time-consuming process. Aside from the size and resolution issues, the graphics that are undergoing conversion may be too large for the computer screen that the converted ETM data will be displayed on. This problem was also encountered in [40] and they decided that breaking up large graphics into smaller individual ones was preferable to simply providing the technician with pan/zoom capabilities. As noted in [40] attempting to solve this problem with panning and zooming is not only inefficient for the technician, but also can prevent the technician from seeing a readable version of the "the big picture."

During conversion to Type MC or MC+, one of the issues that will be faced will be deciding on which graphics should be made dynamic and which should be left as static. The graphics of primary concern are those that are embedded in procedural data. Also of concern are graphics in the Repair Parts and Special Tools List (RPSTL) or IPB. This issue can get especially complicated if, in the interest of reducing redundancy, an attempt is made to reference and use

RPSTL or IPB graphics during maintenance procedures. In the interest of conversion, a good trade off here in terms of cost and ETM performance would be to keep the RPSTL/IPB graphics in a static format that is independent from the graphics used in procedural steps, and then make the graphics embedded in procedural steps dynamic for those tasks that are either highly complex or highly critical.

The generic layer DTD in MIL-D-87269 provides for controlling the level of detail in Type MC and MC+ graphics by allowing for preconditions in <grphprim> elements. All the graphic elements and attributes that are needed for creating IETM graphics are given in the generic layer and therefore should be the same across all IETM applications (although as we saw earlier, this is not the case). Conversion of the graphic itself can be done in a manual fashion using an appropriate graphic editor to divide the original graphic file into a group of files. In the simplest case, if the authoring and presentation system can handle raster graphic overlays, each callout and each primitive could be separated out and stored in separate files. A more desirable method for separating out the callouts, requiring less storage space, would be to create text and arrow vectors, store these in separate layers or files, and then delete the raster callouts. Hotspots are recorded in the generic layer as <grphprim> elements with an identifying "type" attribute (e.g., type="hs" or equivalent). The minsize attribute is then used to denote the coordinates of the active region.

The difficulty of dealing with dynamic graphics, of course, when converting to Type MC or MC+, is putting the overlay and hotspot information into SGML in conformance with the generic layer DTD. Products sold by Intercap and Carberry make editing the graphics easy, but currently, no COTS tools have been identified that would make creation of the SGML equivalent easy. The Lockheed Fort Worth authoring system is government owned and provides these capabilities, for CGM binary (black and white) vector graphics only, through its SGML export capability. Dynamic graphics are intimately tied to the textual data which references them. It would not be useful to have a COTS product that can generate the primitive information in SGML unless it is tied very closely with the rest of the authoring system. This is necessary because aside from figuring out appropriate coordinates for display, creating correct expressions within preconditions, for proper implementation of context dependent filtering, can be very difficult if done by hand.

#### **4.5.6 Conversion Automation**

No technology has yet been created or discovered that can fully convert a technical manual in a completely automated or totally unattended fashion. Automated computer recognition in general has proved very elusive, and only limited domains have proved reasonable targets for automated recognition engines. Unrestricted domains, such as those encompassing schematics, engineering drawings, and maps, contain numerous types of entities or shapes that must be categorized during the conversion process. IETM conversions represent an even greater challenge. Not only must the computer recognize a wide variety of entities in the source file (scanned image, ASCII text file, Type MB, Type MB+, etc.), but also the meaning of the manual must be understood. The degree of successful conversion generally corresponds to the level of document understanding that is achieved.

Currently, four projects in progress deal specifically with automated conversion of page-based data into Type MC+ data. These projects are currently all looking primarily at conversion of NAVAIR and also NAVSEA paper technical manual work packages. These projects were all initiated through CDNSWC.

One of the projects is being performed by SAIC. SAIC is researching and developing methods for the automated conversion of technical manual text into a format that is compliant with the IETM specifications. The next project is being done by ManTech International Corporation. ManTech's project is complementary to the SAIC project in that they are researching and developing software tools for the automated conversion of technical manual graphics into specification-compliant, dynamically configurable, interactive, IETM graphics. The next project is being carried out by Martin Marietta. Their project is independent from the previous projects, and they are researching and developing techniques for the automated conversion of both text and graphics (the entire technical manual) into a format that complies with the IETM specifications. The final project is being done by Aquidneck Data Management. They have researched information constructs within NAVAIR work packages and are currently creating conversion programs to map those constructs into a format compliant with the IETM specifications.

Automating the conversion of technical manual text is likely to be based on the recognition of content coupled with the tagging of that content, recognition of direct and implied actions, recognition of direct and implied cross references, sentence restructuring, etc. Automated recognition of the content required by the content-specific layer in MIL-D-87269 alone requires sophisticated, syntactic and semantic, text processing capabilities. In AI circles, this type of processing is typically referred to as natural language processing. Conversion packages based on natural language processing need not be developed completely from scratch. One product in particular, sold by Oracle, called ConText, analyzes the content and attempts to understand the structure of the English text that it reads. The product consists of a server that processes documents generating content markups and client tools that interpret the markups to perform functions in customer applications. According to the literature, this product can assign subject-matter description to sentences, paragraphs, and even entire documents.

Automating the conversion of technical manual graphics is likely to be based on the recognition of graphic primitives, location pointers, and text. Aside from the obvious difficulties here, a large number of side issues make this an especially difficult image processing problem to solve. In particular, technical manual graphics that are scanned from paper pages may have inconsistent line widths, bleed-through from other pages, dropouts, run-together lines, text blurring, etc. In addition to the recognition problem, another obstacle here is extraction of the graphic primitives and callouts while maintaining the integrity of the original graphic. It is almost impossible to anticipate the variety of problems that will be encountered in the automated conversion of technical manual graphics.

Although these projects are faced with seemingly impossible tasks and involve research at the very frontier of image processing and AI, they are progressing. Furthermore, the results of these projects may have a significant impact on many of the issues discussed in this report. The effects

are likely to be positive in that, as the conversion process becomes more automated, less manual intervention will be required; hence, the overall cost of the endeavor should decrease. At this time, there is really no way of estimating the extent of the benefits that these projects may yield.

#### **4.6 Functional Capabilities of Authoring and Presentation Systems**

Because of the wide variety of possible hypermedia implementations of ETM systems, it is necessary to look at the functional capabilities of the systems that both create and present the ETM data. The functionality provided by these systems is the main difference between Type MA+, MB, MB+, MC, and MC+ systems. These functional differences can have a significant impact on how the data gets converted. In this section, we will take a detailed look at some popular hypermedia systems that are in use today. By looking at these systems, fundamental differences between those of different Types or Classes will become apparent. This section is necessary because there are a lot of misconceptions about the current capabilities of hypermedia systems.

Two important functions that many hypermedia systems currently lack are the ability to implement conditional branching and the ability to evaluate, assert, and react to runtime variables. Proper implementation of a specification-compliant Type MC or MC+ IETM requires that the hypermedia system have this ability. The DTDs given in MIL-D-87269 allow for runtime variables by allowing a property element (whose content is treated as a variable) to have values. The properties and their associated values are then all maintained in a state table at presentation time. Conditional branching is implemented through content-specific elements that are modeled according to the IF-NODE template. These IF-NODEs have expressions that evaluate properties having values in the state table.

A system with the ability to do both conditional branching and runtime variables allows for much more efficient authoring of the troubleshooting procedures and alternative information that are required by most ETM systems. Troubleshooting information is often given in the form of troubleshooting trees. Traversing from one troubleshooting test to another is more efficiently done through conditional branching and variable checking than through simple hyperlinking. Without runtime variables, troubleshooting nodes with multiple output possibilities could not be dynamically traversed based on selections made more than a single node beforehand. Selections would have to be made at every node. The capability to efficiently author alternative information is important and necessary when creating ETMs for multiple levels of maintenance, multiple skill levels, etc.

##### **4.6.1 Lockheed Fort Worth Presentation and Authoring System**

As just noted, not many hypermedia systems have the capability to provide dynamic runtime functions such that all the variable and branching information is extracted strictly from the SGML-tagged data. One system that has this capability is the Lockheed Fort Worth Presentation and Authoring System that was constructed for the F-16. This is a Type MC+ system and was the first system that was built specifically to comply with the current version of the IETM specifications. The IETM data is all maintained and authored to a Sybase database. The authoring methodology is very efficient and requires no knowledge or use of SGML.

The interesting thing about the Lockheed system is that when the IETM data is to be used with the presentation system to perform a maintenance task, the data is first extracted from the database as SGML files. These SGML files comply with most aspects of both the generic and content-specific layer DTDs given in MIL-D-87269. They are completely devoid of any sort of format information or frame tags. The presentation system then reads and interprets the SGML file just as an interpreter would interpret a computer program. The Sybase database is neither required nor used by the presentation system. When this system was designed, the goal was to build a system that required a minimal amount of COTS software in order to reduce the cost of supplying it to subcontractors. The only COTS package that is required is a Sybase database.

During the presentation, both text and graphics are dynamically formatted for screen presentation in accordance with the requirements in MIL-M-87268. This is a significant proof of concept for the SGML encoding of IETM data. The reason this is significant is because it is thought by some that IETM's, which comply with the current version of IETM specifications, cannot be implemented with open standards such as SGML and HyTime. Obviously they can be implemented with open standards and Lockheed has done it. Besides having an appropriate DTD and presentation system, the key is to keep track of runtime variables with a state table, or something similar, maintained in the presentation system.

#### **4.6.2 The World Wide Web**

The largest and most widely supported type of hypermedia system available today are those which are consistent with Type MB+. The most widely supported Type MB+ hypermedia system at this time is the WWW. WWW is currently a perfect example of a scroll-oriented Type MB+ system. The World Wide Web was originally designed and prototyped at CERN (European Laboratory for Particle Physics). The World Wide Web requires that all data be encoded in HyperText Markup Language (HTML). Note that HTML is not the same as HML which is supported by InfoAccess. The World Wide Web and the use of HTML are important to this task, because they are currently undergoing explosive growth on the Internet. It is not difficult for anyone with access to the Internet to create hypermedia documents (with text, graphics, audio, and video) that can be accessed and read by anyone in the world. HTML documents are generally used in conjunction with a presentation system known as Mosaic. Mosaic is a public domain software package that was originally developed by the National Center for Supercomputing (NCSA) and is currently available in versions for X-Windows, Microsoft Windows, and Apple MacIntosh [11].

HTML documents can be created by following the HTML SGML DTD. HTML documents can be created with Type MB+ capabilities. It should be noted, however, that, although an SGML DTD for HTML exists, HTML currently uses a hyperlinking and addressing scheme that does not follow the HyTime standard and also does not follow commonly used SGML practices. Making HTML an accepted and fully conforming application of both SGML and HyTime is addressed to some extent in [12]. In terms of linking, hyperlinks in HTML documents are represented with the <a> tag (where "a" means anchor). The target of the link is then specified in an "href" attribute for that <a> tag. In HTML, the target may appear within the same document or it may be within some other document somewhere else on the Internet. Linking to targets across the Internet, and

therefore around the world, is accomplished using the URL (Uniform/Universal Resource Locator) syntax. In a nutshell, the URL provides a very simple yet very powerful way of linking to any file anywhere in the world on the Internet. The difference between the way links are specified in HTML and the way they would be handled in a HyTime-compliant format begins with the way URLs are used in the href attribute. Specifically, in SGML/HyTime, external files are conventionally accessed through entity references. In other words, instead of including a path and filename as an attribute of an element (as HTML does with href and URLs), the path and filename would instead be given within an entity. This problem is similar to the graphic filename problem mentioned in Section 4.2.

Regardless of a few minor issues, HTML is still classified as scroll-oriented Type MB+, it is extremely popular, it is very widely used, and HTML documents are usually maintained in a neutral format (rather than in some database system that imports/exports this neutral format). HTML does not allow for conditional branching or runtime variables and therefore is not sufficient for implementing Type MC and Type MC+ data. The fact that it is scroll-oriented also makes it more difficult to author data such that it is presented in small blocks, but it is still possible. A newer version of HTML will soon be released; it is expected to have more advanced capabilities. Also our own in-house tests show that it is possible to create a system and task-oriented ETM that looks very much like a Type MC or MC+ system.

It should also be noted that HTML is currently undergoing revision. There are currently three levels of HTML. HTML level one is mandatory for all WWW clients. HTML level 2 is a superset of level 1 and includes forms for user input; it is not clear how wide the current support is for level 2. HTML level 3 is a superset of level 2, and among other things will include support for tables, figures, and equations. As with level 2, it is not clear how wide the current support is for level 3.

#### 4.6.3 Interactive Authoring and Display System (IADS)

One of the hypermedia systems that is government-owned and was made for the Army is IADS. IADS is a perfect example of a frame-based Type MB+ system. The key to IADS is the fact that it relies on seven reserved tag names as listed below in Table 4.6.3-1.

**Table 4.6.3-1 IADS Reserved Tag Names**

IADS Reserved Tag Names			
FILE	FRAME	BUTTON	HOTSPOT
ACTION	GRAPHIC	NOVICE	

These tags may be used with or without an SGML tagged document that was prepared according to a specific DTD. If a DTD is used, every element will be given a style using a stylesheet. On implementation, the data is all placed in FRAME tags and presented as a linear series of these frames. Traversing from one frame to the next requires hitting the NEXT key. Frames can be accessed in a non-linear fashion only through hyperlinks and hotspots. The IADS system does not

allow for conditional branching or runtime variables and therefore is not sufficient for implementing Type MC and Type MC+ data. Although this is the case, it is certainly possible to create a system and task-oriented ETM that looks very much like a Type MC or MC+ system. Two good examples are the Patriot Missile ETM and the Avenger ETM which are both implemented on IADS.

#### **4.7 Multiple Levels of Maintenance within Single Paper Manuals**

Electronic technical manuals of any Type or Class may be created for maintenance at any level (i.e., U, I, D, etc.). Serviceable systems with low to moderate levels of complexity may have all maintenance, part, and troubleshooting information, for all maintenance levels, contained within a single paper TM. Systems with higher levels of complexity, on the other hand, are likely to have separate paper TMs for each maintenance level as well as individual manuals for parts, troubleshooting/fault isolation, and maintenance tasks. Also, especially in the case of a work package format, there may be separate paper TMs for each maintenance-significant system, subsystem, etc. In this subsection we will pay particular attention to single TMs which have maintenance information for multiple maintenance levels.

When a single paper TM contains information for all maintenance levels, the problem of converting it to a frame-oriented ETM has a number of possible solutions, some more efficient than others. Ideally, an ETM that is used for maintenance at a particular level should display only the information that applies to that level of maintenance. In other words, if, for example, a paper manual contains an organizational maintenance task and an intermediate maintenance task on the same paper page for the same system, when using the ETM, only the organizational task should be displayed if doing organizational level maintenance, etc. Implementing this capability is straight forward when the ETM system has alternative-information/context-dependent-filtering capabilities (i.e., Type MC and MC+) but it is not straight forward when using other non-page based ETM systems (i.e., frame-oriented Type MB+).

One possibility for creating a frame-oriented Type MB+ manual here would be to separate the original paper manual information for each maintenance significant level. The resultant ETM would then essentially consist of multiple, independent, electronic manuals (i.e., one for each level). An alternative to having separate Type MB+ ETMs for each level would be to have a single ETM with the option of selecting the appropriate maintenance level while using it. Since Type MB+ systems generally do not have context-dependent filtering capabilities this would be implemented through hertext and hyperlinks. In this case, the information or tasks for each maintenance level would be listed on a menu and the technician would simply select, and be presented with, the desired information.

In a Type MC or MC+ system, an ETM for multiple levels of maintenance could be created very easily. At the start of the maintenance session, the presentation system could prompt the technician, asking for the maintenance level. This could then set a value for a property (i.e., LEVEL) within the state table. The authored data could then have preconditions based on the value of LEVEL. During presentation, the preconditions would be evaluated by the presentation

system, transparent to the user, and only the data or frames with preconditions which evaluate to TRUE would then be displayed.

ETM implementations are currently in their very early stages. Because of this, providing ETM information for multiple levels of maintenance has not yet been encountered, and therefore not considered a problem, in many ETM implementations. Due to the inherent benefits, it is only a matter of time, however, until a true Integrated Weapon System Database (IWSDB) environment is established. Once this environment is established, manuals for all levels of maintenance will be in electronic formats; and how this data is handled will ultimately depend on the capabilities of the authoring and presentation system. This subsection was included to bring attention not only to the issue of single, self-contained, legacy manuals prepared for multiple maintenance levels, but also to point out how Type MC and MC+ systems can efficiently solve this problem.

#### **4.8 Planning an ETM Conversion Project**

Properly planning a conversion project is as important as the conversion process which will ultimately take place. When converting between Types or Classes of ETMs, conversion is more than just data translation. Because systems which comply with the IETM specifications are rare and the specifications themselves are still being developed, it would not be useful to discuss planning or conversion procedures without also discussing issues associated with the systems that the data is being converted for. Discussing conversion issues alone between any two systems is only possible when those systems are well specified.

Even when the source and target systems are well specified, undertaking a conversion project requires a significant amount of planning and preliminary analysis. Therefore, in accordance with this task, conversion to a format which even approaches compliance with the IETM specifications will require much more preliminary analysis than otherwise would be necessary. Initially, only two things are known for sure. First, we are most interested in the steps necessary to convert from one electronic format to another (MB, MB+ to MC). Second, in this report, the electronic data is assumed to be in SGML, and therefore conversion of these electronic formats involves the mapping of one DTD into another.

Before the project begins, a needs assessment should be carried out. It is extremely important to identify whether an electronic manual that is more than simply a page turner is needed at all. If an electronic manual is needed, then the next critical step is to decide on what Type or Class of manual is best suited for the system in question. At this point, cost becomes a factor. In order to proceed, an estimate of the cost of the conversion effort must be determined. A reasonable estimate of the cost cannot be obtained until both a suitable authoring and presentation system are selected. It is a misconception to think that in a conversion project consideration should only be given to the presentation system. The reality is that authoring and presentation systems must be considered together at the very beginning of the conversion project. The capabilities of the authoring system can have a significant impact on the time and hence the cost to carry out the conversion project.

System selection is important because there can be a significant cost boundary between Type MC and MC+ compared to the other levels. This cost boundary exists because currently there are not nearly as many systems with built in Type MC and MC+ functionality compared to the lower levels. There are Type MB+ systems which can be modified to achieve Type MC and MC+ functionality, but it is preferable to have a system that has these capabilities built in. If the source data is either Type MA, MA+, or MB, it is worthwhile to create cost estimates for both a selected Type MB+ target and also a Type MC target system. Details on estimating the cost are given in a later section. It is important to consider during system selection exactly how the information given in Section 4.5 would be represented. Primary consideration should be given to the representation of troubleshooting information and graphics. Making the decision to convert to an ETM of any Type or Class may also depend on such factors as the phase of the defense system's life-cycle and whether or not the LSAR database follows a numbering system similar to MIL-STD-1808. These two items, in particular, are part of a Type MC+ viability decision flow chart that was developed for the Navy and is given in [46].

Having access to an LSA database can greatly enhance an ETM creation/conversion project. In the F-22 program, for instance, they have approximately 30 main databases (out of hundreds all over the country) and the LSA database is considered the major core database. They intend to create a Type MC+ ETM, using a modified version of the F-16 ETM system, by copying much of the information they need from their LSA database and importing it into their ETM authoring system. They also noted that their program was the first to implement the CALS-compliant LSAR standard. The F-22 LSA process is executed in accordance with MIL-STD-1388-1A and is documented in accordance with MIL-STD-1388-2B.

After carrying out the needs assessment, cost estimation, and getting approval to continue, the actual conversion project may begin. At this point, the next set of steps are similar to that which would be required by any normal conversion project. Many conversion service bureaus do nothing but convert data from one format to another. The three most widely known conversion service bureaus that were contacted during this project were Data Conversion Laboratory (DCL), Gateway, and Docucon. At the start of any conversion effort, it is recommended that a conversion service bureau be contacted for advice. DCL, although not as big as either Gateway or Docucon, has performed tens of thousands of conversion projects over the years, and they have created a very detailed workbook for planning conversion projects [13]. The items in this workbook can be generalized to apply to most conversion projects and are most suitable for conversion from one static document format to another such as WordPerfect to Type MB, etc.

Carrying out the conversion project now requires a detailed analysis of how the information in the source document would be mapped into information structures that are both required to make the electronic technical manual and are also supported by the selected target system. Section 4.1 in this report discussed the source document information at a very high level. Section 4.5 discussed the desired target information structures at a more detailed level. In order to successfully carry out a conversion project, it is necessary to determine an exact mapping for every unique item that will change when going from the source to the target. This mapping must be done at the very beginning. If the authoring and presentation systems were properly selected after careful study of Section 4.5, then there should be no significant surprises during the mapping stage. Note that the

mapping should not be confined to paper analysis. Every unique mapping instance should be identified and manually performed using the authoring system, and then its appearance should be verified using the presentation system. In any case, the mapping process must, at the very least, provide detailed guidelines on all the information constructs identified in Section 4.4.

This mapping process will reveal the ultimate complexity of the conversion project. As stated in [13] and shown in Table 4.8-1, there are principally five kinds of mappings, and each has an associated complexity:

**Table 4.8-1 Mapping Complexity**

<b>Mapping</b>	<b>Relative Complexity</b>
1 to 1	Low
many to 1	Low
1 to many	Medium to High
0 to 1	High
0 to many	High to Unsolvable

An example of a 1 to 1 mapping would be converting all <item> tags to <step> tags. An example of a many to 1 mapping would be converting all <item>, <step>, and <p> tags to <para> tags. A 1 to many example would be the reverse of a many to 1 and is obviously more difficult because it may not be apparent which <para> tags should become <p> tags, etc. The 0 to 1 and 0 to many mappings mean that the source document contains no tags for information that is required to be tagged in the target document. A good example given by [13] of a 0 to 1 mapping that may be easily solved is the tagging of part numbers. In this case, a dictionary of part numbers would have to be created and then the entire document would have to be searched for each instance of those part number strings. Each one that is found could be tagged as a <part>. More difficult examples of 0 to 1 mappings include words that are placed into an index. According to DCL, automation of this kind of index conversion will result usually in about 70% to 90% accuracy in normal documents. The 0 to many mappings are the most difficult and may require significant amounts of processing and/or manual intervention.

Conversion of ETM Types and Classes from lower levels to more information-rich levels can be considered highly complex in terms of the conversion process, because it will obviously involve a lot of 0 to 1 and 0 to many mappings. If all authoring and presentation systems were the same price and had a similar user base, then the cost of the conversion project would be based principally on the mapping between the source and target DTDs. In other words, converting from Type MB to Type MC may be no more expensive than converting to MB+, depending on the DTDs that are used. It is also interesting to consider mapping from Type A data. After scanning in the graphics and either using Optical Character Recognition (OCR) or retyping the text, the cost of converting to Type MB or MB+ may not be significantly different than going to Type MC; it just depends on the requirements of the target DTD. Some Type MB documents that are destined for paper output, in fact, require a lot of content tags. These content tags are applied

when the author wishes to use a FOSI and apply special formatting instructions to the data (i.e., put it in bold faced 18 point font).

After a detailed analysis of both the source and targets has resulted in a comprehensive set of mapping guidelines for all the necessary information structures, the next step is to determine how the mapping will take place. In other words, it must be determined which mapping procedures can be done in house by the contractor and also which procedures should be done by a conversion service bureau. It is recommended that a conversion service bureau be contacted to assist with the planning. Conversion service bureaus maintain a wide variety of conversion software tools which they sometimes modify as necessary to carry out conversions. They have the most experience in determining which software tools and what mix of manual and automated effort will provide the most cost-effective solution to the conversion problem at hand.

Conversion from one electronic tagged file format to another may involve the use of autotagging software. The contractor, or whoever is carrying out the conversion, may be interested in using the autotagging software to do the conversions in house. It should be noted that there is a common misconception concerning the capabilities of autotaggers. Autotagging software must be set up or customized to carry out the task at hand. It is noted in [13] that according to their tests, the work required to customize an autotagger versus writing conversion code directly in C was about the same. This is very interesting, because conversion programs written in general purpose programming languages such as C are much more flexible and therefore more capable of handling those unusual cases that always occur and often represent the majority of the effort. The point is that autotagging software may or may not be worth the investment, and some determination should be made as to what exactly that software needs to do before it is purchased. Our opinions are consistent with those expressed in [13], because two of the more popular autotagging software packages that we looked at during this project appeared to require about as much effort to learn as learning a new programming language.

After carrying out the initial mappings and deciding who will carry out which parts of the conversion, the remaining phases involve implementation and quality assurance and are straightforward. The first of these is the setup phase. This may involve purchase of appropriate hardware and software. It also may involve establishing teams to carry out different parts of the conversion process in an efficient manner. Following the setup phase comes the actual implementation phase where the conversion work is carried out. Implementation will be followed by extensive manual cleanup, tailoring, and inspection.

#### **4.9 Methodologies for Conversion**

It is impossible to establish a concise conversion methodology, beyond what was discussed in the previous sections, that is appropriate for all situations. It is advantageous, however, to examine other successful conversion efforts before undertaking a new one. In this section, we will review several significant conversion efforts involving Type MC, MC+, and similar data. The previous sections are useful for planning and determining mapping requirements. This section will focus on the execution stage that should take place following the planning and mapping stages. We will begin by discussing the conversion methodology employed for the F/A-18 which yielded an earlier

version of Type MC and Type MC+ data. Following that, we will look at the conversion methodology used for the F-16 which also resulted in Type MC and Type MC+ data. We then review one of the biggest and most cost-effective conversion efforts of its kind which was done for the LM2500 Gas Turbine and resulted in data similar to Type MB+ and Type MC data. Following this, we look at a conversion effort carried out for the AEGIS Fire Control System by the Port Hueneme Division of the Naval Surface Warfare Center. The last two conversion methodologies we review are those associated with the Newport News Shipbuilding and the Joint Integrated Maintenance Information System (JIMIS). After looking at all of these conversion methodologies, a general conversion methodology for ETM data, based in part on the preceding sections, is discussed.

#### **4.9.1 Conversion for the F/A-18**

The first conversion scenario developed for data which conforms in some sense to organizational level Type MC and MC+ data was carried out by AL/HRG at Wright Patterson Air Force Base, OH. The data was converted for the F/A-18 Flight Control System (FCS) in support of the IMIS program. The DTD (with some Type MC and MC+ capabilities) that was used here was a modified CDM Version 5.3, a forerunner of the CDM in the current version of MIL-D-87269 [9]. According to [9] almost 900 pages of existing paper-based legacy data were converted to this Type MC+ format during this project.

This conversion effort was divided into several distinct parts:

- Conversion of procedural (task) data,
- Conversion of diagnostic data, and
- Database integration and simplification of converted graphics.

The conversion work was divided among several individuals and groups [9]. McDonnell Douglas Government Aerospace - East (MCAIR) converted the procedural (task) text and graphic data for the FCS, its subsystems, and related systems. Systems Exploration, Inc. (now NCI Information Systems, Inc.) authored the dynamic diagnostic data for the FCS. AL/HRG authored the dynamic diagnostics for related systems, simplified graphics, and was in charge of database integration from the three authoring sites.

The textual portions from the task data were converted by re-keying the data using the CDM Smalltalk Authoring System (STAS) created by AL/HRG. The AL/HRG system employed four basic editors:

- Text editor,
- Reference list editor,
- Table editor, and
- Graphic editor.

During authoring, windows are displayed as a collection of these editors. The system determines which editors are needed for authoring particular elements and also indicates when data is

required or optional (similar to the way typical SGML editors work). Once authored, the system can export the data as SGML. At presentation time, the presentation system parses the SGML file to present it.

The MCAIR portion of the project involved the conversion of 60 page-based work packages [40]. These consisted of 854 pages where there were 205 graphic pages and 649 text pages. In the end they delivered 66 SGML tagged flat files, conforming to CDM Version 5.3, which consisted of 3,949,741 bytes of SGML-tagged text data and 1,167 CGM files amounting to 31,754,191 bytes of data. Out of all of these CGM files they had 82 CGM files which they termed "Type C" image files (each file contained more than one image) which contained a total of 277 separate images. Their "Type C" image files are graphics that resulted from mixing and matching graphic primitives.

The conversion methodology developed by MCAIR is detailed in [40]. Their conversion effort was separated into graphics conversion and text conversion. Before conversion, a general preprocessing step was performed on both the graphics and text in the paper manuals which consisted of marking up a copy of each work package by hand to indicate what information would be inserted into the database.

The graphics pre-processing involved assigning control numbers to graphics such that they could easily track the resultant CGM files. It also required a technical writer to circle and label those graphical elements that would be included in the database and those elements that would be ignored. Note that this was carried out after they decided that they would break up their graphics when electronically displayed. Breaking up the graphics was considered to be the best solution for electronic display purposes because they were initially dealing with graphics that were originally formatted for an 8.5 x 11 inch paper page rather than a relatively small (and lower resolution) computer screen.

After pre-processing, the graphic conversion process involved the use of a continuous-feed scanner for scanning in the graphics. After scanning, software then automatically converted the raster scan file into a vector file and then performed OCR on the vector file to recognize the text. This was then followed up with some additional post-processing. They eventually discovered that the converted text (that was within the graphic) was problematic when displayed on the PMA; and also the level of detail in the original graphics increased the size of the stored files and consequently increased the amount of time it took to display the graphic on the PMA. For these reasons, they decided to re-author the graphics for display on a small, low to medium resolution display. They increased the text size, callout arrow sizes and also broke up graphics which had more than one detail per page so that they could be displayed more clearly and on separate screens.

The text pre-processing involved marking up the paper manuals to indicate the appropriate CDM elements that the paper information belonged to and also the text that would be deleted (i.e., references to figures numbers, etc.). The text-pre-processing also involved establishing text to graphic links. After the graphics were broken up and named in the graphic pre-processing step, the technical writer spread out the marked up graphics on a desk and began reading the associated

text. After reading each step, the marked up graphics were checked for any elements that should be linked to that step (the links that were found were then marked on the paper manual). At this time, things like references and effectivities were also marked.

After pre-processing, the text conversion was carried out. Originally they had expected to convert the textual data from its existing Xyvision electronic format to the CDM Version 5.3 DTD using rule-based conversion routines. Their main problem however was that the CDM was changing during this time. Every little change made to the CDM forced them to spend weeks re-programming their conversion software. They eventually entered everything into a database manually using AL/HRGs STAS. In [40] they state that they can automatically create a 90% complete Version 5.3 CDM-compliant SGML file from existing page-based data. Furthermore they state that the remaining 10% is very labor-intensive due to the links required by the CDM that did not exist in the paper (and therefore must be added manually).

#### **4.9.2 Lockheed Fort Worth Conversion for the F-16**

Lockheed Fort Worth has converted over 7,000 pages of page-based technical manual data into an organizational level Type MC+ format for the F-16. Approximately 4,000 of these pages were converted for the AL/HRG IMIS. All the technical manual information was obtained in an electronic file that was fraught with typesetting codes. The original technical manual data was similar to a chapter-based work package and was prepared according to MIL-M-38784 (Manuals, Technical: General Style and Format Requirements) and MIL-M-83495 (Manuals, Technical: On Equipment Set, Organizational Manuals; Detailed Requirements for Preparation of).

The Lockheed conversion was carried out by technical writers. They peaked at 15 technical writers. The technical writers were trained for approximately four hours before they carried out the conversions on their own. Once trained, a technical writer generally required additional assistance for about one month while converting "easy to convert" data.

The first step carried out in this conversion effort was to use a preparer to extract the text from the electronic files. The page-based electronic files sent to the preparer had either procedural steps or descriptive information. The IPB data and the troubleshooting information was not included at this point. Lockheed determined that the electronic file contained many step sequences (compared to instances of descriptive information). They therefore programmed the preparer to insert <step> tags around every independent block of textual information. In addition to this, Lockheed took advantage of the fact that the systems and tasks in the page-based manuals followed a structured numbering system similar to MIL-STD-1808 (System/Subsystem/Subject Number (S/S/SN) Numbering System). This enabled the preparer to automatically insert <task> and <system> tags. Because the tasks were identified, the preparer could insert step sequence tags at the beginning of each task.

Following the parsing step, the technical writers manually replaced the step tags that were placed on the descriptive information with <para> tags. This did not require a significant amount of work because a relatively small percentage of the data was descriptive information. To be precise, after converting 3,733 pages, their final converted IETM SGML data contained 8,209

step tags, 2,351 step sequence tags, 1,010 paragraph tags, and only 219 descriptive information tags. After inserting <para> tags for the descriptive information, the tagged data files were parsed into Sybase database tables.

The graphic files from the paper-based manual were available in various electronic formats. All graphic files were loaded into a graphics package called Autotrol and were converted to CGM binary vector graphics. A catalogue of all graphic images was then created so that they could be easily accessed during authoring.

Once the data was placed into the database, the authoring system was then used for detailed organization and structuring of the data. In other words, the alternative information was delineated, context dependent filtering capabilities added, links established, etc. The authoring system enables all of this to be done in a tightly controlled, high-level fashion. It would simply be too complicated to put this information into the Type MC+ SGML file by hand. Also, the CGM graphic files were loaded into the authoring system and edited. The callouts were extracted and keyed with appropriate tasks and steps for context dependent display.

The IPB was converted in a non-automated fashion. Due to the customer-directed, limited implementation of parts data, building CDM-compliant parts authoring tools was not cost effective. As with the task and descriptive data, all graphics were available in an electronic format that was converted to CGM through Autotrol. Additionally, all part information was entered into the authoring system by hand. Note that the IPB data was all converted independently from the rest of the data. Furthermore, the graphics in the IPB were left as static and were not tailored for context-dependent display.

Converting the diagnostic information was the most difficult part of the entire effort and this can be attributed to the many additional data needs required to support AL/HRG's IMIS Diagnostic Module (IMIS-DM). All troubleshooting data was converted by hand; the troubleshooting tables, in particular, were the hardest to convert. All troubleshooting information consisted of either tables or trees, and all was converted into a logic tree format. Following the content-specific layer for O-level maintenance in MIL-D-87269, each step in the logic tree became a new task.

During the conversion effort, it became apparent that having experienced authors (data converters) was very important. Lockheed estimated that it took approximately 2 to 2.5 hours per page for conversion of the task, parts, and descriptive data. This estimate includes everything, the time required for preparsing, tailoring of illustrations, data organization, etc. Lockheed also estimated that it took approximately 17 hours per page to convert the troubleshooting information (see the discussion of IMIS-DM authoring requirements, above).

Given the nature of legacy data and the multitude of functions provided by the authoring system, it is not expected that the amount of time could be significantly reduced by using any other system or COTS product to convert to Type MC+ data. Aside from searching for sophisticated research solutions to automating conversion, the only way to significantly reduce this time is by reducing the requirements imposed by the IETM specifications and reduce the complexity of the content-specific layer DTD for O-level maintenance. The most significant issue concerning Lockheed's

experience, which reveals the true complexity of MIL-D-87269, is the fact that when their IETM data was exported as SGML, they found that 92% of the SGML file consisted of tags and attributes, while only 8% consisted of actual technical manual text.

### 4.9.3 LM2500 Gas Turbine Conversion Effort

The LM2500 Gas Turbine Systems IETM required the conversion of over 110,000 pages of technical information and documentation [8]. Out of this, approximately 80,000 pages consisted specifically of legacy technical manual data [14] and approximately 14,000 foldouts or graphics [8]. The AN/BSY-2 Combat Systems IETM approach, developed by Martin Marietta, was chosen for the Gas Turbine Systems IETM. One of the important factors in making this selection was the fact that the AN/BSY-2 was the only approach that included training capabilities. All technical information and documentation for the Gas Turbine Systems and the training curricula for the Gas Turbine Technician "C" School is included in the IETM. Although these IETMs do not comply with the CDM structure in the IETM specifications, according to [8], the Gas Turbine Systems IETM was extremely economical and sets the standard for creation of IETMs from legacy data. During this effort, many issues were raised concerning the applicability and inapplicability of the DTDs in MIL-D-87269 for conversion of legacy data. Due to this fact, this section will give a detailed summary of not only the conversion process but also important aspects of the DTDs that were used.

This conversion effort differed in many respects from the F/A-18 and the F-16. One of the more significant differences was the fact that a very large number of paper pages were involved. An enormous amount of data had to be extracted from these paper manuals, and no electronic equivalents were available. The first step in the conversion process involved shipping the paper pages of technical information to two conversion service bureaus. The conversion service bureaus converted all of the text to flat ASCII files and then assigned SGML tags.

The DTD that was used by the conversion service bureaus is described in [14] and [8] as a high-level DTD that consisted of a set of SGML tags from MIL-M-28001B (Markup Requirements and Generic Style Specification for Electronic Printed Output and Exchange of Text). This DTD does not involve content tagging; it provided for tagging down to the paragraph level. Also, all diagrams, figures, and tables were SGML object tagged. The conversion cost was approximately \$4.00 to \$5.00 per page. This DTD was developed in one day in August 1993 by representatives from Martin Marietta-Syracuse, Knolls Atomic Power Laboratory, General Dynamics Electric Boat and the CDNSWC.

The key to their cost-effective conversion was the simplicity of their DTD. It is stated in [14] that all technical manual DTDs consist of a core set of tags that describe the bulk of the manual contents. The following set of tags is listed according to frequency of use, and according to [14], comprise 98% of all the tags used in any technical manual:

- <title>, <head>
- <para>, <para0>, <subpara1>, etc.
- <list>, <item>, <step 1>, <step 2>, etc.

<table>, <row>, <entry>, <tbody>, <thead>  
<figure>, <foldouts>, <graphic>  
<warnings>, <cautions>, <notes>  
<emphasis>, <change>

It is important to note that the DTD, as a whole, was a very small subset of MIL-M-28001B. It is stated in [14] that the DTD given in MIL-M-28001B is so vast with so many elements, that conversion from paper to even a page-based manual (i.e., Type MB) would have to be a manual, and therefore expensive, effort. The tags that were chosen for this effort allowed for rapid, low cost conversion of the paper data.

After the page-based data was SGML tagged, it was uploaded into a database and divided into two files. The first file contained all text and graphics; the second file contained all SGML tags and was termed the Object ID File. The database was an object-oriented database that was run using relational database management software.

The next step in the conversion involved the creation of view packages using the database data. These view packages were created by government subject matter experts rather than the conversion service bureau or the contractor. Creation of the view packages involved the linking of Object File IDs, SGML tags, and the application of a small subset of tags borrowed from the DTDs given in MIL-D-87269 (most are from the generic layer). They created another DTD for the new tags and refer to it as the IETM DTD (the first one is referred to as the Gas Turbine DTD). The view package tags, according to [14] can be split into three categories:

1. Structure tags: tasks, steps, menus, dialog; these tags provide viewpackage structure.
2. Command tags: link, xelemloc, process, video, audio; these are tags which result in an action being initiated.
3. Data tags: text, title, choice; these are tags which contain data in the viewpackage that will be displayed to the user.

Once the view packages are created, they are uploaded via an ASCII text editor back into the original database. Note that these view packages contain all data for presenting dialog boxes and menus as well as for linking to and launching audio, video, and animation. The dialog boxes themselves are created by subject matter experts, using COTS software. Following the view packaging process, the view package that is created is exported from the database, put through a translator, and presented on a COTS hypermedia presentation system made by InfoAccess.

One of the controversial steps taken in this approach was the use of government subject matter experts for the authoring of view packages. Creation of the IETM alone was estimated at costing 2.8 million dollars. This cost, however, may not accurately reflect the cost which other efforts would incur, because government subject matter experts were used during the view packaging process. The view packaging process was a manual process, and therefore it is expected that if this cost would have been taken into account, then the cost of the entire effort would have been higher. Although this is the case, it is important to note that according to a representative from Martin Marietta, approximately 4,000 view packages were created in only 6 months. Therefore,

unless there was an extremely large number of authors, the view packaging process was not significantly complex.

The DTDs that were used in the LM2500 Gas Turbine conversion were designed to facilitate cost effective conversion from paper-based technical manuals while providing the capability for the data to be displayed by any number of hypertext products. Both of these DTDs are currently being used by Newport News Shipbuilding, Naval Underwater Warfare Center, Norfolk Detachment, Naval Education and Training Program Management Support Activity (NETPMSA Pensacola, FL), and Niagara Mohawk Power Corporation. This ETM will be widely used in many sites. It is believed that the only ETM program bigger than the Gas Turbine project is the one at Ford Motor Company which has over 2,700 service sites. The final ETM appears to be equivalent to a frame-based Type MB+ system with some Type MC capabilities.

#### **4.9.4 AEGIS Fire Control System Conversion Project**

The Port Hueneme Division of the Naval Surface Warfare Center is currently creating a Type MC+ IETM for the AEGIS FCS. They are using the Hughes AIMSS authoring and presentation system to create this ETM. They selected AIMSS over the JSTARS IMIS system mainly because of cost reasons (the JSTARS system involves a number of different COTS products on Sun workstations whereas the AIMSS system is self-contained and runs under Microsoft Windows). They have a delivery date of October 1994.

As with any conversion effort, the process was essentially fixed by the capabilities of the AIMSS authoring system. When this project began, the AIMSS system did not have the capability to import data using SGML. The converted data, therefore, had to be stored basically as either text files or graphics files; no other information could automatically be inserted. A new version of the AIMSS system was released in September 1994, however, which has the capability to both import and export data in SGML. Although this is the case, it should be noted that this version of AIMSS is not capable of importing its own export. The AIMSS system has created its own content-specific layer for exporting their data that is different from the one for O-level maintenance given in MIL-D-87269. Their implementation also redefines certain elements that are normally fixed by the generic layer.

Although their SGML capabilities are still being refined, the AIMSS system has an authoring system that is very sophisticated and also easy to use. They provide a unique method of visually linking text, table, and graphic objects for creating frames of information. Text, table, and graphic objects can be created and easily shared by other frames. Those at Port Hueneme consider the AIMSS system to be a Class 4 (Type MC+) IETM system according to the CDNSWC Classes given in Section 2.

The conversion process carried out for the FCS reflects the capability of AIMSS and consisted of the following basic steps:

1. Develop database structure,
2. Perform content design,

- Identify common objects (text, tables, graphics),
  - Establish data flow, and
  - Identify additional data requirements.
3. Convert/reauthor data,
  4. Populate database with objects,
  5. Integrate the data (link objects), and
  6. Perform Test/Quality Assurance.

As a side note, Port Hueneme has noted that the ETM makes it much easier to manage configuration control. Configuration control is difficult in a paper manual when there is a high change rate. The average change rate for the AEGIS FCS is one change per day, and that is considered very high. Configuration control in a technical manual becomes a state table in the Port Hueneme ETM and is much easier to manage.

Following conversion, it was noted that creation of the ETM allowed for significant reductions in data redundancy. Data for the cycle-down transmitter experienced a reduction on the order of 100 to 1. Within FCS maintenance tasks, steps were reduced (due to object sharing) by a factor of 100 to 3. Warnings and cautions numbering in the hundreds were reduced to dozens.

Before beginning the conversion effort, they carried out a preliminary analysis to determine the length of time it would take to create their Class 4 IETM. Their analysis was based on two NAVSEA work packages that consisted of 5 pages and 10 pages respectively. They estimated that:

- Content design requires 10 minutes/page,
- Populating the database and linking requires 10 minutes/page, and
- Testing and QA requires 10 minutes/page.

These estimates are all based on the following assumptions:

- The structure of the data is in place,
- Authors are skilled in IETM conversion, and
- Common objects have already been identified.

In this analysis, they determined that conversion from a paper manual to Class 3 (Type MC) cost the same as conversion from paper to Class 4 (Type MC+). It is significant to note here that these results were noted by the chairman of the Tri-Service IETM Working Group (TSWG) as seriously conflicting with those obtained during the LM2500 Gas Turbine IETM project.

#### **4.9.5 Newport New Shipbuilding Conversions**

Newport News Shipbuilding has converted eight chapters of the Ship Valves Technical Manual from Interleaf into SGML using the Gas Turbine DTD. The converted data is then uploaded into the same COTS hypermedia system that was used by the LM2500. The Interleaf files were a non-SGML equivalent to Type MB data created for paper output. Interleaf is widely used in both

DoD and industry for desktop publishing, document management, and producing camera-ready hard-copy. This conversion effort is similar, in principle, to converting Type MB to a Type MB+ system with some Type MC capabilities.

The Newport News Shipbuilding approach to building an IETM from legacy electronic data was rather straight forward and consisted of the following five steps:

- Standardize format and structure of desired document.
- Insert SGML tags using modified COTS software (tags are based on 28001) either directly or through auto-tagging.
- Extract graphics and convert to TIFF (with CCITT Group IV compression) using commercial software.
- Convert SGML tagged text to Hypertext Markup Language (HML) required by the InfoAccess presentation system.
- Use InfoAccess's GUIDE software to convert HML tagged document to Microsoft Windows graphical user interface files.

The interesting part of the Newport News conversion experience is the speed with which the conversion was carried out. Once the OmniMark conversion programs were written, the conversion of all eight chapters took less than two hours (including cleanup). The key to achieving this speed was the authoring of the Interleaf file. The Interleaf files were recently created and when authored correctly are very "component" oriented. The people at Newport News even went back to the authors and made sure that they used certain techniques in creating their Interleaf file. Correct authoring techniques can be compared to creation of tables in common word processors. Modern word processing software has commands to create tables and then insert elements into the rows and columns. Tables can also, of course, be created using graphical drawing commands instead of the table commands. By using the table command, however, it is much easier to write a conversion program to map the table into SGML (or anything else). Writing conversion programs can be easy or very difficult depending on how the original is authored.

#### **4.9.6 Conversion and the Joint Integrated Maintenance Information System**

The Joint Integrated Maintenance Information System (JIMIS) is a Type MC+ system being developed by Northrop-Grumman Melbourne Systems Division. It was originally intended to be used for the JSTARS program, but may also be used in others, such as the V-22. JIMIS is essentially an integration of COTS software products implemented on Sun Workstations running X-Windows with Motif. The products include Oracle Version 7 RDBMS, ArborText products, and InterCAP (Illustrator 2, X-CHANGE, and Metalink Graphics Tools). Furthermore, it is estimated that only 10% of the software code which constitutes JIMIS was actually developed by Grumman (and consists of mostly filtering software).

Creating ETM data using JIMIS involves the loading of text data files and graphics files into libraries. The loaded files are then used in the building of part data, procedural data, troubleshooting data, and descriptive data. In their case, the part data comes from IPB manuals,

procedural data comes from job guide manuals, troubleshooting data comes from fault-isolation manuals, and descriptive data comes from general system manuals.

The text data files require SGML tags before insertion into one of their libraries. In [35], they give a process for transitioning manuals prepared with Interleaf into JIMIS. They begin with a Preliminary Technical Order (PTO) that may be a Job Guide, General System, or Fault Isolation Manual in an Interleaf format. They copy the electronic Interleaf file, clean it up (deleting figure frame pages, etc.), and save the data into an ASCII file. The ASCII file is then imported into a JIMIS directory and filtered into a series of SGML tagged files. The filtering process also automatically attempts to reduce data redundancy by eliminating exact duplicates of tagged data. Following this, the data is then loaded into one of the "High-Level" libraries given in Table 4.9.6-1 and used for building the procedural, troubleshooting, part, and descriptive data.

**Table 4.9.6-1 JIMIS High Level Libraries**

step	alerts (warning/caution/note)
para	dialog (text questions)
person (personnel required)	equip (support equipment)
refmat (reference materials)	consum (consumables)
fault (fault codes)	outcome
partbase	connection
attachparts (attaching parts)	task ref
location	graphic

Insertion of graphic data into JIMIS is begun after all PTOs for a particular system have been completed. At this point, a graphics team reviews all illustrations in the chosen set of manuals. The illustrations are dissected into individual base graphics and all duplicates are eliminated. Note, according to [35] a base graphic is defined as a single graphic, representing an item(s), used to support a text application. It contains no shading, text, or other markings. Also during this time, the various IPB manuals are consulted and a unique JIMIS Functional Designator (Fun Des) is assigned to every part on the aircraft. Following this, each extracted base graphic is assigned a unique control number which serves as the graphic ID in the JIMIS database. The graphics team then converts the base graphics from AutoCad into an InterCAP format. The converted InterCAP files are then reviewed and edited as required by the graphics team. Upon approval, the base graphic is released for use and transferred into the JIMIS graphics library. Text, callouts, notes, and graphical overlays are all added where required when building part, procedural (tasks), troubleshooting, and descriptive data.

Generally speaking this system is not unlike that of other Type MC+ systems (AIMSS, and Lockheed). Although the textual data is inserted into their libraries as SGML, the JIMIS system is not capable of exporting the fully converted/authored ETM data as an integrated SGML file, this means that the Oracle RDBMS may be required during presentation. It is also of interest to note that the JIMIS system cannot import SGML data that contains branching, looping, and context-dependent filtering information that is prepared according to the CDM. It only imports

small, simple, blocks of tagged ASCII text data. Currently, the Lockheed system is the only Type MC+ system, encountered during this project, that is known to be capable of both importing and exporting CDM-compliant SGML data with embedded branching, looping, and context-dependent filtering SGML constructs.

#### **4.9.7 A General Formula for Conversion**

The previous examples were given to show the variety of methods used, results obtained, and issues involved. As stated elsewhere in this report, there does not exist a concise conversion methodology that would be applicable for converting arbitrary paper and electronic technical manuals into a format that complies with the IETM specifications.

Any conversion methodology that may be employed should be established during the planning stages. Planning a conversion project is covered in Section 4.8. This subsection will look primarily at the implementation of conversion that follows authoring and presentation system selection, legacy document analysis, and the mapping of information and information structures. This implementation methodology will contain aspects from the different methodologies just covered and will not focus on any particular COTS product or government owned system.

Conversion to Type MC, as covered in Section 4.3, at its highest level involves the identification of the system, subsystem, subassembly hierarchy. After identifying this hierarchy, it is useful to identify the task data (procedural and setup), descriptive data, parts data, and troubleshooting (or diagnostic or fault) data. If an entire technical manual or set of technical manuals is to be converted, then all of this information will be present in some form.

After the identification phase, the next logical step is to separate the technical manual data into two categories: the data that exists only on paper and the data that exists in some electronic form. To get data into an electronic format, the paper data will require two levels of processing: text data capture and graphical data capture. Decisions made during the mapping phase (Section 4.5 and 4.8) may significantly affect the initial data capture and/or conversion process. Even when mapping to an exceedingly simple DTD, difficulties will be encountered because it is likely that all the information constructs identified in Section 4.4 will be encountered. If all these issues were handled during the mapping phase, the groups carrying out the conversions are likely to require less consultation with the ETM system implementors. During the LM2500 Gas Turbine conversion effort, the conversion service bureaus found some issues difficult to resolve and, according to a representative from Martin Marietta, telephoned system implementors for guidance, on average, once per week.

Text data capture is carried out in two ways: the text can be either retyped in by hand (sometimes called rekeying) or scanned into a raster format (at a high enough resolution). When scanned into a raster format, the text is automatically recognized through computer character recognition algorithms (typically referred to as OCR). Automated text recognition with OCR packages is not always the most cost-effective solution. The ability of OCR packages to recognize characters depends on the quality, consistency, font, and font size of the characters. Depending on these characteristics, the results of OCR can be either highly accurate or highly inaccurate. Even

accuracies of 99% can equate to 1 mistake per every 100 characters; if each word on average contains 5 characters, this equates to 1 misspelled word out of every 20. This would obviously result in an enormous number of errors in an entire technical manual (including errors in part numbers). It is not uncommon for a conversion service bureau to find it more cost-effective to manually retype in text instead of scanning and OCRing it. If a significant amount of text data is required to be captured, a conversion service bureau should be contacted.

Graphical data capture from paper pages is carried out simply by scanning the technical manual pages, separating the figures from the rest of the page image (cropping), and storing the figures in separate raster files. At this point, no attempt should be made to remove callouts or decompose into graphic primitives. These things are better done during the authoring phase with a suitable authoring system.

Capturing table data may also be necessary. If it is necessary to convert a table to a non-graphical format, then a neutral tagging format such as SGML must be used at the outset. Table tagging can be done by following MIL-M-28001B, MIL-D-87269, or in some other customized fashion. The conversion of simple tables is highly automated within more advanced conversion service bureaus.

Pages which contain mixtures of text, tables, and graphics can be converted with a mixture of the above three steps. It is not uncommon to carry out these conversions in separate phases by separate conversion service bureaus. One bureau may capture only graphics, while the other may do text and tables. In any case, the data capture phase will be necessary whenever the legacy data exists in either a Type MA or MA+ format. The Type MA+ graphical data can be easily extracted, but the text and table issues are the same as for paper and OCR and may or may not work well on Type MA+ data.

If the legacy data is in an electronic format such that the text is accessible in an ASCII format (similar to Type MB or higher), then the data capture phase for text is not necessary. If the graphical images used in the electronic files are accessible in a suitable electronic format, then the graphical data capture phase will not be necessary. Depending on the format of the data, a preparsing step may be desirable as described in Section 4.9.2. This step will be termed an electronic data conversion step. In this section, electronic data conversion will refer to the conversion of any ASCII text file that contains either typesetting codes or follows any kind of tagging scheme (TeX, SGML, Interleaf, etc.) to an SGML file with a desired DTD.

Either after or during the data capture or conversion, automated placement of a high-level set of tags from the desired DTD should take place where possible. The degree of automated tag placement will depend entirely on the information that is available in the source file and the information desired by the target DTD. In the Lockheed experience, many system, task, and step tags were automatically placeable. After automated tag placement, any other tags that can be easily placed, or any cleanup that is required, should be done before loading it into the authoring system.

The above capturing and conversion steps are certainly done on task data and descriptive data. The part and troubleshooting data, as discussed in Section 4.5.4, can be captured and/or converted before, after, or in parallel with the other data. Converting part data is not expected to be difficult. Conversion of troubleshooting data, depending on the source and target formats, is likely to require the most time and the most experienced authors, and is not typically done by conversion service bureaus.

Once the data is in a suitable high-level, structured format, technical writers can use either an appropriate text editor, graphics editor, table editor, SGML editor, suitable COTS products, or the ETM authoring system for the remainder of the conversion process. The remainder of the conversion process depends on the ETM requirements and the capabilities of the authoring system. To reduce the number of questions, the authors must also be fully aware of the results of the mapping stage and the handling of the information constructs (Section 4.4, 4.5, and 4.8).

When converting to Type MC+ data, it is likely that the high-level tagged and structured data will be immediately imported into a specially designed authoring system (rather than an SGML editor). A sophisticated authoring system such as Lockheed or AIMSS will make the remainder of the authoring highly efficient. During the authoring phase, it is also absolutely necessary to have the capability of previewing the data exactly as it would be displayed on the presentation system. The authoring phase is likely to include the linking of cross-references, placement of hotlinks, placement of graphical hotspots, implementation of context dependent filtering capabilities on both text and graphics, implementation of dialogs, etc. Clearly, if the JCALS system is to be used, the need for an efficient authoring system combined with preview capabilities, coupled with the fact that ETMs are produced in a wide variety of ways makes it necessary for JCALS to have a direct and independently developed interface to each ETM system that it must deal with.

#### **4.10 COTS Products for Conversion**

In the previous subsections, we investigated technical manual information, conversion methodologies, and the functionality of authoring and presentation systems. In this subsection we will look specifically at COTS products that are relevant to the authoring, presentation, and transformation of this technical manual information.

A multitude of COTS products are available that are relevant to the creation, display, and conversion of electronic documentation of many kinds. The goal of this project, of course, was to attempt to identify only those products that were most applicable for generation and conversion of electronic technical manuals for DoD. For the sake of clarity, note that complete conversion of a technical manual (whether or not it is in an electronic format) to a different level requires the ability to author, present, and transform data. A complete conversion package must have all three of these capabilities.

The identification of candidate products was carried out by first establishing a list of requirements. Those products that had at least one of the properties given in the list of requirements were identified as candidate products. Following the identification phase, certain selected products

were further evaluated using an evaluation matrix developed specifically for providing a comprehensive look at the capabilities of the package being evaluated.

#### **4.10.1 Requirements**

In order to establish a list of requirements for this undertaking, it became necessary to consider a base set of standards, specifications, and other documents that may be either useful or necessary in creating an electronic manual (or converting one). A set of requirements for COTS products used during the evaluation was then extracted from this base set of documents. Table 4.10.1-1 contains the base set of military specifications, Table 4.10.1-2 contains the set of military standards and handbooks, Table 4.10.1-3 contains miscellaneous government documents, Table 4.10.1-4 contains relevant non-government documents, Table 4.10.1-5 contains document titles under development by the Tri-Service IETM Working Group (TSWG), and Table 4.10.1-6 contains the set of requirements that were the bases for COTS product selection. At this point, it is useful to note that all the selected requirements in Table 4.10.1.6 have narrow, well-defined interpretations, except for MIL-M-87268 and MIL-D-87269. These two specifications made it necessary to look beyond normal "CALs-compliant" tools and consider other, more general, COTS hypermedia systems.

#### **4.10.2 Evaluation Matrix**

As mentioned previously, the evaluation matrix (given in Appendix B) was prepared principally for identifying capabilities of COTS packages in the CALS ETM/IETM environment. To this end we found it necessary to first separate the matrix into two distinct categories. The first category is termed a Product Feature Survey and concentrates on the various functional features that a COTS product may have. The second category is an IETM Specification Compliance Survey and concentrates on those aspects of the IETM specifications that are both required and are also evaluatable (to some extent). This second category was principally included for the sake of evaluating products that can handle Type MC+ data (and also some instances or aspects of Type MC data).

During the development of the matrix, due to the multitude of applicable products and their functional capabilities, it was deemed useful to further subdivide both of the initial categories. The Product Feature Survey category was divided into eight parts:

- PART 1: GENERAL PRODUCT INFORMATION
- PART 2: HARDWARE/SOFTWARE REQUIREMENTS
- PART 3: SUPPORTED INPUT FILE FORMATS
- PART 4: SUPPORTED OUTPUT FILE FORMATS
- PART 5: AUTHORING/VIEWING TOOLS

**Table 4.10.1-1 Military Specifications**

TITLE	NUMBER
Manuals, Interactive Electronic Technical: General Content, Style, Format, and User-Interaction Requirements for	MIL-M-87268
Data Base, Revisable: Interactive Electronic Technical Manuals, for the support of	MIL-D-87269
Quality Assurance Program: Interactive Electronic Technical Manuals and Associated Technical Information; Requirements for	MIL-Q-87270
Digital Representation for Communication of Product Data: IGES Application Subsets and IGES Application Protocols	MIL-D-28000
Markup Requirements and Generic Style Specification for Electronic Output and Exchange of Text	MIL-M-28001
Requirements for Raster Graphics Representation in Binary Format	MIL-R-28002
Digital Representation for Communication of Illustration Data: CGM Application Profile	MIL-D-28003
Manuals, Technical; General Style and Format of (Work Package Concept)	MIL-M-81927
Manuals, Technical: General Style and Format Requirements	MIL-M-38784
Quality Program Requirements	MIL-Q-9858
Drawings, Engineering and Associated Lists	MIL-D-1000

**Table 4.10.1-2 Military Standards and Handbooks**

<b>TITLE</b>	<b>NUMBER</b>
Automated Interchange of Technical Information	MIL-STD-1840
Abbreviations for Use on Illustrations, Specifications, Standards, and in Technical Documents.	MIL-STD-12
Engineering Drawing Practices	MIL-STD-100
Definition of Terms for Automatic Electronic Test and Checkout	MIL-STD-1309
Logistic Support Analysis	MIL-STD-1388-1
Logistic Support Analysis Record, DoD Requirements for a	MIL-STD-1388-2
Human Engineering Design Criteria for Military Systems, Equipment and Facilities	MIL-STD-1472
Quality Assurance Terms and Definitions	MIL-STD-109
Marking Technical Data Prepared By or For the Department of Defense	MIL-STD-1806
Defense System Software Development	DOD-STD-2167
Defense System Software Quality Program	DOD-STD-2168
Automated Data System Documentation	DOD-STD-7935
Department of Defense Continuous Acquisition and Life-Cycle Support (CALs) Program Implementation Guide	MIL-HDBK-59
Directory of DoD Engineering Data Repositories	MIL-HDBK-331
Technical Manuals	MIL-STD-361
System/Subsystem/Subject Number (S/S/SN) Numbering System	MIL-STD-1808
Work Breakdown Structures for Defense Materiel Items	MIL-STD-881

**Table 4.10.1-3 Other Government Documents**

<b>TITLE</b>	<b>NUMBER</b>	<b>SOURCE</b>
Information Security Program Regulations	DOD 5200.1-R	Department of Defense
Industrial Security Manual for Safeguarding Classified Information	DOD 5220.22-M	Department of Defense

**Table 4.10.1-4 Non-Government Documents**

<b>TITLE</b>	<b>NUMBER</b>	<b>SOURCE</b>
DoD Liaison Recommendation for Hazardous Materials Warnings in Technical Data	AIA PUBS-119	Aerospace Industries Association
Information technology - Hypermedia/Time-based Structuring Language (HyTime)	ISO/IEC IS10744:1992	American National Standards Institute
Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML)	ISO 8879	American National Standards Institute
Information Processing Systems - Computer Graphics - Metafile for the storage and transfer of picture descriptive information (CGM)	ISO/IEC 8632:1992	American National Standards Institute

**Table 4.10.1-5 Documents Under Development by the Tri-Service IETM Working Group**

<b>DOCUMENT TITLE</b>	<b>EXPECTED COMPLETION DATE</b>
Electronic Delivery Device Handbook	2nd Quarter FY 1995
View Package Handbook	2nd Quarter FY 1996
System Acquisition Managers' Implementation Handbook	2nd Quarter FY 1996
IETM Test Methodology and Tools	2nd Quarter FY 1996
IETM Tutorial	3rd Quarter FY 1995

**Table 4.10.1-6 Requirements for COTS Identification**

TITLE	NUMBER
Manuals, Interactive Electronic Technical: General Content, Style, Format, and User-Interaction Requirements for	MIL-M-87268
Data Base, Revisable: Interactive Electronic Technical Manuals, for the support of	MIL-D-87269
Digital Representation for Communication of Product Data: IGES Application Subsets and IGES Application Protocols	MIL-D-28000
Markup Requirements and Generic Style Specification for Electronic Output and Exchange of Text	MIL-M-28001
Requirements for Raster Graphics Representation in Binary Format	MIL-R-28002
Digital Representation for Communication of Illustration Data: CGM Application Profile	MIL-D-28003
Information technology - Hypermedia/Time-based Structuring Language (HyTime)	ISO/IEC IS10744:1992
Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML)	ISO 8879
Information Processing Systems - Computer Graphics - Metafile for the storage and transfer of picture descriptive information (CGM)	ISO/IEC 8632:1992

- PART 6: AUTOMATED/SEMI-AUTOMATED CONVERSION TOOLS
- PART 7: MISCELLANEOUS FEATURES
- PART 8: EVALUATION SUMMARY

The IETM specification compliance section was divided into six parts:

- PART 1: CONTENT AND STYLE
- PART 2: DISPLAY ELEMENTS
- PART 3: USER INTERFACE ELEMENTS
- PART 4: WARNINGS, CAUTIONS, AND NOTES
- PART 5: GRAPHICS ELEMENTS
- PART 6: DATABASE REQUIREMENTS

These categories seemed sufficient for identifying those aspects of the conversion, authoring, and presentation packages that were important to this project. One of the unfortunate aspects of this matrix is that it is rather large and extensive. Evaluation of a product and filling out the matrix requires the evaluator to have a very detailed knowledge and understanding of the COTS package. This requirement made it necessary to conduct the evaluations that were done by a variety of methods such as phone calls, site visits, obtaining evaluation software, etc.

#### **4.10.3 Overview of COTS Products Considered**

Identifying products which met at least one of the requirements in Table 4.10.1-6 involved searches of libraries, magazines, journals, and the Internet. Also, many relevant products appeared in such places as Documentation 94, Seybold Seminars Boston 94, and the NAVSEA/NAVSUP International Logistics Symposium. Other useful sources for identifying products were CALS Shared Resource Centers and DoD programs already involved in conversion and creation of electronic technical manuals. Three published sources used for identifying many CALS-related products, in particular, were [15, 16, 17]. Other sources can be found in Appendix D which contains a complete listing of the documents stored in the ManTech library and used in our research.

Appendix A gives a listing which contains, among other things, all of the COTS products which were identified as candidates for meeting one of the above requirements. After reviewing and evaluating many of these products, it became apparent that they could be grouped into ten basic functional categories. For convenience, two additional categories were created, one for listing identified conversion service bureaus, and the other for listing those products that did not fit into the ten main categories. The proposed categories, therefore, consist of both COTS products and services and are given in Table 4.10.3-1. Note that image scanners and products which perform OCR were neither identified nor evaluated during this project. For identifying and comparing OCR packages, the Information Science Research Institute provides very detailed and reliable information [18]. The products, within their appropriate categories, are given in Tables 4.10.3-2 through 4.10.3-13. Note that the products and services are given in no significant order.

**Table 4.10.3-1 COTS Products and Services**

<b>COTS PRODUCTS AND SERVICES</b>	
1.	Electronic presentation systems
2.	General electronic authoring systems
3.	SGML/text editors
4.	Graphics conversion and viewing
5.	SGML, tag-based, and text-based conversion software
6.	Electronic document analysis and design tools
7.	Electronic document managers/databases
8.	HyTime software
9.	Parsers
10.	Page layout tools
11.	Conversion service bureaus
12.	Other

**Table 4.10.3-2 Electronic Presentation Systems**

<b>ELECTRONIC PRESENTATION SYSTEMS</b>	
1.	Interactive Presentation Manager, (Aquidneck Data Company)
2.	SuperBook System, (Bellcore)
3.	DynaText, (Electronic Book Technologies)
4.	OLIAS Browser, (HAL)
5.	AIMSS Reader, (Hughes)
6.	Guide Reader, (InfoAccess)
7.	WorldView 2, (Interleaf)
8.	Pinpoint, (Maxwell Data Management, Inc.)
9.	HyperWriter for SGML, (Ntergaid)
10.	Lector PAT, (Open Text Corp.)
11.	Pathways Viewer, (Westinghouse)
12.	Explorer, (SoftQuad)
13.	Lockheed Fort Worth Presentation System (Government Owned)
14.	IADS Reader (Government Owned)
15.	Mosaic (Public Domain)

**Table 4.10.3-3 General Electronic Authoring Systems**

<b>GENERAL ELECTRONIC AUTHORING SYSTEMS</b>	
1.	Professional Publisher, GUIDE Author, Text Indexer, GUIDE Writer, (InfoAccess)
2.	HyperWriter for SGML, (Ntergaid)
3.	Pathways Tools, Pathways Loader (Westinghouse)
4.	AIMSS, (Hughes)
5.	Lockheed Fort Worth Authoring System, (Government Owned)
6.	IADS Author (Government Owned)

**Table 4.10.3-4 SGML/Text Editors**

<b>SGML/TEXT EDITORS</b>	
1.	ADEPT Editor, (ArborText)
2.	SGML Writer, (Datalogics)
3.	InContext, (InContext Systems)
4.	Interleaf 5<SGML> (WYSIWYG), (Interleaf)
5.	Author/Editor, (SoftQuad)
6.	HotMetal, (SoftQuad)
7.	Intellitag, (WordPerfect)
8.	SGML Smart Editor, (Auto-Graphics)

**Table 4.10.3-5 Graphics Conversion and Viewing**

<b>GRAPHICS CONVERSION AND VIEWING</b>	
1.	Meta Link Author, Meta Link Runtime, Illustrator 2, Quick Edit, Red-Liner, X-Change, (Intercap Corp.)
2.	CADleaf Plus, (Carberry)
3.	OSR Series, (GTX)
4.	Hercules, (Hercules Defense Electronics Systems, Inc.)
5.	IGES Toolkit, IGES Parser/Viewer, IGESXpert, CALSView, (IGES Data Analysis)
6.	Image Author, Image Editor, (InfoAccess)
7.	PreVIEW 4.0, (Rosetta Corp.)

**Table 4.10.3-6 SGML, Tag-based, and Text-based Conversion Software**

<b>SGML, TAG-BASED, AND TEXT-BASED CONVERSION SOFTWARE</b>	
1.	PowerPaste, (ArborText)
2.	FastTAG, (Avalanche/Interleaf)
3.	SGML Hammer, (Avalanche)
4.	OmniMark, (Exoterica)
5.	Interleaf <SGML> Gateway, (Interleaf)
6.	Oracle Context, (Oracle Federal Systems)
7.	TagWrite, (Zandar)
8.	DynaTag, (Electronic Book Technologies)

**Table 4.10.3-7 Electronic Document Analysis/Design Tools**

<b>ELECTRONIC DOCUMENT ANALYSIS/DESIGN TOOLS</b>	
1.	SGML Support Program, (Aquidneck Data Company)
2.	Document Architect, (ArborText)
3.	Interactive Stylesheet Editor, (Electronic Book Technologies)
4.	Interleaf 5<SGML> Toolkit, CALS Application, CALS Toolkit (Interleaf)
5.	NEAR & FAR, (Microstar)
6.	DTD Viewer, (ZiffTech)
7.	DTDocumenter, (SoftQuad)
8.	RulesBuilder, (SoftQuad)
9.	Document Analyzer, (Avalanche/Interleaf)
10.	Hyperleaf Toolkit, Active Link Tool, (Interleaf)

**Table 4.10.3-8 Electronic Document Managers/Databases**

<b>ELECTRONIC DOCUMENT MANAGERS/DATABASES</b>	
1.	ActiveSearch, (Active Systems)
2.	Basisplus, (Information Dimensions)
3.	PAT, (OpenText)
4.	PassagePRO, (Passage Systems, Inc.)
5.	Digital Data Management Systems (PRC)
6.	Parlance Document Manager, (Xyvision, Inc.)
7.	DynaBase, (Electronic Book Technologies)
8.	Impact Search and Retrieval, (Auto-Graphics)

**Table 4.10.3-9 HyTime Software**

<b>HYTIME SOFTWARE</b>	
1.	HyMinder, MarkMinder, (TechnoTeacher)
2.	Explorer web files, (SoftQuad)

**Table 4.10.3-10 Parsers**

<b>PARSERS</b>	
1.	MarkMinder, (TechnoTeacher)
2.	SGMLS, (Public Domain)
3.	SGML Kernel, (Exoterica)

**Table 4.10.3-11 Page Layout Tools**

<b>PAGE LAYOUT TOOLS</b>	
1.	ADEPT Publisher, (ArborText)
2.	DL Composer, (Datalogics)
3.	FrameBuilder, (Frame Technology Corp.)
4.	Interleaf 5 <SGML>, (Interleaf)
5.	CAPS, (XSoft)
6.	Parlance Publisher, (Xyvision, Inc.)
7.	SGML Enabler QuarkXTension, (SoftQuad)

**Table 4.10.3-12 Conversion Service Bureaus**

<b>CONVERSION SERVICE BUREAUS</b>	
1.	Data Conversion Laboratory
2.	Gateway Conversion Technologies
3.	Docucon
4.	Jouve
5.	Maxwell Data Management
6.	Merex, Inc.
7.	I-NET (CSRC)

**Table 4.10.3-13 Other**

<b>OTHER</b>	
1.	ICAT (Automated Technology Systems Corp.)
2.	ECALS, (Computer Sciences Corporation)
3.	WorkSmart, (Information Design Corp.)
4.	Knowledge Pro, (Knowledge Garden)

The above products were identified as having the potential to meet one of the requirements in Table 4.10.1-6. A conversion project that has the goal of creating an electronic technical manual may require all of the tools and systems listed in Table 4.10.3-1. Tools such as those in Table

4.10.3-3 will in some cases have the capability of providing multiple functions that cover many of the remaining product categories. Some companies, on the other hand, cover many categories by providing a different tool for each.

Care must be exercised when selecting tools from different companies to fulfill the needs covered by each category. Not all tools can work together, not all tools support the use of SGML in the same way (different features are supported), etc. Notice that many of the tools that may be needed during a conversion project are provided in the current JCALS system described in Section 3.2. JCALS cannot solve the entire conversion problem, however, because it is not likely to be capable of either implementing the additional features or previewing capabilities which different authoring systems usually provide.

## **4.11 The Cost of Conversion**

In general, the cost of converting paper or electronic technical manuals, is a very difficult issue to address. Typically, different conversion efforts will have radically different requirements. These different requirements can impact the ultimate cost in various ways. In this section, we will look at class-based, time-based, tag based, and other attempts that have been made to estimate costs. We will also point out ambiguities and possible weaknesses that may exist with each approach. We will begin by considering those cost-driving factors that are common to all conversion efforts.

### **4.11.1 Cost Estimation**

The only cost factors that all conversion projects appear to have in common are complexity, volume, and time constraints. Complexity, in this context, refers to those aspects of the conversion that can be handled automatically (if it is deemed cost-effective) versus those that are manual-labor intensive. Costs associated with complexity can be estimated only after the analysis and mapping phases, discussed in Section 4.8, are performed. The volume of data that must be converted is easily determined and also has a direct and obvious impact on the final cost. Time constraints also affect cost; the sooner the data is needed, the more it usually costs.

When dealing with conversion service bureaus, cost estimates are usually made on the basis of a proposed list of items and review of a sample [33]. The final estimate will be valid only for the items that have been explicitly reviewed and approved. Unanticipated items or deviations, of course, cannot be predicted and must be handled as they occur. Depending on the volume to be converted, it is wise to approach a conversion service bureau with all issues that were covered in the mapping phase. With their tools and experience, they may have cost-effective ways of handling different transformations that would not be apparent to the non-expert.

Generally speaking, the only conversion project that can be estimated with any precision is one that converts Type MA to Type MA+ (Class 1). One conversion service bureau recently noted that this conversion could cost on the order of 25 cents per page. Compare this with data given in [19], [20], and [46], which describes this sort of conversion as costing approximately \$1.00 per page, 70 cents per page, and \$2.00 per page respectively. A possible next step in a conversion from paper may involve text capture and image capture. If the text capture can be done with OCR, one bureau estimates the cost to be \$1.75 to \$2.00 per page (for a non-tagged OCR'd

ASCII text file). Proofreading and cleanup would cost extra. The image capture would then be carried out by separating the figures from the scanned images, deskewing, despeckling, etc.

On the other hand, reliably predicting the cost of conversion from Type MA to any Type other than MA+, is likely to be completely impossible (depending on the demands of the target format). Converting to Type MB, for example, by applying appropriate SGML tags, may seem an easy thing to do at this point, but that may or may not be true depending on the Type MB target DTD. Recall that in the LM2500 Gas Turbine experience, as discussed in Section 4.9.3, scanning, text capture, and placing of a high-level set of tags was carried out by conversion service bureaus at approximately \$4.00 to \$5.00 per page (but the tags were easy to place). It is further noted in [20] that converting from paper to SGML-tagged pages costs roughly \$6.00 to \$10.00 per fully SGML tagged page (it is not known which DTD he was referring to). Another attempt at estimation of the cost of essentially converting from Type MA to Type MB is given in [46] and indicates this would cost \$16.00 per page. Normally, when going beyond a simple paper or page-image TM and converting to or between ETM Types or Classes at any other level, the three general cost factors above involve many of the following issues:

- Document analysis costs.
- ETM system evaluation and selection costs.
- Setup costs.
- Manpower requirements (i.e., technical writers, document analysts, programmers, subject matter experts, etc.).
- Software purchasing and licensing agreements.
- Hardware requirements (computers, scanners, disk space, purchasing vs. leasing, etc.).
- Training costs.
- Accounting for time constraints on hardware, software, and delivery schedule.
- Accessibility/Format of source data.
- In house work vs. conversion service bureau work.
- Automated vs. manual text conversion and accuracy required (i.e., OCR vs. re-typing).
- Information-richness of source data.
- Automated and manual placement of SGML tags (depends on mapping complexity).
- Desired level of functionality in target data.
- Maintaining original structure vs. reorganization of the document (i.e., when converting to Type MC or MC+, chapters and sections may have to become a hierarchy of systems/sub-systems/sub-assemblies. Each level of the hierarchy would then have associated procedural, descriptive, parts, and fault information).
- Interpretation of troubleshooting information (depends on mapping complexity).
- Diagnostic method employed.
- Automated vs. manual cross references (in terms of conversion and implementation in the target document).
- Simple vs. complex (spanning) tables.
- Placement and/or conversion of hotlinks and hotspots.
- Raster vs. vectorized graphics.
- Static vs. interactive graphics.

- Handling of binary vs. gray level vs. color figures (including scanning, deskewing, cropping, etc.).
- Handling of standard size paper pages vs. oversized pages.
- Placement of index tags.
- Structural vs. content tagging (depends on mapping requirements).
- Raster vs. SGML representation of mathematical formulas.
- Stylesheet, formatting, or presentation requirements.
- Automated and manual validation procedures.

Although cost is mainly determined by complexity, and complexity is determined by the mapping process, attempts have been made to establish cost guidelines for conversion to different data types on a per page basis. The following, recently determined, cost data is given in [19]:

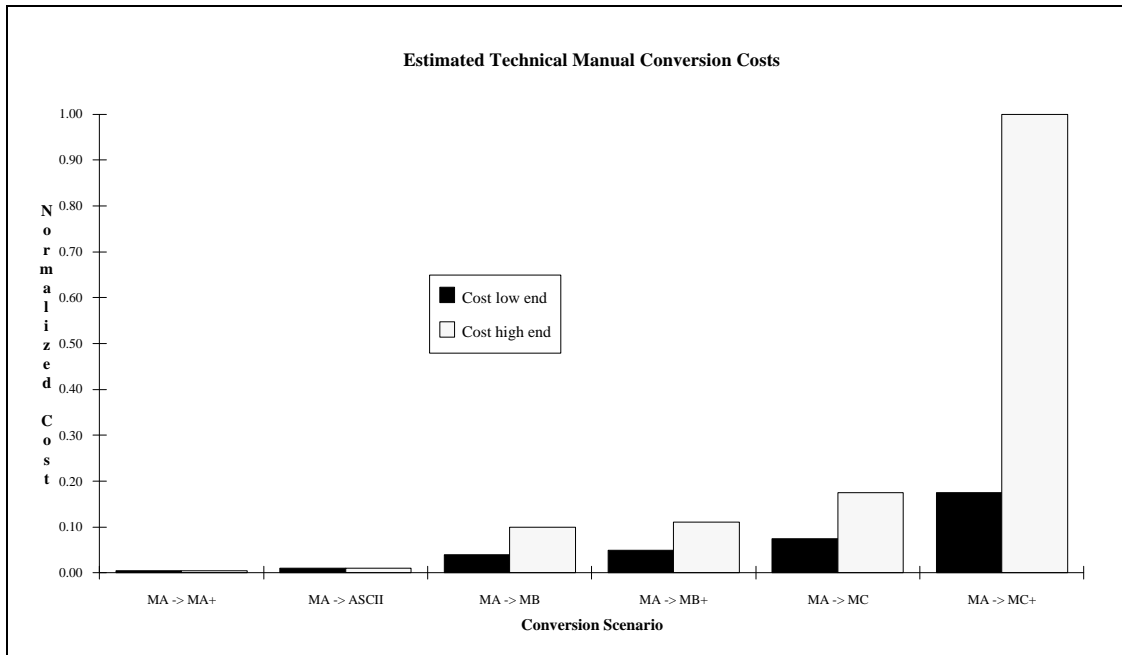
- The cost to convert paper data into Class 1 (equivalent to Type MA+ in this report) with indexing is about \$1.00 per page.
- The cost to convert an ASCII text file into Class 2 (Type MB) can range anywhere from \$8.00 to \$20.00 per page (without considering any infrastructure costs).
- The cost to convert paper data into Class 3 (Type MC) data is about \$15.00 to \$35.00 per page.
- The cost to convert paper data into Class 4 (Type MC+) data is about \$35.00 to \$200.00 per page.

There are obvious advantages to having cost data available in this way. If the cost could be accurately determined on a per page basis when converting between any Type or Class, then it would be straightforward to make cost predictions of page-based source data conversion with respect to volume. Note that according to [19], the higher cost of converting to Class 4 would result when the source data is revalidated (rather than just reformatted for screen presentation). A visual example of the cost data just mentioned is given in Figure 4.11.1-1.

An imperfect but more accurate method for determining page-based cost estimates would be to separate cost estimates in terms of the nature of the data that it contains. In Section 4.5, we saw that conversion of troubleshooting information, in particular, can be substantially more complex and therefore considerably more expensive than any other forms of data. The actual cost data per page may be more accurately determined if the TM or ETM data that is to be converted is divided according to the following three categories:

1. Normal maintenance task and descriptive information conversion,
2. Troubleshooting information conversion, and
3. Part information conversion.

To see the importance of this, one program determined that, in their particular situation, conversion of troubleshooting information required approximately 17 hours per page, on average, whereas all other data required approximately 2 to 2.5 hours per page. These differences in times are significant and clearly lead to corresponding differences in cost.



**Figure 4.11.1-1 Technical Manual Conversion Costs As Estimated In [19]**

Noting the amount of time it takes to convert data on a per page basis, as with cost, may or may not be useful, depending on the source and target data formats as well as the kind of information being converted. In [25], it is noted that an analysis of NAVSEA work packages resulted in the following time estimates:

- Content design requires 10 minutes/page.
- Populating the database and linking requires 10 minutes/page.
- Testing and QA requires 10 minutes/page.

These estimates are all based on the following assumptions:

- The structure of the data is in place.
- Authors are skilled in IETM conversion.
- Common objects have been identified.

This study was based on two NAVSEA work packages of 5 and 10 pages respectively and amounts to 30 minutes per page to carry out a conversion. One further conclusion from this analysis was that conversion from a paper manual to Class 3 costs the same as conversion from paper to Class 4. It is not clear if conversion of troubleshooting information was considered during this study. It is also not clear if the validation issues discussed in [19] were considered.

It is useful to note that both of the time-based examples just given deal with conversion of paper work packages into systems that are considered by each program to be Type MC+ IETM

systems. Conversion of ordinary data in the first system required about 2 to 2.5 hours per page, whereas in the second it required only 30 minutes per page. Both systems offer a high-level, user-friendly interface for authoring the data. Because they are both considered Type MC+ systems, why is there such a significant difference in the time required to convert their data? It is not clear why the times are so dissimilar; the only way to determine this would be to perform a controlled test where the exact same data is converted on each machine using similarly skilled operators.

Clearly, the above examples show that difficulties can arise when trying to estimate the cost of conversion on a per page basis. It is also evident that page-based cost estimates are not applicable to all conversion scenarios. Cost estimates per page would not make sense, for instance, when converting from a frame-oriented Type MB+ ETM to a Type MC or MC+ ETM. In these cases, cost information may seem to make more sense on a per tag basis. One conversion service bureau noted that they commonly charge per tag and per image, giving examples such as:

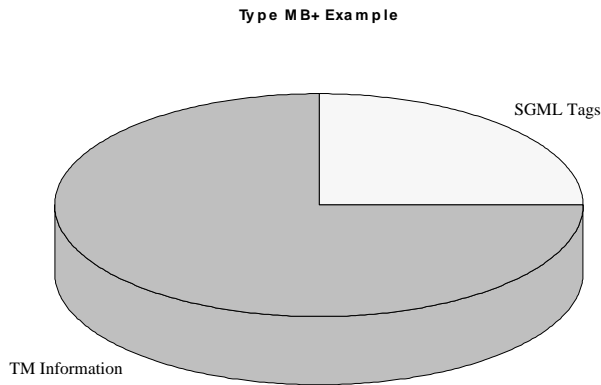
- 6 cents per tag for structural tags.
- 10 cents per tag for content tags.
- 12 cents per tag for index tags.
- 45 to 55 cents per image for scanning of raster images.

Looking at the cost of a conversion project on a per tag basis cannot be accurately determined from this data alone. The reason is that this data does not take into account which of those tags can be placed automatically versus those which must be placed manually (which brings us back to mapping complexity). These prices also do not include the filling of attributes or the use of SGML ENTITYs, etc.

Although cost cannot be determined accurately from the tag data given, this data can be used to provide some insight into the relative cost and complexity of converting to ETMs of different Types or Classes. This can be done because one will usually follow different DTDs with different requirements.

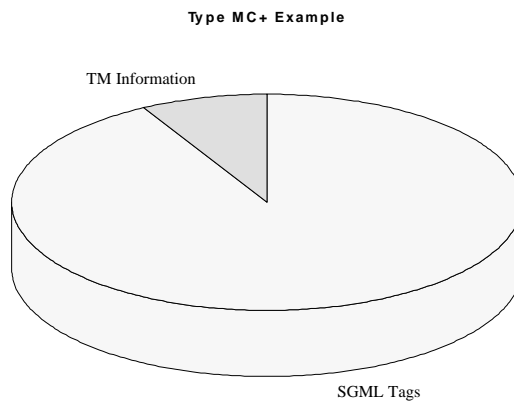
The first example of looking at cost on a per tag basis is the result of internal tests. In one test, for instance, a Type MA Organizational-level TM [26] was converted to a Type MB+ ETM using a scanner, OCR package, image editing package, and the IADS system. In this example, 22 pages of the manual were converted. The pages that were chosen included descriptive information, maintenance task procedures, troubleshooting information, and front matter.

For tagging, a subset of the IADS DTD Version 1.3 was used and consisted of 18 unique SGML elements that were principally structural tags. Note that the IADS DTD Version 1.3 was developed by MICOM for technical manuals. Using this set of 18 tags, the entire marked-up file consisted of a total of 296 tags. Also, the entire SGML file consisted of approximately 29,301 bytes of ASCII text where 25% of the text was due to markup and the other 75% was actual TM information, Figure 4.11.1-2. If the cost of placing each of these tags were 6 cents per tag, this would result in a cost of about 81 cents per page of converted data with an average of just less than 14 tags per page.



**Figure 4.11.1-2 Technical Manual Information vs. Tags in the IADS  
Type MB+ Data Conversion Experiment.**

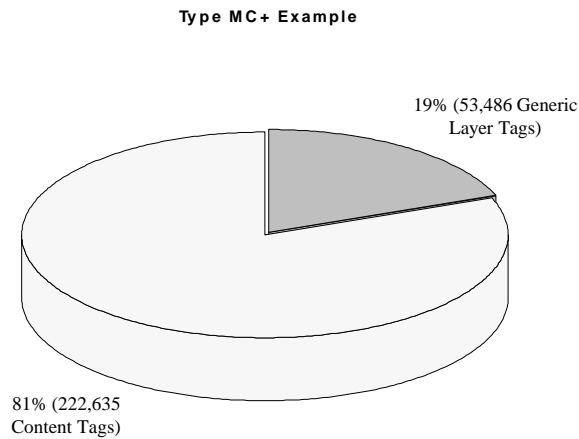
The next example of the relative cost on a per tag basis is non-SGML Type MB Organizational-level TM data that was converted to a Type MC+ ETM. This data was obtained from Lockheed Fort Worth. In this example, the conversion involved just under 4,000 pages of TM data. Without attributes, a total of 276,121 tags were represented in the final Type MC+ data (there were over 50 unique SGML elements, including different binary operators and unary operators). The entire SGML file consisted of approximately 20,294,332 bytes of ASCII text where 92% of the text was due to markup and the other 8% was actual TM information, Figure 4.11.1-3.



**Figure 4.11.1-3 Technical Manual Information vs. Tags in the Type  
MC+ Data Conversion.**

Although the Type MC+ tags would be better described as content tags, for the sake of comparison with the above Type MB+ example, if the cost of placing each of these tags were 6 cents per tag, this would result in a cost of about \$4.43 per page of converted data with an average of just less than 74 tags per page. It is interesting to note that out of the total number of

SGML tags, 53,486 were taken from the content-specific layer DTD in MIL-D-87269, and the other 222,635 tags were taken from the generic layer DTD, Figure 4.11.1-4. This indicates that tagging the content alone, at a price of 6 cents per tag, would be about 86 cents per page with an average of just under 15 tags per page. Above, it is noted that content tags cost 10 cents per tag; this would result in a cost of roughly \$1.44 per page of content-specific tagged data.



**Figure 4.11.1-4 The Number of Generic vs. Content-Specific Elements used in the Type MC+ Data Conversion**

From the tags used in the Type MC+ conversion example, a subset that could be placed in a primitive fashion (in accordance with Type MC) would consist of the ones listed below:

- |           |          |
|-----------|----------|
| alert     | person   |
| consum    | refmat   |
| descinfo  | reqcond  |
| equip     | step     |
| follow-on | system   |
| graphic   | task     |
| input     | techinfo |
| partinfo  | step-seq |

These tags have associated information that is relatively easy to pick out from any form of TM or ETM. In the example just given, the data consists of 42,769 of these tags.

#### **4.11.2 Cost Justification**

The cost of converting to or creating an ETM must be not only estimated but also justified. The justification phase may be carried out in any number of ways, and the term Cost Benefit Analysis (CBA) usually surfaces [34]. According to [22], a CBA is usually used to measure savings and improvement after a solution has been selected, designed, and sometimes even implemented. In this sense, when attempting to decide between ETMs of various Types (Classes), with various systems, a Business Case Analysis (BCA) is more appropriate.

As further defined in [22] [23] [24], a Business Case (BC) is prepared to allow for evaluation and selection of alternative investments for business process improvements. The BCA identifies the best areas of investment and compares the proposed cost of making these investments to the cost of maintaining the current status quo. Furthermore, the BC approach is considered in [23] to be fundamental to justifying proposals for new information technology initiatives.

According to [22, 24], a BCA includes:

1. An analysis of functional process needs or problems.
2. Proposed solutions, assumptions, and constraints.
3. Performance metrics and drivers.
4. Cost allocations and savings.
5. Investments and risks.

Without going into any significant detail, the proposed solutions may consist of deciding whether to convert Type MA manuals to Type MB+, MC, or MC+. Naturally, different costs are associated with each. The type of manual chosen will also affect the way in which the data is authored or converted, the extent to which the conversion can be automated, and even the way in which maintenance is ultimately performed.

Determining cost savings, as noted in item 4, may not be straightforward. Cost savings, because of the use of ETMs, may manifest themselves in a variety of ways that are not easily quantified. One example of a cost reduction that is hard to quantify is the increase in troubleshooting accuracy that is likely to occur. Increases in troubleshooting accuracy, among other things, lead to reduced false removal rates. More details concerning the advantages that some ETMs can have in this area are given in Section 4.5.4. Another cost issue that is hard to quantify is the instant accessibility to information that an ETM can provide. Northwest Airlines, for instance, noted a 50% time reduction for the mechanic to locate appropriate documents [25]. It is also exceptionally difficult to quantify the reduction in cost because of ETMs that incorporate interfaces to internal diagnostic and built-in test equipment.

Cost benefits that should also be taken into account are those due to reduced costs for updates and revisions. TM information is subject to change as mistakes are found and also as the weapon

system's requirements change during the life cycle. With an ETM, changes may be incorporated electronically and will not require reprinting. Where the ETM data is extracted from a single source, only the central ETM or ETM database would need to be changed or modified. With this method, the ETM would always contain the most accurate TI. Some estimates claim that the cost savings from the reduction in printing alone is 30% [34]. Benefits of using SGML tagged data, alone, can be significant and are touched on in [21].

Other cost benefits include reduced printing costs, reduced weight and volume, reduced technician training costs, and multi-user access. By comparison, ETMs have been noted to require approximately 1%-2% of the weight and volume of traditional paper TMs [34]. The weight and volume of paper TMs can be significant. For example, it is noted in [8] that an *Aegis* class ship houses approximately 37 tons of paper, filling roughly 1850 cubic feet of storage space. Obviously, reductions in weight and volume here would be highly desirable.

Clearly, a number of issues can surface when attempting to justify the cost of creating or converting to ETMs. This subsection has attempted to point out some of those issues and has further identified the BCA as being an appropriate method for providing cost justification. The justification primarily arises from the evaluation of alternatives and comparing them to the cost of maintaining the current TM or ETM methodology. Details concerning proper preparation of a BCA for an IWSDB are being done within the Business Case Development Task (Task 7) of this project. The results of that task should be consulted, because they will also assist in the development of a BC for ETM conversion projects.

## 5.0 VALIDATION, VERIFICATION, AND ACCEPTANCE TESTING

This section provides an overview of validation, verification, and acceptance testing issues that may be encountered when converting legacy paper and/or electronic TM or ETM data. Currently, a consistent methodology, applicable for all Types or Classes, for contractor validation, government verification, and government acceptance testing of either authored or converted ETM information, does not exist. After establishing appropriate definitions, this section will discuss these topics as well as some controversial issues associated with them.

### 5.1 Definitions

Before delving into the technical issues associated with validation, verification, and acceptance testing, it is important, for the sake of clarity, to show how these terms are characterized within this document. For the purposes of this report, validation, verification, and acceptance testing of ETM data for any Type or Class will be defined very similar to the way in which they are defined in MIL-Q-87270. To be precise, these terms are defined, within this report, as follows:

**Validation:** Validation is a *contractor* performed QA process in which an ETM or the data that constitutes that ETM is compared in detail against the system supported by the ETM, or against other approved source data, to assure that the presented ETM data is technically accurate in all respects and that it conforms to all requirements of the contract. Validation is the most important of the contractor's QA actions. Validation should be done in accordance with a previously developed validation plan. Validation involves tests of the ETM information against the associated hardware by actual performance of the procedures by contractor personnel using a Government furnished Electronic Display Device (EDD) and other Government approved support equipment in accordance with the validation plan. The main objective of validation is to *ensure* that the contractor has provided, for support of the equipment involved, *accurate* and adequate ETMs and/or ETM data.

**Verification:** Verification is a *Government* performed process carried out by qualified personnel, of the prescribed skill level, from the operating command or facility assigned to operate and maintain the weapon system to which the ETM applies. The purpose of the verification process, which will be carried out on a representative EDD and supporting systems, is to ensure, as a minimum: (a) the suitability of the ETM for the intended maintenance environment; (b) its usability by its intended users; and (c) its compatibility with Government systems. The Government, in general, will perform acceptance tests on the ETM subsequent to approval of changes incorporated into the ETM as a result of verification findings.

**Acceptance Test:** The final test series performed by the *Government* on the ETM, following verification. If the ETM is considered satisfactory as a result of these tests (meets all contractual requirements) it will be accepted by the Government.

## 5.2 Conversion and the Validation Process

Given the definitions in the previous section, it is important to understand that validation, verification, and acceptance testing all depend on the target ETM Type or Class. Throughout this report, it is assumed that the target ETM data is created by way of a conversion process. Furthermore, the conversion methodologies are based on the premise that the (to be converted) source data, whether or not it was on paper or in electronic form, was previously validated by the contractor and both verified and accepted by the Government.

Before going further, it is important to note that in the course of this project, both NIST and the Air Force CTN were researched as possible ETM validation and verification centers. However, neither activity currently supports comprehensive testing of ETM systems or ETM data sets of any complete Type or Class. Both activities do, however, provide testing tools and/or services for verifying conformance with various graphic file formats and for dealing with SGML compliance issues.

The fact that converted data is based on data which has already been validated, verified and accepted has resulted in general confusion as to the scope of the validation which must be carried out by the contractor. For example, consider that in the case of creating (rather than converting to) Type MC+ data, the validation process given in MIL-Q-87270 consists of the following elements:

- a) Preparation and approval of a validation plan.
- b) Engineering review and certification that IETMs and related technical information are ready for validation.
- c) Validation of IETMs and related technical information.
- d) Validation of the IETM Database.
- e) Validation of any automated IETM compilation process.
- f) IETM system validation.
- g) Post validation review (for Navy IETMs only).
- h) Preparation of validation documentation.
- i) Validation certification.

The confusion associated with validation and conversion efforts manifests itself in step c). The technical information that constitutes the converted ETM has already been both validated, verified, and accepted and therefore effects the extent to which some feel that step c) should be carried out. When creating (rather than converting to) Type MC+ data, step c) requires a comprehensive look at all the technical information, i.e., every step of every procedure is validated by an appropriately skilled individual (not one of the ETM authors or system engineers, etc.), every graphic in every step must be properly and comprehensibly displayed on the EDD (which may have size and resolution constraints), troubleshooting procedures must be validated with the weapon system hardware to which the troubleshooting procedures apply, etc.

Obviously, full validation of newly authored technical manual information can be a long, arduous, and expensive process and that is why those that carry out conversion projects would like to be able to avoid it if at all possible. In the F/A-18 conversion effort, for instance, it is noted in [40]

that with legacy data, the expense of validation and verification has already been incurred by the contracting customer and unless this data can be converted with the certainty that the user-supplied data has not changed, the validation/verification process will have to be repeated. Moreover, in [40], repeating this process is pointed out as being very costly for any aircraft weapons platform.

It is interesting to note that, according to [40], they initially thought that their conversion methodology was automated enough and the presentation of the procedural data to the user was similar enough to the original that no validation/verification would be necessary. In the end, however, this was not the case. They eventually came to the conclusion that their ETM data should be validated/verified again before being put into actual use. They also noted that re-validation of their (dynamic) diagnostic data was mandatory because the data was interchanged and presented in an entirely different manner.

### **5.3 The Validation Plan**

When creating (rather than converting to) Type MC+ data, according to MIL-Q-87270, the first step in the validation process is for the contractor to prepare a validation plan. Generally, the validation plan as described in MIL-Q-87270 must contain, at the very least, the following items:

- a. Titles and identification numbers (when available) of all deliverables to be validated.
- b. Schedules for validation steps.
- c. Cognizant contractor organization and personnel responsible for accomplishing the validation effort.
- d. Site locations, support equipment, facilities, test equipment, materials, contractual end items, and tools required during validation.
- e. General characteristics relating to skill level of target audience type personnel to be used in validation of procedural TI.
- f. Identification of next higher assembly(ies)/system(s) required to support the effort.
- g. Associated TI recommended for concurrent or consecutive validation.
- h. Special safety precautions.
- i. Any special environmental requirements.
- j. Record keeping system to be used in validation.
- k. How validation of procedures is to be accomplished.
- l. A proposed fault simulation list for use in validation of troubleshooting procedures.
- m. Approach to evaluation of usability of the IETM as generated by the contractor; specifically, establishment of the adequacy and accuracy of information access procedures, and identification of any potential man machine interface problems involving effective use of the IETM on the Government specified EDD.
- n. Application of results of tests performed during the validation process to the correction of deficiencies.
- o. When required by the contract, a technical approach to validation of the IETM Database, differentiating between computer supported technical procedures (e.g., automated checking routines) and those to be carried out by humans (e.g., reading and comparing data base entries with external data).

- p. When required by the contract, a technical approach to validation of any automated IETM compilation process.

When transforming or converting TM or ETM data to a format which does not comply with the IETM specifications, a validation plan which contains all of the above elements may or may not be necessary, depending on the specific contractual requirements. In any case, however simple or complex the validation process is expected to be, creation of a comprehensive and concise validation plan is certainly a good idea. Some of the more important elements that a validation plan for converted ETM data may ultimately need to address are given in Table 5.3-1.

**Table 5.3-1 Elements to be Included in a Validation Plan for Converted ETM Data**

1.	Validation of the technical information.
2.	Validation of the ETM data as presented on a Government furnished EDD with attempts to identify potential man-machine interface problems.
3.	Validation of any automated work-package or view-package extraction processes.
4.	Validation of the authoring and presentation systems themselves as software systems to be used in conjunction with the ETM data.
5.	Validation of the ETM presentation system in conjunction with the ETM data set.
6.	Validation of the underlying ETM data in terms of structure and function.
7.	Validation of the underlying ETM data in terms of data format.

The first element in Table 5.3-1 that is suggested for a validation plan is certainly the most important. The technical information must be valid and, as seen by the F/A-18 experience in [40], how the contractor chooses to validate the converted data depends not only on the automation involved and the similarity of the presentation, but also on how much confidence the contractor has on their own conversion methodology. Note that validation of Type MC and MC+ data that may be context-filtered is particularly challenging and may require many passes under different conditions to verify that the data both exists and is technically accurate. More on this is given in Section 5.4.

The second element in Table 5.3-1 is important because the EDD that is used in the field may have a smaller screen size, the screen may be of a lower resolution, and it is important to note that these two simple factors may ultimately be much more of a hindrance to the technician than using a paper manual. It is also important to recognize that size, weight, and connectivity requirements may cause additional problems for the technician. This problem, for instance, was encountered by SAIC while creating an ETM for the M916 Truck for the Army. The ETM was displayed on a Contact Test Set (CTS) which was not only heavy (for a portable computer) but also could not be moved easily because it was connected to a data bus on the M916 by way of a very large cable. They solved this problem by using a very thin portable terminal with a notepad-like screen which did not require any external connections and communicated with the CTS via an RF link.

The third element that the validation plan should address is the valid operation of the view packaging process (if one is used). The view-packaging process in this sense refers to the extraction of particular maintenance tasks from a central ETM database to perform maintenance (i.e., similar to selecting an appropriate manual or work-package). A good example of a system with this capability is the Lockheed Fort Worth system as described in Section 4.6.1. The important thing here is to ensure that the ETM data is exported properly and that there is a sound methodology for handling the inevitable broken links that will occur as a result of extracting only subsets of the ETM database.

The view-package extraction process described above brings to mind the concept of extracting ETMs from an IWSDB. The term IWSDB referred to here is considered to be consistent with the definition given in [45]. Although integrated environments are generally desirable, they can create difficulties for validation. Some members of the TSWG indicated that validation problems could arise if all ETM data is extracted from various disparate databases at presentation time. In short, in terms of validation, stand-alone ETMs, where the ETM data can be strictly controlled, are favored.

The fourth element refers to the validation of the authoring and presentation systems as software systems irrespective of the maintenance information that they are intended to display. This element is discussed in Section 5.5. The fifth and sixth elements in Table 5.3.1, although listed separately, are somewhat interrelated and are addressed in Section 5.6. The seventh element is primarily a CALS-related issue and is discussed in Section 5.7.

#### **5.4 Validation of Technical Manual Information and Personnel Issues**

When creating Type MC+ data and no validated/verified/accepted legacy data exists, the validation of the technical information, according to MIL-Q-87270, must be performed by individuals who are of approximately the same education, experience, and skill level as the actual target audience of the IETM. Note that the target audience refers to the maintenance technicians that will actually be using the IETM to perform maintenance. It is further stated in MIL-Q-87270 that those involved in authoring the IETM cannot be expected to provide a realistic test of the validity of the IETM.

The personnel requirements outlined in MIL-Q-87270 are certainly reasonable for newly authored ETM data of any Type or Class. When considering validation of converted data, however, it is important to look at this issue in more detail. According to the definition of validation given in MIL-Q-87270, validation can consist of comparing the final data to some type of “source” data to ensure its technical accuracy. Since, in this report, validated, verified, and accepted information is being converted, there is actually an excellent “source” for validation of the converted data. That “source” is of course the legacy paper and/or electronic manuals that are undergoing conversion.

The question of how the legacy data should be validated now becomes, who is best qualified for comparing the source data to the converted data for the purposes of validation? Subject matter experts in the form of maintenance technicians are certainly the optimal choice for validating newly authored ETM data and for determining readability and comprehensibility of either newly

authored or converted ETM data. However, those same persons may not be the optimal personnel for determining how well the converted (target) ETM data agrees with the source data. To this end, technical writers and possibly even system design engineers (for validating the converted troubleshooting information) from the contractors who built the weapon system itself may be either preferred or required.

Determining the degree to which the target data is consistent with the source data may or may not be difficult and clearly depends on both the source and target TM/ETM Types or Classes. A match can be easily determined, for instance, when the source is Type MA and the target is Type MA+ or MB. A match can be much more difficult to determine, however, when the source is chapter-based Type MA in a non-work package format and the target is Type MC+ and follows the DTDs given in MIL-D-87269. Matches are more difficult to discern because the original manual will be structurally reorganized in terms of systems, subsystems, etc. where each has procedural (tasks), descriptive, troubleshooting, and parts information. Aside from figuring out where the information is located, the problem of determining a match is particularly difficult for procedural steps associated with dynamic graphics and also for troubleshooting information that may have undergone significant re-authoring.

## **5.5 Validation and ETM System Software**

Validating the authoring and presentation systems themselves may fall within the scope of a conversion project if these systems are not provided as Government Furnished Equipment (GFE). Note that in the case of Type MC+ data, the issue of IETM system software, in MIL-Q-87270, is within the scope of the quality assurance program but is not listed as being a part of the validation plan. Regardless of what the most appropriate category is, as with IETM system software in MIL-Q-87270, any authoring and/or presentation system software that is developed for an ETM system of any Type or Class should be developed in accordance with DOD-STD-2167 or DOD-STD-7935 and in either case should conform to the requirements of DOD-STD-2168. Note that, in the future, it is expected that both DOD-STD-2167A and DOD-STD-7935A will be superseded by MIL-STD-SDD. MIL-STD-SDD is a draft DoD standard issued on December 22, 1992. One of its objectives is to create a single DoD software development standard applicable for any type of software by merging DOD-STD-2167A and DOD-STD-7935A and the Data Item Descriptions (DIDs) associated with these standards.

The issue of DoD software quality assurance is currently a subject undergoing some debate. One of the issues concerns ISO 9000 versus DOD-STD-2167A/DOD-STD-7935A and DOD-STD-2168. In [42] it is recommended that DoD software development should conform to ISO 9000 by a certain time. More specifically, it is recommended in [42] that DoD should insist that its software developers commit to ISO 9000-3 and ISO 9001. ISO 9000 is a series of quality assurance standards, and as indicated in [42], are applicable for any business, whether in manufacturing, service, retail, or government, in producing a product or service. It is further pointed out in [42] that the ISO 9000 series standards have been adopted by 60 countries including the U.S., Canada, Japan, and all the members of the European Community and also if a business wishes to sell its products or services within the European market or to certified ISO businesses, it needs to be certified ISO 9000 too.

Although ISO 9000 is clearly important, it is considered by some to be inferior to DOD-STD-2167A and DOD-STD-2168. Specifically, critics claim that ISO 9000 documentation requirements are vague, inconsistent, and incomplete while further claiming that, in comparison, DOD-STD-2167A documentation requirements are known and well-defined. The extent to which this becomes a problem is uncertain at this time but this issue is likely to be directly affected by the June 29, 1994 memorandum signed by Secretary of Defense William Perry. That memorandum is reprinted in [43] and emphasizes the use of performance and commercial specifications and standards over military specifications and standards. Furthermore, the use of military specifications and standards is considered a last resort and can only be authorized with an appropriate waiver.

Assuring the integrity of the ETM system software is further addressed for Type MC+ systems in [27]. In that document, it is recommended that testing of the authoring and presentation systems be done by using test data sets and test scripts that could be made available by a testing agent. It is important to keep in mind, however, that according to MIL-Q-87270, these test scripts must be developed in accordance with DOD-STD-2167 and DOD-STD-2168. In the future, when ETM Types or Classes stabilize, perhaps a testing agent could be established that can either validate/verify or at least provide data for testing both authoring and presentation systems for all ETM Types or Classes.

In this context of ETM system software validation, it is also important to note that in [44] it is shown that the most successful DoD software development projects have utilized independent test and evaluation agencies [36]. These agencies are either separate independent contractors or "in-house" agencies that are independent from the software development agency. The Independent Validation and Verification (IV&V) approach is a DoD directive that promotes objective, impartial, and unbiased testing. IV&V is a general requirement of DOD-STD-2167A and should be conducted by an independent and accredited testing activity.

Currently, a validated/verified/accepted authoring system would essentially be certified as an approved ETM authoring system. Likewise, a validated/verified/accepted presentation system would simply be certified as an approved ETM presentation system. In the future, it is hoped that the certification could go one step further and identify, precisely, a particular ETM Type or Class that the systems are approved for.

## **5.6 Validation and ETM Data Sets**

This section will focus on validation of the ETM data set. Validating an ETM data set involves not only testing of the structure and function of the ETM data but also testing that the data is properly prepared for the particular presentation system that will display it. The latter point is important because, as discussed in detail in Section 4, different ETM presentation systems process and interpret ETM data differently. This, however, should not be a significant problem if the presentation system that will ultimately be used was obtained at the start of the conversion project to ensure compatibility with the created/converted ETM data.

### 5.6.1 Type MC+ ETM Data Sets

Validating the structure and function of ETM data can be particularly difficult when dealing with Type MC and MC+ data. In their SGML formulations, this data is likely to contain runtime variables, conditional branching, expressions, linking, and other forms of complex SGML structures. Validating that the underlying data is properly constructed here is similar to the system software tests in paragraph 3.6.6.2 of MIL-Q-87270. This problem has been studied, in detail, for IETM (Type MC+) data in [27]. Moreover, that study recommends the following tools for testing of SGML IETM data sets (Type MC+ data sets):

- Questionnaire,
- Check List,
- Test Scripts,
- Test-Specific DTDs,
- SGML/HyTime Parser and/or Engine and SGML Viewer Module,
- Reference Implementation,
- Test Data Set,
- Content Analysis Module, and
- Graphics Test Suite(s).

That report also identified four tools that would be helpful in creating the ones mentioned above:

- Data Management Module,
- SGML Editor/Viewer Module,
- Regression Analysis Module, and
- DTD Subset Validator.

Note that the Regression Analysis Module simply takes output files from the same system and detects any historical differences between them.

The Questionnaire and Check List are used mainly to ensure that contractually specified features and requirements are tested. The Test Scripts then direct the tester to the appropriate tools and procedures to carry out the test. According to MIL-Q-87270, the test scripts and testing methods should at a minimum perform the tests summarized below:

1. The IETM data meets all specific requirements of MIL-D-87269.
2. All relationships (established data access links), are in place, valid, and operable.
3. There are no open loops in the fault isolation tasks (i.e., that each task ends in a single resolution, and each retrieval string ends in successful isolation and repair of a discrete fault).
4. There are no unreferenced or inaccessible data entities in the system.
5. Check that all special functions are fully operable.

The reference implementation is perhaps the most important part of these test tools. It is a 100% Tri-Service specification (MIL-M-87268 and MIL-D-87269) compliant IETM (a Type MC+

ETM). Note, in particular, that [27] places the following requirements on the reference implementation:

1. The reference implementation must be completely checked for conformance before using it to check other deliverables' conformance.
2. It must be able to read MIL-D-87269 and MIL-M-87268 conforming IETM's.
3. It must be able to read all graphics formats used in conforming IETM's.
4. It must be able to clearly interpret and describe errors found in the input to the reference implementation.

In [27], the need for the development of the reference implementation is considered to be highly urgent, it is expected to be very difficult, and it is expected to depend on an appropriate SGML/HyTime Parser and a Graphics Test Suite.

The requirements on the reference implementation are very important to this task. One of the most difficult requirements is that it must be able to read MIL-D-87269 and MIL-M-87268 conforming IETMs (Type MC+ ETMs). Given the current set of specifications this will, as alluded to in Section 4, be next to impossible unless there are reference implementations for each MIL-D-87269 compliant variant that exists. The specifications will have to be more specific if DoD intends to have a reference implementation that will be able to read arbitrary specification-compliant files. The only solution here may be to ultimately target a single compliant system that best performs the desired functions and call it the reference implementation. Then be sure that all other Type MC+ ETMs under development are validated using this system as the reference implementation.

### **5.6.2 ETM Data Sets in General**

The testing elements recommended in [27] and discussed in Section 5.6.1 are relevant for Type MB, MB+, MC and MC+ ETM data sets. Generally speaking, the following elements were taken from Section 5.6.1 and are both applicable and straightforward tests which can be performed on Type MB, MB+, MC, and MC+ ETM data sets:

1. Perform syntactical checks for compliance with the tagging conventions used in the selected ETM Type or Class.
2. Confirm parsability of tagged ETM data set (DTD) by SGML/HyTime parser.
3. Confirm validity of graphic file formats.
4. Check that all data access links and branches (if applicable) are valid and operable.
5. Depending on the Type or Class and re-authoring scheme, when applicable, ensure that there are no "open loops" in the fault isolation tasks (i.e., that each task ends in a single resolution, and each retrieval string ends in successful isolation and repair of a discrete fault).
6. Ensure that there is no unreferenced or inaccessible data in the system.
7. Check that all special functions are fully operable.

The idea of a reference implementation, as described in Section 5.6.1, is of consummate importance for all ETM Types and Classes. When the ETM Types or Classes stabilize a reference implementation will be necessary for each Type or Class. In the interest of standardization and interoperability, ETM data from all Types or Classes should not only be similar in appearance and function to the reference implementation, but also the underlying ETM data from each Type or Class should be 100% compatible with that required by the appropriate reference implementation.

### 5.6.3 Software and Types MC, MC+ Data

We will close this section with a brief look at an issue that could affect how the more advanced ETM methodologies mature. In simple terms, the issue concerns the extent to which Type MC and MC+ ETM *data* is considered software rather than simply data. Full SGML implementations of Type MC and MC+ data have many of the capabilities of normal computer programming languages. Recall the runtime variables, looping, and conditional branching capabilities. In DOD-STD-2167A computer software is defined as:

**Computer software (or software).** A combination of associated computer instructions and computer data definitions required to enable the computer hardware to perform computational or control functions.

In DOD-STD-7935A a computer program is defined as:

**Computer Program (program).** A separately compilable, structural (closed) set of instructions most precisely associated with early generations of computers.

In the December 1992 draft version of MIL-STD-SDD there is a distinction between application software, system software, and support software:

**Application software.** Software designed to fulfill specific needs of a user; for example, software for navigation, payroll, or process control. Contrast with: support software; system software.

**Support software.** Software that aids in the development or maintenance of other software, for example compilers, loaders, and other utilities. Contrast with application software; system software.

**System software.** Software designed to facilitate the operation and maintenance of a computer system; for example operating systems, diagnostic software, and other utilities. Contrast with: application software; support software.

Looking at the above definitions, Type MC and MC+ ETM data could, in some sense, be considered the application software which would result in the authoring and presentation systems being considered the support software.

The purpose of looking at this issue is to call attention to the fact that Type MC and MC+ data could be considered software and because of this the issues of ETM data development, interoperability and interchange could, in the future, be addressed in ways that are similar to the way in which software engineering issues (i.e., software reuse, modular/structured programming, etc.) are currently being addressed.

## **5.7 CALS and Converted ETM Data**

In addition to validating the ETM systems and the integrity of the converted information, it is also important to recognize that some attention must be given to validating the underlying data formats. This step may in fact be necessary whenever those underlying formats are intended to comply with military, national, or international standards of any kind. This is actually rather important because systems do exist which claim conformance to various standards but, in reality, they do not. Giving particular attention to the CALS standards, if the graphics are to be in CGM, IGES, and/or CCITT Group IV raster and the text is in SGML, for instance, then the contractor should provide some method for assuring that the files that they deliver actually comply with the appropriate standards. Identifying test tools for these CALS standard formats was one of the tasks within the OSD CALS IWSDDB project. The results of that task are given in [41].

## **5.8 Verification**

As defined in Section 5.1, verification is performed by the Government. When verifying Type MC+ data, according to MIL-Q-87270, the contractor, when specified by contract, may be required to both develop a verification support plan as well as support the verification process. In MIL-Q-87270, this support may, in some cases, consist of:

- a. Technical support,
- b. Assistance in performing verification procedures,
- c. Recording and disposition of verification comments, or
- d. Certification that all discrepancies and defects noted during verification have been corrected or resolved.

It is not expected that the verification guidelines outlined in MIL-Q-87270 would be significantly different from those which would be required for newly created (rather than converted) ETMs of any Type or Class. Currently, no guidelines exist for performing either verification or verification support of an ETM which was created from *conversion* of TM or ETM data that was previously validated, verified, and accepted by the Government.

## **5.9 Acceptance Testing**

Government acceptance testing will generally be performed on ETMs of any Type or Class. It will occur upon final submission of the delivered product. During acceptance testing, the ETM will be tested by the Government for compliance with all requirements of the contract. For informational purposes, Appendix B of MIL-Q-87270 gives an overview of the steps that may be involved in acceptance testing of Type MC+ ETMs delivered to the Navy. To be precise, acceptance tests, performed by the Navy, on a Type MC+ ETM will be conducted in two phases:

1. Tests to assure that the ETM functions satisfactorily when viewed with a test EDD; specifically, that all special functions are available, calling sequences are correct, and in general that all capabilities requested have in fact been provided.
2. A thorough review of the content of each screen to assure that no procedures have been left out, the material is written comprehensibly, the technical information is error-free, troubleshooting information in fact leads to fault isolation in all branches, and that the technician is informed as to the calling sequences for required information and can obtain it quickly and directly.

These tests are outlined in greater detail in Sections 50 and 60 of Appendix B in MIL-Q-87270. Although not stated in the specification, generally speaking, the Navy methodology for performing acceptance testing is applicable, and may not differ significantly, among the other services. Also, aside from the references to MIL-M-87268, the Navy methodology is generic enough such that it can be applied, by the Government, to other ETM Types or Classes with little modification. As with validation and verification, no specific guidelines currently exist for performing acceptance tests on an ETM which was created from *conversion* of TM or ETM data that was previously validated, verified, and accepted by the Government.

## 6.0 FINDINGS AND RECOMMENDATIONS

As discussed throughout this report, the primary difficulty with establishing conversion procedures is the fact that for most Types or Classes of data there are different ETM implementation possibilities. Currently, the three services, and programs within those services, are using ETM systems which meet their individual needs but are generally incompatible with each other. Although this is currently the case, it need not be that way in the future. The development of standards and specifications for ETMs are still in their early stages. Given this, it is of some value to have a number of diverse implementations, such as there are now, which can assist in the development of an appropriate and refined set of final ETM standards and specifications.

Currently, embarking on any project which involves creation of an ETM or creation of ETM data can be difficult. One of the reasons it is difficult is due to the fact that there is not a single authority or office within DoD with a clear mandate to provide the guidance which is needed. Guidance is needed particularly on:

- ETM Type or Class selection (with BCA or CBA data to back it up).
- ETM authoring system selection.
- ETM presentation system selection.
- DTD selection (when applicable).
- Authoring, conversion, and presentation of troubleshooting information (appropriate for selected Type/Class and system).
- Authoring, conversion, and presentation of procedural maintenance tasks (appropriate for selected Type/Class and system).
- Authoring, conversion, and presentation of part information (appropriate for selected Type/Class and system).
- Authoring, conversion, and presentation of descriptive data (appropriate for selected Type/Class and system).
- Authoring, conversion, and presentation of graphics (appropriate for selected Type/Class and system).
- Validation guidelines and procedures for both authored and converted ETM data of all approved Types or Classes.

During this task, many ETM-related COTS products were investigated and evaluated. Some vendors, although they cooperated, noted that they are displeased with the fact that they are quite often contacted by independent entities within or representing DoD who want to evaluate their product(s) for use with ETM data. It is certainly true that the functional capabilities of these COTS products are not always easy to understand and are also changing at a very high rate. If there was a single office within DoD that provided guidance on development of ETMs, that office should also keep abreast of changes in the marketplace by evaluating and re-evaluating relevant COTS products as necessary. Having a single source which is responsible for doing in-depth evaluations of these COTS products would certainly be more efficient than having evaluations carried out (almost) simultaneously by persons who may not even understand what questions should be asked or which features are most important for a proper evaluation. Ideally, the ETM office should also be responsible for conducting certification of authoring systems, presentation

systems, and COTS packages which satisfy the requirements that will eventually be established for all ETM Types or Classes.

It is evident from this report that a lot of important issues must be considered before converting legacy paper or electronic TM data to ETM data of any Type/Class beyond Type MA+ or Class 1. The only DoD specifications which currently exist for creation of ETM data are the IETM specifications (MIL-M-87268, MIL-D-87269, MIL-Q-87270). This current set of IETM specifications is not sufficient for assuring interoperability among IETM implementations. Creating an IETM database or a DTD by following the Content Data Model in MIL-D-87269 is not sufficient for assuring that IETM systems will be capable of exchanging their data without an independently developed intermediate translation step. Furthermore, following the IETM specifications does not even guarantee that two IETM systems which can both import and export their data using the same generic layer and content-specific layer DTDs will be capable of exchanging their data without an independently developed intermediate translation step. Moreover, these intermediate translation steps may or may not be entirely automatable.

Aside from those systems that attempt to comply with the IETM specifications, ETM systems, in general, which import, export, or compile and present SGML data are not necessarily capable of directly exchanging data with other ETM systems with similar capabilities. The main problem here is that the SGML elements which are used for ETM presentation do not have a standardized way of exchanging processing information associated with non-display-specific elements. In other words, creating paper pages from SGML by following MIL-M-28001 can be easily standardized because a FOSI is used to tell the SGML compiler/formatter how the data placed within the tags will appear (i.e., all data within <TITLE> tags will be centered and in bold face) What is needed for developing interoperable ETMs, therefore, is a standardized way of associating processing instructions with arbitrary SGML tags. One alternative, of course, is to simply define a set of DTDs for each appropriate Type or Class of ETM while stating and enforcing the fact that certain elements will have certain processing capabilities (i.e., IF-THEN). The presentation systems would then have to be designed or modified accordingly. A better alternative, perhaps, would be to allow for arbitrary DTDs while having a handful of reserved elements for implementation of the processing capabilities (similar, in some respects, to IADS).

Problems with using SGML were also noted by those who participated in the development of the LM2500 Gas Turbine ETM. In particular, it is stated in [14] and [32] that sets of narrowly defined, content-intensive, low-level SGML tags such as those in the content-specific layer in MIL-D-87269 and also those in the complete set of MIL-M-28001B have resulted in many "CALs-compliant" vendor products that are unable to recognize or exchange one another's data. Furthermore, they state that a standard set of high-level tags would solve this problem and DoD could more quickly realize its CALS objectives of data digitization, commonality, and data exchange.

There are others within DoD and Industry that are aware of the problems of using SGML with ETM data. One proposed solution is to combine MIL-M-28001 with MIL-D-87269 into a single specification which would provide specifications for content-oriented SGML databases as well as a methodology for providing both interactive electronic and printed media. To support this idea,

it is argued that attempting to maintain both specifications will result in unnecessary cost expenditures for programs which are required to use them both. As sensible as this idea appears, some, however, are of the opinion that combining MIL-M-28001 and MIL-D-87269 is not feasible.

From our experience, it does not seem like a good idea to attempt to combine these two specifications now. The main reason is due to the present ambiguities associated with implementation of MIL-D-87269. MIL-D-87269 is not well understood. Those who attempt to comply with it are likely to turn immediately to the DTDs given in its appendices. The DTDs provided require a highly complex and structured way of using SGML for creating interactive electronic documents. When implemented correctly, the resultant SGML file will look very much like a computer program. Furthermore, it is not even widely understood that this SGML file can actually be processed for display without requiring an extra step for imbedding scripting information. A presentation system would not need extra tags for scripting CDM data if it is designed to handle the SGML methodology that is employed (i.e., it would know what to do when it sees an IF-STEP or a LOOP-PARA, etc.). Currently there are no IETM systems in existence that can handle *arbitrary* content-specific layers that are based simply on consistent interpretation of the generic layer and its templates. It does not seem wise to combine MIL-D-87269 with anything until it evolves to a state where it is well understood and can be consistently implemented.

The problems associated with the use of SGML in the above paragraphs clearly indicates that there is more to defining a CALS-compliant ETM than the underlying data formats. Consideration must also be given to the presentation system and how that presentation system both processes and displays the ETM data. Defining different Types or Classes of CALS-compliant ETMs is indeed a very complex problem because there does not yet exist a widely accepted standardized presentation or viewing format for even simple, static, page-based electronic documents [39], let alone for electronic documents that are interactive and dynamic.

Aside from interpreting SGML instances which may contain non-display-specific (processing) elements, defining a CALS-compliant presentation system for any Type or Class of ETMs implies that all compliant presentation systems will be capable of displaying CALS-compliant ETM data in a consistent fashion. The issue of consistent presentation is important. CALS-compliant presentation systems (at least among each Type or Class) should all have a similar look and feel so that technicians will not have to spend time learning how to use different presentation systems for the same Type or Class of ETM.

Currently, it appears that the only cost-effective solution to defining CALS compliant ETMs for each Type or Class is to select single reference implementations, one for each Type or Class, and ensure that all other implementations generate data compatible with the appropriate reference implementation. Reference implementations may impose some limitations in the authoring but the impact will be more positive than negative. One need only look at the World Wide Web as described in Section 4.6.2 to see how positive the impact can be. With the WWW there is essentially only a single DTD (which defines HTML) used by certainly many thousands, and perhaps even millions of people for making interchangeable and interlinkable hypermedia

documents. Yet, with a reference implementation similar to IADS, a single DTD need not be a requirement for ETMs. In this case, arbitrary DTDs can be made while a handful of elements are reserved for implementation of processing (i.e., IF-THEN) operations.

Due to the current state of ETMs and the IETM specifications, the issue of interfacing converted ETM data with the JCALS system, as discussed in Section 3.4, can only be resolved on a case by case basis. JCALS will be treated as a management system and it is envisioned that ETMs will be managed like other types of TMs. Given the current state of ETMs in DoD this will require that independent interfaces be developed between the JCALS system and each ETM authoring/presentation system that it is to be used with.

## REFERENCES

- [1] The U.S. Army Publications and Printing Command Digital Publications Standards Development, by Horace Layton, CALS Expo Nov. 93.
- [2] Technical Manuals, (MIL-STD-361A).
- [3] Manuals, Technical; General Style and Format of (Work Package Concept), (MIL-M-81927B(AS), 4/5/89.
- [4] Technical Manuals, (MIL-STD-361), 7/15/92.
- [5] System/Subsystem/Subject Number (S/S/SN) Numbering System, (MIL-STD-1808), 3/28/92.
- [6] List of NAVAIR Technical Manual Information Constructs by Specification, by Aquidneck Management Associates, Ltd. January 20, 1994.
- [7] Lockheed Fort Worth Company F-16 Interactive Electronic Technical Manual (IETM) Initiatives, by Vinnie Botticelli, Lockheed Fort Worth, February 18, 1994
- [8] The Gas Turbine Systems Interactive Electronic Technical Manual (IETM), Improving Maintenance & Streamlining the Training Pipeline, by CDR Clifford J. Alligood, SC, USN. NAVSEA/NAVSUP International Logistics Symposium, 1994.
- [9] "The F/A-18 IETM Diagnostic Demonstration", by 1Lt. Eric N. Carlson, USAF, Fall 1993 CALS Journal.
- [10] "The New Computer Graphics Metafile Standard: A General Purpose, International Standard for Structured Graphics", by John C. Gebhardt, PhD, and Lofton R. Henderson, in CALS Journal, Fall 1993.
- [11] The Internet Unleashed, by Sams Publishing 1994.
- [12] Making HTML a HyTime Application, by Elliot Kimber, First International Conference on the Application of HyTime, Vancouver, British Columbia. July 26-27, 1994.
- [13] The Data Conversion Project Planning Workbook, Data Conversion Laboratory, 1993.
- [14] Document Type Definitions: A Case For a Common Set of High Level Tags, by Michael Graser, NAVSEA/NAVSUP International Logistics Symposium, 1994.
- [15] "Section II, Product Sources by Standards Implementation and by Industry", in CALS Journal, April 1994.

- [16] The Contractor's Guide to CALS & Concurrent Engineering, by Beazley, William G. Pasha Publications Inc. and Knowledge Base International, 1991.
- [17] TRW CALS COTS Product Guide, by TRW CALS Program Office, TRW, Inc. One Space Park, Redondo Beach, CA, January 31, 1991.
- [18] UNLV Information Science Research Institute, 1994 Annual Report, by Information Science Research Institute, University of Nevada, Las Vegas, 1994.
- [19] Emerging Digital Technical Manual Formats and Management Issues, by Wayne Honea, NAVSEA/NAVSUP International Logistics Symposium, 1994.
- [20] "Implementing Integration Where the Sea Meets the Shore," by Rear Admiral James B. Greene, Jr., in CALS/Enterprise Integration, Summer 1994.
- [21] "What Every CIO Should Know About SGML", by Linda Turner, in CALS/Enterprise Integration, Summer 1994.
- [22] A Guide to Business Case Analysis for STEP Implementation, by PDES, Inc. September 1993.
- [23] Implementation of OLTRS Within DND, Business Case, Prepared for Chief of Financial Services of the Department of National Defense, by Rockwood Informatics Corporation, July 1991.
- [24] Preliminary Model Business Case Approach Paper, by ManTech International Corporation, 1994.
- [25] "Hughes Aircraft Company Presentation," by Ron Newman, InterCAP Graphics Users Association Conference, September 11-14, 1994.
- [26] Technical Manual, Organizational Maintenance for Loader, Scoop Type, DED 4x4, Articulated Frame Steer, 2-1/2 Cubic Yard, TM 5-3805-262-20, Sept. 1987.
- [27] Testing Tools and Test Strategies For IETMs, Unisys Report SD210-1344, September 30, 1993
- [28] Quality Assurance Program: Interactive Electronic Technical Manuals and Associated Technical Information, Requirements For, Military Specification (MIL-Q-87270), November 20, 1992.
- [29] Data Base, Revisable: Interactive Electronic Technical Manuals, For The Support Of, Military Specification (MIL-D-87269), November 20, 1992.

- [30] Testing Tools and Test Strategies For IETMs, Unisys Report SD210-1344, September 30, 1993.
- [31] Manuals, Interactive Electronic Technical: General Content, Style, Format, and User-Interaction Requirements, Military Specification (MIL-M-87268), November 20, 1992.
- [32] CALS on the Deckplate! A Proven Method for Digital Logistics from Cradle to Grave, by Gordon Schaffer, NAVSEA/NAVSUP International Logistics Symposium, 1994.
- [33] Digital Data Types for Electronic Technical Manuals, by ManTech International Corporation, June 1994.
- [34] Interactive Electronic Technical Manuals, Cost /Benefit Analysis, by Hughes Aircraft Company, July 1992.
- [35] "Joint Integrated Maintenance Information System Presentation", by Ken Hamm and Karen Like, InterCAP Graphics Users Association Conference, September 11-14, 1994.
- [36] "Software Quality and Testing: What DoD Can Learn from Commercial Practices", by Lieutenant Colonel Mark R. Kindl, U.S. Army, CrossTalk - The Software Engineering Report, December 30, 1993.
- [37] Military Standard Defense System Software Development, Military Standard (DoD-STD-2167A)
- [38] Software Development and Documentation, Draft Military Standard (MIL-STD-498), December 22, 1992.
- [39] "Electronic Delivery - What Are The Implementation Issues For Corporate Applications?", in The Gilbane Report on Open Information & Document Systems, September 1993.
- [40] MCAIR Data Development for F/A-18 IMIS, Report MDC 92B0107, by McDonnell Douglas Government Aerospace - East, November 1, 1992.
- [41] Final Software Test Tools Report for the OSD CALS IWSDB Project, by ManTech International Corporation, 1994.
- [42] "It's Time for ISO 9000", by Mark Dawood, in CrossTalk, The Defense Software Engineering Report, March 1994.
- [43] "DoD Policy on the Future of MILSPEC" by William J. Perry, in CrossTalk, The Journal of Defense Software Engineering, September 1994, Volume 7 Issue 9.

- [44] DoD Software Test and evaluation Project (STEP) reports [25, Vol.3].
- [45] Integrated Weapon System Database (IWSDB) Implementation Strategy Paper, by ManTech International Corporation, 1994.
- [46] Applying CALS to the Creation, Management, and Use of Technical Manuals, by CALS Resource and Implementation Cooperative, for Navy CALS Acquisition/Implementation Group June 30, 1993.
- [47] The CGM Handbook, by Lofton R. Henderson and Anne M. Mumford, 1993 Academic Press Limited.

**APPENDIX A: VENDOR CONTACT REPORT**

**APPENDIX B: ELECTRONIC TECHNICAL MANUAL CONVERSION PACKAGE  
SURVEY MATRIX**

## **APPENDIX C: SAMPLING OF COMPLETED SURVEYS**

**APPENDIX D: IETM RESEARCH LIBRARY DOCUMENTATION LISTING**

**APPENDIX E: DoD ETM SYSTEMS AND PROGRAMS**

## **APPENDIX F: ABBREVIATIONS AND ACRONYMS**

