

**DTIC FILE COPY**

Document No. 1810-001  
23 January 1990

2

AD-A228 475

**Technical Report: DTD Creation  
for the  
Software Technology for Adaptable, Reliable Systems  
(STARS) Program**

**Contract No. F19628-88-D-0032**

**Task IR65: SGML Document Descriptions**

**CDRL Sequence No. 1810**

**23 January 1990**

**DTIC  
ELECTE  
NOV 09 1990  
S B D**

**Prepared for:**

**Electronic Systems Division  
Air Force Systems Command, USAF  
Hanscom AFB, MA 01731-5000**

**Prepared by:**

**IBM Federal Sector Division  
800 North Frederick Avenue  
Gaithersburg, MD 20879**

**DISTRIBUTION STATEMENT A**

**Approved for public release;  
Distribution Unlimited**

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE January 23, 1990	3. REPORT TYPE AND DATES COVERED Final		
4. TITLE AND SUBTITLE Technical Report: DTD Creation			5. FUNDING NUMBERS C: F19628-88-D-0032	
6. AUTHOR(S) S. Kutoroff				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) IBM Federal Sector Division 800 N. Frederick Avenue Gaithersburg, MD 20879			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Electronic Systems Division Air Force Systems Command, USAF Hanscom AFB, MA 01731-0000			10. SPONSORING/MONITORING AGENCY REPORT NUMBER  CDRL Sequence No. 1810	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) SGML (Standard Generalized Markup Language) is an international standard for representing the elements and structure of electronically stored text. SGML uses Document Type Definitions (DTD's) to unify the structure of various kinds of documents.  This document introduces SGML and examines the DTD's used in the STARS Program.				
14. SUBJECT TERMS  STARS, SGML, Standard Generalized Markup Language			15. NUMBER OF PAGES 18	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

## TABLE OF CONTENTS

DEFINITIONS	1
1. Introduction	1
2. Document Markup History	2
3. SGML Processing	5
4. DTD Analysis	8
5. DTD Creation	11
6. References	14
A. APPENDIX: REPORT DTD	15



Accession For	
NTIS SPA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By <i>per letter</i>	
Distribution	
Availability Codes	
Dist	Special
<i>A-1</i>	

## DEFINITIONS

- AFTOMS : Air Force Technical Order Management System
- attlist : additional information attached to an SGML element
- CALS : Computer-aided Acquisition and Logistic Support
- CD-ROM : Compact Disc Read Only Memory
- DoD : Department of Defense
- DSSSL : Document Style Semantics Specification Language, DP 10179
- DTD : Document Type Definition - the specification of SGML document structure
- element : a structural component of an SGML document
- entity : text substitution facility in SGML documents
- ISO : International Organization for Standards
- MIL-M-28001 : Military Specification, Markup Requirements and Generic Style Specification for Electronic Printed Output and Exchange of Text
- MIL-M-38784 : Military Specification, Technical Manuals: General Style and Format Requirements
- ODA : Office Document Architecture, ISO 8613
- Postscript : a page description language developed by Adobe Systems
- SGML : Standard Generalized Markup Language, ISO 8879:1986 a standard language used to describe the elements and structure of a document by specifying the relationships between its elements.
- SPDL : Standard Page Description Language, DP 10180
- STARS : Software Technology for Adaptable Reliable Systems

## 1. Introduction

Within the Software Technology for Adaptable Reliable Systems (STARS) there exists a need to exchange source code and documents between the prime contractors. Even within this program there are multiple hardware vendors, operating systems, and applications software. The problems experienced by the DoD in exchanging revisable documents are apparent within STARS. Part of the problem has been addressed by the selection of the Standard Generalized Markup Language (SGML, ISO 8879:1986) as the standard for document exchange.

SGML defines a standard means to specify the organization and relationships between the elements of structured documents. SGML is intended to be used in the preparation of documents using descriptive markup to denote document elements. The Document Type Definition (DTD) defines the elements of a document and specifies the relationship of each element to other elements.

The following section on the history of electronic document production explains some of the reasons why descriptive markup is important and why SGML is the appropriate tool for use in the STARS program for document interchange. The remainder of this document introduces the DTDs as used in the STARS program as well as those used in other DoD programs. The features and rationale for the DTDs are discussed and the foundation for a new STARS DTD is presented. The proposed STARS DTD will be delivered in CDRL 1820.

## 2. Document Markup History

Markup refers to the non-content portion of a document. All blank spaces, including margins, word spacing, and line spacing is considered markup. There are examples of text presentation entirely without markup in stone tablets from antiquity and in early paper writing. Content without markup is very difficult to read, as shown below (the examples used below are from [COOMBS, p. 936]):

miltonexpressesthisideamostclearlylaterinthetractcannotpraiseafugitiveand  
cloisteredvirtueunexercisedandunbreathedthatneversalliesoutandseesh  
adversarybutslinksoutoftheracewherethatimmortalgarlandisto berunfor  
not withoutdustandheat similarlywordsworth

Markup is added to a document to improve its readability.

A final copy document has the property of presentational markup where the letter, word, and line spacing is imbedded to make the document pleasing to the eye and enhance its ability to communicate. Placing presentational markup in a document for electronic storage makes the document difficult to alter but easy to retrieve for distribution. To add a sentence to a presentation ready document means that the word and line spacing of the entire document may need revision; however, printing additional copies of a presentation ready document simply requires sending the document to a hardcopy device. Presentation ready documents are also tied to a particular output device and form factor. The previous content is shown below in presentational form:

Milton expresses this idea most clearly later in the tract:

I cannot praise a fugitive and cloistered virtue unexercised and  
unbreathed, that never sallies out and sees her adversary, but  
slinks out of the race where that immortal garland is to be run  
for, not without dust and heat.

Similarly, Wordsworth...

In order to facilitate document preparation, word processors were developed to store documents in a format peculiar to the program's design. Editing the document became easy since word processors adjust all spacing and page formatting to accommodate new or changed text. The file is printed by the word processor itself. Each word processing vendor has a unique file format peculiar to the needs and capabilities of its own software.

Transferring electronic copies of documents becomes more difficult because conversion from one file type to another is required if the sender and the recipient use heterogenous equipment. In the microcomputer world alone, Microsoft Word supports 8 file formats, Word Perfect supports 6, and Claris MacWrite II supports 20. The proliferation becomes more acute when time is factored into the problem. A document delivered electronically may become unreadable, not only because no software conversion tool is available, but also because the media upon which it is stored becomes obsolete, e.g. eight inch CP/M floppy disks from the early 1980's.

The cost of re-keying documents and long term storage is nowhere more significant than to

the United States Department of Defense. Technical manuals for military devices require constant updating and must be available at the depot level for support. To support the weapon systems on a single 10,000 ton Navy cruiser requires 26 tons of paper manuals [Helgerson]. Such documents may be stored and distributed in digital form, for example using CD-ROM, which is compact, light weight, and economical to distribute. Such storage is only possible when a commonly accepted format exists. The answer to many of the problems of document interchange is an accepted standard, such as SGML.

Besides application specific word processors, there are markup systems in which "formatting" information is imbedded in the content. Runoff, TEX, nroff, and SGML are examples of such markup systems. Of these kinds of markup systems, SGML uses descriptive markup and the remainder use procedural markup. Descriptive markup means that the inserted markup commands describe the content to which they apply; procedural markup means that the inserted markup tells the processor what action to perform on the content. The previous text is shown with both a procedural and a descriptive markup notation below:

PROCEDURAL:

Milton expresses this idea most clearly later in the tract:

```
.sk 3 a;.in +10 -10;.ls 0;.cp 2
I cannot praise a fugitive and cloistered virtue unexercised and unbreathed,
that never sallies out and sees her adversary, but slinks out of the race
where that immortal garland is to be run for, not without dust and heat.
.sk 3 a;.in -10 +10;.cp 2;.ls 1
Similarly, Wordsworth...
```

DESCRIPTIVE:

Milton expresses this idea most clearly later in the tract:

```
<lq>
I cannot praise a fugitive and cloistered virtue unexercised and unbreathed,
that never sallies out and sees her adversary, but slinks out of the race
where that immortal garland is to be run for, not without dust and heat.</lq>
Similarly, Wordsworth...
```

The procedural markup example shows inserted commands to skip lines, indent, and change letter spacing for the long quotation. These instructions tell the formatter what to do as they are encountered; however, such instructions are incapable of taking advantage of the features on different output devices. Additionally, to change the format for all long quotes in a document requires that each set of long quote instruction sequences be altered. For this reason, word processors adapted the style sheet to simplify the process of altering the printed style of a document without changing markup.

The advantage of the descriptive markup shown above is that the markup of a source document becomes independent of the output device, whereas in procedural markup it is tied to the form factor and/or the precise output device in use. In the above example, the format instructions for element tagged as <lq> (to denote a long quotation) might be to use indentation and vertical spacing for a simple printer or italics and open/close quotes with a letter quality printer or to change the font and style on a laser printer or a typographic quality printer. Altering descriptive markup may be done without regard for output specific issues such as page breaks, spacing, and character set metrics. A document with descriptive markup may be output to a

variety of devices ranging from high quality typography printers through laser printers, character printers, and Braille printers with changes only to the format association between the descriptive elements and the printing instructions required on the device.

Another element of document processing is revision control and version control. SGML contains facilities for text substitution, file inclusion, exclusion of content based on switches, and inclusion of content based on switches. The result is that a single source document can be prepared which may be processed in different forms as required for many specific needs with the ability to maintain a single source version of many similar documents. Changes can be made to a document while leaving the old content disabled but still in place for reference.

SGML descriptive markup also lends itself to database applications. An SGML tagged source document can be scanned for all occurrences of a particular tag and that data extracted for use in other applications. Technical manuals which have been tagged to the DTD in MIL-M-38784C will have all part numbers and manufacturer identification so tagged for such extraction. The reverse process can also be done with SGML, a database file can be prepared with all data descriptively tagged using SGML and the file can be processed to produce a report using SGML tools. The STARS Catalog prepared on the IBM Repository is such an application of SGML developed as STARS CDRL 0570 in the Q-increment.

An additional advantage of SGML is that it is a recognized and accepted international standard. A document described using SGML may be interchanged freely between heterogeneous computer systems without loss of content. SGML also has provision for non-SGML material, such as graphics, to be inserted in a document. For these reasons the Computer-Aided Acquisition and Logistics Support (CALs) initiative has adopted SGML for its document interchange needs. SGML will allow multiple vendors on a single procurement to prepare documents and exchange them without regard to the computer equipment used and with minimal expense.

Vendors have developed software for a variety of host platforms which processes documents prepared to the SGML standard. Some products are used to facilitate document markup by verifying the document content and placement of tags as the author works. Such smart editors can offer a certain degree of What-You-See-Is-What-You-Get (WYSIWYG) display to SGML document preparation. Computers such as the IBM PC and the Apple Macintosh are the typical host for this software. Other products are available for conversion of existing paper and electronic documents into SGML markup form. Another class of products, such as IBM's TextWrite, handle the entire gamut of production features needed to prepare technical manuals to SGML based military standards such as MIL-M-28001A, Markup Requirements and Generic Style Specification for Electronic Printed Output and Exchange of Text.

### 3. SGML Processing

SGML document processing involves two basic operations:

1. Selecting the document structure and organizing its contents. This is the process of preparing a "document type definition";
2. "Tagging" the text with descriptive markup tags to identify the document's structure.

A pair of tags, composed of a start-tag and an end-tag, identifies and delimits each element of a document. Begin and end tags are not always required; SGML includes markup minimization rules to simplify the process of preparing a source document instance. Begin or end tags can be defined to be optional and implied by context. SGML also supports a feature called SHORTREF which permits a character to represent a tag, for example a quotation mark can imply the tag <lq>. A tag consists of a tag name that refers to the element itself and three special characters used to set it off from the text proper:

```
START-TAG OPEN  <
END-TAG OPEN    </
TAG CLOSE      >
```

A long quotation might be denoted as shown below:

```
<lq>I cannot praise a fugitive and cloistered virtue unexercised and
unbreathed, that never sallies out and sees her adversary, but slinks
out of the race where that immortal garland is to be run for, not
without dust and heat.</lq>
```

```
START-TAG OPEN:  <
TAGNAME:         lq
TAG CLOSE:       >
TEXT:            I cannot praise a fugitive and cloistered virtue ...
END-TAG OPEN:    </
TAGNAME:         lq
TAG CLOSE:       >
```

Tags must appear directly before and after the element to which they refer. They may be placed on the same line as the text, or they may be on separate lines. The following variations are processed in the same way:

```
<lq>I cannot praise a fugitive and cloistered virtue unexercised and
unbreathed, that never sallies out and sees her adversary, but slinks
out of the race where that immortal garland is to be run for, not
without dust and heat.</lq>
```

-or-

<lq>

I cannot praise a fugitive and cloistered virtue unexercised  
and unbreathed, that never sallies out and sees her adversary,  
but slinks out of the race where that immortal garland is to  
be run for, not without dust and heat.

</lq>

It must be pointed out that the tags themselves are not defined by SGML. SGML is the language and syntax in which the structure of the document is declared and through which the document is processed. Users of SGML prepare a document type definition (DTD) to describe the document and therefore the descriptive tags used in the document markup. A single DTD may, of course, define one document or millions of unique but structurally related documents.

The SGML standard defines the syntax and semantics of all SGML documents. It does not define the output of an SGML processor nor does it specify how documents prepared in SGML are to be formatted. These issues are in the scope of future standards which are intended to permit vendors freedom of implementation.

The output formatting for SGML will be handled by the Document Style Semantics Specification Language (DSSSL, ISO DP 10179). The DSSSL draft standard was released for public comment in September of 1989. It is currently undergoing revision based on comments from the standards community. The draft DSSSL standard was released in an incomplete form, lacking a syntax for the semantic entities required to specify a document. There is also a movement in the standards community to require DSSSL to support the Office Document Architecture (ODA, ISO 8613) standard.

DSSSL must satisfy a wide range of user requirements. The most difficult problem with DSSSL is in its approach. Due to its history within the standards organizations DSSSL cannot define a programming language, it may only define a specification language. DSSSL must allow for specification of every detail of document production including: widows, orphans, multiple fonts, table of contents, index, figures, tables, etc. Yet DSSSL is not limited to the production of books and reports, it must support the needs of the entire graphic arts industry. The DSSSL project editor, Ms. Sharon Adler of IBM, has stated that DSSSL will support any kind of printing, including milk cartons.

The output from a DSSSL process will be in the Standard Page Description Language (SPDL, ISO DP 10180). The SPDL draft standard was released for public comment in September of 1989. It too, is currently undergoing revision. SPDL was co-edited by Dr. Steve Strassen, representing Xerox, and Matthew Foley, representing Adobe Systems. SPDL in its initial form had little in common with Adobe's Postscript, a page description language already widely used by laser printers. Due to changes in the industry shortly after release of the draft proposal for SPDL there is now support from Adobe Systems to make SPDL compatible with Postscript.

The three standards SGML, DSSSL, and SPDL are international standards and as such they must be written with no bias to any specific human language. The multi-lingual requirement complicates the standards and the process by they are written. Existing and yet to be developed font information interchange standards will play important roles in SGML, DSSSL, and SPDL.

At this time, SGML users define an output specification loosely based on DSSSL to define



#### 4. DTD Analysis

An SGML document can be interchanged between SGML processors because the DTD provides the document structure and is interchanged with the source document. The STARS prime contractors have independently selected how the SGML documentation policy is to be addressed in the STARS program. To accommodate this diversity the primes exchange the following materials with each document delivery:

1. the DTD used for the document,
2. the SGML source document instance prepared to the above DTD,
3. Postscript output from the document format process, and
4. a version of the document formatted with only ASCII carriage control

The preparation and acceptance of a standard DTD will simplify the document delivery process.

Within the STARS program there are two major DTDs in use, both provided by the IBM Team. The REPORT DTD was prepared in 1988 by Science Applications International Corporation under STARS Foundation contract N00014-87-C-2386. The REPORT DTD is listed in Appendix A. The GDOC DTD was provided with the IBM mainframe based SGML translator facility (DCF) and it has been used for their Q and R increment reports. GDOC is not listed in this report due to copyright restrictions.

Within the DoD there are organizations which recognize the need for an electronic publishing standard such as SGML. Principle among them is the Computer-Aided Acquisition and Logistics Support (CALs) initiative. CALs has prepared a DTD published in MIL-M-28001A which is intended to be used for the publication of technical manuals. The REPORT DTD was designed as a compromise between the older CALs 28001 DTD and the known limitations of the STARS Foundation Interim SGML system.

Separately, the Air Force Technical Order Management System (AFTOMS) has published the DTD for technical manuals as MIL-M-38784C. The DTDs for 28001A and 38784C implement the same basic document structure and differ primarily in organization. Both of these standards are in review and subject to change.

The following table lists, for comparison, a selection of the tags used in the subject DTDs (see references):

Report	GDOC	28001A	38784C	Note
report	gdoc	doc	doc	
titlepg	titlep	titleblk		
	abstract	purpose	purpose	
		forward	forward	
	preface	preface	preface	
contents	toc	contents	contents	
iluslist	figlist	iluslist	iluslist	
front	frontm	front	front	
bodym	body	body	body	(1)
	tlist	tablist	tablist	
section	h1	chapter	chapter	(2)
para	p	paratext	paratext	(3)
head	h2, h3, h4, h5, h6	para0	para0	(3)

seqlist	ol	seqlist	seqlist
bullet	ul	randlist	randlist
item	li	item	item
deflist	gl	deflist	deflist
term	gt	term	term
def	gd	def	def
rear	backm	rear	rear
appendix	appendix	appendix	appendix

- (1) The Report DTD uses <bodym> instead of <body> as a consequence of a formatter limitation.
- (2) 28001A and 38784C both also support a <section> tag. These DTDs also have a number of other options for large scale document structure.
- (3) 28001A and 38784C have a more complex structuring at this level than indicated in the chart.

Neither REPORT, GDOC, MIL-M-28001A, or MIL-M-38784C supports the kind of documents required of the STARS program. REPORT is simplistic and omits many features required in the STARS program. The REPORT DTD is supplied with the STARS SGML Text Composition System (STCS) which was delivered in November of 1988. STCS was developed using a reusable parser, it is slow and does not implement the complete SGML standard. Users of the REPORT DTD are forced to use text figures for tables and as the only means of inserting any kind of a diagram. The REPORT DTD also uses a clumsy approach to subheadings. In its favor, the REPORT DTD is simple enough to be used and understood by persons with limited SGML experience and has some tag commonality with 28001A. The REPORT DTD is used by Science Applications International Corporation as subcontractor to IBM.

The GDOC DTD is only used by IBM for their deliverables. It offers a very comprehensive set of tags, but these tags are unique to a DTD which is copyright by IBM. Documents prepared using GDOC and the IBM software have a professional appearance when printed on a Postscript(tm) printer. Engineers at IBM have an efficient arrangement for the transfer of files between the IBM mainframe and their personal computers used for document editing and printing.

Unisys has used the REPORT DTD with Author/Editor software from SoftQuad. Author/Editor prepares SGML source instance documents to a given DTD by allowing only valid SGML constructs at all times within a document. Author/Editor runs on the Apple Macintosh computer; however, similar products such as TextWrite from IBM and WriterStation from Datalogics are available for PC compatible computers.

Boeing has expressed interest in another option for producing SGML tagged documents. Documents can be converted to SGML markup using a product such as FasTAG from the Avalanche Development Company of Boulder, Colorado. This class of products is used to reverse engineer SGML markup from final copy documents prepared using other word processing systems. This option may be attractive given the cost, availability, and complexity of SGML software. Microcomputer hosted SGML editors such as Author/Editor and WriterStation cost \$1000 each and generally require the purchase of additional software to support multiple DTDs and to produce fully formatted output.

The STARS DTD can be developed such that it is compatible with 38784C and 28001A for

all but a limited number of tags unique to STARS. For example, a tag to enclose Ada source code within a technical report would be appropriate in the STARS DTD as would a tag to enclose text from a terminal screen dump.

Adaptation of MIL-M-38784C or similar DTDs is not without problems. These standards are intended to substitute SGML based electronic publication for the existing print oriented standard. Emphasis is placed on developing markup which supports matching the required appearance of a printed document, therefore the markup is not fully descriptive of content.

Many elements of MIL-M-38784C are intended to support paragraph numbering and similar requirements for printing technical manuals. These markup requirements complicate the authoring process to the point that SGML context sensitive editors are required to reduce the error rate of manual tagging. Such complexity may discourage document markup by engineers and scientists.

Finally, any DTD prepared for STARS users will only support document creation and not document production since those standards are still in development. An output specification will be needed for each software system used to produce printed documents. This may slow the acceptance of a common DTD.

## 5. DTD Creation

Development of a DTD proceeds from analysis of document content to definition of SGML elements, entities, and attributes. The document structure analysis looks for common elements and structural elements which should be descriptively tagged. Paragraphs, quotations, definitions, footnotes, titles, figures, and chapters are all structural elements that might be identified for tagging.

The analysis of a document may be very detailed, for example it may be useful in certain circumstances to tag sentences. In other applications, say a memo or letter, the primary structural elements would involve identification of the sender and recipient rather than the content of the letter or memo.

The following is a fragment of the 839 total lines which make up the DTD for MIL-M-38784C. The fragment details the structure of paragraphs and their related elements, entities, and attributes:

```
<!ENTITY % bodyatt
    "id ID #REQUIRED
    inschlvl NUTOKEN #IMPLIED
    delchlvl NUTOKEN #IMPLIED
    label NMTOKEN #IMPLIED
    texttype NUMBER #IMPLIED
    itemid NMTOKEN #IMPLIED
    config NUTOKENS #IMPLIED
    skilltrk CDATA #IMPLIED
    hcp %yesorno; '0'
    esds %yesorno; '0'
    xrefid IDREF #IMPLIED
    sssn NMTOKEN #IMPLIED
    unit NMTOKEN #IMPLIED
    module NMTOKEN #IMPLIED
    lru NMTOKEN #IMPLIED
    assem NMTOKEN #IMPLIED
    subassem NMTOKEN #IMPLIED
    ssubassm NMTOKEN #IMPLIED
    compon NMTOKEN #IMPLIED
    partno NMTOKEN #IMPLIED
    annum NMTOKEN #IMPLIED
    exrefid IDREF #IMPLIED
    xreftype NMTOKEN #IMPLIED" >

<!ENTITY % stepatt
    "id ID #REQUIRED
    inschlvl NUTOKEN #IMPLIED
    delchlvl NUTOKEN #IMPLIED
    label NMTOKEN #IMPLIED
    texttype NUMBER #IMPLIED
    itemid NMTOKEN #IMPLIED
    hcp %yesorno; '0'
    esds %yesorno; '0'
    xrefid IDREF #IMPLIED
```

```
exrefid IDREF #IMPLIED
xreftype NMTOKEN #IMPLIED" >

<!ENTITY % asyntxt "(stemg | endemg | %change; | ftnref | indxflag | stemph
| endemph | %tracks;)" >

<!ENTITY % list "(seqlist | randlist | deflist | cfplist)" >

<!ENTITY % nums "(partno | partdesc | smrcode | nsn | modelno | ssn | refdes
| serno | docno | figindex | lin)" >

<!ENTITY % secur
"security (uc | c | s | ts) #IMPLIED
restrict NMTOKENS #IMPLIED
release NMTOKENS #IMPLIED
codeword NMTOKENS #IMPLIED
scilevel %yesorno; '0'
diglyph NMTOKENS #IMPLIED" >

<!ENTITY % spcpara "((warning | bpwarn)?, (caution | bpcaut)?,
(note | bnote?)" >

<!ENTITY % text "((#PCDATA | %asyntxt; | %nums | tool | testeq | material |
torquevl | xref | graphic | subscript | superscript)*, ftnote*)"
>

<!ENTITY % fig "(graphic+, legend?, title?)" >

<!ENTITY % chart "(title, (tabdef | stdtable), thead?, tbody)" >

<!ENTITY % table "(title, (tabdef | stdtable), thead?, tbody)" >

<!ENTITY % tab "(colhdddef*, colbddef*)" >

<!ENTITY % steptxt "((%tracks;)*, %spcpara;, paratext, (%list; | paratext)*,
note?)" >

<!ENTITY % steps "((%steptxt;, (step2, step2+)?), (%steptxt;, (step2, step2+)?))" >

<!ELEMENT para0 - o (title, %spcpara;, paratext, (%list; | paratext)*, note?,
(step1, step1+)?, (subpara1, subpara1+)?) +(figure | chart | table);
<!ATTLIST para0 %bodyatt;
%secur; >

<!ELEMENT subpara1 - o (title, %spcpara;, paratext, (%list; | paratext)*,
note?, (step1, step1+)?, (subpara2, subpara2+)?) >
<!ATTLIST subpara1 %bodyatt;
%secur; >

<!ELEMENT subpara2 - o (title, %spcpara;, paratext, (%list; | paratext)*,
note?, (step1, step1+)?, (subpara3, subpara3+)?) >
<!ATTLIST subpara2 %bodyatt;
```

```
        %secur; >

<!ELEMENT subpara3 - o (title, %spcpara;, paratext, (%list; | paratext)*,
                        note?, (step1,step1+)?) >
<!ATTLIST subpara3 %bodyatt;
            %secur; >

<!ELEMENT step1 - o (%steps;) >
<!ATTLIST step1 %stepatt;
            %secur; >

<!ELEMENT step2 - o (%steptxt;, (step3, step3+)?) >
<!ATTLIST step2 %stepatt;
            %secur; >

<!ELEMENT step3 - o (%steptxt;) >
<!ATTLIST step3 %stepatt;
            %secur; >
```

Given the rationale in the previous section, the STARS DTD will take its element names and structure from MIL-M-38784C. However, the STARS DTD will eliminate elements not needed in STARS publications, such as the warnings, cautions, and part numbers needed for hardware technical manuals. The changes will reduce complexity of the DTD and make it easier to comprehend. Entity definitions in MIL-M-28001A and MIL-M-38784C will be copied only when relevant to STARS reports.

The STARS DTD is currently in development. The complexity of MIL-M-38784C requires that a conforming SGML parser be used to validate changes to the DTD and be available to verify documents prepared and delivered to the DTD. Such a parser has been ordered and will enable publication and review of the proposed STARS DTD by all the STARS prime contractors. The proposed STARS DTD will be delivered in CDRL 1820.

## 6. References

Coombs, J.H., Renear, A.H., DeRose, S.J. "Markup Systems and the Future of Scholarly Text Processing", Communications of the Association for Computing Machinery, Volume 30 Number 11, November 1987, pp 933- 947.

Helgerson, L.W. "Disseminating Digital Data: Why CD-ROM?", CD-ROM End User, Volume 1, Number 8, December 1989, pp 40-43.

International Business Machines. GDOC.DTD (c), 1988. As made available on the IBM Team STARS Repository computer.

International Organization for Standards. Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML), ISO 8879:1986, Switzerland.

International Organization for Standards. Information Processing - Text Communication - Document Style Semantics Specification Language (DSSSL), ISO/IEC DP 10179:1989.

International Organization for Standards. Information Processing - Text Communication - Standard Page Description Language (SPDL), ISO/IEC DP 10180:1989.

International Organization for Standards. Information Processing - Text and Office Systems - Operational Model for Text Descriptions and Processing Languages, ISO/IEC JTC1/SC18/WG9 N807, 17 March 1989.

Science Applications International Corporation. REPORT.DTD, prepared under contract N00014-87-C-2386 to the Naval Research Laboratories for the STARS Foundation Project, 1988.

MIL-M-28001A (DRAFT COPY), as provided for review by Electronic Publishing Committee, Computer-Aided Acquisition and Logistics Support (CALs).

MIL-M-38784C Review material, as provided by VITRO Corporation.

STARS CDRL 0570 (and related materials), Ada Foundations, 17 March 1989.

## APPENDIX A.

### APPENDIX: REPORT DTD

The DTD for the REPORT is shown below.

```
<!DOCTYPE report [  
  
<!ENTITY %misc      "bullet | seqlist | chart | figure | graphic | note" >  
<!ENTITY %subbody   "head | para | %misc; " >  
<!ENTITY %text      "( #PCDATA | indxflag )*" >  
  
<!ELEMENT report    - - ( front? , bodym , rear? ) >  
<!ELEMENT front     - - ( titlepg , contents? , iluslist? , deflist? ) show ( ) >  
<!ELEMENT titlepg   - - ( title | docno | date |  
                        reldate | author | address )* >  
  
<!ELEMENT indxflag  - - ( #PCDATA ) >  
<!ELEMENT title     - 0 ( %text; ) >  
<!ELEMENT docno     - 0 ( %text; ) >  
<!ELEMENT date      - 0 EMPTY >  
<!ELEMENT reldate   - 0 ( #PCDATA ) >  
<!ELEMENT author    - 0 ( %text; ) >  
<!ELEMENT address   - 0 ( %text; ) >  
<!ELEMENT contents  - 0 EMPTY >  
<!ELEMENT iluslist  - 0 EMPTY >  
<!ELEMENT deflist   - - ( term , def )* >  
<!ELEMENT term      - 0 ( %text; ) >  
<!ELEMENT def       - 0 ( %text; ) >  
<!ELEMENT bodym     - - ( section | %subbody; )* >  
<!ELEMENT section   - 0 ( sectitle? , ( %subbody; )* ) >  
<!ELEMENT sectitle  - - ( %text; ) >  
<!ELEMENT head      - - ( %text; , ( %subbody; )* ) >  
<!ELEMENT para      - 0 ( %text; | %misc; )* >  
<!ELEMENT bullet    - - ( item+ ) >  
<!ELEMENT item      - 0 ( %text; ) >  
<!ELEMENT seqlist   - - ( item+ ) >  
<!ELEMENT chart     - - RCDATA >  
<!ELEMENT figure    - - RCDATA >  
<!ELEMENT graphic   - - CDATA >  
<!ELEMENT note      - 0 ( %text; ) >  
<!ELEMENT rear      - - ( appendix* , deflist? , index? ) >  
<!ELEMENT appendix  - - ( apdxtitl , ( %subbody; )* ) >  
<!ELEMENT apdxtitl  - - ( %text; ) >  
<!ELEMENT index     - 0 EMPTY >  
]>
```