Maritime Interactive Electronic Technical Manuals

D:23 RECOMMENDED DTD AND STYLE VIEW

Version 2.1, 06. December 1996.

NR Norsk Regnesentral
ANVENDT DATAFORSKNING
Norwegian Computing Center/Applied Research and Development

RAPPORT/REPORT

Report no :912

F1: IT-Standards is a part of the Information Technology for Ship Operation Programme funded by the Resarch Council of Norway

Norwegian Computing Center:
Tryggve E. Sørensen

Oslo
December 1996

NR Norsk Regnesentral
ANVENDT DATAFORSKNING

Rapport/Report

Tittel/Title: Maritime Interactive Electronic Technical Manuals,
D:23 RECOMMENDED DTD AND STYLE VIEW

Version 2.1, 06. December 1996.

Dato/Date: December
År/Year: 1996
ISBN: 82-539-0412-6
Publikasjonsnr: 912
Publication no:

Forfatter/Author: Norwegian Computing Center:
Tryggve E. Sørensen

Approved by: Alf-Einar Løberg, Kongsberg Norcontrol Training AS

Sammendrag/Abstract:

This report contains a Document Type Definition (DTD) that allows SGML encoding of technical manuals for ships equipment. It also contains style views for DynaText (Electronic Book Technologies, Inc.) which will give a presentation of documentation encoded according to this DTD. The DTD has been developed by NR in cooperation with Kongsberg Norcontrol Training AS, which has developed the DynaText style views.

This work is carried out as part of the Information Technology for Ship Operation Programme (MITD) which is a research programme funded by The Research Council of Norway. This deliverable (D23) is prepared by Norwegian Computing Center/ Tryggve Sørensen for Kongsberg NorControl Training AS and DNV Research AS as part of the F1:IT-Standards project.

A diskette supplements this written report containing electronic versions of the Document Type Definition and Style View.

The report together with accompanying diskette is freely available, and can be obtained by contacting either Stian Ruud at DNV Research AS or Alf-Einar Løberg at Kongsberg Norcontrol Training AS.

Emneord:
Indexing terms:
SGML, Interactive Documentation, DTD, CALS, Standard
Målgruppe/Target group:

Tilgjengelighet/Availability: Open

Prosjektdata/Project data:

Prosjektnr/Project no: 06 50 02

Antall sider/No of pages: 74

Satsningsfelt:

Research field: Electronic Publishing, Standards, SGML

Norsk Regnesentral/Norwegian Computing Center
Gaustadalléen 23, Postboks 114 Blindern, N-0314 Oslo, Norway
Telefon 22 85 25 00, telefax 22 69 76 60

0 Introduction

This report contains a Document Type Definition (DTD) that allows SGML encoding of technical manuals for ships equipment. It also contains style views for DynaText (Electronic Book Technologies, Inc.) which will give a presentation of documentation encoded according to this DTD. The DTD has been developed by NR in cooperation with Kongsberg Norcontrol Training AS, which has also developed the DynaText style views.

This work is carried out as part of the Information Technology for Ship Operation Programme (MITD) which is a research programme funded by The Research Council of Norway. This deliverable (D23) is prepared by Norwegian Computing Center/Tryggve Sørensen for Kongsberg NorControl Training AS and DNV Research AS as part of the F1:IT-Standards project.

A diskette supplements this written report containing electronic versions of the Document Type Definition and Style View.

The report together with accompanying diskette is freely available, and can be obtained by contacting either Stian Ruud at DNV Research AS or Alf-Einar Løberg at Kongsberg Norcontrol Training AS.

1 Scope

1.1 Specificity of the scope

This document recommends an SGML document type definition for the development of interactive electronic technical manuals (IETM) that accompanies ship equipment. Vendors of ship equipment write the documentation that is delivered with their products. This document proposes how they can structure and represent their documentation using the Standard Generalized Markup Language SGML – ISO 8879.

This document also provides recommendations for the presentation of interactive electronic technical manuals that have been encoded according to the recommended DTD. Presentation guidelines are based on usage of the electronic publishing tool DynaText from Electronic Book Technologies Inc. in the form of styleviews accompanying the SGML document type definitions included herein.

2 Conformance

An interchanged IETM is conformant if it can be correctly interpreted. This means that:

- parsed correctly by an SGML parser

2.1 Profiles

Profiles may be defined by the using application. A profile defines the use of standards and specifications given in this document and the standard and specifications referenced from this document. This will typically include (subsets of or entire) CALS specifications, subsets of formats specified in this document, international standards and de-facto standards.

2.2 Syntax

An MITD IETM shall be represented using SGML with the document type definition defined in this document and using the accompanying SGML declaration.

The style sheet included in (Appendix) this recommendation is **not** a mandatory part for documents conformant with a MITD IETM. Rather, the style sheet gives an example for how to take a SGML source document to the presentation using an SGML publishing tool.

2.3 Semantics

This recommended DTD provides the infrastructure to specify an encoded representation for expressing the logical structure and content of IETMs for use by suppliers of parts to maritime installations, in particular ships.

Structuring documents this way enables long term storage and retrieval of parts of the documentation in the future.

3 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this recommended MITD IETM DTD. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this recommended MITD IETM DTD are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards. The same applies to ITU-T Recommendations.

3.1 International Standards

ISO 8879:1986, Information Processing – Text and office systems – Standard Generalized Markup Language (SGML).

ISO/IEC 8632:1992 Information technology -- Computer graphics -- Metafile for the storage and transfer of picture description information (CGM).

ISO/IEC 10918:1994 Information technology -- Digital compression and coding of continuous-tone still images (JPEG).

CCITT/Group 3/4 ITU Recommendations T.4 and T.6.

3.2 Specifications

MIL-M-28001B, Military Specification Markup Requirements and Generic Style Specification for Electronic Printed Output And Exchange of Text.

MIL-D-87269, Military Specification, Data Base, Revisable: Interactive Electronic Technical Manuals, For the Support of, 20 November 1992.

CALS 38784C.

3.3 Proprietary standards

Drawing Interchange Format: DXF, Autodesk Inc.

Tagged Image File Format (TIFF). Developed by Aldus Corporation (now owned by Adobe) and Microsoft.

4 Definitions

For the purposes of this MITD IETM draft standard the following definitions apply

- Definitions as in ISO 8879:1986
- Definitions in this clause

4.1 Method of describing CDM Generic Layer template elements

Element names in UPPERCASE. (Example). These elements are not used in source documentation.

Element names in lowercase. Used in source documentation.

5 Document content structure

5.1 System component view

The documentation structure follows closely a system decomposition model where a system is recursively broken down into subsystems of finer and finer granularity. On each level of the system defined in the decomposition, the following information types will be adequate to describe all aspects of the item:

- Descriptive information, giving a general overview of the actual component. Security issues.
- Operational tasks describing daily operational procedures on the actual component.
- Maintenance information.
- Part list.

5.2 Documentation structure

In order to obey the structure defined in this recommended MITD DTD and outlined in Section 5.1 on page 7, the creation of manuals must account for this structuring method. Thus, it is important that the documentation team identifies which parts of the documentation should go into the categories defined above.

5.3 Conversion of existing material

There may be considerable work in adapting existing manuals to the documentation structure defined in this recommended MITD DTD. It may well be the case that it represent a big change in how the operators find their manuals compared to traditional paper-based manuals.

The conversion and adaptation process will mainly be performed through the following procedures:

- Re-structuring content to comply with system/sub-system decomposition model.
- Converting content from old representation into SGML format according this recommended MITD DTD.
- Converting/adapting non-textual information, like figures and pictures into formats complying with those of Section 6 on page 7.
- Inserting reference to non-SGML data into the SGML representation of documentation.

These activities will render a format of the content which is prepared for electronic publishing using the publishing tools of choice.

6 Inclusion of graphics

The following standards are recommended for representing graphic images in an MITD IETM:

6.1 Vector graphics

Vector graphics representations are useful for schematic illustrations where a high degree of accuracy is needed, and where details of the illustrations are maintained even when scaling the figures on a computer screen. This form of representation will render the best quality printout of illustrations.

- Computer Graphics Metafile: ISO/IEC 8632:1992 (CGM)
- Drawing Interchange Format: DXF, Proprietary Autodesk Inc. DXF is Autodesk’s format for allowing the transfer of AutoCad 3-D drawings between CAD and illustration applications.

6.2 Raster graphics

Raster graphics images are useful for the representation of photographic material or photostatic copies. These representations can be used to store pictures of components, special work procedures etc.

- CCITT/Group 3/4 ITU Recommendations T.4 and T.6.
- Tagged Image File Format (TIFF). Proprietary format developed by Aldus Corporation (now owned by Adobe) and Microsoft.
- Joint Photographic Expert Group: ISO/IEC 10918 (JPEG)

6.3 Other standards and formats

The following issues are subject to further development within the area of maritime IETM representation:

- 3D illustrations will probably become an important part of future documentation.
- STEP standards for exchange of product data information is not currently considered in this document. However, these standards may have relevance for future extensions.
- Audio/video standards bring another dimension to IETMs. This document does not specify any standards for representation of audio and video.

6.4 Entering graphical references

Graphic objects are represented using a non-SGML representation like the graphical standards given above. These objects are entered into the SGML documentation format using special constructs defined in this recommended MITD DTD, namely the *graphic* element.

7 MITD Document Type Definition

7.1 Introduction

The recommended MITD DTD consists of two parts as specified in MIL-D-87269, the Generic Layer and the Content Specific Layer. The Generic Layer Specification defines template element classes from which actual elements in the Content Specific Layer are derived. The Generic Layer follows that of MIL-D-87269 with the modifications as listed below. The Content Specific Layer holds all application specific elements, which in this recommended MITD DTD respect, are used for representing maritime interactive electronic technical manuals. The Generic Layer is a specialization of that in Appendix B MIL-D-87269.

This recommended MITD DTD Generic Layer has been modified from that in MIL-D-87269 in the following manner:

1. Logical structure is simplified as to closer resemble its scope of use.
2. Additional elements from MIL-STD-28001B base element set are included in this recommended MITD DTD.

The listing of elements is divided in two parts. The first part lists entity and element definitions of the Generic Layer. These elements shall not be used in source documentation. Rather, they work as templates or meta-elements for definition of Content Specific Layer elements.

In the second part, Content Specific Layer entities and elements are listed. These are used for tagging the source documents.

For a simplified view of the element content model structures, these have been visualized using the graphical symbolics of Near&Far (© Microstar Software Ltd.). The table below explains the meaning of the symbols found in the figures.

Symbols used in graphical representation of element content models.

Symbol	Symbol meaning
▬	Element.
⯈	Entity.
□	Element must occur exactly once.
▣	Zero or one instance of element is possible.
⧉	Zero or many instances of element is possible
◲	One or many instances of element is possible
⊏	Ordered: Elements must come in specified sequence.
≺	Selection: Exactly one of the elements in the selection is possible
⊰	Unordered: Elements can occur in arbitrary sequence.
○	Inclusion. Element can occur anywhere within content model.
⊘	Exclusion Element is not allowed in content model.

7.1.1 Presentation of Generic Layer entities

The Generic Layer entities are described by:

1. Entity name: Name of entity as referenced by DTD structures.
2. Entity expansion: The replacement text of an entity.

3. Purpose: The intended purpose for use of this entity.

7.1.2 Presentation of Generic Layer elements

Each element in the Generic Layer element section is presented with information as follows:

1. Element Name: If name is written in UPPERCASE font, it represents a template element only for use as a reference for the creation of content specific elements. Otherwise, the element name represents a real element that is applicable in source documents.
2. Content model: Logical content allowed for elements of this template.
3. Element use: Purpose of using this element.

7.1.2.1 Attribute lists associated with Generic Layer elements

The Generic Layer elements definitions are followed by attribute lists. These attribute lists are coded as entities in order for easy inclusion in Content Specific Layer element attribute lists.

Real element declarations with attributes associated, will have their attributes listed in attribute tables as specified below in Section 7.1.4.1.

7.1.3 Presentation of Content Specific Layer entities

The Content Specific Layer entities are described by:

1. Entity name: Name of entity as referenced by DTD structures.
2. Entity expansion: The replacement text of an entity.
3. Purpose: The intended purpose for use of this entity.

7.1.4 Presentation of Content Specific Layer elements

Each element in the Content Specific Layer is presented as follows:

1. Element Name: Name used in tagging associated information in source documents complying with this draft standard.
2. Element content model: Definition of logical content
3. Purpose: The intended purpose for use of this entity.

7.1.4.1 Attribute lists associated with Content Specific Layer elements

If attributes are defined with Content Specific Layer elements, these attributes are found in the attribute table following the element description table. The attribute table contains these entries:

1. Attribute Name stating the name of the attribute as represented in the document source.
2. Value Type identifying the allowed type of attribute values. The type can be a list of value or a reserved SGML type identifier.
3. Default value states the value taken by the attribute in case where it is not specified in the document source.
4. Purpose further explains the meaning of the attribute.

7.2 Content data model Generic Layer

7.2.1 Generic Layer entities

This section lists entities referenced in the Generic Layer. Additional entities are associated with template elements (see next section) supporting element attribute lists. These entities are listed together with their template elements since they are so closely related to them, and thus more naturally should appear there.

Entity Name	Entity Expansion
dietmdb-a	PUBLIC "-//USA-DOD//DTD Content Data Model Generic Layer//EN">
Purpose	
Defines the Content Data Model Generic Layer public identifier	

Entity Name	Entity Expansion
text	"text"
Purpose	
Referencing primitive elements in the Generic Layer, and within any Content Specific DTD. This is the definition of plain text, used in element definitions in the Content Specific DTD.	

Entity Name	Entity Expansion
table	"table"
Purpose	
Referencing primitive elements in the Generic Layer, and within any Content Specific DTD. This is the definition of table used in element definitions in the Content Specific DTD.	

Entity Name	Entity Expansion
tabl	"(title? , tgroup+)"
Purpose	
Defines the general content style of tables. From CALS 38784C table content model.	

Entity Name	Entity Expansion
graphic	"graphic | grphprim"
Purpose	
Defines the basic graphic elements.	

Entity Name	Entity Expansion
audio	"audio"
Purpose	
Defines the basic audio element.	

Entity Name	Entity Expansion
video	"video"
Purpose	
Defines the basic video element.	

Entity Name	Entity Expansion
link	“link | hylink”
Purpose	
Possible links are normal link and HyTime link	

Entity Name	Entity Expansion
primitive	" %text; | %table; | %graphic; | %audio; | %video; "
Purpose	
This entity provides a simple method for referencing the primitive elements defined in the Generic Layer, in by one entity name.	

Entity Name	Entity Expansion
linkendlist	“(descinfo | partinfo | text | table | graphic | audio | video | para | partbase | entry)”
Purpose	
Elements that can be at the end of links.	

Entity Name	Entity Expansion
a.root	“HyTime NAME HyDoc boslevel NUMBER #IMPLIED”
Purpose	
The a.root entity enables a constent specific layer to comply with the support requirements for HyTime. This entity is only used in the root element og the DTD.	

7.2.2 Generic Layer elements

This section defines the generic layer element templates. These should not be used for tagging up source documents, but rather these templates define semantic rules for creating content specific elements. These semantic rules make up the minimum set of constraints on content specific elements.

Associated with each element template is an attribute entity list. This is an entity that expands to a list of attributes which should be defined for all content specific element following this template element.

There are two general rules to follow when creating a content specific element. First, the element’s content model must comply with the template’s content model. Second, the template’s attribute entity must be included in the element’s attribute list. The attribute entities for all templates include the attributes “id”, “cdm”, “ref”. The “cdm” attribute indicates which template the element is employing. The “id” and “ref” attributes are used for non-redundant referencing and linking.

Element Name	Element Content Model
NODE	((%link;)*, (NODE ∣ NODE-SEQ ∣ %primitive;)*)

Purpose

The NODE template provides the capability to create composite structures within the content specific layer. Composite structures may contain subcomponents that employ the NODE or NODE-SEQ templates. The NODE subcomponents may be composite structures themselves or they may be primitive NODES (text, tables, graphics, audio and video). Composite structures create hierarchy within the CDM. When composite nodes contain other composite nodes there is an implied hierarchy. The composite node in the content-model is at a lower level in the hierarchy (e.g. a Task-Node contains Step-Nodes in its content model).

Entity Name	Entity Expansion
a.node	“id ID #IMPLIED name CDATA #IMPLIED type CDATA #IMPLIED itemid CDATA #IMPLIED cdm NAME #FIXED ‘node’ ref IDREF #IMPLIED”

Purpose

This entity defines the default attributes list for the associated NODE template element. This entity must be included in the attribute list for all content specific elements based on this template element.

The attributes that this entity expands to, are used as follows:

id Element identifier. For cross-referencing to this element. The identifier name must be uniqe within the document file.

name Available for application.

type Available for application.

itemid Available for application.

cdm Content Data Model. Identifies which template element is used as basis for deriving a Content Specific Element.

ref Cross reference to another element.

Element Name	Element Content Model
NODE-SEQ	(NODE)+

Purpose

The NODE-SEQ template is the mechanism for creating interactive sequences with the user.

Entity Name	Entity Expansion
a.node-seq	“id ID #IMPLIED cdm NAME #FIXED ‘node-seq’ ref IDREF #IMPLIED”

Purpose

This entity defines the default attributes list for the associated NODE-SEQ template element. This entity must be included in the attribute list for all content specific elements based on this template element.

The attributes that this entity expands to, are used as follows:

id Element identifier. For cross-referencing to this element. The identifier name must be uniqe within the document file.

cdm Content Data Model. Identifies which template element is used as basis for deriving a Content Specific Element.

ref Cross reference to another element.

Element Name	Element Content Model
link	(#PCDATA)

Purpose

The "link" element provides the capability for creating relational links within the CDM. "Link" is included within the content model of the NODE template, therefore, any content pecific element employing the NODE template may include relational links.

The 'endtypes' attribute identifies the type of primitive or element that the link is pointing to, and the 'linkends' attribute contains the unique identifier attribute (id) of the element being pointed to.

Attribute list for element ‘link’

Attribute Name	Value type	Default value	Purpose
id	ID	#IMPLIED	Element ID. For cross referencing.
endtypes	%linkendlist;	#REQUIRED	Types of elements by which link can reference.
linkends	IDREFS	#REQUIRED	Element IDs of link ends (anchors).

Element Name	Element Content Model
hylink	(#PCDATA)

Purpose

The "hylink" element provides the capability for creating HyTime-compliant relational links within the CDM. "Hylink" is included within the content model of the NODE template, therefore, any content specific element employing the NODE template may include relational HyTime links. The 'anchrole' attribute identifies the type of primitive or element that the link is pointing to, and the 'linkends' attribute contains the unique identifier attribute (id) of the element being pointed to.

Attribute list for element 'hylink'

Attribute Name	Value type	Default value	Purpose
HyTime	NAME	#FIXED 'ilink'	HyTime architectural form
id	ID	#IMPLIED	Identifier
anchrole	NAMES	#FIXED 'hotspot target'	Anchor role
linkends	IDREFS	#REQUIRED	Link endpoints
reftype	CDATA	#FIXED 'linkends linkendtypes #SEQ'	Reference type
extra	NAMES	'A A'	
intra	NAMES	'A A'	
endterms	IDREFS	#IMPLIED	
aggtrav	NAMES	agg	

Element Name	Element Content Model
linkendtypes	(link, (link | descinfo | partinfo | text | table | graphic | audio | video | process | para | task | partbase | dialog | entry)*)

Element Name	Element Content Model
nmlist	(#PCDATA)

Attribute list for element 'nmlist'

Attribute Name	Value type	Default value	Purpose
HyTime	NAME	nmlist	HyTime architectural form
nametype	(entity | element | unified)	#REQUIRED	Name type
obnames	(obnames | nobnames) |	obnames	
docorsub	ENTITY	#IMPLIED	Document or subdocument
dtdorlpd	NAMES	#IMPLIED	

Element Name	Element Content Model
nameloc	(nmlist)*

Attribute list for element ‘nameloc’

Attribute Name	Value type	Default value	Purpose
HyTime	NAME	nameloc	HyTime architectural form
id	ID	#REQUIRED	Identifier
ordering	(ordered | noorder)	ordered	
set	(set|notset)	notset	
aggloc	(aggloc | agglink | nagg)	agglink	

Element Name	Element Content Model
text	((%link;)*, (#PCDATA | text)+)

Purpose

A “text” unit is basically a text string of “parsable character data” or PCDATA. Within a text string, there may be embedded “text” elements which allow the referencing of other elements or parts of elements through the link/location mechanism explained in the HyTime section of this document. Those embedded “text” references are inserted in the text string that contained them. For example, the string may contain a reference to a standard system name, a standard part nomenclature or a standard task name. By using this mechanism, standard terminology can be referenced consistently throughout the data base, and any changes to the standard terminology can be made in one location and automatically updated throughout the data base.

Attribute list for element ‘text’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.

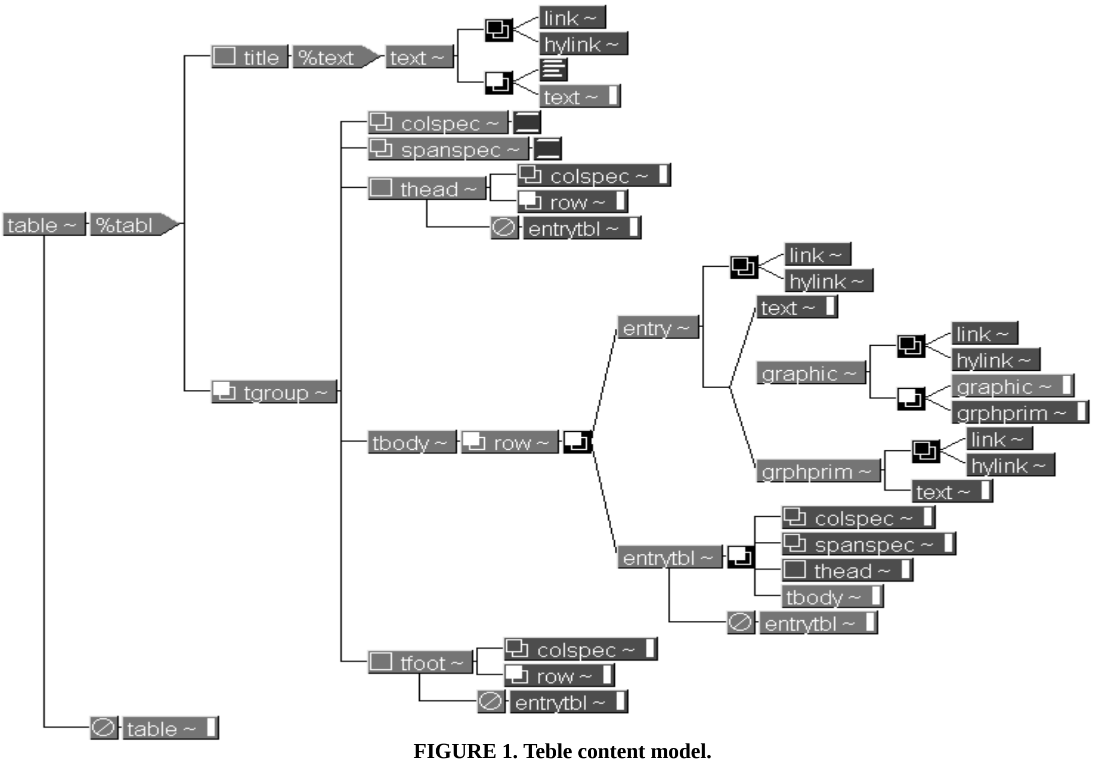


FIGURE 1. Teble content model.

Element Name	**Element Content Model**
table	(%tabl;) -(table)

Purpose

CALS 38784c table element declaration. For application of attributes associated with this element refer to CALS 38784c.

For an explanation of all attributes of the table element, see MIL-M-28001B.

Attribute list for ‘table’

Attribute Name	**Value type**	**Default value**	**Purpose**
tabstyle	NMTOKEN	#IMPLIED	Table style
tocentry	NUMBER	“1”	Table of content entry
verified	NUMBER	“0”	
shortentry	NUMBER	#IMPLIED	
frame	(top , bottom , topbot , all , sides , none)	#IMPLIED	
colsep	NUMBER	#IMPLIED	Column separator
rowsep	NUMBER	#IMPLIED	Row separator
orient	(port , land)	#IMPLIED	Orientation
pgwide	NUMBER	#IMPLIED	Page wide
id	ID	#IMPLIED	Identifier
inschlvl	NUTOKEN	#IMPLIED	Specifies the change level(s) at which information was inserted. An audit trail can be maintained by listing multiple change levels separated by spaces.

Attribute list for ‘table’

Attribute Name	Value type	Default value	Purpose
delchlvl	NUTOKEN	#IMPLIED	Specifies the change level(s) at which information was deleted. An audit trail can be maintained by listing multiple change levels separated by spaces.
label	NMTOKEN	#IMPLIED	If used, it specifies the number or symbol of the ftnote and overrides autogeneration of the number or symbol by the processing system.
hcp	NUMBER	“0”	Hardness Critical Process - If the value consists only of zeros, there is no hardness critical information. If another value is given, the element contains hardness critical information.
esds	NUMBER	“0”	
sssn	CDATA	#IMPLIED	Identifies system/subsystem/subject number (MIDAS) of an equipment part.
unit	CDATA	#IMPLIED	The value of this attribute would be the appropriate unit number of the equipment/ part described in the text of the element type.
module	CDATA	#IMPLIED	The value of this attribute would be the appropriate module number of the equipment/part described in the text of the element type.
lru	CDATA	#IMPLIED	The value of this attribute would be the appropriate line replaceable unit number of the equipment/part described in the text of the element type.
assem	CDATA	#IMPLIED	The value of this attribute would be the appropriate assembly name of the equipment/part described in the text of the element type.
subassem	CDATA	#IMPLIED	The value of this attribute would be the appropriate subassembly name of the equipment/part described in the text of the element type.
ssubassm	CDATA	#IMPLIED	The value of this attribute would be the appropriate sub-subassembly name of the equipment/part described in the text of the element type.
compon	CDATA	#IMPLIED	The value of this attribute would be the appropriate component name of the equipment/part described in the text of the element type.
partno	CDATA	#IMPLIED	The value of this attribute would be the appropriate part number of the equipment/ part described in the text of the element type.
refdes	CDATA	#IMPLIED	The value of this attribute would be the appropriate reference designator of the equipment/part described in the text of the element type.
texttype	NUMBER	#IMPLIED	

Attribute list for ‘table’

Attribute Name	Value type	Default value	Purpose
applictype	IDREFS	#IMPLIED	References the unique applicability definition (<applicdef id=’xxx’>). An example might be that the type of applicability would be aircraft tail numbers. This attribute is optional as it may be derived depending on the context in which the element is used.
applicrefid	IDREFS	#IMPLIED	References unique identifier(s) assigned to applicability identifier(s) (<applicid id=’xxx’>). An example might be a particular aircraft tail number(s).
skilltrk	NMTOKENS	#IMPLIED	Designation of the skill level of the user at which the current element of information is aimed. A particular set of values common to all documents has not been created. Currently, the relevant values are set by contract.
contype	(desc , proc)	#IMPLIED	Identifies the content type. When used with steps, the implied value is procedural. When used with all other element types, the implied value is descriptive.
assocfig	IDREFS	#IMPLIED	Identifies associated figure(s) with the text data through the use of the “id” attribute in the <figure> tag.
assoctab	IDREFS	#IMPLIED	Identifies associated table(s) with the text data.
security	(u , c , s)	“u”	Security classification of the element. Declared Value = u (unclassified), c (confidential), s (secret), ts (top secret)

Element Name	Element Content Model
tgroup	(colspec* , spanspec* , thead?, tbody, tfoot?)

Purpose

In the 38784C DTD, tfoot comes before tbody. This causes trouble for Frame+SGML editor in creating table styles. Therefore these element’s positions have been swapped.

For an explanation of all attributes of this element, see MIL-M-28001B.

Attribute list for ‘tgroup’

Attribute Name	Value type	Default value	Purpose
cols	NUMBER	#REQUIRED	Number of columns in the table or chart.
tgroupstyle	CDATA	#IMPLIED	A unique table group style defined in the FOSI.
colsep	NUMBER	#IMPLIED	Column separator
rowsep	NUMBER	#IMPLIED	Row separator
align	(left , right , center , justify , char)	“left”	Alignment

Attribute list for 'tgroup'

Attribute Name	Value type	Default value	Purpose
charoff	NUTOKEN	"50"	For align="char", horizontal character offset is the percent of the current column width to the left of the (left edge of the) alignment character.
char	CDATA	""	If align = "char", the value is the single alignment character on which the first to occur of this character in the entry is aligned. Entries not containing this character are aligned to the left of this position.
security	(u , c , s)	"u"	Security classification.

Element Name	Element Content Model
colspec	EMPTY
Purpose For an explanation of all attributes of this element, see MIL-M-28001B.	

Attribute list for 'colspec'

Attribute Name	Value type	Default value	Purpose
colnum	NUMBER	#IMPLIED	Number of column, counting from 1 at left of the chart or table.
colname	NMTOKEN	#IMPLIED	Name of column, used to specify the position in a row, or the start or end of a horizontal span of columns (<spanspec>).
align	(left , right , center , justify , char)	#IMPLIED	Text horizontal position within the column.
charoff	NUTOKEN	#IMPLIED	For align="char", horizontal character offset is the percent of the current column width to the left of the (left edge of the) alignment character.
char	CDATA	#IMPLIED	If align = "char", the value is the single alignment character on which the first to occur of this character in the entry is aligned. Entries not containing this character are aligned to the left of this position.
colwidth	CDATA	#IMPLIED	Either proportional measure of the form number*, i.e., "5*" for 5 times the proportion , or "*" (="1*"); fixed measure, i.e., 2pt for 2 point, 3pc for 3 pica; or mixed measure, i.e., 2*+3pt. Coefficients are positive numbers with up to2 decimal places.

Attribute list for ‘colspec’

Attribute Name	Value type	Default value	Purpose
colsep	NUMBER	#IMPLIED	Default for all items in this column (within the closing group) of the table or chart. If other than zeros, display the internal column ruling to the right of each item; if only zeros, do not display it. Ignored for the last column, where the frame setting applies.
rowsep	NUMBER	#IMPLIED	Default for all items in this column (within the enclosing group) of the table or chart. If other than zeros, display the internal horizontal row ruling below each item. If only zeros, do not display it. Ignored for the last row of the table, since overridden by the frame setting.

Element Name	Element Content Model
spanspec	EMPTY

Purpose

For an explanation of all attributes of this element, see MIL-M-28001B.

Attribute list for ‘spanspec’

Attribute Name	Value type	Default value	Purpose
namest	NMTOKEN	REQUIRED	Name of leftmost column of span. Names are identified in colspec of the current tgroup.
nameend	NMTOKEN	REQUIRED	Name of rightmost column of span. Names are identified in colspec of the current tgroup.
spanname	NMTOKEN	IMPLIED	Name of a horizontal span.
align	(left , right , center , justify , char)	”center”	Text horizontal position within the column.
charoff	NUTOKEN	#IMPLIED	For align=”char”, horizontal character offset is the percent of the current column width to the left of the (left edge of the) alignment character.
char	CDATA	#IMPLIED	If align = “char”, the value is the single alignment character on which the first to occur of this character in the entry is aligned. Entries not containing this character are aligned to the left of this position.

Attribute list for ‘spanspec’

Attribute Name	Value type	Default value	Purpose
colsep	NUMBER	#IMPLIED	Default for all items in this column (within the closing group) of the table or chart. If other than zeros, display the internal column ruling to the right of each item; if only zeros, do not display it. Ignored for the last column, where the frame setting applies.
rowsep	NUMBER	#IMPLIED	Default for all items in this column (within the enclosing group) of the table or chart. If other than zeros, display the internal horizontal row ruling below each item. If only zeros, do not display it. Ignored for the last row of the table, since overridden by the frame setting.

Element Name	Element Content Model
thead	(colspec* , row+) -(entrytbl)

Purpose

For an explanation of all attributes of this element, see MIL-M-28001B.

Attribute list for ‘thead’

Attribute Name	Value type	Default value	Purpose
valign	(top , middle , bottom)	“bottom”	Vertical alignment
security	(u , c , s)	“u”	Security classification.

Element Name	Element Content Model
row	(entry ∣ entrytbl)+

Purpose

For an explanation of all attributes of this element, see MIL-M-28001B.

Attribute list for ‘row’

Attribute Name	Value type	Default value	Purpose
rowsep	NUMBER	#IMPLIED	Row separator
security	(u , c , s)	“u”	Security classification

Element Name	Element Content Model
entry	((%link;)*, (text ∣ graphic ∣ grphprim))

Purpose

For an explanation of all attributes of this element, see MIL-M-28001B.

Attribute list for ‘entry’

Attribute Name	Value type	Default value	Purpose
id	ID	#IMPLIED	Identifier
ref	IDREF	#CONREF	Reference
colnum	NUTOKEN	#REQUIRED	Number of column, counting from 1 at left of the chart or table.
row	NUTOKEN	#REQUIRED	

Element Name	Element Content Model
entrytbl	(colspec* , spanspec* , thead? , tbody)+ -(entrytbl)

Purpose

For an explanation of all attributes of this element, see MIL-M-28001B.

Attribute list for ‘entrytbl’

Attribute Name	Value type	Default value	Purpose
cols	NUMBER	#REQUIRED	Number of columns in the table or chart.
tgroupstyle	NMTOKEN	#IMPLIED	A unique table group style defined in the FOSI.
colname	NMTOKEN	#IMPLIED	Column name
spanname	NMTOKEN	#IMPLIED	Span name
colsep	NUMBER	#IMPLIED	Column separator
rowsep	NUMBER	#IMPLIED	Row separator
align	(left , right , center , justify , char)	#IMPLIED	Alignment
charoff	NUTOKEN	#IMPLIED	For align=”char”, horizontal character offset is the percent of the current column width to the left of the (left edge of the) alignment character.
char	CDATA	#IMPLIED	If align = “char”, the value is the single alignment character on which the first to occur of this character in the entry is aligned. Entries not containing this character are aligned to the left of this position.
security	(u , c , s)	“u”	Security classification

Element Name	Element Content Model
tbody	(row+)

Purpose

For an explanation of all attributes of this element, see MIL-M-28001B.

Attribute list for ‘tbody’

Attribute Name	Value type	Default value	Purpose
valign	(top , middle , bottom)	“top”	Vertical alignment
security	(u , c , s)	“u”	Security classification

Element Name	Element Content Model
tfoot	(colspec* , row+) -(entrytbl)

Purpose

For an explanation of all attributes of this element, see MIL-M-28001B.

Attribute list for ‘tfoot’

Attribute Name	Value type	Default value	Purpose
valign	(top , middle , bottom)	“top”	Vertical alignment
security	(u , c , s)	“u”	Security classification

Element Name	Element Content Model
graphic	((%link;)*, (%graphic;)+)

Purpose

This element describes graphics in terms of primitives or references to other graphics, thus providing the ability to create composite graphics.

Attribute list for ‘graphic’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.
minsize	NUTOKENS	#IMPLIED	Available for application.
penshape	CDATA	#IMPLIED	Available for application.
penpatt	CDATA	#IMPLIED	Available for application.
transfrm	NUTOKENS	#IMPLIED	Available for application.
window	NUTOKENS	#IMPLIED	Available for application.

Element Name	Element Content Model
grphprim	((%link;)*, (text))

Purpose

This element defines a primitive graphic which may be contained in the content model or referenced by the ‘ref’ attribute. The graphic is represented in one of the valid formats (cgmbin, cdmchar, cgmclear, fax, iges, dxf, gks), and the format is indicated by the coding attribute. The ‘type’ attribute may identify a graphic as a “hotspot”, thus making it selectable during presentation. The minsize attribute specifies the minimum height requirements for display of the graphic. Any transformations or manipulations of the graphic, other than those described by the notations, can be defined using the penpatt, penshape, transfrm, or window attributes on the graphic primitive element.

Attribute list for ‘grphprim’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.
coding	NOTATION (cgmchar | cgmbin | cgmclear | fax | faxtile | iges)	‘cgmbin’	The format used representing the graphic. (Default is CGM binary representation).
minsize	NUTOKENS	IMPLIED	Minimum size.
penpatt	CDATA	IMPLIED	Pen pattern.
penshape	CDATA	IMPLIED	Pen shape.
transfrm	NUTOKENS	IMPLIED	Transform.
x-location	NUTOKEN	IMPLIED	X location.
y-location	NUTOKEN	IMPLIED	Y location.
window	NUTOKENS	IMPLIED	Window.
external-ptr	ENTITY	IMPLIED	External Pointer
picid	NUTOKEN	IMPLIED	Picture ID

Element Name	Element Content Model
audio	((%link;)*)

Purpose

This element will be used to include an audio sequence into technical information. The model is incomplete pending the Hytime completion.

Attribute list for ‘audio’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.
external-ptr	IDREF	#REQUIRED	External pointer

Element Name	Element Content Model
video	((%link;)*)

Purpose

This element will be used to include an video sequence into technical information. The model is incomplete pending the Hytime completion.

Attribute list for 'video'

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.
external-ptr	IDREF	#REQUIRED	External pointer

7.3 Content data model Content Specific DTD

This section contains a content specific DTD for maritime IETMs. The creation of a content specific DTD represents the second layer of the CDM. It identifies all the content specific elements and their relationships for a given application. In this instance, the application happens to be the display of organizational level data to the technician.

The DTD employs the characteristics defined by the templates of the "Generic Layer." The use of the generic layer primitives means that we do not have to redefine the text, table, graphic, audio, or video, elements within this document.

This document breaks down O-level data into a hierarchy based upon the system/subsystem structure of the ship system. It identifies four different types of information which may be referenced within the document. They are; procedural, descriptive, parts, and fault information. Each type of information is referenced by the system where it is most appropriate.

7.3.1 Content Specific DTD entities

Entity Name	Entity Expansion
dietmdb-b	PUBLIC "-//USA-DOD//DTD Content Data Model Content Specific Layer//EN"

Purpose

Public identifier for the Content data model Content Specific DTD.

Entity Name	Entity Expansion
list	"(seqlist | randlist | deflist)"

Purpose

This entity defines various types of lists used in the documentation. Currently 'sequential lists', 'random lists', and 'definition lists' have been defined.

Entity Name	Entity Expansion
special	"(caution | instruction | note | warning)"

Purpose

Special is used where paragraph styles that should stand out from ordinary paragraphs are needed.

Entity Name	Entity Expansion
sub-prims	" %text; | %special; | %list; | %table; | %graphic; | %audio; | %video; "

Purpose

This lists all types of components that can reside in elements derived from the NODE template element.

7.3.2 Content Specific elements

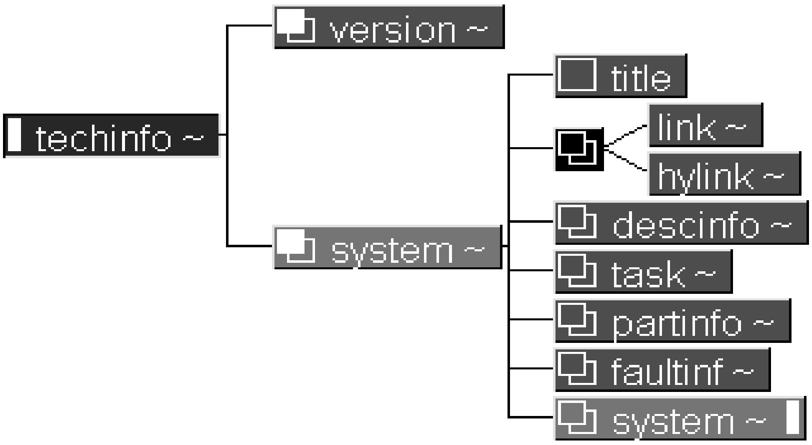


FIGURE 2. Techinfo content model.

Element Name	Element Content Model
techinfo	(version+, (system)+)

Purpose

This element declaration represents the top layer of the information contained in the DTD. The content model contains the top level system, such as "DC2000", "Gantry Crane", "Main Engine", etc. The hierarchy begins at this level.

The ‘techinfo’ element is derived from the NODE template element.

Attribute list for ‘techinfo’

Attribute Name	Value type	Default value	Purpose
%a.root;			This expands to the a.root entity in the Generic Layer description, which includes HyTime functionality.
%a.node;			This expands default attribute list for NODE template element.

Element Name	Element Content Model
system	(title?, (%link;)*, descinfo*, task*, partinfo*, faultinf*, system*)

Purpose

The system element defines the vehicle/system/subsystem/ subassembly hierarchy for the ship system. A system element must be created for any component (ie., vehicle, system, subsystem, subassembly) which has associated technical information (ie., descriptive, procedural, fault, or part information).

The system element employs the 'NODE' template from the generic layer.

Attribute list for 'system'

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.

Element Name	Element Content Model
version	(%text;)?

Purpose

The version declaration is an attempt at controlling different versions of technical information in the database.

The 'version' element employs the NODE template element from the Generic Layer.

Attribute list for 'version'

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.
revision	NMTOKEN	#REQUIRED	Revision number
revdate	NUMBERS	#REQUIRED	Revision date

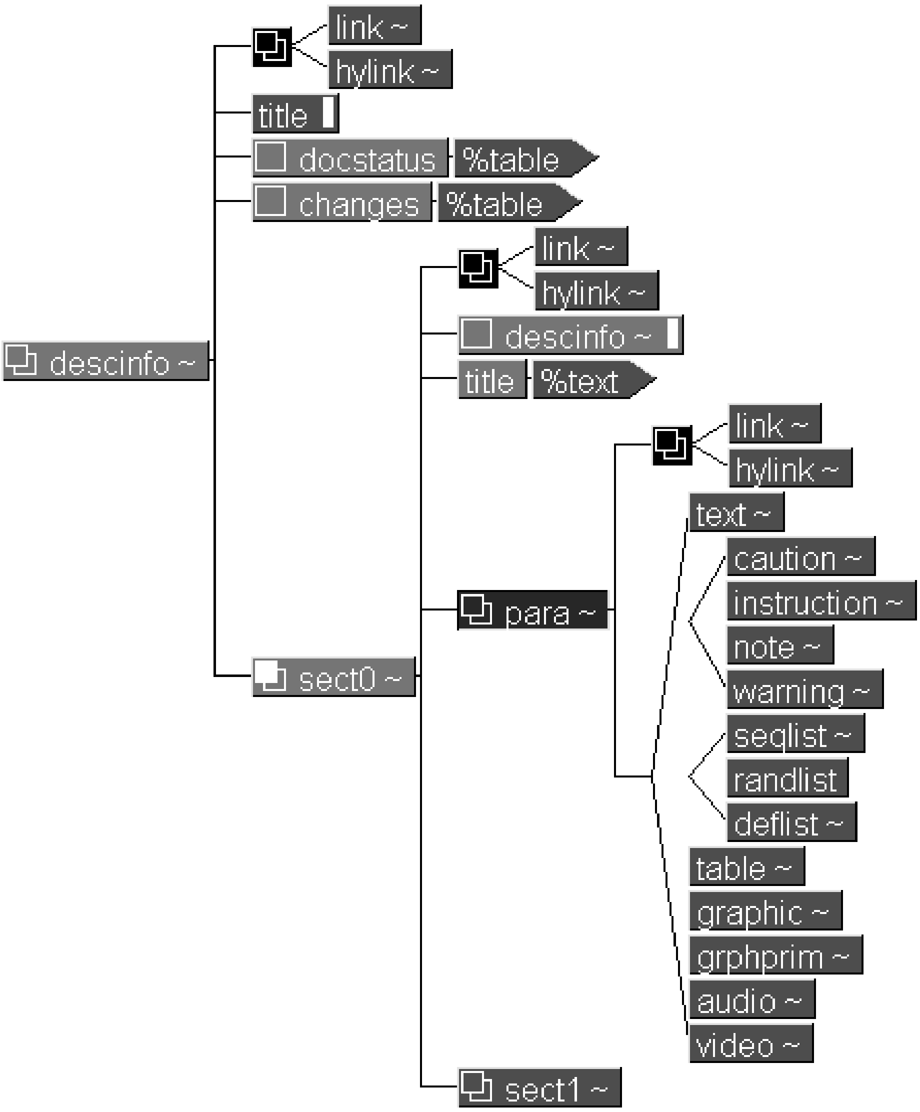


FIGURE 3. Descinfo content model.

Element Name	Element Content Model
descinfo	((%link;)*, title, docstatus?, changes?, sect0+)

Purpose

The element 'descinfo' is used to define general purpose, non-procedural, narrative information such as theory of operation, schematics, etc which are associated with a system component. 'Descinfo' is very flexible. It can be used to describe any arbitrary, hierarchical hypertext-like node containing sub-paragraphs ('paraseq), ('text', 'table', 'graphic', 'audio', 'video', 'process'), etc.

'Descinfo' contains relational links to other elements, a title, a document status table, a list of changes table, and one or more sections (see para declarations).

The descinfo element employs the NODE template from the generic layer.

Attribute list for ‘descinfo’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.

Element Name	Element Content Model
docstatus	(%table)

Purpose

Docstatus is a table of the document status, in the from of a table. This can be used to indicate the current state of the document, like draft, temporary, finished etc.

Element Name	Element Content Model
changes	(%table)

Purpose

Changes is a table listing changes to this document. Could hold updates of previous versions of the document.

Element Name	Element Content Model
para	((%link;)*, (%sub-prims;))

Purpose

The ‘para’ element defines the information which may be contained within the descriptive information as any primitive element defined in the generic layer. It is the highest level paragraph element, and works in a general way by enabling all sorts of primitive elements to be included.

The ‘para’ element employs the NODE template from the Generic Layer.

Attribute list for ‘para’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.

Element Name	Element Content Model
sect0	((%link;)*, descinfo?, title, para*, sect1*)

Purpose

This is the highest level section element. It can optionally contain another description info element, but basically holds a title, a sequence of paragraphs and subsections

The "sect0" element employs the NODE template from the Generic Layer.

Attribute list for ‘sect0’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.

<table>
<tr><th>Element Name</th><th>Element Content Model</th></tr>
<tr><td>title</td><td>(%text;)</td></tr>
<tr><td colspan="2"><b>Purpose</b>
Title holds descinfo/section/subsection etc. titles.</td></tr>
</table>

<table>
<tr><th>Element Name</th><th>Element Content Model</th></tr>
<tr><td>sect1</td><td>((%link;)*, title, para*, sect2*)</td></tr>
<tr><td colspan="2"><b>Purpose</b>
This is the second level section in a section hierarchy. A sect0 element will be the ‘parent’ node to this element.
The sect1 element employs the NODE template element from the Generic Layer.</td></tr>
</table>

Attribute list for ‘sect1’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.

<table>
<tr><th>Element Name</th><th>Element Content Model</th></tr>
<tr><td>sect2</td><td>((%link;)*, title, para*)</td></tr>
<tr><td colspan="2"><b>Purpose</b>
This is the third level section in a section hierarchy. A sect1 element will be the ‘parent’ node to this element.
The sec2 element employs the NODE template element from the Generic Layer.</td></tr>
</table>

Attribute list for ‘sect2’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.

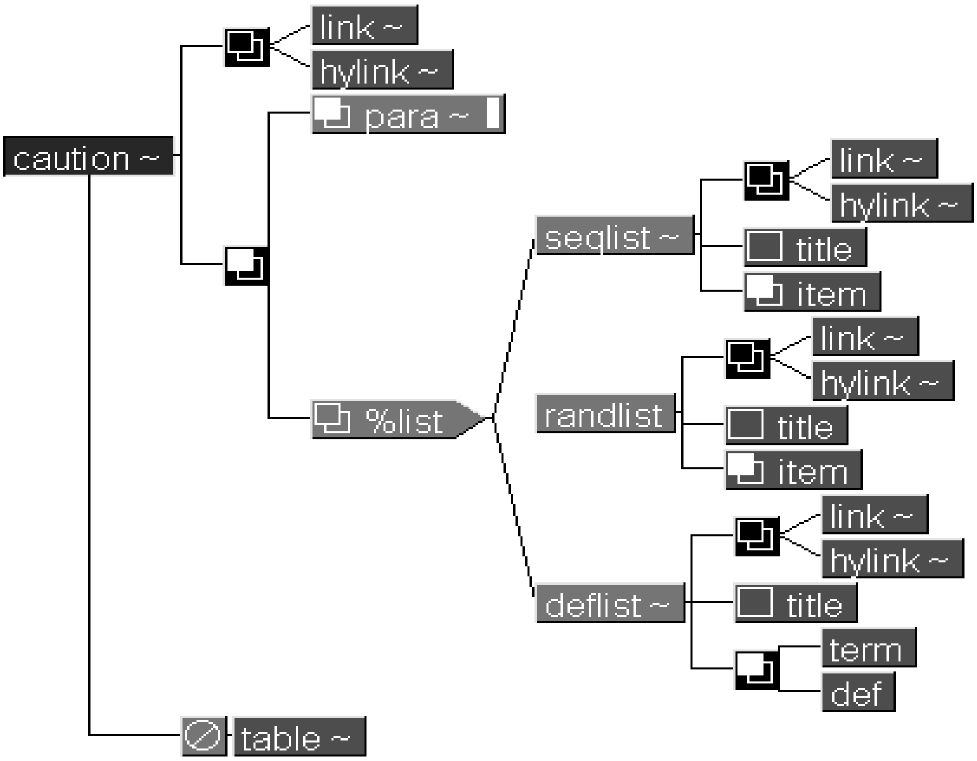


FIGURE 4. Caution content model.

Element Name	**Element Content Model**
caution	((%link;)*, (para+, (%list;)*)+) -(table)

Purpose

This element identifies a special paragraph that identifies a caution. These are needed when content must be separated from ordinary paragraph content. The 'caution' element have been adopted from MIL-PRF-28001 base element set. See 38784C for example of use.

The 'caution' element employs the NODE template element from the Generic Layer.

Attribute list for 'caution'

Attribute Name	**Value type**	**Default value**	**Purpose**
%a.node;			This expands default attribute list for NODE template element.

Element Name	**Element Content Model**
instruction	((%link;)*, (para+, (%list;)*)+) -(table)

Purpose

This element identifies a special paragraph. These are needed when content must be separated from ordinary paragraph content. The 'instruction' lement identifies an instruction.

The 'instruction' element employs the NODE template element from the Generic Layer.

Attribute list for ‘instruction’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.

Element Name	Element Content Model
note	((%link;)*, (para+, (%list;)*)+) -(table)

Purpose

This element identifies a special paragraph. These are needed when content must be separated from ordinary paragraph content. The ‘note’ element have been adopted from MIL-PRF-28001 base element set. See 38784C for example of use. The ‘note’ lement identifies a note.

The ‘note’ element employs the NODE template element from the Generic Layer.

Attribute list for ‘note’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.

Element Name	Element Content Model
warning	((%link;)*, (para+, (%list;)*)+) -(table)

Purpose

This element identifies a special paragraph. These are needed when content must be separated from ordinary paragraph content. The ‘warning’ element have been adopted from MIL-PRF-28001 base element set. See 38784C for example of use. The ‘warning’ lement identifies a warning.

The ‘warning’ element employs the NODE template element from the Generic Layer.

Attribute list for ‘warning’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.

Element Name	Element Content Model
seqlist	((%link;)*, title?, item+)

Purpose

These definitions have been collected from MIL-M-28001B base element name set and adopted into the generic data model of MIL-D-87269.

The ‘seqlist’ element should be used where a sequential numbered list is needed. Attributes control the style of numbering of list items.

The ‘seqlist’ element employs the NODE template from the Generic Layer.

For an explanation of attributes, see MIL-M-28001B.

Attribute list for ‘seqlist’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.
prefix	CDATA	#IMPLIED	Numbering prefix
numstyle	(arabic | romanuc | romanlc | alphauc | alphalc)	#IMPLIED	Numbering style

Element Name	Element Content Model
randlist	((%link;)*, title?, item+)

Purpose

These definitions have been collected from MIL-M-28001B base element name set and adopted into the generic data model of MIL-D-87269.

The ‘randlist’ element should be used where a random non-numbered list is needed. Attributes control the style of numbering of list items.

The ‘randlist’ element employs the NODE template from the Generic Layer.

For an explanation of attributes, see MIL-M-28001B.

Attribute list for ‘randlist’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.
prefix	CDATA	#IMPLIED	Numbering prefix.
numstyle	(arabic | romanuc | romanlc | alphauc | alphalc)	#IMPLIED	Numbering style.

Element Name	Element Content Model
item	(para+)

Purpose

The ‘item’ element is used to hold an list item, and is thus part of the various list content models.

Element Name	Element Content Model
deflist	((%link;)*, title?, (term, def)+)

Purpose

The ‘deflist’ element is used to hold a definition list. These sorts of lists are characterized by having a term followed by an explanation or definition.

The ‘deflist’ element employs the NODE template from the Generic Layer.

Attribute list for ‘deflist’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.

Element Name	Element Content Model
term	(%text;)

Purpose

The ‘term’ element is used to hold the first part of a term/defninition pair in definition lists.

Element Name	Element Content Model
def	(para+)

Purpose

The ‘def’ element is used to hold the second part of a term/defninition pair in definition lists i.e. the definition of the term.

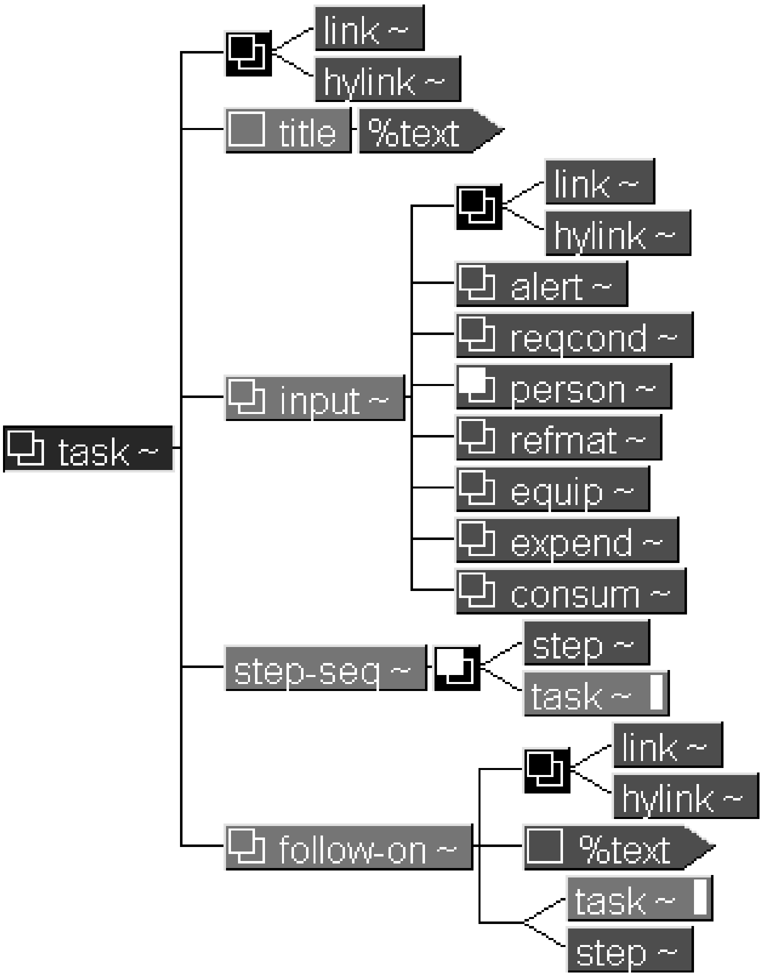


FIGURE 5. Task content model.

Element Name	Element Content Model
task	((%link;)*, title?, input*, step-seq, (follow-on)*)

Purpose

The ‘task’ element defines a maintenance procedure, such as a removal, repair, replacement, test, adjustment, etc. associated with a ‘system’ component. A ‘task’ element contains a list of preconditions which define the task's applicability, relational links to other information elements and input conditions for beginning the task, precautionary messages (i.e., warnings, cautions and notes), a sequence of procedural steps, a list of follow-on conditions which must be accomplished sometime following the completion of the task, and a list of post-conditions which define any state changes to be made after the task is accomplished.

The ‘task’ element employs the NODE template from the generic layer.

Attribute list for ‘task’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.
esttime	NUTOKEN	#IMPLIED	Estimated time in minutes required for the corresponding task to be completed

Attribute list for ‘task’

Attribute Name	Value type	Default value	Purpose
operability	CDATA	#IMPLIED	See MIL-D-87269 Appendix D
servicedes	CDATA	#IMPLIED	Service designator. See MIL-D-87269 Appendix D

Element Name	Element Content Model
input	((%link;)*, alert*, reqcond*,person+, refmat*, equip*,expend*, consum*)

Purpose

The ‘input’ element identifies all the set-up conditions which must be met prior to beginning a task. An ‘input’ contains applicability preconditons, relational links to other elements, and the personnel, consumables, equipment and required conditions for accomplishing the task.

The ‘input’ element employs the NODE template from the generic layer.

Attribute list for ‘input’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.

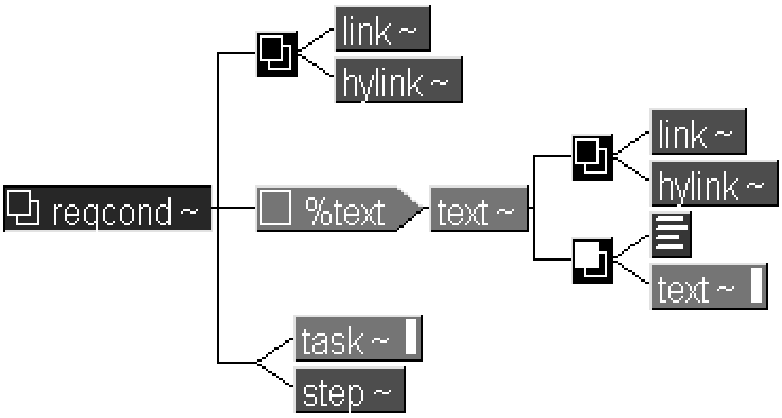


FIGURE 6. Reqcond content model.

Element Name	Element Content Model
reqcond	((%link;)*, (%text;)?, (task ∣ step))

Purpose

A required condition ‘recond’ identifies a maintenance condition (eg., Aircraft Safe For Maintenance), which must be satisfied before beginning a task. It also identifies the task(s) or step(s) which accomplish the required condition if it is not satisfied.

A ‘reqcond’ contains a set of relational links, an optional text element which describes the maintenance condition, a list of task(s) or step(s) which provide instructions for accomplishing the maintenance condition.

The "reqcond" element employs the 'NODE' template from the generic layer.

Attribute list for 'reqcond'

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.

Element Name	Element Content Model
refmat	((%link;)*, (%text;)?)

Purpose

The 'refmat' element identify reference material for a task.

The 'refmat' element employs the NODE template from the Generic Layer.

Attribute list for 'refmat'

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.
desig	CDATA	#REQUIRED	See MIL-D-87269 Appendix D.

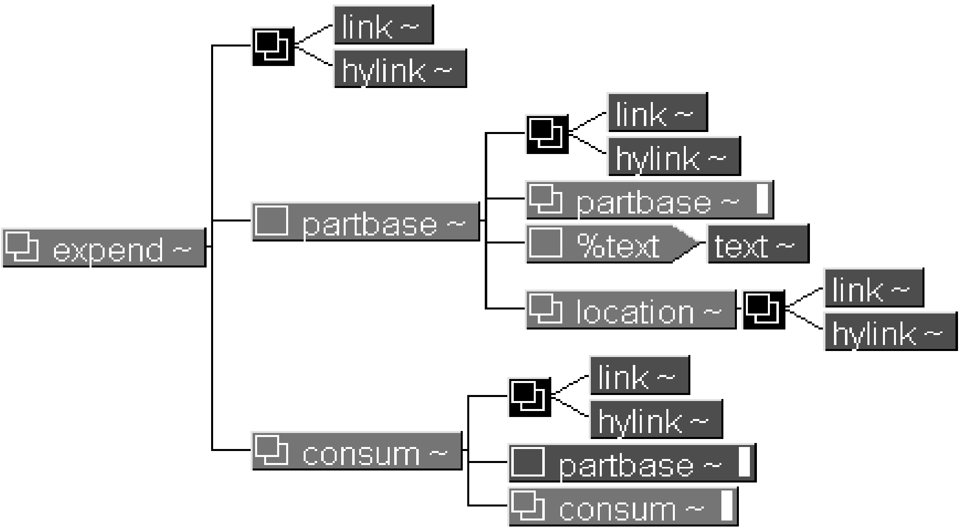


FIGURE 7. Expend content model.

Element Name	Element Content Model
expend	((%link;)*, (partbase)?, (consum)*)

Purpose

The 'expend' element identify expendables for a task.

The 'expend' element employs the NODE template from the Generic Layer

Attribute list for ‘expend’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.
quantity	NUTOKEN	#REQUIRED	Used in the person, equip, expend and consum elements, the value of this attribute signifies the amount of the appropriate consumable, equipment, or people required for the associated task/step.

Element Name	Element Content Model
person	((%link;)*, (%text;)?)

Purpose

This element is used to identify the personnel requirements for a given task. The 'type' attribute will be used to identify the kind of technician required. The 'quantity' attribute identifies the number of that type of technician required for the task.

The person element employs the 'NODE' template from the generic layer.

Attribute list for ‘person’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.
quantity	NUTOKEN	#REQUIRED	Used in the person, equip, expend and consum elements, the value of this attribute signifies the amount of the appropriate consumable, equipment, or people required for the associated task/step.

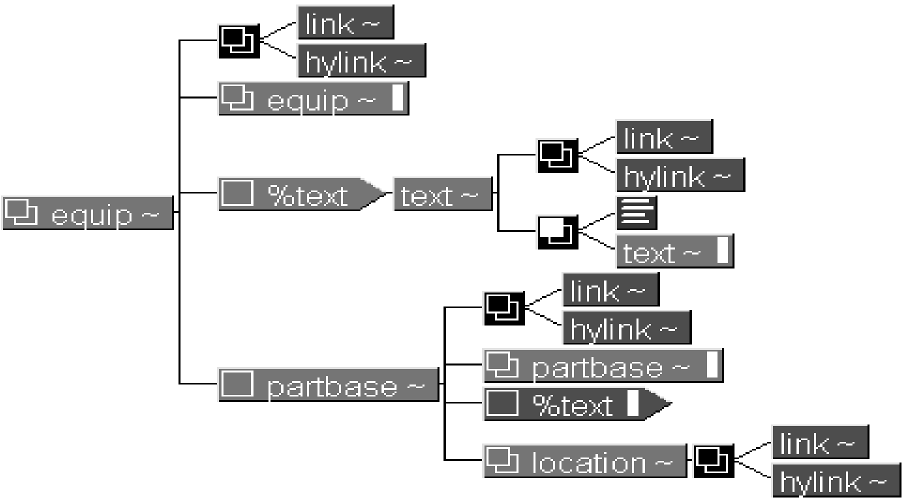


FIGURE 8. Equip content model.

Element Name	Element Content Model
equip	((%link;)*, (equip)*, (%text;)?, (partbase)?)

Purpose

This element identifies all the support equipment required for the completion of the task. An ‘equip’ contains applicability preconditons, relational links to other elements, and any alternate equipment. The quantity attribute identifies the number of equipment items required to complete the task.

The equip element employs the 'NODE' template from the generic layer.

Attribute list for ‘equip’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.
quantity	NUTOKEN	#IMPLIED	Used in the person, equip, expend and consum elements, the value of this attribute signifies the amount of the appropriate consumable, equipment, or people required for the associated task/step.

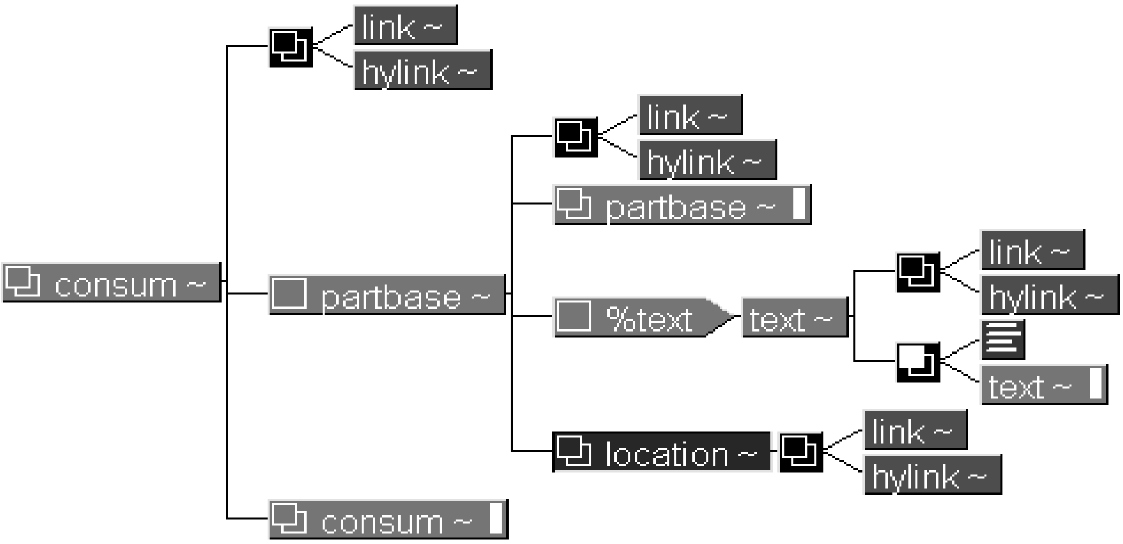


FIGURE 9. Consum content model.

Element Name	Element Content Model
consum	((%link;)*, (partbase)?, (consum)*)

Purpose

The ‘consum’ element identifies all the consumables required for the completion of the task.

A ‘consum’ contains applicability preconditons and relational links to other elements. The ‘consum’ element contains many attributes which identify what the consumable is (govstd, mfgcode, milspec), and the amount required (quantity, unit-of-measure) for accomplishing the task.

The ‘consum’ element employs the ‘NODE’ template from the generic layer.

Attribute list for ‘consum’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.
govstd	CDATA	#IMPLIED	Used in the consum element, the value of this attribute signifies a document that establishes engineering and technical requirements for processes, procedures, practices, and methods that have been adopted as standards. It also establishes requirements for selection, application, and design criteria for materials.
mfgcode	CDATA	#IMPLIED	Used in the consum element, the value of this attribute indicates the in-house code a manufacturer uses to represent parts.
milspec	CDATA	#IMPLIED	Used in the consum element, the value of this attribute represents the exact specification for each item bought by the government.

Attribute list for ‘consum’

Attribute Name	Value type	Default value	Purpose
quantity	NUTOKEN	#REQUIRED	Used in the person, equip, expend and consum elements, the value of this attribute signifies the amount of the appropriate consumable, equipment, or people required for the associated task/step.
unit-of-measure	NMTOKEN	#IMPLIED	Used in the consum element, this attribute identifies the type of unit measurement used to quantify the number of consumables needed for the current application. (e.g., “inches”, “meters”, “pounds”, etc.).

Element Name	Element Content Model
alert	((%link;)*, (%text;)+, (%graphic;)*)

Purpose

The ‘alert’ element identifies an alert that may accompany a task or step. The ‘type’ attribute will identify the kind of alert, either Warning, Caution, Note. An ‘alert’ contains applicability preconditions, relational links, text elements which make up the content of the alert message, and optional ‘graphic’ icons to be displayed.

The alert element employs the ‘NODE’ template from the generic layer.

Attribute list for ‘alert’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.

Element Name	Element Content Model
step	((%link;)*, (alert)*, (%sub-prims;)*, step-seq?)

Purpose

Steps are the primary component of a maintenance procedure. They describe the actions to be performed in order to successfully complete the task. A “step” contains a list of preconditions which delimit the step’s applicability, relational links, precautionary alerts, an optional sequence of substeps, and a list of postconditions which define the state changes to be made after the step is accomplished.

The step element employs the ‘NODE’ template from the generic layer.

Attribute list for ‘step’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.
esttime	NUTOKEN	#IMPLIED	Used in the task and step elements, the value of this attribute indicates the amount of time, in minutes, required for the corresponding task/step to be completed.

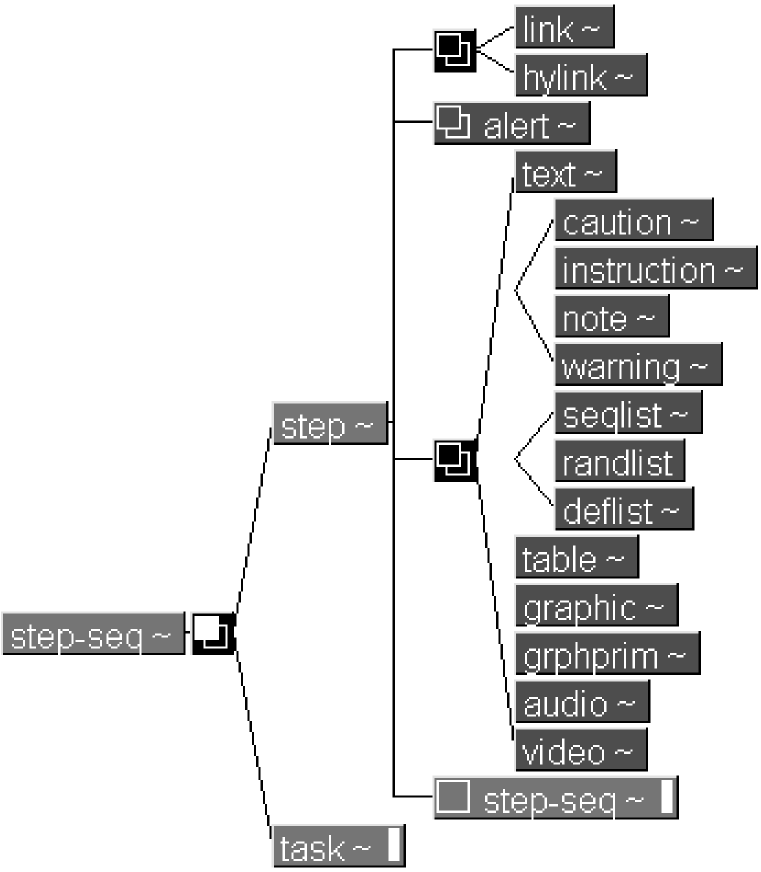


FIGURE 10. Step-seq content model.

Element Name	**Element Content Model**
step-seq	(step | task)+

Purpose

The ‘step-seq’ element provides the capability to create sequences of steps.

This element employs the NODE-SEQ template from the generic layer.

Attribute list for ‘step-seq’

Attribute Name	**Value type**	**Default value**	**Purpose**
%a.node;			This expands default attribute list for NODE template element.

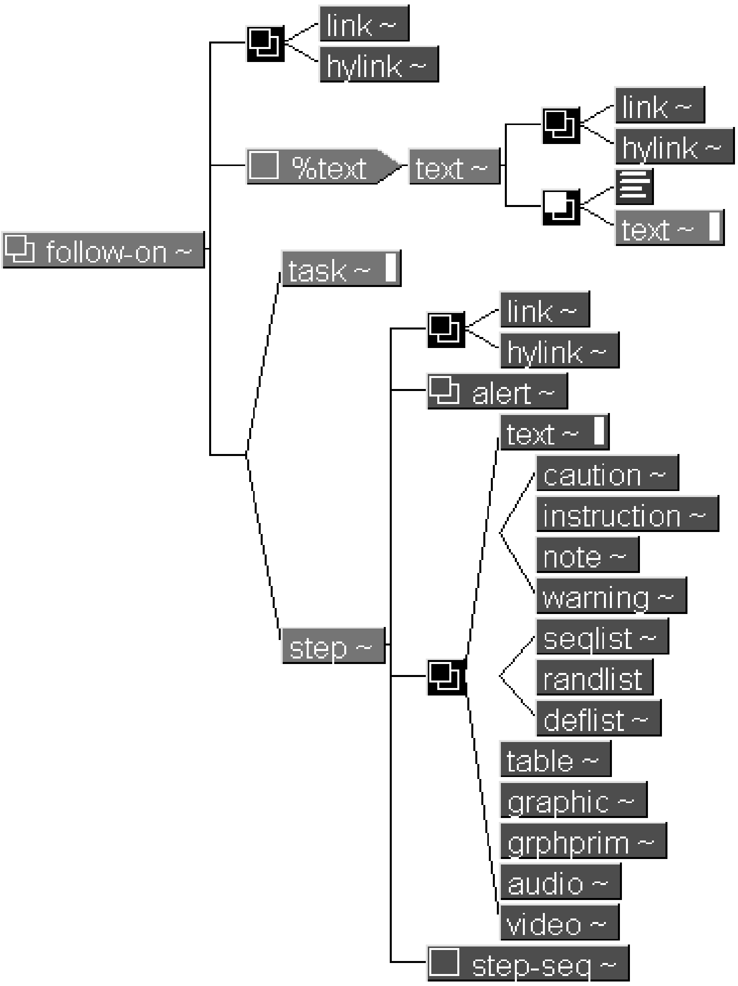


FIGURE 11. Follow-on content model.

Element Name	Element Content Model
follow-on	((%link;)*, (%text;)?, (task ǀ step))

Purpose

A follow-on condition (follow-on) is a maintenance condition which must be accomplished sometime following the completion of a task to clean-up or undo actions performed during the task. For example, in order to fix a component a task might require that an access panel be removed. The panel would then need to be replaced as a follow-on action. This task might be performed sometime after the repair task is completed, but not immediately after the repair task. Other maintenance tasks might be performed in the same area before the follow-on task is accomplished.

A ‘follow-on’ element contains a set of preconditions which define the follow-on maintenance condition which must be satisfied, relational links, an optional text element which describes the follow-on condition, a list of task(s)/step(s) which provide instructions for accomplishing the follow-on condition, and a set of post-conditions which define the state changes to be made once the follow-on condition is accomplished.

The ‘follow-on’ element employs the NODE template from the generic layer.

Attribute list for ‘follow-on’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.

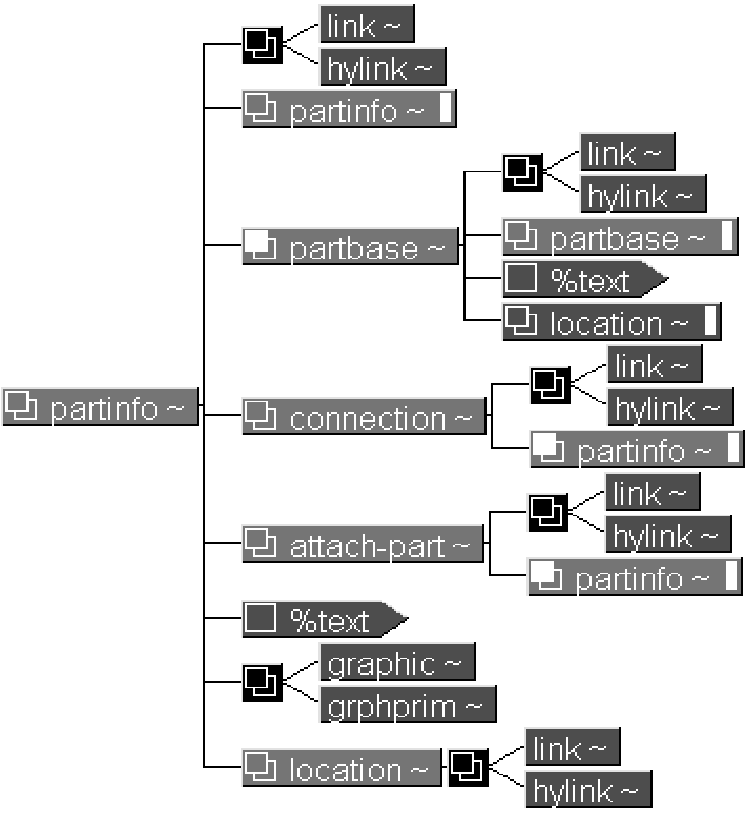


FIGURE 12. Partinfo content model.

Element Name	Element Content Model
partinfo	((%link;)*, (partinfo)*, (partbase)+, (connection)*, (attach-part)*, (%text;)?, (%graphic;)*, (location)*)

Purpose

‘Partinfo’ describes the maintainer’s view of the part information. Each ‘partinfo’ element is related to a ‘partbase.’ However, several ‘partinfo’ items could be related to the same ‘partbase.’ A ‘partinfo’ element contains a list of preconditions, relational links, and alternate parts information (the ‘partinfo’ in the content model). “Partinfo” also identifies the components of the part (partbase), any connecting parts (connection), attaching parts (attach-part), a formal name for the part (text), a picture of the part (graphic), and the location of the part in reference to the weapon system (location).

The ‘partinfo’ element employs the NODE template.

Attribute list for ‘partinfo’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.
indexnum	NUTOKENS	#IMPLIED	Used in the partinfo element, the value of this attribute contains the index number for the part which represents a callout in a graphic output onto paper.

Attribute list for 'partinfo'

Attribute Name	Value type	Default value	Purpose
lru	NUTOKEN	#IMPLIED	Used in the partinfo element, this attribute signifies an essential support item that is removed and replaced at field level to restore the end item to its operationally ready condition. Allowable values are: Item is a LRU Y Item is not a LRU N
mtbf	CDATA	#IMPLIED	Used in the fault and partinfo elements, the value of this attribute signifies, for a particular interval, the total functional life of a population of an item divided by the total number of failures within the population during the measurement interval. The definition holds for time, rounds, miles, events, or other measure-of-life units.
refdes	NMTOKEN	#IMPLIED	Used in the partinfo element, this attribute is an identifier assigned according to a numbering scheme for parts of a system which reflects the hierarchical assembly of the system.
replvl	CDATA	#IMPLIED	Used in the partinfo element, this attribute represents the minimum quantity of a part in stock that will trigger a reorder or stock action.
unitsper	NUTOKEN	#IMPLIED	Used in the partinfo element, this attribute represents the number of units required per assembly of a system or component.
usablon	NUTOKENS	#IMPLIED	Used in the partinfo element, this attribute identifies the different configurations in which a part or assembly may appear within a system or vehicle.

Element Name	Element Content Model
partbase	((%link;)*, (partbase)*, (%text;)?, (location)*)

Purpose

The 'partbase' element describes the supply system's view of the part information. It describes the item in terms of its part number ('partnum').

It allows for the declaration of preconditions for partbase information and relational linking to other information from the partbase element.

The 'partbase' element employs the NODE template from the generic layer.

Attribute list for ‘partbase’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.
cage	NUTOKENS	#REQUIRED	Used in the consum, equip and partbase elements, this attribute is a five-character code assigned by the Defense Logistics Services Center (DLSC) to the design control activity or actual manufacturer of an item contained in the Cataloguing Handbook H4/H8 series.
fsc	CDATA	#REQUIRED	Used in the partbase element, the value of this attribute contains applicable Federal Stock Classification (FSC) codes.
partnum	CDATA	#REQUIRED	Used in the partbase element, this attribute signifies any number, other than a government activity stock number, used to identify an item of production or supply.
smr	CDATA	#REQUIRED	Used in the partbase element, SMR codes are alphabetic or alphanumeric symbols used at the time of provisioning to indicate the source of supply of an item, its maintenance implications, and its recoverability characteristics. The provisioning activity may require the contractor to recommend these codes. Approved codes are defined in: AR 700-82, OPNAVINST 4410.2, AFR 66-45, MCO 4400.120, and DSAR 4100.6.
nsn	CDATA	#IMPLIED	Used in the equip, consum and partbase elements, the value of this attribute is a number, assigned under the Federal Cataloguing Program and/or North Atlantic Treaty Organization (NATO) codification of equipment system to each approved item, which provides a unique identification of an item of supply within a specified Federal Supply Classification (FSC). The field consists of a three-character prefix, a thirteen-character National Stock Number (NSN), and a four-character suffix code. For applicable codes, see DOD 4100.38-M.
pmic	CDATA	#IMPLIED	
cac	NUTOKEN	#IMPLIED	
qpei	NUTOKEN	#IMPLIED	
hci	(Y1 | N1)	“N1”	Used in the partbase element, the value of this attribute represents a code which indicates that an item could degrade system survivability in a nuclear, biological, or chemically hostile environment if hardness were not considered.
lox	(Y2 | N2)	“N2”	
esds	(Y3 | N3)	“N3”	
qec	(Y4 | N4)	“N4”	
magnetic	(Y5 | N5)	“N5”	

Element Name	**Element Content Model**
connection	((%link;)*, (partinfo)+)

Purpose

The ‘connection’ element defines a connection between two ‘partinfo’ elements.

The ‘connection’ element employs the NODE template from the generic layer.

Attrbute list for ‘connection’

Attribute Name	**Value type**	**Default value**	**Purpose**
%a.node;			This expands default attribute list for NODE template element.
version	IDREF	#REQUIRED	
status	(u | a)	‘a’	

Element Name	**Element Content Model**
attach-part	((%link;)*, (partinfo)+)

Purpose

The ‘attach-part’ element defines the attaching parts for a ‘partinfo’ element.

The ‘attach-part’ element employs the NODE template from the generic layer.

Attrbute list for ‘attach-part’

Attribute Name	**Value type**	**Default value**	**Purpose**
%a.node;			This expands default attribute list for NODE template element.
version	IDREF	#REQUIRED	
status	(u | a)	‘a’	

Element Name	**Element Content Model**
location	((%link;)*)

Purpose

The location element provides information for physical assessment. It will contain x, y, z location(s) for a system with respect to the x, Fuselage Station (FS), y, Buttock Line (BL), and z, Water Line (WL) reference system. Where appropriate BL may be replaced by Wing Station (WS).

It allows for the declaration of preconditions for a physical location and relational linking to other information from the location element.

The location element employs the ‘NODE’ template from the generic layer.

Attribute list for ‘location’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.
version	IDREF	#REQUIRED	
status	(u | a)	‘a’	
location-x	NUTOKENS	#IMPLIED	Used in the location element, the value of this attribute contains a number representing a position on the Fuselage Station (FS), which is used as the x-axis of the weapon system.
location-y	NUTOKENS	#IMPLIED	Used in the location element, the value of this attribute represents a position on the Buttock Line (BL), which is used as the y-axis of the weapon system.
location-z	NUTOKENS	#IMPLIED	Used in the location element, the value of this attribute contains a number representing a position on the Water Line (WL), which is used as the z-axis of the weapon system.

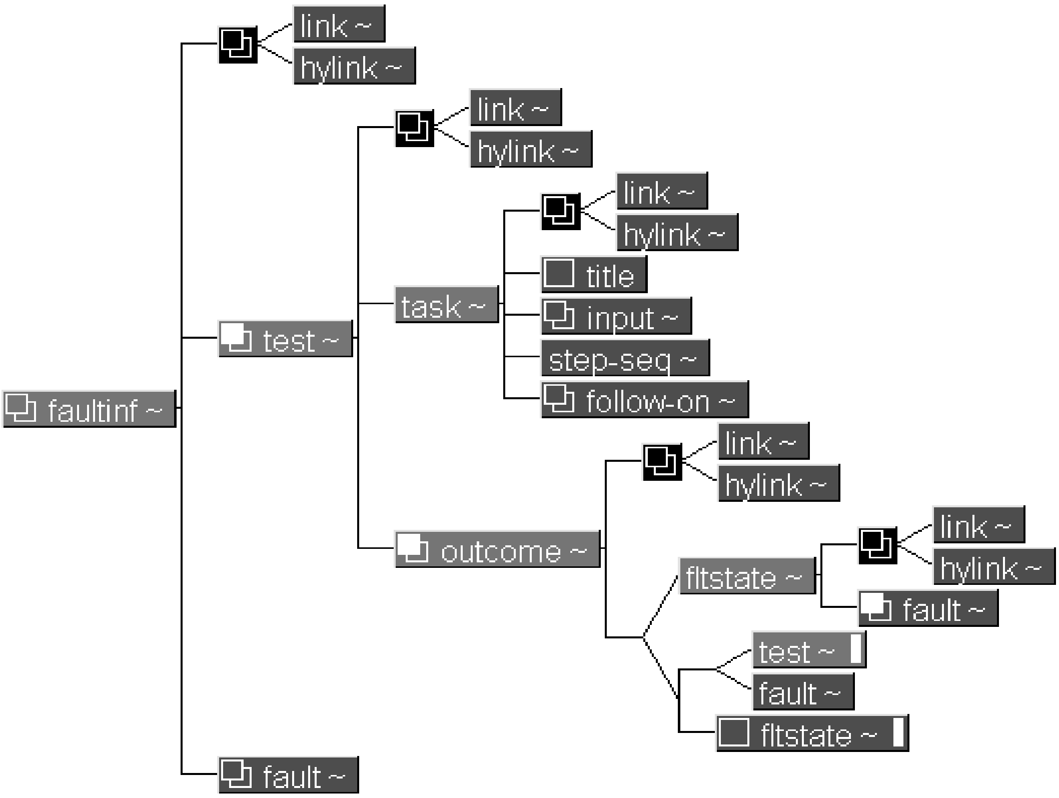


FIGURE 13. Faultinf content model.

Element Name	**Element Content Model**
faultinf	((%link;)*, (test)+, (fault)*)

Purpose

The ‘faultinf’ element identifies all the fault isolation information associated with a system. ‘Faultinf’ can be used to support dynamic troubleshooting models or static troubleshooting trees.

It contains a list of preconditions, relational links to other elements, and lists of tests and faults associated with a system.

The ‘faultinf’ element employs the NODE template from the generic layer.

Attrbute list for ‘faultinf’

Attribute Name	**Value type**	**Default value**	**Purpose**
%a.node;			This expands default attribute list for NODE template element.
version	IDREF	#REQUIRED	
status	(u ǀ a)	‘a’	

Element Name	Element Content Model
test	((%link;)*, (task), (outcome)+)

Purpose

The ‘test’ element identifies a prescribed task to perform and is the usual way of entering the troubleshooting process. The result of a test is an outcome; a test will have two or more outcomes. This element identifies the task needed to complete the test and all the possible outcomes as a result of the test. It contains a list of preconditions and relational links to other information. A ‘test’ element identifies the task which will accomplish the test. All the possible outcomes are contained within the test.

The ‘test’ element employs the NODE template.

Attribute list for ‘test’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.
version	IDREF	#REQUIRED	
status	(u | a)	‘a’	
agent	CDATA	#IMPLIED	Used in the rect and test element, this attribute contains character data describing who performs a maintenance action. It can be either a ‘human’ agent, or some valid computer system (e.g., 1553 bus) called ‘machine’. The default value is ‘human’.
range	CDATA	#IMPLIED	Used in the test element, this attribute represents the boundaries for valid choices or outcomes, according to the element containing the range.

Element Name	Element Content Model
outcome	((%link;)*, (fltstate) | ((test | fault), (fltstate)?)))

Purpose

The ‘outcome’ element identifies a result of a test. In a dynamic troubleshooting model, the outcome will contain a faultstate that identifies an implicated or exculpated set of faults. In a static troubleshooting model, the outcome will contain another test or a fault. The faultstate element will identify the implicated or exculpated faults for the outcome. The test and fault elements identify the next step in a static fault tree.

The ‘outcome’ element employs the NODE template from the generic layer.

Attrbute list for ‘outcome’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.
version	IDREF	#REQUIRED	
status	(u | a)	‘a’	

Element Name	Element Content Model
fltstate	((%link;)*, (fault)+)

Purpose

The ‘fltstate’ element identifies a set of implicated or exculpated faults. Implicated faults are faults suspected of being bad; exculpated faults are faults known to be good. Each implicated fault will have an associated weight based on its likelihood of causing the discrepancy. It contains a list of preconditions, and relational links to other appropriate information. The ‘type’ attribute will designate whether the list of faults are “implicated” or “exculpated.”

The ‘fltstate’ element employs the NODE template from the generic layer.

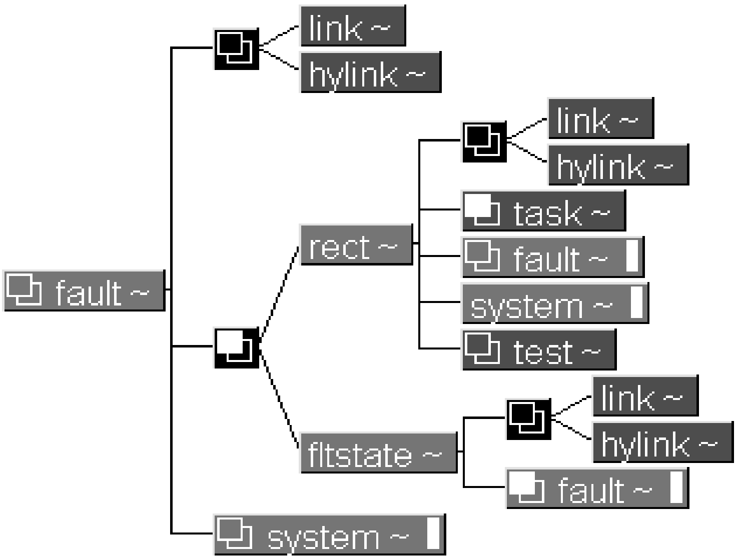


FIGURE 14. Fault content model.

Element Name	Element Content Model
fault	((%link;)*, (rect | fltstate)+, (system)+)

Purpose

The ‘fault’ element identifies the cause of a discrepancy on the weapon system. The fault will identify the appropriate rectification to correct the discrepancy. When transitioning between maintenance levels the fltstate element is used. The rectifications contain tasks which will correct the discrepancy. The system and part information elements will create a back-link to the part that has failed. The ‘fltstate’ represents the system at the next level of maintenance.

The ‘fault’ element employs the NODE template.

Attribute list for ‘fault’

Attribute Name	Value type	Default value	Purpose
%a.node;			This expands default attribute list for NODE template element.
version	IDREF	#REQUIRED	

Attribute list for ‘fault’

<table>
<tr><th>Attribute Name</th><th>Value type</th><th>Default value</th><th>Purpose</th></tr>
<tr><td>status</td><td>(u | a)</td><td>‘a’</td><td></td></tr>
<tr><td>mtbf</td><td>CDATA</td><td>#IMPLIED</td><td>Used in the fault and partinfo elements, the value of this attribute signifies, for a particular interval, the total functional life of a population of an item divided by the total number of failures within the population during the measurement interval. The definition holds for time, rounds, miles, events, or other measure-of-life units.</td></tr>
</table>

Element Name	Element Content Model
rect	((%link;)*, (task)+, (fault)+, (system), (test)*)

Purpose

The ‘rect’ element identifies the prescribed task that will repair the fault causing the discrepancy and all other faults that could be fixed by the rectification. Upon completion of the task, a test is performed to verify the effect of the rectification. The ‘system’ element provides a reference to the system which will be repaired by the rectification. The test element identifies all check-out tests required before completing the maintenance session.

The ‘rect’ element employs the NODE template from the generic layer.

Attribute list for ‘rect’

<table>
<tr><th>Attribute Name</th><th>Value type</th><th>Default value</th><th>Purpose</th></tr>
<tr><td>%a.node;</td><td></td><td></td><td>This expands default attribute list for NODE template element.</td></tr>
<tr><td>version</td><td>IDREF</td><td>#REQUIRED</td><td></td></tr>
<tr><td>status</td><td>(u | a)</td><td>‘a’</td><td></td></tr>
<tr><td>action</td><td>(swap | maint)</td><td>#REQUIRED</td><td>Used in the rect element, this attribute contains character data describing the type of maintenance action required to rectify, or fix, a fault. The action can be a ‘swap’, which means it is a removal/replacement action, or it can be a ‘maint’ action, which means it is an adjustment, alignment, or similar action. The default value is ‘swap’.</td></tr>
<tr><td>agent</td><td>CDATA</td><td>#IMPLIED</td><td>Used in the rect and test element, this attribute contains character data describing who performs a maintenance action. It can be either a ‘human’ agent, or some valid computer system (e.g., 1553 bus) called ‘machine’. The default value is ‘human’.</td></tr>
</table>

8 Style Sheets

8.1 Introduction

Here we include styles that will allow the presentation of source documents tagged using the MITD DTD specified in this document, using DynaText (Copyright Electronic Book Technologies Inc.). The style sheets presented here does not cover the complete DTD. The elements *faultinf* and *partinfo*, including their sub-elements, will not be formatted.

These style sheets are **not** the same as the much referred term in other CALS specifications, namely FOSI. Further, these style sheets cannot take the role of a FOSI. A FOSI is, technically speaking, another DTD that allows detailed description of how to print a document that is tagged by one of the CALS base element set DTDs. The FOSI provides for a “device independent” way of giving such a presentation description. In contrast, style sheets are “device dependent” because they are relevant only to the application they were written for (in this example DynaText). There is no correlation between FOSIs and style sheets such that given a style sheet, we can derive a FOSI (automatically), and vice versa.

The following explains each style sheet in more detail.

8.1.1 Fulltext.v

This style sheet defines how the main presentation window of a DynaText book is presented. This window is (normally) located at the right side of the DynaText application window.

8.1.2 Toc.tv

This style sheet defines the format of the table of contents window of a DynaText book presentation. This window is (normally) located at the left side of the DynaText application window.

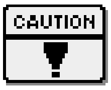
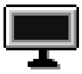
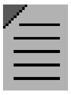
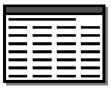
8.1.3 Tables.v

This style sheet defines the format of table presentations in DynaText. This style sheet presents tables in separate windows, which means the table is indicated by an icon in the fulltext view. By activating (double-clicking) the icon, the table window is opened showing the table contents.

8.1.4 Icons

The table below shows icons used by the style sheets included in this document.

Icons used by style sheets

Icon	Explanation
	Bookmark symbol. Indicates that there is a bookmark link to this position
	Caution. Bitmap that is shown together with the CAUTION element presentation
	Figure. Bitmap indicating that there is a figure inserted into the document at this position. Activating this bitmap will present the picture or figure in its own window
	Note. This bitmap indicates that a user entered note is present at this position.
	Table. This bitmap indicates that there is a table at this position. Activating the bitmap will present the table in its own window.

8.2 Fulltext.v

```
<!-- -->


<sheet >



<?INSTED COMMENT: UNGROUPED STYLES FOLLOW>

<style name=”#DEFAULT”>
       <break-before>      Line          </>
</style>

<style name=”#ROOT”>
       <break-before>      Line          </>
</style>

<style name=”#SDATA”>
       <font-family>       attr(font)    </>
       <font-weight>       Medium        </>
       <font-slant>        Roman         </>
       <character-set>     attr(charset) </>
       <break-before>      None          </>
       <text-before>char(attr(code))</>
</style>

<style name=”#TAGS”>
       <font-family>       Courier       </>
       <font-weight>       Bold          </>
</style>

<style name=”ALERT”>
       <font-weight>       Demibold      </>
</style>

<style name=”CAUTION”>
       <font-weight>       Bold          </>
       <space-before>      12            </>
       <space-after>       12            </>
       <icon-position>     Inline        </>
       <icon-type>         caution       </>
</style>

<style name=”CAUTION,PARA”>
       <break-before>      Line          </>
</style>

<style name=”DEF”>
       <break-before>      None          </>
</style>

<style name=”DEF,PARA”>
       <left-indent>       +=120         </>
       <break-before>      None          </>
</style>

<style name=”DEFLIST”>
       <space-before>      8             </>
       <hrule>             After         </>
</style>

<style name=”DESCINFO”>
       <space-before>      36            </>
       <break-before>      Line          </>
       <break-after>       Line          </>
</style>

<style name=”DESCINFO,TITLE,TEXT”>
       <font-weight>       Demibold      </>
       <font-size>         18            </>
       <foreground>        dark red      </>
       <line-spacing>      18            </>
       <hrule>             Before        </>
</style>

<style name=”EQUIP”>
       <score>             Under         </>
       <space-before>      6             </>
       <text-before>EQUIPMENT:</>
</style>

<style name=”EQUIP,TEXT”>
       <score>             None          </>
       <break-before>      Line          </>
</style>
```

```
<style name=”GRPHPRIM”>
       <select>            GRPHPRIM_Attr(Window)</>
</style>

<style name=”GRPHPRIM,TEXT”>
       <break-before>      None          </>
</style>

<style name=”GRPHPRIM_CURRENT”>
       <foreground>        purple        </>
       <space-before>      12            </>
       <space-after>       12            </>
       <justification>     Center        </>
       <hide>              Children      </>
       <break-before>      None          </>
       <script>            ebt-attr(type) filename=”join(‘..\..\figures\’attr(ref))” scale=true</>
       <inline>            attr(type) filename=”join(‘..\..\figures\’attr(ref))” scale=true</>
</style>

<style name=”GRPHPRIM_NEW”>
       <foreground>        purple        </>
       <left-indent>       +=20          </>
       <first-indent>      -=20          </>
       <vertical-offset>   +=5           </>
       <space-before>      12            </>
       <space-after>       6             </>
       <icon-position>     Inline        </>
       <break-before>      None          </>
       <script>            ebt-attr(type) filename=”join(‘..\..\figures\’attr(ref))” scale=true</>
       <icon-type>         fig2          </>
</style>

<style name=”INPUT”>
       <space-before>      8             </>
</style>

<style name=”LINK”>
       <foreground>        blue          </>
       <score>             Under         </>
       <break-before>      None          </>
       <script>            ebt-link target=idmatch(ID,attr(linkends))</>
</style>

<style name=”NOTE”>
       <font-weight>       Demibold      </>
       <space-before>      4             </>
       <space-after>       4             </>
       <text-before>NOTE:</>
</style>

<style name=”PARA”>
       <space-before>      2             </>
       <break-before>      Line          </>
</style>

<style name=”RANDLIST”>
       <space-after>       4             </>
</style>

<style name=”RANDLIST,ITEM”>
       <font-family>       Symbol        </>
       <character-set>     Symbol        </>
       <left-indent>       +=20          </>
       <first-indent>      -=20          </>
       <space-before>      2             </>
       <break-before>      Line          </>
       <text-before>·</>
</style>

<style name=”RANDLIST,ITEM,PARA”>
       <font-family>       MS Sans Serif </>
       <character-set>     ISO8859       </>
       <break-before>      None          </>
       <break-after>       None          </>
</style>

<style name=”REQCOND”>
       <score>             Under         </>
       <space-before>      6             </>
       <text-before>REQUIRED CONDITIONS:</>
</style>

<style name=”REQCOND,STEP”>
</style>

<style name=”REQCOND,TEXT”>
```

```
       <score>             None         </>
       <break-before>      Line         </>
</style>

<style name=”SECT0”>
       <space-before>      16           </>
</style>

<style name=”SECT0,TITLE,TEXT”>
       <font-weight>       Demibold     </>
       <font-size>         16           </>
       <foreground>        dark red     </>
       <line-spacing>      16           </>
</style>

<style name=”SECT1”>
       <space-before>      8            </>
</style>

<style name=”SECT1,TITLE,TEXT”>
       <font-weight>       Demibold     </>
       <font-size>         14           </>
       <foreground>        dark red     </>
       <line-spacing>      14           </>
</style>

<style name=”SECT2”>
       <space-before>      8            </>
</style>

<style name=”SECT2,TITLE,TEXT”>
       <font-weight>       Demibold     </>
       <foreground>        dark red     </>
       <line-spacing>      14           </>
</style>

<style name=”SEQLIST”>
       <space-after>       4            </>
</style>

<style name=”SEQLIST,ITEM”>
       <left-indent>       +=20         </>
       <first-indent>      -=20         </>
       <space-before>      2            </>
       <break-before>      Line         </>
       <text-before>cnum().</>
</style>

<style name=”SEQLIST,ITEM,PARA”>
       <break-before>      None         </>
       <break-after>       None         </>
</style>

<style name=”STEP”>
       <left-indent>       +=20         </>
       <first-indent>      -=20         </>
       <space-before>      16           </>
       <break-before>      Line         </>
       <break-after>       Line         </>
       <text-before>CNUM()\.</>
</style>

<style name=”STEP-SEQ,TASK”>
       <space-before>      16           </>
</style>

<style name=”SYSTEM”>
       <line-spacing>      12           </>
       <space-before>      16           </>
       <break-before>      Line         </>
       <break-after>       Line         </>
</style>

<style name=”SYSTEM,LINK”>
       <space-before>      1            </>
       <inline>            raster filename=”join(‘..\..\figures\’attr(LINKENDS))” scale=true</>
</style>

<style name=”SYSTEM,TITLE,TEXT”>
       <font-size>         20           </>
       <line-spacing>      20           </>
       <space-after>       6            </>
       <justification>     Center       </>
       <hrule>             Surround     </>
       <vrule>             Surround     </>
       <break-before>      Line         </>
       <break-after>       Line         </>
```

```
</style>

<style name=”TABLE”>
       <space-before>      20           </>
       <space-after>       12           </>
       <icon-position>     Inline       </>
       <script>            ebt-reveal stylesheet=tables.v ROOT=TABLE</>
       <icon-type>         table3       </>
</style>

<style name=”TABLE,TITLE”>
       <foreground>        purple       </>
       <vertical-offset>   +=5          </>
       <justification>     Left         </>
       <break-before>      None         </>
</style>

<style name=”TASK”>
       <space-before>      32           </>
</style>

<style name=”TASK,TITLE”>
       <font-weight>       Demibold     </>
       <font-size>         14           </>
       <foreground>        dark blue    </>
       <text-before>PROCEDURE: </>
</style>

<style name=”TASK,TITLE,TEXT”>
       <break-before>      Line         </>
</style>

<style name=”TECHINFO”>
       <font-family>       MS Sans Serif</>
       <font-size>         12           </>
       <line-spacing>      12           </>
</style>

<style name=”TERM”>
       <foreground>        dark blue    </>
       <hrule>             Before       </>
</style>

<style name=”TEXT”>
       <break-before>      None         </>
       <select>            TEXT_Attr(Type)</>
</style>

<style name=”TEXT_”>
       <break-before>      None         </>
</style>

<style name=”TEXT_COMMENT”>
       <font-size>         10           </>
       <left-indent>       +=70         </>
       <first-indent>      -=54         </>
       <break-after>       Line         </>
       <text-before>COMMENT:</>
</style>

<style name=”TEXT_OPERATION”>
       <font-weight>       Demibold     </>
       <font-size>         10           </>
       <left-indent>       +=70         </>
       <first-indent>      -=54         </>
       <break-after>       Line         </>
       <text-before>ACTION:</>
</style>

<style name=”TEXT_RESULT”>
       <font-size>         10           </>
       <left-indent>       +=70         </>
       <first-indent>      -=54         </>
       <break-after>       Line         </>
       <text-before>RESULT:</>
</style>

<style name=”TGROUP”>
       <hide>              All          </>
</style>

<style name=”VERSION”>
       <font-size>         10           </>
       <foreground>        blue         </>
       <space-before>      2            </>
       <justification>     Right        </>
       <hide>              Children     </>
       <break-before>      Line          </>
       <script>            ebt-reveal stylesheet=fulltext.v</>
       <text-before>Copyright Notice</>
</style>

<style name=”VERSION,TEXT,TEXT”>
       <font-size>         12            </>
       <line-spacing>      12            </>
       <justification>     Center        </>
       <break-before>      Line          </>
</style>



</sheet>
```

8.3 Toc.tv

```
<!-- -->


<sheet >



<?INSTED COMMENT: UNGROUPED STYLES FOLLOW>

<style name=”#DEFAULT”>
       <break-before>      Line          </>
</style>

<style name=”#ROOT”>
       <break-before>      Line          </>
</style>

<style name=”#SDATA”>
       <font-family>       attr(font)    </>
       <font-weight>       Medium        </>
       <font-slant>        Roman         </>
       <character-set>     attr(charset) </>
       <break-before>      None          </>
       <text-before>char(attr(code))</>
</style>

<style name=”#TAGS”>
       <font-weight>       Bold          </>
</style>

<style name=”#TOC-ANNOT”>
       <font-family>                     </>
       <font-weight>       Bold          </>
       <foreground>        red           </>
       <line-spacing>      12            </>
</style>

<style name=”DESCINFO”>
       <title-tag>         TITLE,TEXT    </>
</style>

<style name=”DESCINFO,TITLE,TEXT” group=””>
       <font-weight>       Demibold      </>
       <font-size>         16            </>
       <foreground>        dark red      </>
       <left-indent>       +=10          </>
       <line-spacing>      16            </>
       <space-before>      8             </>
       <break-before>      Line          </>
       <break-after>       Line          </>
</style>

<style name=”SECT0”>
       <left-indent>       +=10          </>
       <title-tag>         TITLE,TEXT    </>
</style>

<style name=”SECT0,TITLE,TEXT”>
       <font-weight>       Bold          </>
       <font-size>         14            </>
       <foreground>        dark red      </>
       <left-indent>       +=10          </>
       <line-spacing>      14            </>
       <space-before>      6             </>
       <break-before>      Line          </>
       <break-after>       Line          </>
</style>
```

```
<style name=”SECT1”>
       <foreground>        dark red      </>
       <left-indent>       +=20          </>
       <title-tag>         title,text    </>
</style>

<style name=”SECT1,TITLE,TEXT”>
       <font-weight>       Demibold      </>
       <font-size>         12            </>
       <line-spacing>      12            </>
       <space-before>      4             </>
       <break-before>      Line          </>
       <break-after>       Line          </>
</style>

<style name=”SECT2”>
       <foreground>        dark red      </>
       <left-indent>       +=20          </>
       <title-tag>         title,text    </>
</style>

<style name=”SECT2,TITLE,TEXT”>
       <font-size>         12            </>
       <line-spacing>      12            </>
       <space-before>      2             </>
       <break-before>      Line          </>
       <break-after>       Line          </>
</style>

<style name=”SYSTEM”>
       <font-family>       MS Sans Serif </>
       <left-indent>       +=10          </>
       <title-tag>         TITLE,TEXT    </>
</style>

<style name=”SYSTEM,TITLE,TEXT”>
       <font-weight>       Demibold      </>
       <font-size>         16            </>
       <line-spacing>      16            </>
       <space-before>      12            </>
       <break-before>      Line          </>
       <break-after>       Line          </>
</style>

<style name=”SYSTEM,TITLE,TEXT,TEXT,LINK”>
       <hide>              All           </>
</style>



</sheet>
```

8.4 Tables.v

```
<!-- -->


<sheet >



<?INSTED COMMENT: UNGROUPED STYLES FOLLOW>

<style name=”#DEFAULT”>
       <break-before>      Line          </>
</style>

<style name=”#ROOT”>
       <break-before>      Line          </>
</style>

<style name=”#SDATA”>
       <font-family>       attr(font)    </>
       <font-weight>       Medium        </>
       <font-slant>        Roman         </>
       <character-set>     attr(charset) </>
       <break-before>      None          </>
       <text-before>char(attr(code))</>
</style>

<style name=”#TAGS”>
       <font-weight>       Bold          </>
</style>
```

```
<style name=”ENTRY”>
       <select>            ENTRY_attr(colnum)</>
</style>

<style name=”ENTRY,TEXT”>
       <left-indent>       +=2           </>
       <break-after>       Line          </>
</style>

<style name=”ENTRY,TEXT,TEXT”>
       <break-after>       Line          </>
</style>

<style name=”ENTRY_1”>
       <left-indent>       mult(tableinfo(cals, left-indent, 5),0.75)</>
       <width>             mult(tableinfo(cals, width, 5),0.75)</>
       <column>            True          </>
</style>

<style name=”ENTRY_2”>
       <left-indent>       mult(tableinfo(cals, left-indent, 5),0.75)</>
       <width>             mult(tableinfo(cals, width, 5),0.75)</>
       <column>            True          </>
</style>

<style name=”ENTRY_3”>
       <left-indent>       mult(tableinfo(cals, left-indent, 5),0.75)</>
       <width>             mult(tableinfo(cals, width, 5),0.75)</>
       <column>            True          </>
</style>

<style name=”ENTRY_4”>
       <left-indent>       mult(tableinfo(cals, left-indent, 5),0.75)</>
       <width>             mult(tableinfo(cals, width, 5),0.75)</>
       <column>            True          </>
</style>

<style name=”ENTRY_5”>
       <left-indent>       mult(tableinfo(cals, left-indent, 5),0.75)</>
       <width>             mult(tableinfo(cals, width, 5),0.75)</>
       <column>            True          </>
</style>

<style name=”ENTRY_6”>
       <left-indent>       mult(tableinfo(cals, left-indent, 5),0.75)</>
       <width>             mult(tableinfo(cals, width, 5),0.75)</>
       <column>            True          </>
</style>

<style name=”ENTRY_7”>
       <left-indent>       mult(tableinfo(cals, left-indent, 5),0.75)</>
       <width>             mult(tableinfo(cals, width, 5),0.75)</>
       <column>            True          </>
</style>

<style name=”ENTRY_8”>
       <left-indent>       mult(tableinfo(cals, left-indent, 5),0.75)</>
       <width>             mult(tableinfo(cals, width, 5),0.75)</>
       <column>            True          </>
</style>

<style name=”GRPHPRIM”>
       <select>            GRPHPRIM_Attr(Window)</>
</style>

<style name=”GRPHPRIM_CURRENT”>
       <foreground>        purple        </>
       <space-before>      12            </>
       <space-after>       12            </>
       <justification>     Center        </>
       <hide>              Children      </>
       <break-before>      None          </>
       <script>            ebt-attr(type) filename=”join(‘..\..\figures\’attr(ref))” scale=true</>
       <inline>            attr(type) filename=”join(‘..\..\figures\’attr(ref))” scale=true</>
</style>

<style name=”LINK”>
       <foreground>        blue          </>
       <score>             Under         </>
       <break-before>      None          </>
       <script>            ebt-link target=idmatch(ID,attr(linkends))</>
</style>

<style name=”ROW”>
       <space-after>       2             </>
       <hrule>             Surround      </>
       <vrule>             Children      </>
       <break-after>       Line          </>
</style>

<style name=”TABLE”>
       <font-family>       MS Sans Serif </>
       <space-before>      10            </>
       <space-after>       15            </>
       <hide>              Off           </>
       <break-after>       Line          </>
</style>

<style name=”TABLE,TITLE”>
       <font-weight>       Demibold      </>
       <font-size>         12            </>
       <foreground>        dark blue     </>
       <space-after>       8             </>
       <break-before>      Line          </>
       <break-after>       Line          </>
</style>

<style name=”TBODY”>
       <font-weight>       Medium        </>
</style>

<style name=”TECHINFO”>
       <hide>              All           </>
</style>

<style name=”THEAD”>
       <font-weight>       Bold          </>
</style>



</sheet>
```

A Example of use

This appendix describes a scenario for usage of the recommended MITD DTD along with the accompanying DynaText style views. However, a basic understanding of SGML ISO 8879 and organization of electronic documentation based on this standard is required. Also note that there are numerous ways of producing SGML documentation. This example show only one way of doing it.

In order to proceed with this example the following tools must be installed:

- FrameMaker+SGML from Adobe
- DynaText from Electronic Book Technologies
- Image/graphic tools that can create CGM, TIFF, and other desired formats

In this example we refer to files that are found on an accompanying distribution diskette that is included with this document. It can be obtained by contacting the project manager (Stian Ruud, DNV Research).

A.1 Creation of SGML document instances

A.1.1 Importing DTD into Frame+SGML

Frame+SGML does not use the DTD directly. Instead, DTDs are imported into Frame+SGML, which converts it to an internal representation called EDD (Element Definition Document). This EDD sets up the same rules for structure and attributes as the DTD itself. In addition, it associates traditional Frame styles to DTD elements. In this manner, writers will be able to distinguish elements from each other based on Frame formatting styles.

This is the first preparatory step in bringing documentation on SGML form. Next, we have two choices: Either new documentation will be created from scratch, or we have to convert existing documents written in FrameMaker or other text processing tools to Frame+SGML.

The recommended MITD DTD file is located in *A:\mitd\dtd\mitd.dtd* on the distribution diskette.

1. Import the DTD and save the EDD in a new file
2. Create the desired format rules in the EDD.

A.1.2 Writing from scratch

With the EDD installed, Frame is the natural choice for the writer. This way, he or she will be given the benefits of the SGML structuring rules making sure that the new documentation follows the defined structure.

Documents can be saved in intermediate form, even not conforming to the DTD. However the user is warned where and how the documentation does not conform to the DTD.

Another important benefit, is handling of tables. The recommended MITD DTD defines a table structure which is automatically handled by Frame+SGML and DynaText, saving lots of work with style configuration.

1. Create a new document.
2. Import the element definitions from the EDD: *File -Import -Element Definitions*
3. Write document
4. Save the new document as a Framemaker+SGML document.
5. Open the Structure View and the Element List and you are ready to start.
6. Set SGML application file: *File - Set SGML Application*.
7. Save as SGML. Remember to choose SGML file format and extension “sgm”.

A.1.3 Importing existing document into Frame+SGML

Existing documentation will most likely exist in various formats, like Word, FrameMaker, Interleaf, WordPerfect etc. The first task is to import documents into Frame, which is catered for by a collection of conversion filters delivered with Frame.

Next, a so-called conversion table will associate different parts of the Frame document with elements from the recommended MITD DTD. Some manual intervention will probably be necessary in order to make the conversion table complete, but Frame does its best to give a good starting point for customizing.

1. Open the unstructured document. It is assumed that paragraph formats are used in the document.
2. Choose *File - Developer Tools - Generate Conversion Table*.
3. Save the new document.
4. Edit the conversion table in order to wrap the paragraph formats into the desired SGML elements.
5. Activate the unstructured document (leaving the conversion table open).
6. Activate the *File - Utilities - Structure Current Document* function to commence conversion.
7. Open the EDD file created in the first step, and import the element definitions: *File -Import -Element Definitions*

A.2 Pressing DynaText books

In order to prepare a SGML file for DynaText presentation, we need the DTD and a set of styles specifying how each element in the DTD will look in DynaText. This is the same principle as used in Frame+SGML format rules, but these tools use different formats for the specification.

A set of style views for DynaText are delivered with the distribution diskette, making the production simpler. These files are found in the directory A:\mitd\styles\.

This example presents a very simple way from SGML to readily viewable DynaText documents. However, DynaText is a versatile tool with lots of possibilities. The on-line manual is the best place to consult for a thorough covering av the aspects and possibilities of DynaText publishing.

A.2.1 File structure

DynaText requires the source files used in production, to be structured into a hierarchy.

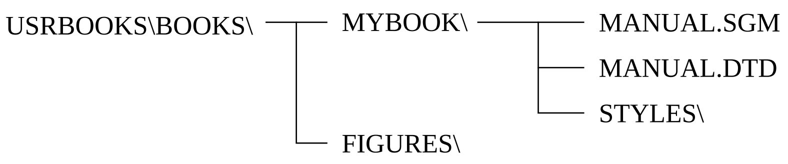


FIGURE 15. DynaText file organization for manual.

The figure above shows how a manual is entered into a directory called MYBOOK. Both the SGML file and the DTD file must be in the same directory. The style views in the STYLE directory are also placed in the directory with the SGML and DTD files. Finally, figures are placed into the FIGURES directory.

A.2.2 Run the mkbook command

You are now ready to run the *mkbook* command. This is a DOS command which must be run from a DOS shell command line window.

It transforms the SGML document into a form viewable by the DynaText browser, along with index files for fast searching. The viewing format is handled without any intervention since the STYLES directory is in place with all necessary style views.

A.2.3 Run DynaText browser

Open the new book in the browser. It will be presented according to the STYLES files.

A.2.4 Customizing styles

The tool InStEd is a style view editor, allowing customizing of style sheets. This step is not necessary if you want to use the delivered styles unchanged.

But if you want to bring special effects or another style which is more distinct for your company, you are free to change the styles.

List of Entities

List of Elements

List of Figures