ThML: Theological Markup Language

For the Christian Classics Ethereal Library

Harry Plantinga

Version 1.02, February 23, 2001

Abstract

This document describes the *Theological Markup Language* (ThML), an XML markup language for theological texts. ThML was developed for use in the Christian Classics Ethereal Library (CCEL), but it is hoped that the language will serve as a royalty-free format for theological texts in other applications. Key design goals are that the language should be (1) rich enough to represent information needed for digital libraries and for theological study involving multiple, related texts, including cross-reference, synchronization, indexing, and scripture references, (2) based on XML and usable with World Wide Web tools, (3) automatically convertible to other common formats, and (4) easy to learn and use. ThML is defined as an XML DTD that extends the Voyager DTD for HTML.

Introduction

The study of theology involves uses of texts that are infrequent in other areas of study. Theological books usually make many references to the Bible, including quotations, commentary, explanations, and citations; special processing for scripture references can aid study. Theological study also often involves texts available in multiple variations or translations, sometimes in Greek or Hebrew, which may have to be synchronized and displayed in parallel columns. It may involve hymns and multiple media. It involves the use of cross-reference systems such as Strongs numbers, various sorts of indexes, and the synchronization of multiple texts in various ways, as for example layers of commentary on a text. Theological study often makes use of several texts related by subject or scripture reference, and tools that support library-wide searching by subject, scripture reference, name, or date would also be desirable. It should be possible for students to relate, combine, or comment on parts of texts in their own document, without altering the originals. A markup language for theological texts should support these applications.

For digital libraries, bibliographic information similar to that stored in a library card catalog should also be represented in the text. This information can be loaded into a software system similar to a library's Public Access Catalog (PAC) to provide an interface for locating books, though unlike a traditional library, searches can also be based on the *contents* of books. This information should be stored in a standard format so that it can be exported to other PACs and search engines, and searches can look through Internet-based digital libraries and PACs as well as documents on the local computer.

Multiple formats for electronic resources are a reality in today's computing environment. Theological study would also benefit from a markup language that is formally defined, with software for converting to and from other formats. And the language should be royalty free, so that texts can be prepared and distributed by small publishers, scholars, students, and other interested parties as well as large publishers. Ideally, no special purpose software for theological study should be required. This would be possible if all the desired applications could be supported by a web browser, for example. As a result, users would not be "locked in" to a particular proprietary software system. And the language should integrate with the World Wide Web, so that public domain texts can be downloaded from on-line libraries such as the CCEL. It should not be obvious to the user whether a text is on a CD-ROM, the local hard drive, or elsewhere on the Internet, except perhaps by access time.

Existing markup languages do not meet all of these needs. Word processor formats don't represent semantic information about a text—an area in which HTML is also weak. A lack of semantic information makes searching, indexing, and converting to other formats and using for other purposes more difficult. The Text Encoding Initiative (TEI)[1] application of SGML is semantically rich for literary analysis but not easy to learn or tuned for theological study. It doesn't offer special handling of scripture references or Strongs-like reference systems, for example. Also, the language is very large and the overhead required to learn and process the language is high. Commercial formats, including STEP[2] and the Logos Library System (LLS)[3], are not designed for integration with the Internet, and preparing texts for these systems requires expensive software, beyond the means of most individuals. Publication in one of these formats may also be controlled by the company or consortium in question. As a result, few public-domain or on-line texts are available in these formats.

This paper describes the *Theological Markup Language*, or ThML, which is a markup language for theological texts designed for use in the Christian Classics Ethereal Library (CCEL), a library of classic Christian texts on the Internet.[4] ThML is designed to meet the requirements outlined above. It borrows some elements from TEI, and it is designed to have all of the capabilities of STEP (version 0.9). ThML is an XML application, so ThML documents can be used directly with next-generation web browsers. These are programmable, so stylesheets and scripting languages will be able to add functionality such as that described above as well as format for display or printing on various devices. It will also be possible to convert ThML documents to other formats such as HTML webs, plain text files, PDF[5], RTF, and others.[6] ThML documents can be prepared with an XML editor, a plain text editor, or in Microsoft Word using a template and converting to ThML automatically. ThML resources are indexed on the ThML web page, http://www.ccel.org/ThML/.

After a couple of years' experience with a digital library and thought about the information that needs to be represented in theological texts, the first version of ThML was designed in the summer of 1998, and it has been undergoing scrutiny and improvement since then. This paper describes version 1.0 of the language. Preliminary tools exist for converting Microsoft Word documents to ThML and for converting ThML documents to HTML webs. One open source software project (OpenBible) for Unix is using ThML and a couple of other groups are considering its use. There is a mailing list for those interested in defining and using the language. Several ThML documents exist and have been validated, and others are underway, including Schaff's *History of the Christian Church* and *Calvin's Commentaries*.

Designing a Theological Markup Language

A language for theological study must handle the markup of text into headings, paragraphs, block quotes, emphasized text, and other basic structural elements that are common to all books and can be represented in a markup language such as HTML. Markup needs for theological study that go beyond these basics include the special handling of scripture references, numbering and synchronization schemes such as Strongs numbers, handling multiple versions or translations of the same text, footnotes, index entries, lexicons, and representing page breaks in the original text.

For the use of digital libraries, bibliographic data about the text should be represented. Markup of subject index entries, scripture references and commentary, names, citations, and dates can be used to build library-wide indexes of those items and assist searching. The original pagination of the document

---

[1] See http://www.uic.edu/orgs/tei/

[2] The Standard Template for Electronic Publishing (STEP) format was developed by members of the Bible Software Industry Standards Group (BSISG, http://www.bsisg.com), including Parsons Technology. It is a multi-vendor standard for Bible software.

[3] The format used by Logos, Inc. (and licensees) for Bible and theological book software.

[4] It is available at http://www.ccel.org.

[5] Portable Document Format, used by Adobe Acrobat software.

[6] Handheld computers may provide a better user interface than a desktop or laptop computer for reading books. Book-reading software and formats tuned for these devices will doubtless become common, if web browsers don't meet the need.

should be represented for bibliographic references. Also, it should be possible to specify a "chunking" of the document for delivery over a low-bandwidth connections such as modems.

Since the primary means of delivering digital libraries such as the CCEL are the world wide web and CD-ROM, the language should use web-based technology, including XML and Unicode, and be usable with web browsers. Therefore, the design goals for ThML are these:

- It should be usable with the world wide web and web-tool-based delivery on CD-ROM
- It should represent information useful for theological study of multiple, related texts, including cross-reference, synchronization, indexing, and handling scripture references.
- It should be rich enough to represent all information needed for conversion to other formats commonly used for electronic publication of books
- It be rich enough to support high quality printing
- Subject to these constraints, it should be as easy as possible to learn and use.

Document repurposing is much more easily achieved when purely semantic markup is used and description of appearance is restricted to stylesheets. This approach gives several benefits: it makes documents more flexible, allowing different stylesheets for the web, typesetting, large-type editions, etc. Mark-up editors are forced to think more about the semantics of the markup, rather than just trying to "make it look right" for a particular use. ThML makes possible purely semantic markup, with a level of semantic representation appropriate for digital libraries and theological study (though not for heavy literary analysis, as TEI). However, since the language is based on HTML, it is also possible to use it as presentational markup. The advantage of this approach is flexibility and familiarity; the disadvantage is that editors must be aware of the issues in order to produce good documents.

Theological Markup Language

Based on XML and HTML

Since Theological Markup Language is based on HTML and XML, it supports all of the markup of HTML, a rich linking language in XLink, and stylesheet support in CSS and XSL. HTML may be used for markup of emphasis, paragraphs, headings, lists, tables, block quotes, images and multimedia, scripts, etc. Links may make use of the extended pointer and link types associated with XML, and formatting will be specified in XSL. These facilities will be used wherever possible, to make the language easier to learn for those who already use HTML and easier to use with the World Wide Web.

Theological Markup Language is defined as a set of extensions to the Voyager Strict DTD of HTML. The DTD is presented in Appendix A and available from the ThML web page, http://www.ccel.org/ThML/dtd/ThML10.dtd. Since the Voyager DTD for HTML is an XML application, there are certain restrictions on the HTML that can be used. Most importantly, all element and attribute names must be lower case, all elements properly nested, and XML-style empty tags used (e.g. `<hr/>`).

Document Structure

ThML documents are contained in one global ThML element, and like HTML documents, they contain a head and a body section.

```
<ThML>
   <ThML.head> … </ThML.head>
   <ThML.body> … </ThML.body>
</ThML>
```

The `ThML.body` element contains all of the contents of the print edition of the book on which the text is based. The `ThML.head` element contains bibliographic and meta information about the text, both the electronic publication and the print edition upon which it is based, if any. The information in this section

is not generally a part of the original book. It may include bibliographic data, keywords, information about the program used to generate the text, etc.

Divisions of the Text

Structural divisions in the body of the text are marked with <div*n*> tags as in this example:

```
<div1 type="Book" title="Admonitions on Things Internal" n="2">
<div2 type="Chapter" n="1" title="Of the Inward Life">
<div3 type="Section" n="I">
</div3> </div2> </div1>
```

<div1> is used for top level parts of a text, including the title page, preface, table of contents, chapters, index, etc. Additional levels are used for lesser structural divisions of the document. These structural divisions show the structure of the original text, and they are also used to prepare a table of contents and allow splitting of a text info files and access to a text by section. The optional `title` attribute is used in constructing a table of contents and may be used for running heads or other identification purposes. The optional `type` and `n` attributes may be used to specify the type and number of the division. If they are present, they will also be used to identify the section in the table of contents.

In programs that process ThML files for the CCEL, the <div*n*> elements are used to generate a table of contents and to split a book into multiple files. By default, a new file will be started with each new <div*n*> element, unless the resulting file would be too small. This "minimum desired file size" can be set with the `filebreak="nnn"` attribute. Thus, `filebreak="0"` causes a break for each div and `filebreak="10000000"` prevents splitting the document at that point (at least for documents of less than 10000000 bytes length).

HTML <p> elements are used for normal paragraphs. They (and all other body elements) may be formatted with `class` attributes and stylesheets, as they would in HTML. <h1> through <h6> elements may be used for headings of varying levels of importance. Also, ThML adds an <argument> element, which may be used for a chapter or section subhead or topic of discussion.

Page Breaks

It is often useful to know the page breaks from the print edition of a book. They may be used as targets for subject index entries identified by page number or to display a text with the pagination of the print edition. Page breaks are marked by the insertion of <pb /> tags, with the `n` attribute giving the page number of the upcoming page (<pb n="37" /> or <pb n="xii"/>). These elements should appear at the *start* of the identified page. Many electronic texts will also have images of pages available on line. The `pb` element will also take an href attribute specifying a URI for an image of the page (<pb n="37" href="gif/0021a.gif" />).

Scripture References

In theological texts, scripture passages may be cited, quoted, or explained. Citations refer to a passage, and they are represented in ThML with the <scripRef> element. That element specifies that the enclosed text is a scripture reference. A ThML processor might link the text to the passage and include it in a scripture reference index, for example. Quoted scripture is marked with the <scripture> element. Commentary on scripture, sermons or hymns based on scripture, and other material that would be useful to someone studying a passage is marked with the <scripCom/> element. In addition, a <scripContext> element is available to set the default version, book, or chapter for succeeding references. All four elements may optionally have the attributes `version`, `passage`, and `parsed`, containing the version of the references, the passage referred to, and a machine-friendly parsed version of the passage.

A reference such as "Romans viii. 27,28; x. 8-13" for the NIV could be marked in a number of ways. A full specification would appear as

```
<scripRef version="NIV" passage="Rom. viii. 27,28; x. 8-13"
parsed="NIV|Romans|8|27|8|28;NIV|Romans|10|8|10|13">Romans viii. 27,28; x. 8-
13</scripRef>
```

or perhaps

```
<scripRef version="NIV" passage="Rom. viii. 27,28"
parsed="NIV|Romans|8|27|8|28">Romans viii. 27,28</scripRef>; <scripRef
version="NIV" passage="x. 8-13" parsed="NIV|Romans|10|8|10|13">x. 8-
13</scripRef>
```

However, a number of shortcuts are possible. If a `parsed` attribute is not used, processing software can parse the passage and add it. The `version` can be inferred from the most recent `<scripContext>` element or left unspecified. The passage attribute can even be omitted when the enclosed text is an unambiguous scripture references. So, the above passage could have been marked as

```
<scripRef>Romans viii. 27,28; x. 8-13</scripRef>
```

The passage attribute is a list of scripture references separated by commas or semicolons. Each reference may consist of a book name (or abbreviation), a chapter, and a verse. The chapter and verse are separated by a semicolon or period. If two references are separated by a dash, all of the intermediary verses are included as well. In the case of books with only one chapter, a reference consists of a book name or abbreviation and a verse. Book names should be as they appear in the version in use or a unique prefix of at least two letters of the name. Abbreviations that are not prefixes may also be accepted by programs that process ThML documents.[7]

The parsed attribute consists of a number of references separated by semicolons. Each reference includes the version, book, from chapter, from verse, to chapter, and to verse separated by the vertical bar ('|') character. The version may be omitted. If the from chapter is zero, the whole book is identified. If the from verse is zero, the whole chapter is specified. If the to chapter and verse are zero, a single verse (from chapter:from verse) is specified.

Note that different translations of the Bible use different versification schemes. For example, Psalm 9 in the King James Version is split into two—Psalms 9 and 10—in the Septuagint. In order to interpret a reference, the versification scheme used must be known. Scripture references will be assumed to be compatible with the versification scheme used by the KJV, ASV, NASB, NIV, and TLB unless otherwise specified. However, specifying that the scripture version for a passage is the Septuagint, for example, implies that the references is based on the Septuagint scheme. References to the Apocrypha without a version specified are interpreted according to the scheme used in the NRSVA (New Revised Standard Version with Apocrypha).

Also, context is sometimes necessary in interpreting a reference. A passage may refer to Romans 8:28 at one point and later to verses 29 and 30 and chapter 10:8-13. It will be up to the editor to identify and mark scripture references, although software may be provided to identify possible scripture references and suggest an interpretation such as `<scripRef passage="Rom. 8:29,30">`verses 29 and 30`</scripRef>`. The `<scripContext version="NIV" passage="Romans 8" />` element may be used to set the default version, book, or chapter for upcoming references to the parser.

Quotations of scripture are marked with the `<scripture>` element. A passage may be represented as in this example:

```
<scripture passage="Mark 7:16" version="NKJV">If anyone has ears to hear, let them
hear!</scripture>
```

This markup may also be used for a translation or version of a book or a whole Bible, perhaps as in the example below. Scripture marked in this way could be automatically retrieved by book, chapter, and verse with an appropriate program.

---

[7] The section marking scripture references draws from the *STEP Programming Specification* and the *STEP Publishing Specification* (available at http://www.bsisg.com).

```
<scripContext version="Calvin's Translation, in English" /> …
<scripContext passage="Romans 8" /> …
<scripture passage="28">We further know, that to those who love God all things co-operate
for good, even to those who are called according to <I>his</I> purpose:</scripture>
<scripture passage="29">for those whom he has foreknown, he has also predetermined to be
conformed to the image of his Son, that he might be the first born among many
brethren;</scripture>
```

Explanation or commentary on a passage involves a semantic relationship between the explanation and the passage explained. This relationship should be represented in the text in order to be able to build an index of scripture commentary. For example, it would be useful to be able to see everything the early church fathers said or preached on a passage. Commentary or explanation of a passage will marked with a `<scripCom/>` element, as in this example:

```
<scripCom passage="Mark 7:16" />Mark 7:16. This admonition seems to apply to most
everyone . . .
```

The `<scripCom/>` element also takes an optional `type` attribute. The `type` is used in the CCEL for classifying references into one of a number of fixed types, including Commentary, Treatises, Sermons, Hymns, Prayers, Drama, etc. These then appear in a classified list of references for the whole library.

Cross Referencing Schemes and Synchronization

Cross referencing is the specification and linking of related passages in separate texts. Cross referencing in theological texts takes many forms. They include simple links such as those that can be handled by an HTML anchor; numeric or symbolic indexing schemes such as dates, Strongs numbers, scripture references, or subject index entries; annotation such as footnotes and commentary; different translations of the same text, etc. In this section we will define markup for handling symbolic cross-referencing schemes other than those that can be handled as ordinary links, scripture references, or annotation.

Standardized symbolic cross reference schemes such as dates, keywords, or Strongs numbers aren't really links to other documents, because any two documents with compatible cross reference schemes can be linked together and no particular documents are specified. Therefore XLL links, element IDs, etc. don't capture the semantics of such information. We will define a new `sync` element to represent this information. For example, the element

```
<sync type="Strongs" value="G42" />
```

might be used to represent a Strongs number at a location in a text.

Software tools may be provided to use this information in a variety of ways. For example, a program would be able to find other passages on related topics or create an index using the Strongs numbering. Multiple different manuscripts of the same original text could be aligned this way, and displayed in parallel columns, with appropriate software.

The scheme name given in the `type` attribute are not pre-defined; applications may invent new synchronization types for specific purposes. For example, the *Rule of Benedict* is available in several different manuscripts in two different traditions. If a common synchronization scheme were defined and manuscripts marked up, any two or more could be selected and aligned as parallel columns on the screen, or alternate forms of a passage could be located.

Annotation

In a ThML document, footnotes, endnotes, etc. are all marked with the `<note>` tag, following the syntax used by TEI Lite[8] for the most part. The note element may take the following attributes: `place`, `resp`, `target`, `targetEnd`, and `anchored`. The `place` attribute specifies how it appears in the text (e.g. end, foot, inline, or margin). The `target` (and `targetEnd`) attributes refer to the start (and end) of the text

[8] See http://www_tei.uic.edu/orgs/tei/lite/.

being annotated, if the note does not occur in the text at its reference point. These attributes allow the notes to be gathered at the end of a chapter or file if desired. The `resp` attribute identifies the person responsible for the note—for example, the author, editor, or a person's initials. The `anchored` attribute make take the value `yes` (default) or `no`, specifying whether the note is anchored at an exact location; margin notes typically are not anchored.

The `<note>` element can also be used to store commentary, margin scrawls, and the like for a text in a separate file. In that case, the `target` and `targetEnd` attributes would be references to a point in another document.

Foreign Languages

The primary language for a document is specified in the header. Passages in other languages may be marked with the `foreign` tag and the `lang` attribute. For example, the Greek passage `<foreign lang="el">`logos`</foreign>` may be marked as shown. "`lang`" attribute values are as specified in ISO 639.[9] Some examples are Dutch: nl, English: en, French: fr, German: de, Greek: el, Hebrew: he, Latin: la, Spanish: es, Portuguese: pt, Russian: ru. Note that the `lang` attribute may be used with most elements. For example, to specify that a scripture reference passage is given in German, one could use `<scripRef lang="de">`…`</scripRef>`.

If the language uses characters not available in the ISO-8859-1 (Latin-1) character set, they may be represented in Unicode, as in this Greek example (ëïãïò) and this Hebrew example (äìäé), or using the Latin-1 character set and a suitable font, for example, `<foreign lang="el" style="Font-family: SIL Galatia">`λογος`</foreign>`. The Greek and Hebrew fonts recommended for use with the CCEL are the freeware SIL Galatia and SIL Ezra fonts and related software from the Summer Institute of Linguistics[10], used here in a Greek example (λογος) and a Hebrew example (יהוה). Foreign languages represented with special fonts may eventually be converted to Unicode programmatically.

Attributions, Citations, Dates, Names, and Unclear text

Attributions of authors of poetry, letters, etc. may be marked with the `<attr>` element. Also, names may be marked with the `<name>` element. A different representation of the name for the index may be specified with the `title` attribute: `<name title="Ter Steegen, Gerhard">`G. Ter Steegen`</name>`. (The `title` attribute may be used with any element; it is displayed in index entries and tables. In web-based presentation of a book, it may also be presented as a "tool-tip".) The `<unclear>` element may be used to mark words or passages that could not be transcribed with certainty or passages that a proofreader judges dubious.

Citations of other works such as books or treatises may be marked with the `<citation>` element. That element may also take an `href` attribute to specify a URI for the cited work, if available. The `<date>` element may be used to mark dates that occur in the text. A `value` attribute may be used to specify the date in ISO format, as in this example: `<date value="1997.12.25">`last Christmas`</date>`. These elements are used in the CCEL to aid searching and indexing: the insertIndex element (described in the next section) may be used to insert an index of attributions, works cited, dates, or personal names, and searches for names, attributions, dates, etc. can also be supported.

Verse

Theological books often contain verse—poetry, hymns, or versified presentation of material such as the Psalms. A stanza, verse, or other unit of verse is encoded in a `<verse>` element. Lines are marked with `<l>` elements. Lines that are formatted with varying levels of indentation may be formatted with the class attribute, as in the example below.

---

[9] See http://www.sil.org/sgml/iso639a.html.

[10] Available from SIL at http://www.sil.org/computing/fonts/silgreek/ and http://www.sil.org/computing/fonts/silhebrew/

```
<verse>
         <l class="t1">O God, a world of empty show,</l>
            <l class="t2">Dark wilds of restless, fruitless quest</l2>
         <l class="t1">Lie round me wheresoe'er I go: </l>
               <l class="t3">Within, with Thee, is rest.</l3>
</verse><verse>
         <l class="t1">And sated with the weary sum</l>
            <l class="t2">Of all men think, and hear, and see, </l2>
         <l class="t1">O more than mother's heart, I come, </l>
               <l class="t3">A tired child to Thee. </l3>
</verse><verse>
         <l class="t1">Sweet childhood of eternal life! </l>
            <l class="t2">Whilst troubled days and years go by, </l2>
         <l class="t1">In stillness hushed from stir and strife, </l>
               <l class="t3">Within Thine Arms I lie. </l3>
</verse><verse>
         <l class="t1">Thine Arms, to whom I turn and cling</l>
            <l class="t2">With thirsting soul that longs for Thee; </l2>
         <l class="t1">As rain that makes the pastures sing, </l>
               <l class="t3">Art Thou, my God, to me. </l3>
</verse>
                                    <attr><name>G. Ter Steegen</name></attr>
```

Hymns

ThML also has support for hymns and hymnals. The `<hymn>` element may contain elements `<meter>`, `<author>`, `<tune>`, `<composer>`, `<incipit/>`, and `<music>`, in any number and order, as well as other inline elements such as paragraphs, headings, `<verse>`, etc. The `<author>` and `<composer>` elements take `authorID` and `composerID` attributes, which may contain a standard identifier such as the CCEL personIDs, as well as the `type` attribute. The tune element contains the tune name; it may also have a `tuneID` attribute containing the CCEL `tuneID` for the tune.

The `<meter>` element contains the meter as represented in the hymnal. It may also take the `standard` attribute, giving the meter in a standardized form. The `<incipit/>` element may be used to represent the first line of the melody as it would be if transposed to C, using f and s for accidentals, e.g.

```
<incipit value="CEfGGGFAGFEf" />
```

Finally, the `<music/>` element gives a link to an electronic format for the music, perhaps a PDF page image or a midi file. The element takes the `href` attribute, giving the target, and the `type` attribute, giving the MIME type of the target. It may also take the inline attribute, with the value yes or no, and the actuate attribute, with the value auto or man. For images, specifying `inline="yes"` loads the image onto the page; otherwise it is loaded in a separate window. Specifying `actuate="auto"` makes the image appear automatically; otherwise it appears when a link is clicked. For audio files, specifying `inline="yes"` brings a player control onto the page. So, for example, to specify that a midi file be played automatically when a page is loaded, and that a player control be displayed, one could use

```
<music href="hymn.midi" type="application/midi" inline="yes" actuate="auto"/>
```

Index Entries and Indexes

A point in the text may be marked for insertion into an index using the `<index>` element. For example, one might mark a passage for inclusion in a subject index this way:

```
<index type="subject" subject1="Christian Life"
subject2="Sanctification" title="Apotheosis"/>
```
Apotheosis (or Deification) is an ancient theological word commonly used in Eastern theology to describe the process by which a Christian becomes more like God.

The `title` attribute is used as the identifier in the Table of Contents. The `type` attribute is used to identify the index that this reference is to be added to; values used for the CCEL include `subject` (the subject index for the book) and `globalSubject` (the library-wide subject index). Other values may be used for specialized indexes.

Up to four levels of subjects may be specified. The terms used for the `subject1`, `subject2`, `subject3`, and `subject4` attributes may be chosen by the editor or selected from a fixed set, as determined by the application. The CCEL will use the Library of Congress classification scheme, but editors may also choose subject entries that seem appropriate; these will then be translated into LCCS subjects. An optional `subjectID` attributed may be used in place of the `subject1`…`subject4` attributes; it specifies a code for a set of subjects. For the CCEL, the `subjectID` value would be a Library of Congress call number, such as BV4138.

A document may have several user-selected types of index entries. An XML element (`<insertIndex type="subject" />`) is also provided to specify that a sorted, hierarchical index of all the "subject" (e.g.) index entries should be inserted at that point, with links to the appropriate locations in the text. Certain additional index types are also understood: `<insertIndex type="name" />` inserts an index of all names marked with the `<name>` element; if the `title` attribute is present, it is used as the index entry. Similarly, indexes may be inserted for citations, dates, foreign words and phrases, images (`<img>`), names, scripture references (`<scripRef>`), and scripture commentary (`<scripCom>`).

Terms, Definitions, and Glossaries

Some documents contain one or more glossaries. Glossaries may be marked up with the `<glossary>` element and `<term>` and `<def>` elements. The `glossary` element may take an optional `type` attribute, to specify how the glossary may be used.

```
<glossary>
   <term>Apotheosis</term>
   <def>An ancient theological word used to describe the process by which a Christian becomes
   more like God</def>
</glossary>
```

Software tools may be provided for composing two documents (which may be the same), using the glossary in one and the text in another. Words of the text defined in the glossary could be footnoted, underlined and linked, or defined in a separate window.

Additions and Deletions

The `<ThML.body>` section of the electronic text should have all of the contents of the print edition. However, for display purposes, it may be desirable to add to or delete from the print edition. For example, it may be desirable to delete the original subject index and add one that is generated automatically. The `<added>` element is used to mark sections that have been added and do not appear in the print edition, and the `<deleted>` element is used to mark the sections from the print edition that should not appear when the book is presented, even though they remain in the XML for those who want to see exactly how the print edition looked. For example, a subject index might be marked this way:

```
<added>
   <H1>Subject Index</H1>
   <insertIndex type="subject" />
</added>
<deleted>
   [original subject index here]
```

```
   </deleted>
```

The `added` and `deleted` elements take the optional attributes `resp`, which identifies the person responsible for the changes; `reason`, identifying the reason for the change, and `date`, the time of the change.

Header Information

The head section of a ThML etext has the information *about* the text, information which does not necessarily appear in the text. It consists of a Dublin Core bibliographic record for the document and a few other elements that are useful for digital libraries. The `<ThML.head>` element has the most detailed (and least frequently used) markup; in practice it will normally be generated from a template. A sample template may be found in Appendix B. Note that a few items of information are duplicated between the ThML.head elements and the Dublin Core bibliographic record. This is the case because ThML requires these elements to be present in the header, but all elements in the Dublin Core are optional.

The head section may start with some HTML elements, such as `<title>` and `<meta/>`. There follow three sections that are unique to ThML: `<generalInfo>`, `<electronicEdInfo>` and `<printSourceInfo>`.

The `<generalInfo>` section contains information about the text that is not specific to the electronic edition or the print edition on which it is based. The section is optional, as are its four elements:

```
<generalInfo>
   <description>          Textual description of book
   <firstPublished>       Date first published in any edition
   <pubHistory>           Any available information on the publication history
   <comments>             Any other comments
</generalInfo>
```

The optional printSourceInfo section contains information specific to the print source from which the electronic text was derived, if there is one. The elements it may contain are these:

```
<printSourceInfo>
   <published>            Publication information (required)
   <copyLocation>         E.g. Buswell library, 231.4 b29h c.2.
   <image type="…" href="…"/>  Electronic image of the book
</printSourceInfo>
```

The electronicEdInfo section contains information specific to the electronic edition, such as publication information, editorial practices and status, etc. It may contain the following elements (required elements are marked with *):

```
<electronicEdInfo>
   <publisherID>          *Publisher code of electronic edition, as assigned by the CCEL, e.g. ccel
   <authorID>             *Author ID as assigned by publisher
   <bookID>               *Book ID as assigned by publisher
   <version>              *Edition or version of electronic edition, e.g. 1.1
   <editorialComments>    Comments about editorial practices: whether spelling was normalized,
                          what was done with end-of-line hyphens, corrections that were made,
                          tagging practices, etc.
   <revisionHistory>      A list of published editions and changes between them
   <status>               Current status of text—e.g. This text still needs proofreading
   <DC>                   *DC record for electronic edition
   <comments>             Other comments
</electronicEdInfo>
```

The Dublin Core (DC) element contains any combination of the 14 Dublin Core elements. All are optional and may be repeated in any order. The semantics are as described in the Dublin Core documentation[11], with a few additions. The elements are as follows:

```
<DC.Title>        <DC.Creator>      <DC.Subject>     <DC.Description>
<DC.Publisher>    <DC.Contributor>  <DC.Date>        <DC.Format>
<DC.Identifier>   <DC.Source>       <DC.Language>    <DC.Relation>
<DC.Coverage>     <DC.Rights>
```

These elements all have three optional attributes: `lang`, `scheme`, and `sub`. The `lang` attribute is the language in an ISO639-1 representation. The `scheme` attribute identifies the representational scheme. in The `sub` element represents subtypes, such as the Created subtype of the Date element. So, for example, the publication date might be represented as

```
<DC.Date sub="Created" scheme="ISO8601">1999-12-23</DC.Date>
```

Special usages for the CCEL include the following:

`<DC.Creator>`Thomas, à Kempis, 1380-1471`</DC.Creator>`
- This is the unified form of the name

`<DC.Creator scheme="CCEL">`kempis`</DC.Creator>`
- This is the CCEL personID for the person

`<DC.Subject scheme="LCCN">`BV4821.A1 1949`</DC.Subject>`
`<DC.Subject scheme="LCSH">`Meditations.`</DC.Subject>`
- For the CCEL, the Library of Congress Call Number and Subject Heads should be included.

`<DC.Subject scheme="CCEL">`Classic; Recommended`</DC.Subject>`
Some special keywords (separated by semicolons) control the use of the document in the CCEL:
Biography – it is a biography of a person, who is identified as shown below
Classic – include on list of classics
Fiction – a work of fiction
Non-Fiction – [Default]
Recommended – include on list of recommended readings
Hymns – include on list of hymns and hymnology
Reference – include on list of reference items (e.g. dictionaries, encyclopedias, commentaries)

`<DC.Subject scheme="target">`anselm`</DC.Creator>`
- The target of a biography is identified by the CCEL personID

`<DC.Description>` In preparing this edition of The Imitation of Christ, the aim was to achieve a simple, readable text which would ring true to those who are already lovers of this incomparable book and would attract others to it.`</DC.Description>`
- The description should be a couple of sentences or a paragraph. In the CCEL, it will be used in indexes and links to the book.

`<DC.Contributor sub="Transcriber" scheme="CCEL">`whp`</DC.Contributor>`
- info about the transcriber, typist, or person responsible for markup can be represented here
`<DC.Identifier`
`scheme="CCEL">`http://www.ccel.org/ccel/kempis/sample/1.15.htm`</DC.Identifier>`
- CCEL standard identifier, as described below
`<DC.Identifier`
`scheme="URL">`http://www.ccel.org/ThML/sample/About.htm`</DC.Identifier>`
`<DC.Source sub="ElectronicEdition">`Project Gutenberg`</DC.Source>`

[11] http://purl.org/dc/

```
<DC.Source sub="PrintEdition">Milwaukee: Bruce Publishing Company, 1949,
    c1940.</DC.Source>
<DC.Language scheme="ISO639-1">en</DC.Language>
```

Handling Multiple Volumes

To handle multi-volume works such as a commentary or encyclopedia, each volume may be handled as a separate ThML document. An additional ThML document may then be prepared which combines all of the volumes into one larger unit. The document might look like this:

```
<ThML>
 <ThML.head>
    <!-- information about the whole series -->
  </ThML.head>
  <ThML.body>
    <!-- document for entire series, e.g. a series-wide table of contents-->
  </ThML.body>
  <!ENTITY volume1 SYSTEM "volume1.xml" />
  &volume1;
  <!ENTITY volume2 SYSTEM "volume2.xml" />
  &volume2;
  <!-- rest of volumes here -->
</ThML>
```

Alphabetical List of ThML Body Elements

The following list contains the elements that may be used for ThML markup in the body section of a text. Elements that occur in HTML may also be used, but they are not listed here. Elements concerning hymns are also not listed here; see the section on hymns above.

Name	Use	Example
added	Text added to print edition	`<added reason="Automatic TOC" resp="whp" date="1999-12-23"><insertContents level="2" /></added>`
argument	Section topic	`<argument>`…`</argument>`
attr	Attribution, e.g. poem author	`<attr>`G. Ter Steegen`</attr>`
citation	Reference to another work	`<citation title="Imitatio Christi. English." href="/k/kempis/imitation/">`Imitation of Christ`</citation>`
date	Any date or time	`<date value="1998.12.25">`last Christmas`</date>`
def	Definition from glossary	`<def>`An ancient theological…`</def>`
deleted	Text from print edition that should not be displayed	`<deleted reason="Replaced with automatic TOC" resp="whp" date="1999-12-23"><H1>`Table of Contents`</H1>`…`</deleted>`
div*n*	Major divisions in text	`<div2 type="Chapter" n="I" title="Of the Inward Life">`
index	Index entry	`<index type="subject" subject1="Christian Life" subject2="Sanctification" title="Apotheosis">`The word apotheosis . . . `</index>`
insertIndex	Insert index here	`<insertIndex type="foreign" />`
foreign	Foreign language passage	`<foreign lang="he" dir="rtl">`שלום`</foreign>`
glossary	Mark a glossary	`<glossary type="lexicon">`

		<term>…</term><def>…</def></glossary>
l	Line of verse	<l>O God, a world of empty show,</l>
name	A person's name	<name title="Ter Steegen, Gerhard">G. Ter Steegen</name>
note	Footnotes, endnotes, etc.	<note place="foot" resp="editor" target="#p1" targetEnd="#p2">…</note>
pb	Page break in print edition	<pb n="37" href="page37.gif" />
scripCom	Commentary on scripture	<scripCom passage="Rom. 8:28" version="LXX" />In this verse…
scripContext	Set scripture context for later references	<scripContext passage="Romans 8" version="NRSV" />
scripRef	Scripture reference	<scripRef passage="Rom. 8:28" version="NRSV">Rom. 8.28</scripRef>
scripture	Scripture passage	<scripture passage="Rom. 8:28" version="NIV">All things work…</scripture>
sync	Synchronization point	<sync type="Strongs" value="G42" />
term	Term from glossary	<term>Apotheosis</term>
unclear	Unclear or uncertain text	<unclear>modem</unclear>
verse	Poetry, verse	<verse>…</verse>

Table 1: ThML Body Elements

Using ThML in the CCEL

The way that ThML is used in the CCEL is in some cases more specific than the description above. For example, the <note> element merely identifies text as a note of some sort. The software used in the CCEL moves this text to the end of a section and links it with a footnote marker. Some of the special uses of ThML have been documented above; some others are given here.

- authorID, bookID, and publisherID attributes are keys in the CCEL database, limited to 12 characters.
- CCEL URIs are of the form [http://www.ccel.org/]ccel/authorID/bookID[_version].fmt where ".fmt" identifies the desired format, e.g. .htm, .txt, .xml, etc. The URL may also have |id1 or #id2 appended. In the former case, only the element whose ID is id1 is returned. The latter case works like an HTML hash. Both forms may be combined. Three special IDs are also available: _TOC, a machine-generated table of contents, _About, a machine-generated page of information about the document, and _Pnnn the specified page.

Conclusion

Theological study requires text with relatively rich markup. The Theological Markup Language has been designed to address these needs powerfully and without too much complexity. ThML was designed to be a rich enough representation to support powerful indexing and user interface features and to allow conversion into other popular formats without loss. It is also hopefully a language that can be learned without extraordinary effort. It is a fundamental element of the Christian Classics Ethereal Library system, and it will make the library far more functional and useful.

Appendix A: ThML DTD v. 1.0

The ThML DTD is based on the Voyager DTD for HTML. It uses the voyager DTD and four additional files: ThML.dtd, ebook.mod, bible.mod, and hymn.mod. The last three are modules that add support for general electronic books, scripture references, and hymns.

File: ThML.dtd

```
<!--
    ThML: Theological Markup Language for electronic texts and
    digital libraries.

    ThML is an extension of the Voyager version 4.0 Strict DTD.

    Author: Harry Plantinga
    Version 1.0, 1999-08-18
    Namespace = http://www.ccel.org/Profiles/ThML

    Filename: ThML.dtd
-->

 <!-- Name files we will want to include -->

 <!ENTITY % Voyager.entities SYSTEM "html-entities.mod">
 <!ENTITY % Voyager.chars SYSTEM "html-chars.mod">
 <!ENTITY % Voyager.base SYSTEM "html-base.mod">
 <!ENTITY % Voyager.phrases SYSTEM "html-phrases.mod">
 <!ENTITY % Voyager.lists SYSTEM "html-lists.mod">
 <!ENTITY % Voyager.blocktext SYSTEM "html-blocktext.mod">
 <!ENTITY % Voyager.object SYSTEM "html-object.mod">
 <!ENTITY % Voyager.img SYSTEM "html-img.mod">
 <!ENTITY % Voyager.forms SYSTEM "html-forms.mod">
 <!ENTITY % Voyager.tables SYSTEM "html-tables.mod">

 <!ENTITY % ThML.ebook SYSTEM "ebook.mod">
 <!ENTITY % ThML.bible SYSTEM "bible.mod">
 <!ENTITY % ThML.hymn  SYSTEM "hymn.mod">

 <!-- Customize entities that control the content model -->

 <!ENTITY % block.forms "form | fieldset">
 <!ENTITY % media "|object | img | map">
 <!ENTITY % fontstyle "tt | i | b | big | small">
 <!ENTITY % phrase "em | strong | dfn | code | q | sub | sup |
                    samp | kbd | var | cite | abbr | acronym" >
 <!ENTITY % lists "ul | ol | dl">
 <!ENTITY % inline.forms "input | select | label | textarea | button">
 <!ENTITY % blocktext "pre | hr | blockquote | address">
 <!ENTITY % misc "ins | del | script | noscript | added | deleted |
            index | pb | scripContext | scripRef | scripCom | scripture">
 <!ENTITY % extra.inline "| %fontstyle; | %phrase; | %inline.forms; |
            date | foreign | name | note | sync | unclear">
 <!ENTITY % extra.block "| %lists; | %blocktext; | %block.forms; | table |
            glossary | argument | verse | attr | insertIndex | hymn">
```

```
 <!--
   Now include the files using the names we declared
   earlier, starting with the entities file and ending
   with the base file. The entities file defines the
   standard content models for elements declared in the
   following files.
 -->

 %Voyager.entities;
 %Voyager.chars;

 %ThML.bible;
 %ThML.hymn;
 %ThML.ebook;

 %Voyager.phrases;
 %Voyager.lists;
 %Voyager.blocktext;
 %Voyager.object;
 %Voyager.img;
 %Voyager.forms;
 %Voyager.tables;
 %Voyager.base;

<!ELEMENT ThML.head (title?, base?, generalInfo?, printSourceInfo?,
    electronicEdInfo, (script|style|meta|link)*)>

<!ELEMENT ThML.body (div1|added|deleted)*>

<!ELEMENT ThML (ThML.head, ThML.body, ThML*)>
```

File: ebook.mod

```
<!--
  Electronic book/Digital library support for Voyager.
  From the ThML DTD, v. 1.0.  File: ebook.mod

  Author: Harry Plantinga (hplantin@calvin.edu), January 1, 2000.
  This file may be copied according to the terms of the Artistic License.
-->

 <!--================ ebooks ================================-->

 <!-- Block-level additions -->
 <!ELEMENT attr %Inline;>          <!-- attribution, signature, etc. -->
 <!ATTLIST attr %attrs; >

 <!ELEMENT argument %Inline;>      <!-- chapter subhead or topic -->
 <!ATTLIST argument %attrs; >

 <!ELEMENT l %Inline;>             <!-- line of poetry -->
 <!ATTLIST l %attrs;>

 <!ELEMENT glossary (term|def+)+>  <!-- list of definitions -->
 <!ATTLIST glossary %attrs;
    type     CDATA  #IMPLIED>

 <!ELEMENT sync EMPTY>             <!-- synchronization point -->
 <!ATTLIST sync %attrs;
    type     CDATA  #REQUIRED
    value    CDATA  #REQUIRED>

 <!ELEMENT verse (l|%misc;)*>      <!-- group of poetic lines -->
 <!ATTLIST verse %attrs;
    type     CDATA  "stanza"
    n        CDATA  #IMPLIED>

 <!--================================================================-->
 <!-- Inline additions -->
 <!ELEMENT date %Inline;>          <!-- any reference to a date -->
 <!ATTLIST date %attrs;
    value    %Datetime;  #IMPLIED>

 <!ELEMENT index EMPTY>            <!-- index entry -->
 <!ATTLIST index %attrs;
    type     CDATA  "subject"
    subject1 CDATA  #REQUIRED
    subject2 CDATA  #IMPLIED
    subject3 CDATA  #IMPLIED
    subject4 CDATA  #IMPLIED
    target   IDREF  #IMPLIED>

 <!ELEMENT insertIndex EMPTY>      <!-- add a computed index here -->
 <!ATTLIST insertIndex %attrs;
    level    CDATA  #IMPLIED
    type     CDATA  #REQUIRED>

 <!ELEMENT foreign %Inline;>       <!-- foreign-language phrase -->
 <!ATTLIST foreign
    lang   %LanguageCode; #REQUIRED
    dir    (rtl|ltr)      #IMPLIED
```

```
    %coreattrs;
    %events;>

 <!ELEMENT name %Inline;>          <!-- a person's name -->
 <!ATTLIST name %attrs; >

 <!ELEMENT unclear %Flow;>         <!-- text that may be incorrect -->
 <!ATTLIST unclear %attrs; >


 <!--=================================================================-->
 <!-- Controls (inline or block level)  (for misc) -->
 <!ELEMENT added (#PCDATA | %block; | %inline; | %misc; | div1 | div2 |
           div3 | div4 | div5 | div6)*>  <!-- text added to the source doc --
>
 <!ATTLIST added %attrs;
    resp     CDATA      #IMPLIED
    reason   CDATA      #IMPLIED
    date     %Datetime; #IMPLIED>

 <!ELEMENT deleted (#PCDATA | %block; | %inline; | %misc; | div1 | div2 |
           div3 | div4 | div5 | div6)*>  <!-- text deleted from source doc --
>
 <!ATTLIST deleted %attrs;
    resp     CDATA      #IMPLIED
    reason   CDATA      #IMPLIED
    date     %Datetime; #IMPLIED>

 <!ELEMENT note %Flow;>            <!-- footnote, endnote, etc. -->
 <!ATTLIST note %attrs;
    place    (foot|end|inline|margin|interlinear) "foot"
    resp     CDATA      #IMPLIED
    target   IDREF      #IMPLIED
    targetEnd IDREF     #IMPLIED
    anchored (yes|no)   "yes">

 <!ELEMENT pb EMPTY>               <!-- page break in print edition -->
 <!ATTLIST pb %attrs;
    n        CDATA  #IMPLIED
    href     %URI;  #IMPLIED>


 <!--=================================================================-->
 <!-- Other additions (not in Block or Inline models) -->
 <!ELEMENT term %Inline;>          <!-- term of definition list -->
 <!ATTLIST term %attrs;>

 <!ELEMENT def %Flow;>             <!-- definitions -->
 <!ATTLIST def %attrs;>

 <!--=================================================================-->
 <!-- Structure (divs) -->
 <!ELEMENT div1 ((%block; | %misc;)*, div2*)> <!-- top divisions -->
 <!ATTLIST div1 %attrs;
    type     CDATA  #IMPLIED
    n        CDATA  #IMPLIED>

 <!ELEMENT div2 ((%block; | %misc;)*, div3*)>
 <!ATTLIST div2 %attrs;
    type     CDATA  #IMPLIED
```

```
   n        CDATA  #IMPLIED>

<!ELEMENT div3 ((%block; | %misc;)*, div4*)>
<!ATTLIST div3 %attrs;
   type     CDATA  #IMPLIED
   n        CDATA  #IMPLIED>

<!ELEMENT div4 ((%block; | %misc;)*, div5*)>
<!ATTLIST div4 %attrs;
   type     CDATA  #IMPLIED
   n        CDATA  #IMPLIED>

<!ELEMENT div5 ((%block; | %misc;)*, div6*)>
<!ATTLIST div5 %attrs;
   type     CDATA  #IMPLIED
   n        CDATA  #IMPLIED>

<!ELEMENT div6 (%block; | %misc;)*>
<!ATTLIST div6 %attrs;
   type     CDATA  #IMPLIED
   n        CDATA  #IMPLIED>


<!--=================================================================-->
<!-- head elements -->

<!-- generalInfo: info about the book, for all editions -->
<!ELEMENT generalInfo (description?, firstPublished?, pubHistory?,
   comments?)>
<!ELEMENT description %Flow;>     <!-- paragraph-length desc of document -->
<!ELEMENT firstPublished %Flow;>  <!-- date first published -->
<!ELEMENT pubHistory %Flow;>      <!-- any info about publication history --
>
<!ELEMENT comments %Flow;>        <!-- any other general comments -->

<!-- printSourceInfo: info about the print source for this ebook -->
<!ELEMENT printSourceInfo (published, copyLocation*, image*)>
<!ATTLIST image
   type     CDATA  #REQUIRED
   href     %URI;  #REQUIRED>
<!ELEMENT published %Flow;>       <!-- full bibliographic reference -->
<!ELEMENT copyLocation %Flow;>    <!-- location of print source -->
<!ELEMENT image EMPTY>            <!-- any on-line images of print source --
>

<!-- electronicEdInfo: info about this electronic text -->
<!ELEMENT electronicEdInfo (publisherID, authorID, bookID, version,
   editorialComments?, revisionHistory?, status?, DC, comments?)>
<!ELEMENT publisherID (#PCDATA)>  <!-- publisher code -->
<!ELEMENT authorID (#PCDATA)>     <!-- author code assigned by publisher -->
<!ELEMENT bookID (#PCDATA)>       <!-- book code assigned by publisher -->
<!ELEMENT version (#PCDATA)>      <!-- version code -->
<!ELEMENT editorialComments %Flow;> <!-- editorial practices -->
<!ELEMENT revisionHistory %Flow;> <!-- revision history of this etext -->
<!ELEMENT status %Flow;>          <!-- how clean/complete is this etext? -->

<!--=================================================================-->
<!-- Dublin Core record -->
<!ELEMENT DC (DC.Title | DC.Creator | DC.Subject | DC.Description |
   DC.Publisher | DC.Contributor | DC.Date | DC.Type | DC.Format |
```

```
   DC.Identifier | DC.Source | DC.Language | DC.Relation |
   DC.Coverage | DC.Rights)*>

<!-- attributes for DC elements: sub (subtype), scheme, lang -->
<!ENTITY % DCAtts
  "sub    CDATA #IMPLIED
   scheme CDATA #IMPLIED
   lang   CDATA #IMPLIED" >

<!ELEMENT DC.Title       (#PCDATA)> <!ATTLIST DC.Title       %DCAtts;>
<!ELEMENT DC.Creator     (#PCDATA)> <!ATTLIST DC.Creator     %DCAtts;>
<!ELEMENT DC.Subject     (#PCDATA)> <!ATTLIST DC.Subject     %DCAtts;>
<!ELEMENT DC.Description (#PCDATA)> <!ATTLIST DC.Description %DCAtts;>
<!ELEMENT DC.Publisher   (#PCDATA)> <!ATTLIST DC.Publisher   %DCAtts;>
<!ELEMENT DC.Contributor (#PCDATA)> <!ATTLIST DC.Contributor %DCAtts;>
<!ELEMENT DC.Date        (#PCDATA)> <!ATTLIST DC.Date        %DCAtts;>
<!ELEMENT DC.Type        (#PCDATA)> <!ATTLIST DC.Type        %DCAtts;>
<!ELEMENT DC.Format      (#PCDATA)> <!ATTLIST DC.Format      %DCAtts;>
<!ELEMENT DC.Identifier  (#PCDATA)> <!ATTLIST DC.Identifier  %DCAtts;>
<!ELEMENT DC.Source      (#PCDATA)> <!ATTLIST DC.Source      %DCAtts;>
<!ELEMENT DC.Language    (#PCDATA)> <!ATTLIST DC.Language    %DCAtts;>
<!ELEMENT DC.Relation    (#PCDATA)> <!ATTLIST DC.Relation    %DCAtts;>
<!ELEMENT DC.Coverage    (#PCDATA)> <!ATTLIST DC.Coverage    %DCAtts;>
<!ELEMENT DC.Rights      (#PCDATA)> <!ATTLIST DC.Rights      %DCAtts;>


<!-- entities -->

<!ENTITY % inline.ebook "| date | foreign | name | note | sync | unclear">
<!ENTITY % block.ebook  "| glossary | argument | verse | attr |insertIndex">
<!ENTITY % misc.ebook   "| added | deleted | index | pb">
```

File: bible.mod

```
<!--
  Support for Bibles, Bible references, citations, commentary in Voyager.
  From the ThML DTD, v1.0.  File: bible.mod

  Author: Harry Plantinga (hplantin@calvin.edu), January 1, 2000.
  This file may be copied according to the terms of the Artistic License.
-->

 <!--========================  Bibles  ============================-->
 <!--=========== references, citations, commentary ==================-->

 <!ENTITY % scripturePassage "CDATA"> <!-- e.g. Rom. 8:28-30, 9.10-1, Rev2-->
 <!ENTITY % scriptureParsed  "CDATA"> <!-- version|bk|fch|fv|tch|tv -->
 <!ENTITY % scriptureVersion "(NIV | KJV | NKJV | RSV | NRSV | ASV |
    NEB | VUL)">
 <!ENTITY % scripComType     "(Commentary | Hymn | Sermon | Outline |
    Meditation | Treatise | Study | Poem | Citation)">

 <!ELEMENT scripture (#PCDATA | %block; | %inline; | %misc;)*>
 <!ATTLIST scripture %attrs;
   version %scriptureVersion;  #IMPLIED
   passage %scripturePassage;  #IMPLIED
   parsed  %scriptureParsed;   #IMPLIED>

 <!ELEMENT scripContext EMPTY>
 <!ATTLIST scripContext %attrs;
   version %scriptureVersion;  #IMPLIED
   passage %scripturePassage;  #IMPLIED
   parsed  %scriptureParsed;   #IMPLIED>

 <!ELEMENT scripRef %Inline;>
 <!ATTLIST scripRef %attrs;
   version %scriptureVersion;  #IMPLIED
   passage %scripturePassage;  #IMPLIED
   parsed  %scriptureParsed;   #IMPLIED
   target  %URI;               #IMPLIED>

 <!ELEMENT scripCom EMPTY>
 <!ATTLIST scripCom %attrs;
   type    %scripComType;      "Citation"
   version %scriptureVersion;  #IMPLIED
   passage %scripturePassage;  #IMPLIED
   parsed  %scriptureParsed;   #IMPLIED
   target  %URI;               #IMPLIED>


 <!ENTITY % misc.bible "| scripContext | scripRef | scripCom | scripture">
```

File: hymn.mod

```
<!--
  Hymn support library support for Voyager.
  From the ThML DTD, v. 1.0.  File: hymn.mod

  Author: Harry Plantinga (hplantin@calvin.edu), January 1, 2000.
  This file may be copied according to the terms of the Artistic License.
-->


<!ELEMENT hymn (title | meter | author | date | argument | tune |
                verse | index | scripRef | scripCom | p | h1 | h2 |
                h3 | h4 | h5 | h6 | blockquote | hr | note)+>
<!ATTLIST hymn %attrs;
    n          CDATA  #IMPLIED>

<!ELEMENT meter %Inline;>
<!ATTLIST meter %attrs;
    standard   CDATA  #IMPLIED>

<!ELEMENT author %Inline;>
<!ATTLIST author %attrs;
    type       CDATA  #IMPLIED
    authorID   CDATA  #IMPLIED>

<!ELEMENT tune (composer | incipit | music | date | meter |title|copyright)+>
<!ATTLIST tune %attrs;
    tuneID     CDATA  #IMPLIED>

<!ELEMENT composer %Inline;>
<!ATTLIST composer %attrs;
    type       CDATA  #IMPLIED
    composerID CDATA  #IMPLIED>

<!ELEMENT incipit EMPTY>
<!ATTLIST incipit %attrs;
    value      CDATA  #IMPLIED>

<!ELEMENT music EMPTY>
<!ATTLIST music %attrs;
    type       CDATA  #IMPLIED
    inline     (yes|no) "no"
    actuate    (auto|man) "man"
    href       %URI;   #REQUIRED>

<!ENTITY % block.hymn "| hymn">
```

Appendix B: ThML.head Sample for the CCEL

```
<!DOCTYPE ThML PUBLIC "-//CCEL//DTD Theological Markup Language//EN"
"dtd/ThML.dtd">
<ThML>
<ThML.head>
<title>ThML Sample Document</title>
<generalInfo>
   <description>This is a sample ThML document. A brief description of the
   document goes here.</description>
   <firstPublished>ca. 1400</firstPublished>
   <pubHistory>
```

This book had been published in over 6000 editions by 1900 -- more than one per month for 500 years. It has been called the most-published of all books other than the Bible.

```
   </pubHistory>
   <comments>A copyright renewal search did not find record of copyright
   renewal for the source edition of this text.</comments>
</generalInfo>
<printSourceInfo>
   <published>Milwaukee: Bruce Publishing Company, 1949, c1940.</published>
   <copyLocation>Book in private collection of Harry
      Plantinga.</copyLocation>
   <image type="front" href=""/>
   <image type="spine" href=""/>
</printSourceInfo>
<electronicEdInfo>
   <publisherID>ccel</publisherID>
   <authorID>kempis</authorID>
   <bookID>sample</bookID>
   <version>1.15</version>
   <editorialComments>Unambiguous end-of-line hyphens removed.
   </editorialComments>
```

Publisher code of electronic edition, as assigned by the CCEL
Author ID as assigned by publisher
Book ID as assigned bypublisher
Edition or version of electronic edition
Comments about editorial practices: whether spelling was normalized, what was done with end-of-line hyphens, corrections, tagging practices, etc.

```
   <revisionHistory>
   <ul>
      <li>v1.15, 1999-01-24, modified for Voyager -- lower case tags,
               XML</li>
      <li>v1.14, 1999-01-12, adds colored text, smart quotes, a scripRef in a
               footnote, and markup in the header -- here!</li>
      <li>v1.13, 1998-12-10, modified to use DC in header instead of
               xmarc.</li>
      <li>v1.12, 1998-12-05, modified to use xmarc in header instead of many
               of those old header elements.  Looks better, I think.</li>
      <li>v1.0,  1998-08-03, Initial ThML version</li>
      <li>Original, 1994,    Initial Word version</li>
   </ul>
   </revisionHistory>
```

A list of published editions and changes

```
   <status>
   Preliminary released version has been proofread and is pretty clean.
   Names and scripture references have been marked, but only token index
   entries added. Uses a preliminary version of ThML; subject to change.
   </status>
```

Current editorial/publication status of text

```
   <DC>
      <DC.Title sub="Main">The Imitation of Christ</DC.Title>
      <DC.Title sub="Alternative">Imitatio Christi. English.</DC.Title>
      <DC.Creator>Thomas, &agrave; Kempis, 1380-1471</DC.Creator>
      <DC.Creator sub="Alternative">Kempis, Thomas &agrave;</DC.Creator>
```

```
      <DC.Subject scheme="LCSH">Meditations.</DC.Subject>
      <DC.Subject scheme="CCEL">Classics; Recommended</DC.Subject>
      <DC.Description> In preparing this edition of The Imitation of Christ,
         the aim was to achieve a simple, readable text which would ring true
         to those who are already lovers of this incomparable book and would
         attract others to it.</DC.Description>
      <DC.Publisher>Grand Rapids, MI: Christian Classics Ethereal
         Library</DC.Publisher>
      <DC.Publisher sub="Address">ccel@www.ccel.org</DC.Publisher>
      <DC.Publisher scheme="CCEL">CCEL</DC.Publisher>
      <DC.Contributor sub="Transcriber" scheme="CCEL">whp</DC.Contributor>
      <DC.Date sub="Created" scheme="ISO8601">1998-12-10</DC.Date>
      <DC.Type>Text.Monograph</DC.Type>
      <DC.Format scheme="IMT">text/xml</DC.Format>
      <DC.Identifier
         scheme="CCEL">ccel/kempis/sample/1.15.htm</DC.Identifier>
      <DC.Identifier>http://www.ccel.org/ThML/sample/About.htm</DC.Identifier>
      <DC.Subject scheme="LCCN">BV4821.A1 1949</DC.Subject>
      <DC.Source sub="ElectronicEdition">Project Gutenberg</DC.Source>
      <DC.Source sub="PrintEdition">Milwaukee: Bruce Publishing Company,
         1949, c1940.</DC.Source>
      <DC.Language scheme="ISO639-1">en</DC.Language>
      <DC.Relation></DC.Relation>
      <DC.Coverage></DC.Coverage>
      <DC.Rights>Public Domain</DC.Rights>
   </DC>
   <comments></comments>
</electronicEdInfo>
</ThML.head>
<ThML.body>
```