# Twenty Years of Theological Markup Languages: A Retro- and Prospective

by Rozalija Grace

**ABSTRACT**: ThML—the first open, XML-based markup language designed specifically for digital libraries handling theological collections—was conceived in 1998, sparking a period of development in discipline-specific markup languages for theology that lasted until the early 2000s, when the dominance of the TEI standard led the field to stagnate. Despite the disappearance of the active developer communities behind most of the projects and technical improvements in TEI, however, ThML and other languages developed during that period remain in use. After presenting a brief history of theology-specific markup, this article seeks to understand what its persistence tells us about the discipline-specific needs of biblical and theological studies that are still not being met by TEI, and offers insights as to the lessons that may be drawn from these projects for the future of theological markup.

In 1998, Harry Plantinga, a computer science professor at Calvin College, began work on the Theological Markup Language (ThML)—the first open, XML-based markup language designed specifically for digital libraries handling theological collections.[1] Over the past two decades, ThML has provided critical technical infrastructure to a wide range of projects, both academic and popular, that have disseminated theological literature to millions of users. In 1999, a consortium of leading organizations in biblical studies and Bible publishing started development on projects that became the Open Scripture Information Standard (OSIS)—another XML-based language intended for specialized theological use. This markup, too, made its way into a wide array of projects, and it was particularly favored by Bible translators and publishers. Throughout the early 2000s, these two flagship initiatives were joined by a host of smaller endeavors with much more limited adoption: OpenSong (XMM), Unified Scripture Format XML (USFX), Zefania XML, and others. By 2006, however, interest in discipline-specific markups for theology began to wane as wider adoption of the Text Encoding Initiative (TEI) encoding schema—a technical standard meant to promote compatible styles and expressions of XML across disciplines—increased in appeal to a variety of digital humanists. By the end of the 2000s, the period of active markup development specific to theology projects had largely come to an end.

Nevertheless, the tools developed over the preceding years, particularly ThML and OSIS, remain in use even until today, despite their now limited interoperability and lack of active development and support. This essay briefly surveys the history of these projects and their distinctive features in an effort to understand why they have not been replaced. Analysis shows that, despite TEI's refinement over the past twenty years, many of the critiques made by developers of discipline-specific markups for theology remain unaddressed, such that TEI remains a suboptimal tool for certain kinds of discipline-specific work. The paper concludes by identifying elements of ThML and OSIS that might still be profitably incorporated into TEI, either as custom extensions for current projects or, potentially, as canonical additions to future TEI versions.

## A BRIEF HISTORY OF THEOLOGY-SPECIFIC (XML) MARKUP

Just as the Bible was the first book to be set by movable type, so it was also one of the earliest texts to receive specific attention for computer encoding. By the late 1990s, decades of work had produced a plethora of

[1] Harry Plantiga, "ThML: Theological Markup Language for the Christian Classics Ethereal Library," *Christian Classics Ethereal Library*, February 23, 2001, http://www.ccel.org/ThML/ThML1.02.pdf.

"markup languages" for setting biblical text in newly digitized publishing processes, with each publisher generally employing its own proprietary standard.

By the turn of the century, however, the World Wide Web Consortium's (W3C) Extensible Markup Language (XML) began rapidly to overtake such idiosyncratic systems. XML was simultaneously human- and machine-readable, readily compatible with the Web, and easily customized for a wide variety of uses. Its overarching "syntax" for how documents are to be described in machine processing is more a standard for the creation of markup languages than a markup language in itself. Various projects realize their own languages through XML by adding to its basic set of "tags" marking "elements" and "attributes" in the text, in order to be able to describe the features they deem important. Basic tags included in all XML-based languages mark such "elements" as titles, paragraphs, quotations, and points of emphasis. Implementations of XML in the form of specific markup languages customize the available "tags" to enable more sophisticated and precise descriptions of the physical features of a document, the literary devices found in the text, variants among manuscripts in particular locations, or whatever may be of interest within a particular discipline or a specific project.

Combined with this flexibility, the openness of XML's technical standards promised not only a way out of the publishers' Babel, but also an extension of the benefits of computer-aided publishing to smaller enterprises, such as missionary groups printing Bibles in indigenous languages or digital libraries using the Internet to disseminate materials for theological study.

### ThML

The release notes for Harry Plantinga's 2001 Theological Markup Language (ThML) identified several objectives of a markup language for theological study, stating that it should support:

1) special processing for scripture references
2) synchronization and alignment of texts in multiple versions (variants, translations, etc.)
3) representation of hymns and mixed media
4) cross-referencing of Strong's numbers, diverse indices, commentaries, etc.
5) library-wide searching of texts by multiple facets (subject, scripture reference, etc.)
6) user annotation of texts without alteration to the original file
7) direct incorporation of bibliographic metadata
8) royalty-free use and redistribution of the markup language itself
9) web-based access without need of software beyond the web browser

"Existing markup languages," Plantinga concluded, "do not meet all of these needs." He dismissed word processor formats and HTML as lacking adequate representation for semantic information and rejected commercial formats of the time (such as STEP and the Logos Library System) as prohibitively expensive and not fully Internet ready. While these judgments would likely be shared by any digital humanist today, what stands out from Plantinga's assessment of the state of the art at the beginning of the 21st century is his critique of the Text Encoding Initiative (TEI).

TEI describes both a technical standard and a scholarly community of practice, inaugurated in 1987 with the aim of providing a common framework for text encoding across humanities disciplines. In 2002, the "P4" release of the TEI Guidelines embraced XML as the underlying architecture for TEI encoding, establishing an array of hundreds of XML elements to provide a thorough toolkit for all manner of digital humanities (DH) projects while ensuring the greatest possible interoperability among resulting data and documents. Its broad success in this endeavor is shown by the fact that today it is the *de facto* standard within humanities, used for everything from the British National Corpus and the New Zealand Electronic Text Centre to the Perseus Project and Epidoc.

Plantinga readily acknowledged the influence of TEI on his work, but in light of the broad adoption of TEI, his reasons for not adopting it directly command attention twenty years on. "The Text Encoding Initiative," he concluded, "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 Strong's-like reference systems . . . [It] is very large and the overhead required to learn and process the language is high." It was to supply these perceived deficiencies that ThML was developed and deployed for the Christian Classics Ethereal Library (CCEL) hosted at Calvin College. Although Plantinga has never made any deliberate effort to missionize ThML as a standard beyond the CCEL, its example prompted adoption by a number of other projects, such as the OpenBible software project for Unix, the Go Bible mobile app, and most notably the SWORD project of the Crosswire Bible Society.

These applications reaped benefits from several elements in the ThML schema that facilitated work on Bibles and related texts. Most obvious is the element <verse>, providing a ready and unambiguous encoding for this common unit, which is almost unaccountably absent from TEI. Plantinga also offered the element <argument> for chapter or section subheads or summaries of topics of discussion, as are frequently found in Bibles (and other early modern books) to summarize the contents of chapters. The easy differentiation of these from other kinds of headings and titles made their inclusion or suppression in various displays easy to implement. The elements <scripRef>, and <scripture> helped to distinguish references and allusions to scripture from direct scripture quotations, as well as neatly separating both from all other forms of intertextuality, in order to build more readily automated indices of the use of scriptural texts in Patristic writings and elsewhere. Likewise, <scripCom> conveniently designated passages that functioned as commentaries on scripture, with type attributes allowing fine grains of distinction in identifying commentary as diverse as prose treatises and sections of canonical hymns.

Representation of hymns was a major focus of Plantinga's design. Not only did they receive their own <hymn> element, but this was permitted to take child elements designating <meter>, <author>, <tune>, <composer>, an <incipit> (providing opening measures transposed to the key of C), and <music> (embedding recordings or sheet music). All of these features are now richly deployed in a sister-project to the CCEL—hymnary.org—which has received sponsorship from the Hymn Society of the United States and Canada as well as funding from the National Endowment for the Humanities.

Perhaps ThML's most distinctive feature is the element <sync>, which Plantinga designed for "dates, keywords, or Strong's numbers [that] aren't really links to other documents . . . Therefore XLL links, element IDs, etc., don't capture the semantics of such information." [2]

Although now largely restricted in use to the CCEL and hymnary.org, ThML has influenced many subsequent projects and earned indirect notice from the academic community through the features it has made possible at those websites (in large part by making them easy to implement by student and community volunteers who have encoded hundreds of books). The shortness and simplicity of the ThML schema is especially notable in reading appraisals like that of Brian Kooy, who, reviewing the CCEL for *Reference Reviews*, described it as "a well-executed and technically rich site with . . . features unmatched by any other online digital collection of its kind." He paid particular attention to the search options "for definitions [and] for scripture passages," which were directly enabled by the specific design features of ThML mentioned above.[3]

## XSEM AND OSIS

One year after Plantinga began development of ThML, Dennis Drescher approached SIL International—a Christian organization that promotes linguistic research in support of translation and mission efforts—with

[2] XLL links are now more commonly known as XLinks—an XML-based standard for the formatting of links to Web content.
[3] Brian K. Kooy, "Christian Classics Ethereal Library," *Reference Reviews* 25.7 (2011): 12.

the idea of developing an XML-based system for scriptural markup. The result was a consortium formed with the United Bible Societies, the American Bible Society, the Society of Biblical Literature, and several smaller stakeholders, known as the Bible Technologies Group. The group's initial effort was the XML Scripture Encoding Model (XSEM), which differed from Plantinga's approach in a number of respects. Notable differences were that XSEM was based on a W3C XML Schema rather than a Document Type Definition (DTD), and XSEM focused exclusively on markup for scripture, as opposed to other forms of theological literature.[4] XSEM was short-lived as an independent standard, however, and was quickly expanded into a more ambitious undertaking—the Open Scripture Information Standard (OSIS)—from 2000 on.[5] The range of application was extended beyond XSEM's narrow focus in order to include both "scripture and related text," and plans were laid to market OSIS as an industry standard. The OSIS FAQ sheet even provided a thirty-second "elevator speech" for pitching it to supervisors at publishing companies.[6]

Like ThML, OSIS took inspiration from TEI, inheriting large numbers of elements directly and retaining TEI form attributes to enable processors to make equivalences.[7] Also like ThML, but in marked contrast to TEI, it aimed at the slimmest possible profile. The user manual boasted that, apart from construction of the document header, "the average user can produce a professional quality encoding of a Bible text with fifteen (15) or fewer elements."[8]

A much richer set than fifteen was available, however. Like ThML, a <verse> element was offered, joined by a <chapter> element. The latter was admitted by OSIS' documentation to be a case of "syntactic sugar"—an element marking a function easily marked in another way—but it did clarify document structures for lightly-trained encoders already working with <verse>.

Another simplification for encoders was the attribute @canonical, which could be applied to any element to distinguish core material in a work from editorial apparatus and inclusions, regardless of whether those inclusions came from preparation of the markup or were present in the source (such as distinguishing the text of a Bible chapter from headings introducing subsections of a chapter by theme). This could be reflected otherwise through the use of @resp, of course, but @canonical provided a more convenient shorthand for a very common binary distinction. OSIS also provided preset  types such as "bookGroup," "book," "commentary," "concordance," "devotional," "majorSection," "section," and "subSection," the latter three offering particular functionality in representing common exegetical subdivisions of Biblical books that are not expressly indicated in the chapter-verse system, such as the "Servant Songs" within Isaiah.

Similarly, <note> types were included in the base schema for "allusion," "background," "crossReference," "devotional," "exegesis," "explanation," "study," "translation," and "alternative/variant," facilitating the process of manipulating textual notes at fine degrees of granularity. Special elements were included for epistles (<salute>, <signed>, <closer>), as well as special attributes for <l> (used to mark lines of poetry) to represent particularities in the structure of Old Testament verse (@doxology, @selah). The <divineName> was given its own element, to allow easy transformations when it was necessary to suppress representation, substitute, or transform by special rules in the stylesheet. Other names benefitted likewise from special attributes for the standard <name> element (@holiday, @nonhuman, @ritual). The @see attribute aided in the encoding of scriptural cross-references in automatically generated indices.

---

[4] Dennis Drescher, "XSEM: XML Scripture Encoding Model," SIL, September 6, 2001, https://scripts.sil.org/xsem.
[5] Patrick Durusau, *OSIS Users Manual*, American Bible Society (https://web.archive.org/web/20120226061942/http://img.forministry.com/7/7B/7BB51FB8-84B3-4FF3-939ED473FA90A632/DOC/OSIS2_1UserManual_06March2006_-_with_O'Donnell_edits.PDF), 2.
[6] Bible Technologies Group, *OSIS FAQs* (https://web.archive.org/web/20131107020309/http://www.bibletechnologies.net:80/vsItemDisplay.dsp&objectID=B25323FC-E8BB-49C2-A6C98E74A377F721&method=display).
[7] Durusau, *OSIS Users Manual*, 2.
[8] Durusau, *OSIS Users Manual*, 11-12.

The most technically significant OSIS distinctives, however, were three other features. The <date> element, normally limited to describing year/month/day/hour/minute/second, was augmented with types for times independent of the clock, such as sunrise and sunset, the Catholic canonical hours, and Muslim calls to prayer, and the element was also given a modified syntax enabling easy encoding of dates as repeating at intervals. This allowed lectionaries, prayer books, and other such works to be encoded with machine-readable indications of the ritual dates and times on which specific texts were to be used.

The @editions attribute allowed units of the text at virtually any size to be quickly designated for inclusion or suppression in particular generated expressions of the encoded work. For example, the books of the Apocrypha might be encoded with <div=book edition="Catholic Study"> to include them when Catholic and Study editions of the Bible are rendered from the XML file, but not when Protestant editions are produced.

OSIS' most enduring legacy has been its "Trojan milestone" (also known as CLIX) method for encoding elements that cross the boundaries of other elements (as when a single verse falls across two paragraphs). This was a variation on previous methods using anchors and standard milestones that "has the advantage that milestones representing a given type of element have the same name as the element, and automatically have the same attributes."[9] This method has since been incorporated by other markups, including TEI. It represented a significant innovation at the time, particularly for enabling the simultaneous mapping of different versification schemes, as found, for example, in the differences in numbering of Psalm verses between Jewish and Christian traditions.

Though the website of the Bible Technologies Group went offline in 2006 and active development of OSIS ceased, it has had a long afterlife. It remains in currency with the SWORD project and pops up from time to time in the work of individual scholars, like Robie and Bulkeley.[10]

Both OSIS and ThML continue to receive interest from initiatives working specifically on scriptural and theological materials.[11] On the publication and software development front, too, both languages continue to be represented. Kayode Sowole, who won the University of Lagos' "Best Student App" award for mobile development with Wazobia (an app that allows offline reading of the Bible in multiple Nigerian languages), cited both ThML and OSIS as "the most significant part of my preparation" and as having been necessary for the development of the app.[12]

## CONTINUING LIMITATIONS OF TEI

TEI largely became the "de facto standard for encoding DH texts" from 2002 on, following its conversion from SGML to XML.[13] It is now widely regarded as "standard practice" for digital humanities projects.[14] Nonetheless, it remains true, as Plantinga originally asserted, that "[m]ultiple formats for electronic resources are a reality in today's computing environment." Given TEI's ambition to provide an overarching structure capable of accommodating the widest range of needs on the part of textual scholars, it is therefore worth asking why

[9] Durusau, *OSIS Users Manual*, 54.
[10] Jonathan Robie, "Querying Greek texts in XML: Part 1," *BiblicalHumanities.org*, November 13, 2015, http://biblicalhumanities.org/xquery/tutorial/greek/2015/11/13/querying-000.html; Tim Bulkeley, "Returning to E-commentary," *Sansblogue*, "Returning to e-commentary," April 8, 2016, https://bigbible.org/sansblogue/bible/biblical-interpretation/returning-to-e-commentary/.
[11] See, for example, Leonid Dubinsky, "Digital Judaica Done Right," *DigitalJudaica*, http://www.digitaljudaica.org/judaica/html/index.html.
[12] Dayo Adesulu, "Meet UNILAG Undergraduate who Developed Wazobia Bible Application," *Vanguard*, March 27, 2014, https://www.vanguardngr.com/2014/03/meet-unilag-undergraduate-developed-wazobia-bible-application/.
[13] Desmond Schmidt, "The End of XML," *Multi-Version Documents*, March 18, 2018, http://multiversiondocs.blogspot.com/2018/03/the-end-of-xml.html.
[14] UNL Center for Digital Research in the Humanities, "Best Practices for Digital Humanities Projects," https://cdrh.unl.edu/articles/best_practices.

it has not been able to fully supersede either ThML or OSIS, which continue to find user communities despite their age, lack of active support, and limited interoperability across TEI-dominated fields of scholarship.

Such supersession might have been expected for reasons beyond the simple desire for interoperability in texts and tools. For one thing, many of the key institutions and individuals that contributed to theology-specific projects were also engaged with TEI. The Society of Biblical Literature, for example, was a participant in the TEI Consortium throughout the years it participated in the Bible Technologies Group, and Patrick Durusau, who worked for the SBL on OSIS, also served on the TEI Character Set Working Group.[15] As one would expect, these close connections facilitated technology transfer, and TEI was enriched by ideas emanating from other projects. The P5 release, for instance, incorporated the "Trojan milestone" method for encoding overlap in non-hierarchical structures, citing the work of Steven DeRose.[16] DeRose served as the chair of the Bible Technologies Group and developed the Trojan milestone method as part of the OSIS development team.

Despite this cross-fertilization and its general growth and refinement over the past twenty years, however, TEI has not overcome the two major critiques offered by Plantinga back in 1998—it is difficult to learn and apply consistently, and it is not tuned for theological work. The following sections address each of these critiques from a discipline-specific perspective in light of more recent literature on TEI, before the conclusion offers notes toward possible resolutions and the enrichment thereby of both TEI as a standard and theological digital humanities as a community.

## DIFFICULTY AND INTEROPERABILITY

The first of the two reasons Plantinga gave back in 1998 for not embracing TEI for his digital theological library was that it was "not easy to learn" and that this would inhibit the willingness and ability of others to contribute to the project. Twenty years on, TEI has not become appreciably easier. That the language itself retains a steep learning curve is, of course, to be expected in light of the inherent complexity of the vast range of tasks it seeks to accomplish, but it is disquieting that more tools have not appeared to ease the end-user into the process. Hence, ease-of-use and interoperability quickly begin to fuse into a multifaceted yet singular concern, because low ease-of-use is partly the result of a paucity of software tools for facilitating the markup process, and partly the result of the proliferation and consequent ambiguity of the element set, which in turn inhibits software development.

There are, at present, 567 TEI elements—twenty-three more than there were in 2012, when Desmond Schmidt observed the perverse effects that this daunting range has on the Initiative's *raison d'être*. Beyond simply intimidating the non-specialist and thus excluding potentially valuable contributors to DH projects (just as Plantinga had worried), the breadth of the element set works to ensure ambiguity in the selection of elements for even basic editorial tasks.[17] An almost absurd example familiar to biblical scholars is that of selecting an element to encode verses in standard chapter/verse reference systems. The P5 guidelines suggest the use of the <ab> element ("anonymous block"—typically used as a catch-all for non-paragraph structures with paragraph-like functions) for encoding "canonical verse divisions of Biblical text," but counsel in the very next paragraph that the <seg> element ("segment"—typically used for otherwise unspecified structures below the level of a sentence) may be preferred for the same purpose in order to avoid conflicts with <p> ("paragraph") structures. Meanwhile, an example of reference system encoding elsewhere on the same page demonstrates the use of the  element ("division"—typically used as a generic descriptor for structures

[15] Society of Biblical Literature, "Biblical Scholars, Standards and the SBL," SBL Forum, 2.6 (2004), https://www.sbl-site.org/publications/article.aspx?ArticleId=45.
[16] Steven DeRose, "Markup Overlap: A Review and a Horse," Proceedings of Extreme Markup Languages 2004, http://xml.coverpages.org/DeRoseEML2004.pdf.
[17] Desmond Schmidt, "The Role of Markup in the Digital Humanities," *Historical Social Research / Historische Sozialforschung* 37 (2012): 125, 128.

above the level of a paragraph) for Bible verses. Given that encoding for theological projects often aims at the most intensive levels of interoperability in order to facilitate sophisticated cross-indexing, parallel alignment, and commentary attachment, the unpredictability of any given text's encoding of something as basic as a verse unit is a severe impediment to exchange of research data.

Of course, it is not expected by TEI that any project will use the full range of elements, and customization by narrower selection of element modules is actively encouraged. Two difficulties commonly arise from this. One is that customization of TEI is even more daunting from a technical standpoint than using it. Because the customizations do not come pre-packaged for consumer use but instead arise ad hoc from the user base, projects that do not have in-house tech specialists and cannot afford to contract any often attempt to shoehorn their work into an ill-fitting but ready-made customization, as evidenced by the widespread use of TEI Lite, which was intended originally as a technical demonstration and not meant to be used for real-world projects at all. This creates an interoperability divide between well-resourced and/or tech-savvy projects and those which are not. Even in ideal cases, however, customization at high levels of technical proficiency often serves only to solidify incompatibilities when projects select different elements for similar or even identical purposes when building their customizations. Selecting narrower ranges of elements for particular projects may increase consistency, and therefore interoperability, within those projects, but it does little to achieve those goals *across* projects. Neither should the present system of customization be overestimated in its effectiveness on this point at even the project level. UCLA's introductory tutorial for digital humanities warns its reader that "even the same individual working on different days can use tags differently. The range of interpretation is difficult to restrict, and individual acts of tagging are rarely consistent."[18] It thus seems difficult to disagree with Schmidt, noting that the foundational goal of TEI was a ready data exchange and maximized interoperability that no one has achieved, owing to the subjectivity inherent in TEI. This, of course, is only the double-edge of the sword XML designers originally forged—ambiguity being the inescapable result of its core goals of "flexibility, extensibility, [and] modularity."[19]

Jerome McDonough has given as much thought as anyone to finding a way out of this bind. "One possible response . . ." he has suggested, "would be to say that perhaps our community cares less about interoperability than we thought . . . and [that] the adoption of metadata standards that impeded interoperability is merely a reflection of that underlying reality, and not a major problem to resolve."[20] The fact that significant projects like CCEL and SWORD continue to use what are now highly idiosyncratic encodings would seem to offer some support for this conclusion. At the same time, however, they also lend support to McDonough's following suggestion—that libraries can favor interoperability by more severely restricting the range of encoding options and thus "decreasing the possibility for local variation in encoding."[21] This was, in fact, an implicit design goal of both ThML and OSIS as against TEI, and necessarily contributed to their isolation as TEI's influence grew—an outcome McDonough recognized when he speculated that "removing local capacity for variation will also tend to reduce the number of institutions who are willing to use such a markup language."[22] McDonough ultimately rejected this approach as not flexible enough to engage successfully with publishers and other communities and instead favored a shift in priorities that would see crosswalk development elevated to the same level of respect and resource-commitment presently enjoyed by the development of independent markups.[23] This, too, can be found among theological markup communities today, with projects frequently developing tools

[18] UCLA Center for Digital Humanities, "Text Encoding: Mark-up and TEI," *Intro to Digital Humanities: Concepts, Methods, and Tutorials for Students and Instructors*," September 2013, http://dh101.humanities.ucla.edu/?page_id=60.
[19] Jerome McDonough, "XML, Interoperability and the Social Construction of Markup Languages: The Library Example," *digital humanities quarterly* 3.3 (2009), http://digitalhumanities.org/dhq/vol/3/3/000064/000064.html
[20] McDonough, "XML," paragraph 30.
[21] McDonough, "XML," paragraphs 32-33.
[22] McDonough, "XML," paragraph 33.
[23] McDonough, "XML," paragraph 34.

for transformation between ThML, OSIS, Zefania XML, and even older non-XML systems like General Bible Format (GBF) as a way of combating the Babel that has arisen. This prevalence of crosswalking, however, ultimately further removes disincentives from the use of narrowly tailored standards and has arguably helped to facilitate their proliferation.

## LACK OF SPECIFIC TOOLS

These small encoding languages have survived principally because they offer tools fitted to theological work that are still not to be found elsewhere. Given the difficulties already noted in encoding verses in TEI, for example, and the common use of chapter/verse structures to organize not only the Bible, but the Qur'an and other important texts, it is somewhat unaccountable that there is no specific element to be found for this among the 567 elements defined by TEI. ThML and OSIS, as we have seen, both offer one. This and many other features of those markups could, potentially, be seen simply as streamlining the encoding process for aspects of a text that can also be represented in a technically equivalent fashion by standard TEI. In many cases, this is true, but the streamlining should not be underestimated, especially when it helps to reduce ambiguity in element selection.

TEI's emphasis on flexibility and extensibility has made it extremely powerful for the encoding of literary material, where unexpected permutations of format or style are common. For literary texts, it is unquestionably the present best practice.[24] It is also appropriately favored where there is strong emphasis on exact transcription of source documents and manuscripts, where one must also expect the unexpected.[25] That TEI should predominate even in biblical studies is not without reason. Theology, ethics, and other such disciplines, however, extract little benefit from the flexibility and extensibility offered by TEI, as they are generally concerned with texts in the abstract more than specific documents (with their attendant idiosyncrasies) and particularly with texts that have long-established standards of structure and reference already in place. In these cases, trying to apply overly broad tools to represent such structures creates needless vagaries hindering interoperability and can even compromise the integrity of the text, insofar as the attempt to map traditional textual structures to more general categories in the markup can introduce problematic forms of interpretation, not unlike translation into another language.

The streamlining enabled by the highly specific elements and attributes of ThML and OSIS had a practical benefit as well, in that it lowered barriers to learning the system for individuals with little technical background. This category generally includes scholars in theology and religion, whose departments are often severely under resourced compared even to other humanities faculties, such as English (which generally dominates discussions of digital humanities). Additionally, theology and religious studies departments and centers are, in Western countries, focused on the study of Abrahamic traditions. If usefully interoperable encoding is possible only for projects able to recruit and support dedicated encoding specialists, these traditions will remain significantly overrepresented in the corpus of encoded documents as compared with Eastern religions, indigenous religions, and New Religious Movements. TEI thus carries the potential to parochialize digital theology as a result of the need for intensive institutional support.

For many years, it has been expected that the development of new software tools would enable wider participation by reducing the need for specific technical skills, but this has not happened at nearly the speed or to nearly the extent that was generally anticipated, largely because, as Schmidt recognized, even when individual projects successfully customize appropriate, consistent, and streamlined implementations of

[24] UCLA Center for Digital Humanities, "Text Encoding."
[25] H. A. G. Houghton, "The Electronic Scriptorium: Markup for New Testament Manuscripts," pp. 31–60 in *Digital Humanities in Biblical, Early Jewish, and Early Christian Studies*. Claire Clivaz, Andrew Gregory, and David Hamidović, eds. (Leiden: Brill, 2013).

TEI, the need to accommodate the full range of TEI elements when building broadly compliant tools inhibits software development.[26] The result is that significant costs, whether in money or in needed volunteer labor, are imposed on projects for custom-building software solutions. This burden is particularly onerous in theological DH, where even projects backed by colleges and universities are often located in chronically under-resourced departments and many projects are driven by the volunteer efforts of professional practitioners, ranging from pulpit ministers with interest in formal theology to missionaries translating texts into indigenous languages.

Lowering the technical bar for participating in encoding was not the only advantage of ThML and OSIS. In many cases, they offered meaningful technical features for which TEI in its standard form does not provide, such as direct representation of verses as distinct from other structures, encoding of recurring dates and non-clock times for lectionaries, semantically accurate synchronization of Strong's numbers, semantic distinction of the Divine Name, detailed representation of hymns as components of larger textual settings, or close granularity in indexing diverse uses of scripture as distinct from other intertextual features. Twenty years on from Plantinga's original decision not to use TEI, the now-standard digital humanities markup language continues to miss specifics of theological texts, even for quite common features.

## FUTURE POSSIBILITIES

The way forward is not clear. Despite the advances made by the OSIS team, overlap of elements remains a serious problem without an elegant solution under existing XML frameworks, and it is only one of many stumbling blocks.[27] There are signs that XML itself may be reaching crucial limits in its development. In the meantime, however, it remains ubiquitous, and TEI continues to reign as *the* markup language of choice for textual encoding. Under these conditions, it would seem that a few useful lessons can be learned from theology-specific markup experiments.

There is real value to offering a streamlined markup with low technical overhead for learning—at least at its basic levels of execution. Given the nature of its constituencies, theological digital humanities work would likely benefit from a TEI customization scoped down to approximately the size and complexity of OSIS, but this would have to be, like OSIS itself, a coordinated effort. Basic agreement would have to be reached on the representation of verses, the encoding of overlap, and other such issues—agreements for which OSIS can perhaps offer a starting point in discussions. To this point, TEI customizations have largely been themed around the kind of material being represented, and it may be something of a shift to think of customizations in terms of a clientele served, or of the nature of the outputs to be produced. This can have not only technical benefits, however, but also cultural ones. Phrasing a markup language as general purpose can create the impression that it is for technical specialists, who bring that particular expertise to a variety of teams, and that it may largely be set aside by others. The presentation of a specifically *theological* markup language, however, suggests a disciplinary skill to be acquired by all serious participants in the discipline and may encourage more uptake of technical skills among scholars.

If TEI is to fully capitalize on the lessons of ThML and OSIS, however, it must not only be scaled back in some areas to focus on the particular purposes of theological work, but it must also be expanded in other areas to meet the unique needs of that work. TEI holds as an objective that it should provide modularity and extensibility to cover all kinds of textual encoding work. If that aim is to be realized, it would do well to look at the features that have made ThML and OSIS so stubbornly enduring within their narrow communities, and to take their longevity as proof of the distinctive value of many of their features. Some will be implementable through compliant customizations in the near future; others may await incorporation into a more distant P6 release. In either case, a TEI customization offering a smaller but more precisely defined toolkit for the

[26] Schmidt, "The Role of Markup," 128-129.
[27] Schmidt, "The Role of Markup," 129.

purposes of working with biblical and theological texts could finally allow TEI to fill the niches that ThML and OSIS have been forced to hold these many years and thus to enhance interoperability within the theological community, if not outside of it as well.

## REFERENCES


Adesulu, Dayo. "Meet UNILAG Undergraduate who Developed Wazobia Bible Application." *Vanguard*, March 27, 2014, https://www.vanguardngr.com/2014/03/meet-unilag-undergraduate-developed-wazobia-bible-application/.

Bible Technologies Group. 2013. *OSIS FAQs*. Retrieved from https://web.archive.org/web/20131107020309/http://www.bibletechnologies.net:80/vsItemDisplay.dsp&objectID=B25323FC-E8BB-49C2-A6C98E74A377F721&method=display.

Bulkeley, Tim. "Returning to E-commentary." *Sansblogue,* April 8, 2016, https://bigbible.org/sansblogue/bible/biblical-interpretation/returning-to-e-commentary/.

DeRose, Steven. "Markup Overlap: A Review and a Horse." *Proceedings of Extreme Markup Languages* 2004. http://xml.coverpages.org/DeRoseEML2004.pdf.

Drescher, Dennis. "XSEM: XML Scripture Encoding Model." *SIL*, September 6, 2001, https://scripts.sil.org/xsem.

Dubinsky, Leonid. "Digital Judaica Done Right." *DigitalJudaica*, http://www.digitaljudaica.org/judaica/html/index.html.

Durusau, Patrick. *OSIS Users Manual*. American Bible Society, 2006, https://web.archive.org/web/20120226061942/ http://img.forministry.com/7/7B/7BB51FB8-84B3-4FF3-939ED473FA90A632/DOC/OSIS2_1UserManual_06March2006_-_with_O'Donnell_edits.PDF.

Houghton, H. A. G. "The Electronic Scriptorium: Markup for New Testament Manuscripts." Pages 31–60 in *Digital Humanities in Biblical, Early Jewish, and Early Christian Studies*. Edited by Claire Clivaz, Andrew Gregory, and David Hamidović. Leiden: Brill, 2013.

Kooy, Brian K. "Christian Classics Ethereal Library." *Reference Reviews* 25.7 (2011): 11–12.

McDonough, Jerome. "XML, Interoperability and the Social Construction of Markup Languages: The Library Example." *digital humanities quarterly* 3.3 (2009), http://digitalhumanities.org/dhq/vol/3/3/000064/000064.html.

Plantiga, Harry. "ThML: Theological Markup Language for the Christian Classics Ethereal Library." *Christian Classics Ethereal Library*. February 23, 2001. http://www.ccel.org/ThML/ThML1.02.pdf.

Robie, Jonathan. "Querying Greek Texts in XML: Part 1." *BiblicalHumanities.org*. November 13, 2015. http://biblicalhumanities.org/xquery/tutorial/greek/2015/11/13/querying-000.html.

Schmidt, Desmond. "The End of XML." *Multi-Version Documents*. March 18, 2018. http://multiversiondocs.blogspot.com/2018/03/the-end-of-xml.html.

_____. "The Role of Markup in the Digital Humanities." *Historical Social Research / Historische Sozialforschung* 37 (2012): 125–46.

Society of Biblical Literature. "Biblical Scholars, Standards and the SBL." *SBL Forum*, 2.6 (2004). https://www.sbl-site.org/publications/article.aspx?ArticleId=45.

UCLA Center for Digital Humanities. "Text Encoding: Mark-up and TEI." *Intro to Digital Humanities: Concepts, Methods, and Tutorials for Students and Instructors.* September, 2013. http://dh101.humanities.ucla.edu/?page_id=60.

UNL Center for Digital Research in the Humanities. "Best Practices for Digital Humanities Projects." https://cdrh.unl.edu/articles/best_practices.