

<?xml-model href="http://www.tei-c.org/release/xml/tei/custom/schema/relaxng/tei\_jtei.rng" type="application/xml" schematypens="http://relaxng.org/ns/structure/1.0"?>

<?xml-model href="http://www.tei-c.org/release/xml/tei/custom/schema/relaxng/tei\_jtei.rng" type="application/xml" schematypens="http://purl.oclc.org/dsdl/schematron"?>

<TEI xmlns="http://www.tei-c.org/ns/1.0" rend="jTEI" xml:lang="en">

<teiHeader>

<fileDesc>

<titleStmt>

<title type="main">ATOP</title>

<title type="sub">Another TEI ODD Processor</title>

<author xml:id="SB">

<name>

<forename>Syd</forename>

<surname>Bauman</surname>

</name>

<affiliation> <roleName>Senior XML programmer/analyst</roleName> at the <orgName>Northeastern University Digital Scholarship Group</orgName> </affiliation>

<email>s.bauman@northeastern.edu</email>

</author>

<author xml:id="MH">

<name>

<forename>Martin</forename>

<surname>Holmes</surname>

</name>

<affiliation> <roleName>Programmer/analyst</roleName> at the <orgName>Humanities Computing and Media Centre</orgName> at the <orgName>University of Victoria</orgName> </affiliation>

<email>mholmes@uvic.ca</email>

</author>

<author xml:id="HBS">

<name>

<forename>Helena</forename>

<surname>Bermúdez Sabel</surname>

</name>

<affiliation>

<roleName>Software developer at Jinntec</roleName>

</affiliation>

<email>helena.bermudez@jinntec.de</email>

</author>

<author xml:id="DM">

<name>

<forename>David</forename>

<surname>Maus</surname>

</name>

<affiliation> <roleName>Head of Research &amp; Development</roleName> at the <orgName>State and University Library Hamburg</orgName> </affiliation>

<email>david.maus@sub.uni-hamburg.de</email>

</author>

</titleStmt>

<publicationStmt>

<publisher>TEI Consortium</publisher>

<date>2023</date>

<availability>

<licence target="https://creativecommons.org/licenses/by/4.0/">

<p>For this publication a Creative Commons Attribution 4.0 International license has been granted by the author(s), who retain full copyright.</p>

</licence>

</availability>

</publicationStmt>

<sourceDesc>

<p>No source, born digital.</p>

</sourceDesc>

</fileDesc>

<encodingDesc>

<appInfo>

<application ident="TEIGarage" version="1.1.1">

<desc> A <ref target="https://annuel2.framapad.org/p/TEI\_ATOP\_jTEI\_2022\_(Newcastle)\_paper">Framapad document</ref> was used as the original scratch pad for writing this paper. It was converted to Markdown using the Framapad export feature. That Markdown was converted to rudimentary TEI by TEIGarage 11.0.16+8, 1.1.1. </desc>

</application>

</appInfo>

</encodingDesc>

<profileDesc>

<langUsage>

<language ident="en">English, mostly using American spellings</language>

</langUsage>

<textClass>

<keywords>

<term>TEI</term>

<term>ODD</term>

<term>ATOP</term>

<term>XSLT</term>

<term>Stylesheets</term>

</keywords>

</textClass>

</profileDesc>

<revisionDesc>

<change who="#SB" when="2023-04-14T12:00:00">Started conversion to jTEI</change>

<change who="#SB" when="2023-04-14T11:30:00">Converted to TEI using TEIGarage</change>

<change who="#SB" when="2023-04-14T11:00:00">Finalized some edits in Framapad</change>

<change who="#MH #HBS #DM" from="2023-04-09" to="2023-04-13">Edited and made suggestions in Framapad</change>

<change who="#SB" when="2023-04-08">Wrote bulk of paper in Framapad</change>

</revisionDesc>

</teiHeader>

<text>

<front>

<div type="abstract" xml:id="abstract">

<p>The TEI XSL Stylesheets codebase is a tool that tackles the transformation of TEI XML documents to and from various formats. This includes the crucial activity of generating schemas from TEI ODD, the TEI XML–conformant specification format that allows one to write a schema language (for example, TEI P5 itself is written in ODD) or to customize TEI P5 using a literate programming approach. Because of the difficulties of maintaining this current set of stylesheets, a task force was created with the mission of developing, from scratch, an ODD processor that reads in a series of one or more TEI ODD customization files, merges them with a TEI language (likely, but not necessarily, TEI P5 itself), and generates RELAX NG and Schematron schemas. This paper presents a rationale for this task and the initial steps of this work.</p>

</div>

</front>

<body xml:id="atop">

<head>ATOP: Another TEI ODD Processor</head>

<p>This paper is an introduction to a TEI enterprise to create a utility for generating schemas (RELAX NG and Schematron) from descriptions of encoding systems written in TEI ODD, including chaining (i.e., customizations, customizations of customizations, etc.). The paper begins with <ref target="#terminology" type="crossref">a glossary</ref> of common terms used throughout the paper. <ref target="#rev" type="crossref">Section 3</ref> reviews what TEI ODD is and gives a broad overview of how it is currently processed. The <ref target="#hx" type="crossref">next section</ref> addresses why this is an important endeavor, explains how it came to be, and describes the very limited intended goals. <ref type="crossref" target="#work">Section 5</ref> discusses how the work is taking place and where it can be found. The <ref type="crossref" target="#plans">sixth section</ref> addresses our decisions and work so far. To conclude, we present some of the issues we have encountered with the current ODD language specifications (<ref target="#issues" type="crossref">section 7</ref>).</p>

<div xml:id="disclosures">

<head>Disclosures</head>

<p>Two of the authors, Syd Bauman and Helena Bermúdez Sabel, are currently members of the TEI Technical Council. A third, David Maus, is the author of <ref target="https://schxslt.github.io/schxslt/">SchXslt ([ʃˈɛksl̩t] — An XSLT-based Schematron processor)</ref>, mentioned below, which is quickly becoming the favorite Schematron processor not just in the TEI community, but in much of the XML world. The fourth, Martin Holmes, is the author of the first paper to directly address the problem head-on (<ref type="bibl" target="#holmes2021">Holmes 2021</ref>), and thus spurred the entire endeavor. All of the authors have been involved in trying to maintain the current TEI Stylesheets, and therefore have a vested interest in the success of this project.</p>

</div>

<div xml:id="terminology">

<head>Terminology</head>

<p>This is a list of terminology, acronyms, and initialisms used in this paper.</p>

<list>

<item><emph>ATOP</emph>: Another TEI ODD Processor. (<q>Another</q> because it is intended to be a replacement for the existing one within the TEI Stylesheets.)</item>

<item><emph>FLOSS</emph>: Free, Libre, Open-Source Software. All TEI software is FLOSS.</item>

<item><emph>ODD</emph> (One Document Does it All): <quote source="#teicODD1">a TEI XML format used to express schema fragments, prose documentation, and reference documentation for any XML markup scheme as a single document. It is the language in which the TEI is itself defined, and which should also be used to express a customization of the TEI scheme.</quote> <ref xml:id="teicODD1" target="#teicODD" type="bibl">TEI Consortium, n.d.b</ref> </item>

<item><emph>ODD Customization</emph>: an ODD file which describes modifications to a base schema, most commonly a TEI schema. </item>

<item><emph>PLODD</emph>: Pruned and Localized ODD. An ATOP-specific term referring to the final state of an ODD customization undergoing transformation, immediately before it is turned into RELAX NG and Schematron schemas.</item>

<item><emph>tagset</emph>: an inventory of elements, attributes, and values constituting a custom markup language.</item>

<item><emph>RELAX NG</emph>: an industry-standard grammar-based schema language. RELAX NG schemas can be generated from TEI ODD files.</item>

<item><emph>Roma/roma</emph>: Roma (capital R) is a web-based service that provides a user-friendly interface for creating ODD customization files and generating schemas and documentation from them; roma (lower-case r) is a command-line tool for generating schemas and documentation from ODD customizations.</item>

<item><emph>Schematron</emph>: a validation language based on assertions about document structure and patterns. Schematron serves as a useful adjunct to RELAX NG, and is also supported by ODD.</item>

<item><emph>TEI Stylesheets</emph>: a library of tools for converting between TEI and other document formats, primarily using XSLT, originally created by Sebastian Rahtz, and now maintained by the TEI Technical Council and other volunteers.</item>

<item><emph>TEIGarage</emph>: a web service which offers various transformations to and from TEI, based on the TEI Stylesheets.</item>

<item><emph>XSLT</emph>: eXtensible Stylesheet Language Transformations, an XML language designed to read in an XML document and write out a transformed version of it. For example, XSLT can be used to transform TEI into other document formats.</item>

<item><emph>XSpec</emph>: an XML language for writing and running unit tests for XSLT, Xquery, and Schematron.</item>

</list>

</div>

<div xml:id="rev">

<head>TEI ODD Processing: A Broad Overview</head>

<p>The TEI system is, in general, a tagset for describing the structure of, and other interesting information relating to, extant texts. While it is intended to promote the interchangeability and processability of transcriptions of texts of interest to humanities scholars, it can be used for a variety of other text-encoding purposes. For example, this paper is written in TEI. But a portion of the TEI system (TEI Consortium 2023, <ref target="https://tei-c.org/Vault/P5/4.7.0/doc/tei-p5-doc/en/html/TD.html">chapter 22: Documentation Elements</ref>) is dedicated not to describing extant texts, but rather to describing markup languages intended for describing extant texts (or anything else, for that matter). Thus, because TEI is currently represented in XML, ODD has elements and attributes for describing an XML encoding language. For example, ODD has constructs for:</p>

<list rend="bulleted">

<item>describing elements and their attributes;</item>

<item>providing content models for elements;</item>

<item>constraining attribute values, including with controlled vocabularies;</item>

<item>providing examples and discussion of the use of elements and attributes;</item>

<item>organizing elements and attributes into classes, for easier use.</item>

</list>

<p>The TEI system itself is, of course, described using this subsystem, which is affectionately called ODD (see <ref target="#burnard2004" type="bibl">Burnard and Rahtz 2004</ref>, <ref target="#rahtz2013" type="bibl">Rahtz and Burnard 2013</ref>, <ref target="#burnard2013" type="bibl">Burnard and Rahtz 2013</ref>, <ref target="#teicODD" type="bibl">TEI Consortium, n.d.b</ref>).</p>

<p>While ODD can be used for a variety of purposes, its two main uses are (1) defining the TEI Guidelines, and (2) creating customizations of the TEI tagset (see <ref type="bibl" target="#bauman2019">Bauman 2019</ref>). The remainder of this section is an overview of the current system for generating schemas and documentation from TEI ODD customization files.</p>

<p>The <title>TEI Guidelines</title> are written in TEI ODD. That is, the source for the <title>Guidelines</title> you read on the web or via an ePub or PDF viewer is a single TEI document. It has a <gi>TEI</gi> start-tag at the beginning, a <gi>TEI</gi> end-tag at the end, and a <gi>teiHeader</gi> just like any other TEI document. It is filled with <gi>div</gi>, <gi>head</gi>, and <gi>p</gi> elements as you might expect. But it also contains <gi>elementSpec</gi> elements for describing elements and <gi>moduleSpec</gi> elements for grouping those element specifications, and a host of other constructs dedicated to describing an XML markup language in detail. This single TEI document is stored in over eight hundred separate files for the convenience of those maintaining it, but those files comprise a single TEI document.</p>

<p>This single TEI document that defines an XML language is not intended to be used on its own, however. This is in large part because the TEI covers many fields of study, and a user from one field may not be interested in the TEI elements and attributes that support another. Thus a customization file—also written in TEI ODD—is used in conjunction with the TEI Guidelines. The customization file expresses which parts of TEI will be used, and which will not, and adds additional explanations, constraints, and examples where needed. One of the most widely-used customizations is TEI Lite, a large and complex customization that produces a slimmed-down version of TEI intended <quote source="#teiLite">to meet 90% of the needs of 90% of the TEI user community</quote> (<ref target="#teicFAQ" type="bibl" xml:id="teiLite">TEI Consortium, n.d.a, <q>What is TEI Lite?</q></ref>). A customization that is very commonly used for experimentation (but not appropriate for use in actual encoding projects) is TEI All, a very small customization that simply says <q>use all bits of TEI without any further explanations, examples, or constraints</q>.</p>

<p>Given a TEI customization, the main tasks we wish to accomplish are to convert the combination of the TEI customization and the TEI source document into (1) customized documentation and (2) usable schemas. Customized documentation is a version of the <title>TEI Guidelines</title>, or at least of the reference section, that reflects the changes expressed in the customization ODD. For example, if the customization ODD says <q>I only need the required modules plus the <gi>metDecl</gi> element</q>—that is, it does not include the specialized modules for drama, feature structures, manuscript description, etc.—then only ~61 different element types would be listed as possible children of <gi>p</gi>, and only ~197 as members of the class <ident>att.global</ident>. This makes documentation much easier to read when compared to TEI All, in which ~207 elements would be listed as possible children of <gi>p</gi>, and ~590 as members of the class <ident>att.global</ident>.</p>

<p>We need to generate usable schemas from the TEI system because no software exists that will directly validate a document against an XML language written in TEI ODD, let alone provide assistance in writing a document that conforms to it. The TEI ODD language is specifically designed to be converted to schema languages for which such software is commonly available: RELAX NG (for the closed schema language constructs) and ISO Schematron (for the open schema language constructs). Furthermore, the ODD language can be converted to other languages. </p>

<p>Following the principles of literate programming (<ref target="#knuth1984" type="bibl">Knuth 1984</ref>), the process of generating custom documentation from a customization file and the TEI Guidelines is called <hi rend="italic">weaving</hi>, and the process of generating schemas is called <hi rend="italic">tangling</hi>. The TEI Consortium provides software (called <name>roma</name>, not to be confused with <name>Roma</name>,<note>See <ref target="https://roma.tei-c.org/">https://roma.tei-c.org/</ref>.</note> the web interface to that software) which performs these tangle and weave functions. See <ptr target="#fig01" type="crossref"/>.</p>

<figure xml:id="fig01">

<graphic url="tei\_odd\_customization\_detail\_iconed.png" width="676px" height="538px"/>

<head type="legend">Generation of customized documentation and schemas from TEI</head>

</figure>

</div>

<div xml:id="hx">

<head>History: Who, What, and Why</head>

<p>The software currently used to convert TEI ODD into custom documentation and custom schemas is a large set of XSLT stylesheets commonly referred to by the somewhat generic name <name>the TEI Stylesheets</name>.<note>You might think they are called that because the name of the GitHub repository where they live is <name>Stylesheets</name>, but in fact the repository is named after the software.</note> The Stylesheets also convert TEI into many other formats (including DocBook, HTML, PDF, NLM, and OpenDocument), and many of those formats into TEI. These stylesheets (<ref target="https://github.com/TEIC/Stylesheets/">https://github.com/TEIC/Stylesheets/</ref>) are both an extraordinarily impressive accomplishment (that of Sebastian Rahtz, to be precise) and somewhat of an obstacle to the growth of the TEI language. The Stylesheets are magnificent in that they perform a large set of intricate tasks reasonably well, and in that they were written very quickly. However, they are very difficult to maintain. The source of the problem was summed up by Martin Holmes in 2021: <quote source="#holmes">Sebastian was such a genius he really found it impossible to imagine how dim the rest of us actually are.</quote> For a detailed discussion of the problems the Stylesheets pose see the rest of Holmes’s paper (<ref type="bibl" target="#holmes2021" xml:id="holmes">Holmes 2021</ref>). But to summarize here for our purposes:</p>

<list rend="bulleted">

<item>The Stylesheets are <emph>very</emph> complex. This is in large part because they do a <emph>lot</emph> more than convert TEI ODD into custom documentation and custom schemas, which are the core activities the TEI Technical Council needs to support.</item>

<item>They are written in an outdated dialect of XSLT (basically XSLT 1.0 with a few 2.0 constructs). XSLT 2.0 and 3.0 provide many new constructs which contribute to clarity, concision, and easier documentation.</item>

<item>They are full of redundancy (e.g., there are two functions named <code>tei:includeMember()</code> which are used during ODD processing that do exactly the same thing).</item>

<item>They are almost completely undocumented.</item>

</list>

<p>Thus fixing bugs in, let alone adding new features to, the Stylesheets is quite difficult, in part because it is very difficult to be confident that a change does not introduce other problems. Furthermore, it is common to find new bugs when fixing known bugs. David Maus went so far as to say in an issue comment <quote source="#comment">I consider the Stylesheets as broken beyond repair. They work … to a certain degree[,] but they are, from my point of view, effectively unmaintainable.</quote> (<ref target="#maus2021" type="bibl" xml:id="comment">Maus 2021</ref>)</p>

<p>The natural consequence to the Stylesheets being simultaneously so difficult to maintain and being so central to the TEI Technical Council’s mission is that the Council spends a lot of its time trying to address issues in the Stylesheets. There is a monthly meeting devoted to the task, and it is not nearly enough. It is worth mentioning, though, that people other than those on the Council can contribute to the Stylesheets. In fact, the Stylesheets repository was deliberately (and some would say unwisely) separated from the repository used for the TEI Guidelines themselves in order to encourage and foster work on the Stylesheets by programmers other than those on the Council. This has not been entirely successful, though. Fewer than 4% of commits to the primary development branch are by people who have never served on the Council.</p>

<p>Two of the authors (Bauman and Holmes) had been fantasizing about rewriting the Stylesheets from scratch for several years. In 2021 at the virtual TEI conference Holmes presented the paper cited above (<ref type="bibl" target="#holmes2021">Holmes 2021</ref>), laying out the argument that the Stylesheets are becoming a hindrance to progress within the TEI. It seemed that most Council members were quite sympathetic to Holmes’s position. Bauman saw this as an opportunity, and asked the Council to create a task force for the purpose of creating a new ODD processor (ATOP) from scratch that generates RELAX NG and Schematron schemas from the ODD language. This deliberately left out many of the tasks the Stylesheets perform. However, the Council feels that the only tasks performed by the Stylesheets that are core parts of its mission are (1) generation of the Guidelines themselves, (2) generation of custom documentation, and (3) generation of custom schemas. But even considering only those three core tasks, two are not covered by the charge to the new task force. This is for a variety of reasons, the largest of which is that another group is tackling the other activities that are part of the Council’s core mission: generating the Guidelines themselves and generating customized documentation.</p>

<p>Thus ATOP will be an ODD processor that takes as input a chain of one or more ODD files and produces as output both a RELAX NG schema (representing the grammar expressed by the chained ODDs) and a Schematron Schema (representing the additional ISO Schematron constraints in the chained ODDs), although the Schematron constraints may be embedded in the RELAX NG, thus creating only a single file of output. That is to say, ATOP will not even perform ODD to DTD or ODD to W3C Schema processing, let alone TEI Lite to ePub3, DOCX to TEI, or ODD-with-Processing-Model to stylesheets.</p>

</div>

<div xml:id="work">

<head>Work Processes</head>

<p>Work on ATOP is being performed by a task force that consists of the four authors. While we have no hard limit to how often or how rarely we meet, so far we have been meeting for one hour per week via videoconference call (with screen share capability), and have found this pace quite comfortable. We also occasionally make use of a dedicated channel in the TEI Slack space.</p>

<p>Almost all work takes place on a TEIC GitHub repository (<ref target="https://github.com/TEIC/atop">https://github.com/TEIC/atop</ref>). This repository includes the code (including tests), issues and pull requests, a glossary of terms, documentation of goals, namespaces, error codes, and minutes of our meetings.</p>

<p>We have deliberately set no hard deadlines for ourselves, putting the goal of maintainable code over and above the goal of finishing our work rapidly. (After all, the world has been making do with the current Stylesheets for many years). We had an aspirational goal of having working code for the Council to start testing in summer 2023, and an initial full release by the end of 2023. However, it neither surprises nor upsets us that we have not met those temporal goals. We released an interim package in October 2023 which may already be useful to some ODD creators: see <ref target="#deliverables" type="crossref">section 6.5</ref> for further details.</p>

</div>

<div xml:id="plans">

<head>Plans and Progress</head>

<div xml:id="comparison">

<head>Comparison to Current Process</head>

<p>The current Stylesheets include over 330 XSLT files; of those only approximately 25 to 33 are part of the process we are duplicating. However, the division of labor among these is, at best, confusing, and has been described (several years after the death of the primary author) on a TEI wiki page (<ref target="https://wiki.tei-c.org/index.php?title=Mapping\_ODD\_processing">https://wiki.tei-c.org/index.php?title=Mapping\_ODD\_processing</ref>).</p>

<p>Another problem with the current suite is inconsistent terminology. In order to try to avoid this pitfall, we agreed to a defined glossary of terms that we developed within the first month of the project (<ref target="https://github.com/TEIC/atop/wiki/Terminology">https://github.com/TEIC/atop/wiki/Terminology</ref>).</p>

<p>We anticipate that ATOP will not be <emph>nearly</emph> as complicated as the current suite, in large part because it will do so much less. It <emph>may</emph> end up as complicated as the portion of the current suite that handles the same process, but probably not; and even if it does, it will be well-documented, clear, comparatively easy-to-understand code.</p>

</div>

<div xml:id="workflow">

<head>Workflow</head>

<p>Our intended workflow will involve at least four stages (see <ptr target="#fig02" type="crossref"/>). Each stage will likely begin with a validation step. In most cases invalidity would result in the entire process being aborted. There may be some cases for which a warning will be issued and processing continues. One advantage to this methodology is that we will, by necessity, generate a rigorous definition of a derived ODD file and of a Pruned Localized ODD (PLODD). It may mean that intermediate files need be written out at every stage (which some might consider a disadvantage); however, writing those intermediate files will be very useful in debugging the system.</p>

<figure xml:id="fig02">

<graphic url="datapath.png" width="184px" height="497px"/>

<head type="legend">Stages for generating schemas from a customization</head>

</figure>

<list>

<item>Stage 1: <hi rend="bold">Assemble Customization ODD</hi>. References to external components, if any, are read into the ODD; prose may be dropped at this stage (but most metadata and all glosses and descriptions are kept).</item>

<item>Stage 2: <hi rend="bold">Customization ODD + Base ODD → Derived ODD</hi>. The output of stage 1 is read in and whatever base ODD it refers to is validated and read in. The ODDs are combined by following the instructions in the assembled customization ODD and applying them to the base ODD. All internationalization information (i.e., all languages used for glosses, descriptions, and alternate identifiers) is kept. The output is referred to as a <term>derived ODD</term>.</item>

<item>Stage 3: <hi rend="bold">Customization ODD + Derived ODD → Derived ODD</hi>. This stage is very similar to stage 2, except that the customization ODD used is the next ODD in the chain, and the output of the previous stage is used as if it were the base ODD. This stage is repeated 0 or more times as needed.</item>

<item>Stage 4: <hi rend="bold">Derived ODD → Pruned Localized ODD</hi>. The output of the last stage is read in. It is pruned, in that most if not all information that is not needed for the creation of a schema is removed. It is localized, in that all but one language for each set of glosses, descriptions, and alternate identifiers is selected, and the rest are removed. The output is referred to as a PLODD.</item>

<item>Stage 5: <hi rend="bold">PLODD → RELAX NG + Schematron</hi>. The output of the previous stage is validated and (if it passes) read in, and transpiled into a RELAX NG schema with embedded ISO Schematron rules.</item>

</list>

<p>The user is welcome to try to convert the RNG to DTD or XSD, but such is out of scope for ATOP; our goal is only to generate RELAX NG and Schematron schemas.</p>

</div>

<div xml:id="consistency">

<head>Consistency and Conventions</head>

<p>We have spent some time developing a set of coding conventions to help ensure consistency of the code we produce. These conventions focus mostly on naming and usage. For example, if a subroutine is going to return a node, we use a template for that routine; if it returns something other than a node, we use a function.</p>

<p>We are using XML validation technology (in particular, Schematron) to enforce addition of documentation, along with consistency in our coding conventions:</p>

<list>

<item>of variable names (e.g., <code>$atop:vVersion</code>)</item>

<item>of parameter names (e.g., <code>$atop:pPatternPrefix</code>, <code>$atop:tpFun</code>)</item>

<item>of function names (e.g., <code>atop:unique-ident()</code>)</item>

<item>in global use of <code>@as</code> attribute</item>

</list>

</div>

<div xml:id="testing">

<head>Component Testing</head>

<p>We test each component rigorously while it is being developed. This involves developing an XSpec suite to test functions and named templates, and perhaps for larger components. We are also working on a sizable suite of test ODDs. We have already gathered, through a request on TEI-L, well over fifty customization ODDs written by TEI users—our in vivo test suite. In addition, we plan to generate both language ODDs and customization ODDs specifically designed to test particular features of the ODD language—our in vitro test suite. Furthermore, since we have (deliberately) written the stage 5 program first, we are writing another in vitro test suite for processing of PLODDs.</p><!-- EDITED TO HERE. -->

<p>When testing the final RELAX NG result, we plan to perform both comparison testing (checking that the output file matches a previously agreed-upon expected-output file) and behavioral testing (checking that the output file flags a known-to-be-correct file as valid, and a known to be incorrect file as invalid). The latter is necessary because there are often multiple RELAX NG patterns that will match the same XML constructs. For example, the content models <code>( blort+ )</code>, <code>( blort, blort\* )</code>, and <code>( blort, blort?, blort\* )</code> all mean the same thing: <q>one or more <gi>blort</gi>s</q>.</p>

</div>

<div xml:id="deliverables">

<head>Deliverables</head>

<p>The main deliverable will be a set of XSLT programs that convert a chain of one or more ODDs into RELAX NG and Schematron. Ancillary deliverables will be a driver or glue program for an end user to run, a schema with which to test the input document(s), a schema with which to test the intermediate document(s) (although they may be hidden from users unless they request them), and documentation of the entire system. Each of these will, of course, be FLOSS, probably under the same license as other TEI software.</p>

<p>The main programs will be written in XSLT 3 (using XPath 3.1) without any extended functionality. Even if and when XSLT 4 is available, we plan to stick with XSLT 3. These programs should run in any standard XML-friendly environment, including the GNU/Linux command line and oXygen. We make use of oXygen, and the end product should run easily in oXygen, but nothing should <hi rend="italic">require</hi> oXygen.</p>

<p>The <soCalled>glue</soCalled> or <soCalled>driver</soCalled> program that tests the input and runs the required XSLT programs in the right order will probably be written in Apache Ant.</p>

<p>In October 2023 we released an interim package containing components to run the final stage of the process (<ref target="https://github.com/TEIC/atop/releases">https://github.com/TEIC/atop/releases</ref>). As mentioned above, besides the transformation files, the release includes the schemas that enable a user to test whether the input files conform to the PLODD specification before running the transformation. The released components may already be useful to ODD creators who directly author complete ODD files rather than starting from a customization of P5. </p>

</div>

</div>

<div xml:id="issues">

<head>Issues with the Current ODD Language Specifications</head>

<p>It is not surprising that our work runs into places where ODD must be more clearly specified. A brief commentary on some of these issues follows:<note>These examples are taken from a presentation by the authors delivered at the joint MEC (Music Encoding Conference) and TEI Conference 2023 (Paderborn, September 6–9, 2023). See <ref type="bibl" target="#baumanetal2023">Bauman et al. 2023.</ref></note>

<list> <label>Schematron query language binding</label>

<item>There is no mechanism in the ODD language for users to specify the query-binding language used in their Schematron contexts and tests. Driven by the ATOP Task Force, a new element, <gi>constraintDecl</gi>, will be created. </item>

<label>Duplicate definitions of the same attribute</label>

<item>An XML element cannot have two attributes with the same name<?oxy\_delete author="heb" timestamp="20240211T144110+0100" content=" "?>, but the current ODD schema allows sibling <gi>attDef</gi> elements with the same <att>ident</att>.</item>

<label>Documentation elements allowed almost anywhere</label>

<item>The class <ident>model.oddDecl</ident> is a member of <ident>model.inter</ident> (which groups elements that can appear either within or between paragraph-like elements). This means that, for example, the element <gi>rdg</gi> (reading) may contain elements such as <gi>classSpec</gi>, <gi>constraintSpec</gi>, <gi>dataSpec</gi>, or <gi>elementSpec</gi>. </item>

<label>Schematron Contexts</label>

<item>Schematron constraints depend on the definition of two XPath patterns: the <emph>context</emph> and the <emph>test</emph>. The context is defined in the <att>context</att> attribute of the element <gi>sch:rule</gi>, while the tests are defined within the elements <gi>sch:assert</gi> or <gi>sch:report</gi> which are direct descendants of <gi>sch:rule</gi>. In ODD, <gi>sch:rule</gi> is optional, on the basis that if the context is not explicit, it should be inferred by the processor. However, the TEI Guidelines do not specify how the context should be inferred when it is not explicit, and in some contexts, unambiguous inference is impossible.</item>

</list>

</p>

<p>We are raising issues in the TEI (<ref target="https://github.com/TEIC/TEI/labels/atop">https://github.com/TEIC/TEI/labels/atop</ref>) and Stylesheets (<ref target="https://github.com/TEIC/Stylesheets/labels/atop">https://github.com/TEIC/Stylesheets/labels/atop</ref>) GitHub repositories as we work. In some cases our findings will likely result in changes that require fixes to previous stylesheets. </p>

</div>

</body>

<back>

<div type="bibliography">

<listBibl>

<bibl xml:id="bauman2019"><author>Bauman, Syd</author>. <date>2019</date>. <title level="a">A TEI Customization for Writing TEI Customizations</title>. <title level="j">Journal of the Text Encoding Initiative</title>, <biblScope unit="issue">12</biblScope>. <ref target="<http://journals.openedition.org/jtei/2573>"><http://journals.openedition.org/jtei/2573></ref>. doi:<idno type="doi">10.4000/jtei.2573</idno>.</bibl>

<bibl xml:id="baumanetal2023"><author>Bauman, Syd</author>, <author>Helena Bermúdez Sabel</author>, <author> David Maus</author>, and <author> Martin Holmes</author>. <date>2023</date>. <title level="a">ODD ODDities: Some Problems with the ODD Language Unearthed by an Attempt to Write a Processor</title>. In <title level="m"><q>Encoding Cultures</q>: Joint MEC TEI Conference 2023. Book of Abstracts</title>, edited by Raffaele Viglianti, Johannes Kepper, Peter Stadler, and Joachim Veit, <biblScope unit="page">185–87</biblScope>. <ref target="https://doi.org/10.5281/zenodo.10427826">https://doi.org/10.5281/zenodo.10427826</ref>.</bibl>

<bibl xml:id="burnard2004"><author>Burnard, Lou</author>, and <author> Sebastian Rahtz</author>. <date>2004</date>. <title level="a">RelaxNG with Son of ODD</title>. In <title level="m">Proceedings of the Extreme Markup Languages® 2004</title>. <ref target="https://web.archive.org/web/20061113180648/http://www.mulberrytech.com/Extreme/Proceedings/html/2004/Burnard01/EML2004Burnard01.html">https://web.archive.org/web/20061113180648/http://www.mulberrytech.com/Extreme/Proceedings/html/2004/Burnard01/EML2004Burnard01.html</ref>.</bibl>

<bibl xml:id="burnard2013"><author>Burnard, Lou</author>, and <author> Sebastian Rahtz</author>. <date>2013</date>. <title level="a">A Complete Schema Definition Language for the Text Encoding Initiative</title>. In <title level="m"> XML London 2013: Conference Proceedings</title>, <biblScope unit="page">152–61</biblScope>. London: XML London. <ref target="<http://xmllondon.com/2013/xmllondon-2013-proceedings.pdf>"><http://xmllondon.com/2013/xmllondon-2013-proceedings.pdf></ref>. doi:<idno type="doi">10.14337/XMLLondon13.Rahtz01</idno>.</bibl>

<bibl xml:id="holmes2021"><author>Holmes, Martin</author>. <date>2021</date>. <title level="a">Are the TEI Stylesheets Really Broken beyond Repair?</title> Text Encoding Initiative Conference (Online), October 25, 2021. <ref target="https://zenodo.org/record/6522554">https://zenodo.org/record/6522554</ref>.</bibl>

<bibl xml:id="knuth1984"><author>Knuth, Donald E.</author> <date>1984</date>. <title level="a">Literate Programming</title>. <title level="j">The Computer Journal</title> (British Computer Society) 27 (2): 97–111. <ref target="http://www.literateprogramming.com/knuthweb.pdf">http://www.literateprogramming.com/knuthweb.pdf</ref>; <idno type="doi"> doi:10.1093/comjnl/27.2.97</idno>.</bibl>

<bibl xml:id="maus2021"><author>Maus, David</author>. <date>2021</date>. Comment on pull request 477, <title level="a">Gracefully fail if target if @facs does not exist,</title> November 5, 2020. TEIC Stylesheets GitHub site. <ref target="[https://github.com/TEIC/Stylesheets/pull/477#issuecomment-805138553](https://github.com/TEIC/Stylesheets/pull/477" \l "issuecomment-805138553)">https://github.com/TEIC/Stylesheets/pull/477#issuecomment-805138553</ref>.</bibl>

<bibl xml:id="rahtz2013"><author>Rahtz, Sebastian</author>, and <author>Lou Burnard</author>. <date>2013</date>. <title level="a">Reviewing the TEI ODD System</title>. <title level="m">DocEng ’13: Proceedings of the 2013 ACM Symposium on Document Engineering</title>, 193–96. New York: ACM. <ref target="https://doi.org/10.1145/2494266.2494321">https://doi.org/10.1145/2494266.2494321</ref>.</bibl>

<bibl xml:id="tei2023"><author>TEI Consortium</author>. <date>2023</date>. <title level="m">TEI P5: Guidelines for Electronic Text Encoding and Interchange</title>. Version 4.7.0. Last updated November 16. N.p.: TEI Consortium. <ref target=" https://tei-c.org/Vault/P5/4.7.0/doc/tei-p5-doc/en/html/"> https://tei-c.org/Vault/P5/4.7.0/doc/tei-p5-doc/en/html/</ref>.</bibl>

<bibl xml:id="teicFAQ"><author>TEI Consortium</author>. <date>n.d.a</date>. <title level="a">Frequently Asked Questions</title>. Accessed March 27, 2024. <ref target="<https://tei-c.org/about/frequently-asked-questions/>">https://tei-c.org/about/frequently-asked-questions/</ref>.</bibl>

<bibl xml:id="teicODD"><author>TEI Consortium</author>. <date>n.d.b</date>. <title level="a">Getting Started with P5 ODDs.</title>. Accessed March 27, 2024. <ref target="https://tei-c.org/guidelines/customization/getting-started-with-p5-odds/">https://tei-c.org/guidelines/customization/getting-started-with-p5-odds/</ref>.</bibl>

</listBibl>

</div>

</back>

</text>

</TEI>