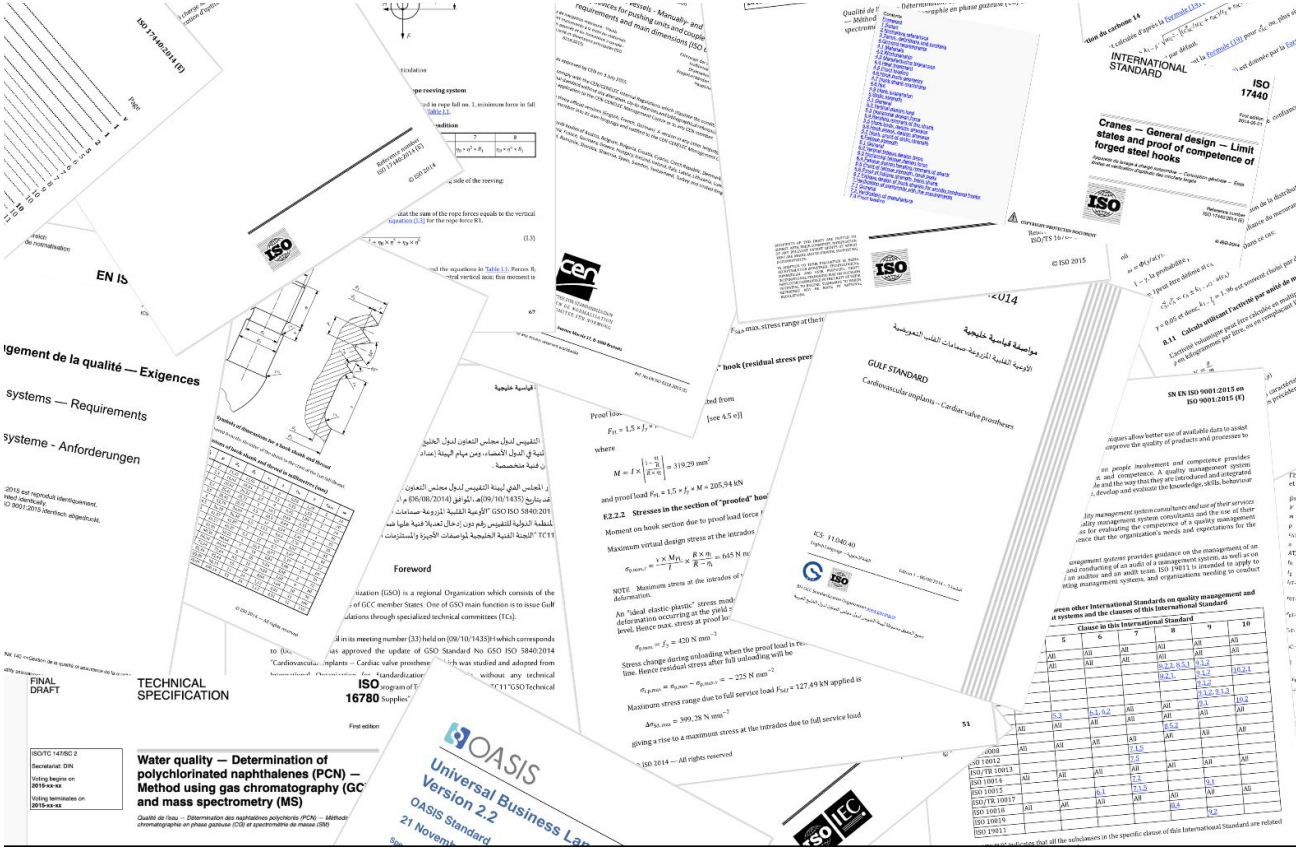
Using XSL-FO and AHRTS at RealtaOnline.com

G. Ken Holman

The world of publishing standards documents to PDF/HTML



Using XSL-FO and AHRTS at RealtaOnline.com

XML Prague 2022 - XSO-FO, CSS, and Paged Output Meetup

20220527-1450z

Publishing services to National Bodies

- when making standards available to their national customers, national bodies must add national-oriented cover pages to content downloaded from ISO, IEC, CEN, and CENELEC feeds
- source standards documents available in PDF, ISO-STS XML, and NISO-STS XML
- publishing management automation improves the attended or unattended process of fetching and publishing standards with minimal human intervention to trigger the process and review results
- company provides implementation support to third-party providers of services to national bodies
- company donates server time to OASIS Open Technical Committees and editors using DocBook

Input XML ISO-STS, NISO-STS, and Réalta handling of adoptions

```
<standard>
  <front>
    <iso-meta originator="xxx">
      ... international metadata ...
    </iso-meta>
    <reg-meta originator="yyy">
      ... regional metadata ...
    </reg-meta>
    <nat-meta originator="zzz">
      ... national metadata ...
    </nat-meta>
    <sec originator="{xxx/yyy/zzz}">
      ...
    </sec>
    <sec originator="{xxx/yyy/zzz}">
      ...
    </sec>
  </front
  <body>
    <sec>... int'l content ...<sec>
  </body>
</standard>
```

ISO-STS serial declaration of adoptions of the content.

```
<adoption>
  <adoption-front>
    <std-meta>
      ... national metadata ...
    </std-meta>
    <sec>
      ... national front matter ...
    </sec>
  </adoption-front>
  <adoption>
    <adoption-front>
      <std-meta>
        ... regional metadata ...
      </std-meta>
      <sec>
        ... regional front matter ...
      </sec>
    </adoption-front>
    <standard>
      <front>
        <std-meta>
          ... international metadata ...
        </std-meta>
        <sec>
          ... international front matter ...
        </sec>
      </front>
      <body>
        <sec>... int'l content ...<sec>
      </body>
    </standard>
  </adoption>
</adoption>
```

NISO-STS wrapping approach for adoptions of the content.

```
<adoption>
 <adoption-front>
  <std-meta>...national metadata...</std-meta>
  <sec>...national front matter...</sec>
 </adoption-front>
 <adoption>
  <adoption-front>
   <std-meta>
    <custom-meta-group originator="realta">
     <custom-meta>
      <meta-name>realta-fetch cen xml lang-en
              back-after-adoption</meta-name>
      <meta-value>73679</meta-value>
     </custom-meta>
    </custom-meta-group>
   </std-meta>
  </adoption-front>
  <standard>
   <front>
    <std-meta>
     <custom-meta-group originator="realta">
      <custom-meta>
       <meta-name>realta-merge pdf</meta-name>
       <meta-value>iso-12345.pdf</meta-value>
      </custom-meta>
     </custom-meta-group>
    </std-meta>
   </front>
   <body/>
  </standard>
 </adoption>
</adoption>
```

Réalta directives to fetch from feeds or merge local PDF or XML

Using a footnote to build content from bottom of the page

GSO ISO 7176-19:2008

ISO 7176-19:2008

مواصفة قياسية خليجية

الكراسي المتحركة – الجزء التاسع عشر: الأجهزة المتحركة على عجلات المستخدمة كمقاعد في السيارات

Gulf Standard

Wheelchairs — Part 19: Wheeled mobility devices for use as seats in motor vehicles

ICS: 11.180.10

English Language – اللغة الإنجليزية Edition 2 – 2 الطبعة

All rights reserved.

BH GSO BSI ISO/IEC ASTM IEEE OIML SASO KWS UAE.S OS QS 123:2014

مواصفة قياسية خليجية

ISO 5 840:2005

الأوعية القلبية المزروعة-صمامات القلب التعويضية

Gulf Standard

Cardiovascular implants--Cardiac valve prostheses

ICS: 11.040.40

English Language – اللغة الإنجليزية Edition 1 – 06/08/2014 – 1 الطبعة

BH GCC Standardization Organization www.gso.org.sa

جميع الحقوق محفوظة لهيئة التقييس لدول مجلس التعاون لدول الخليج العربية

GSO ISO 7176-19:2008
ISO 7176-19:2008 (E)

Gulf Standard

COPYRIGHT PROTECTED DOCUMENT

© ISO 2008 © GSO 2010

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.ch
Website www.iso.ch

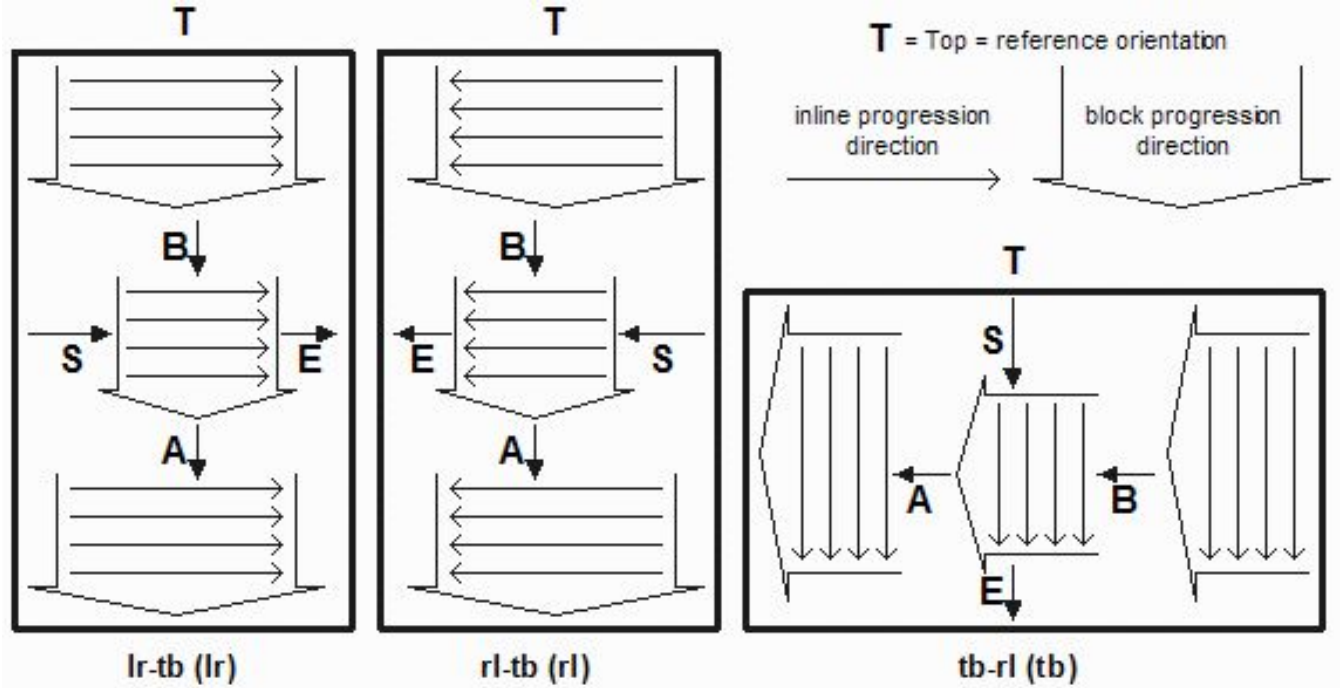
Published in Switzerland

ii

© ISO 2008 © GSO 2010 — All rights reserved

This approach allows arbitrary height of bottom content to hug the bottom of the body region.

"start", "end", "before", and "after", never "left", "right", "top", and "bottom"



Beware unexpected impact of font metrics

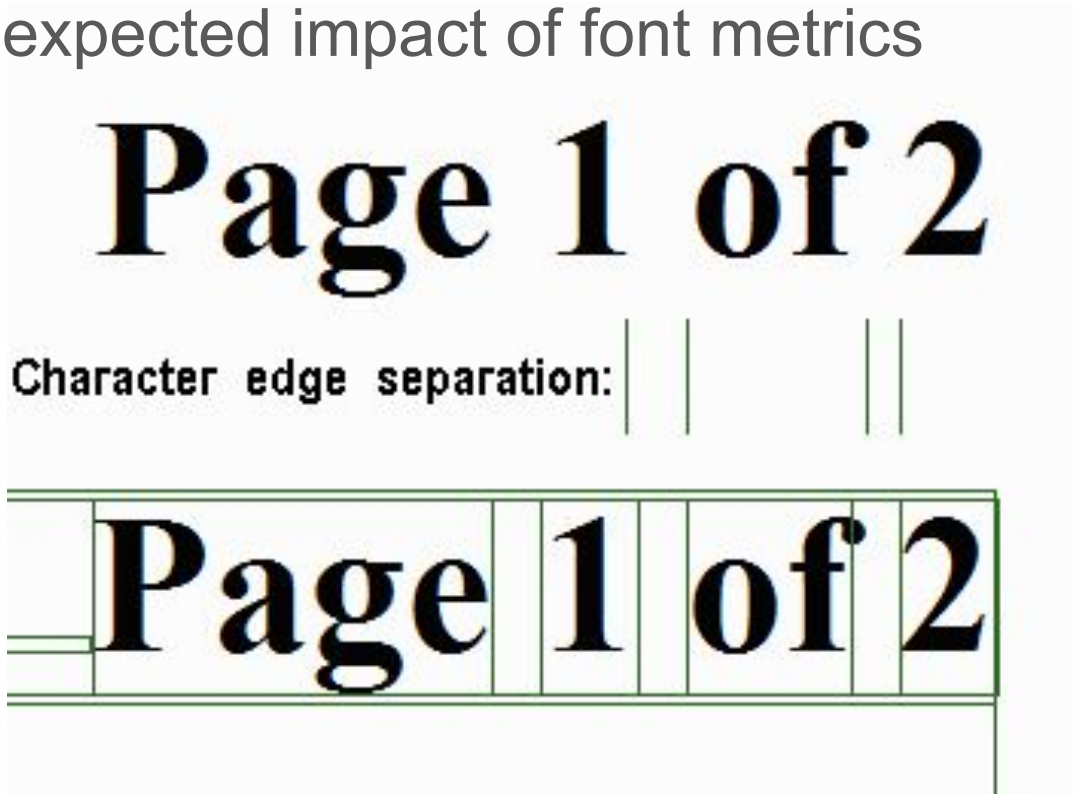
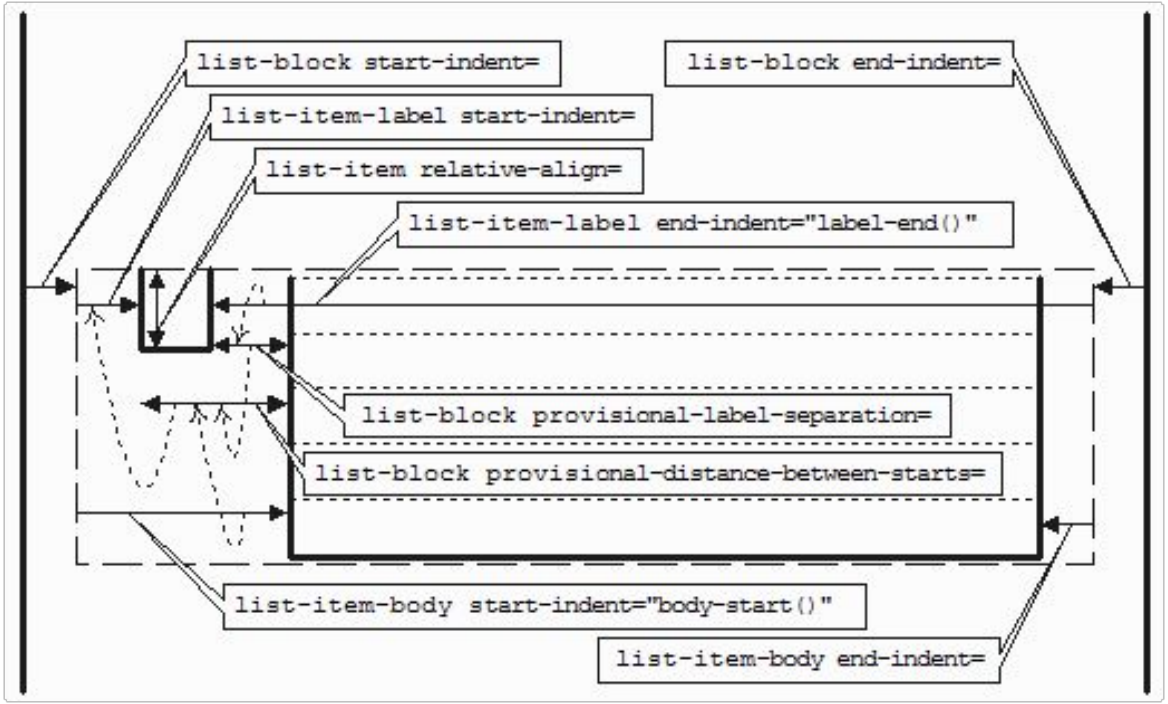


Image captured using Antenna House interactive border reveal feature

Using list properties for consistent list labeling

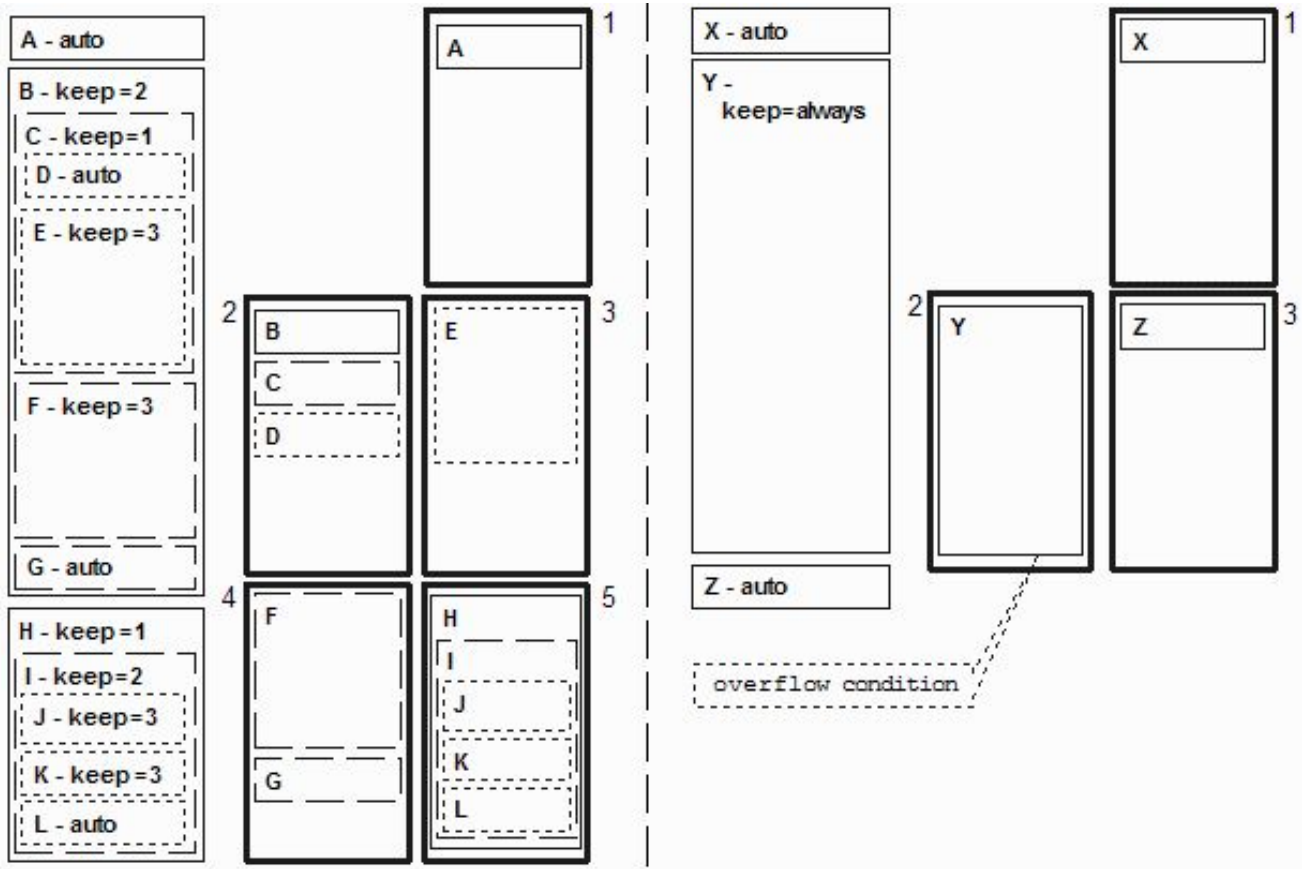


The list label remains consistent regardless of any changing in the width of the column.

Keeps can be very important

Some customers want to avoid breaking in tables and multiple-image graphics, thus needing to use keeps.

e.g. A low strength keep for a set of two graphics, but a higher strength keep to keep each graphic title together with the image when there is not room for both on the page.

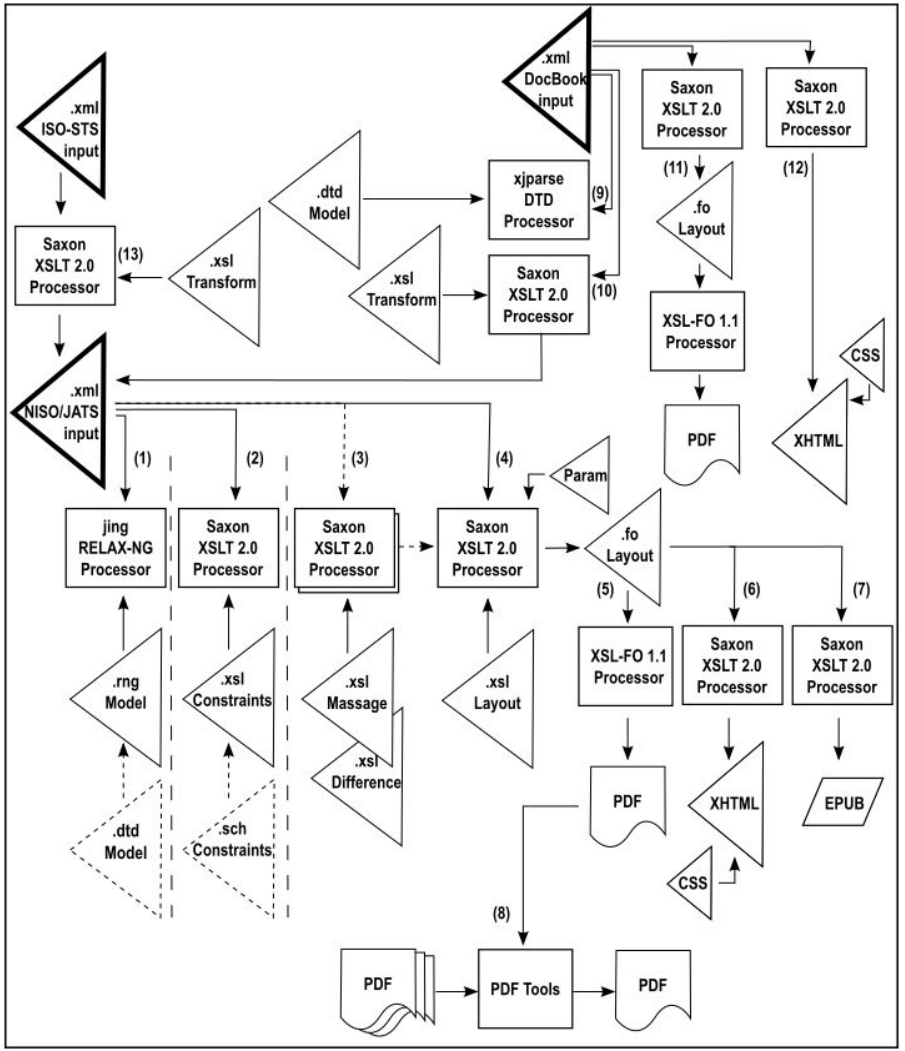


Elaborate processing controlled using Apache Ant

Core process publishes NISO-STS or JATS to augmented XSL-FO
- input ISO-STS converted to NISO-STS

Pure XSL-FO subset used to create PDF
- formatter ignores foreign namespaces

Augmentations in XSL-FO leveraged to create XHTML (and eventually EPUB)
- no maintenance headache of interpreting STS/JATS semantics for two outputs
- CSS classes for every construct allows client-specific differences in appearance
- no client-specific XHTML transformation logic



Process steps

1. Pass 1 of 2 for content validation - confirming the input XML does not violate NISO STS or JATS model constraints
2. Pass 2 of 2 for content validation - confirming the input XML does not violate ISO, CEN, client-specific, or internal constraints
3. Optional multiple transformations massaging input XML including for additional rendering value-add features, and for translating differencing markup to a core format
4. Publishing input XML to intermediate XSL-FO
5. Intermediate XSL-FO to output PDF
6. Intermediate XSL-FO to output XHTML with all styles in embedded CSS and with all images embedded as binary data
7. Intermediate XSL-FO to output EPUB (in development; initial results ready but not nearly complete while awaiting user requirements)
8. Merge of output PDF with fetched PDF from external sources (e.g. adoptions, amendments, etc.)
9. OASIS and DocBook input processing - confirming the input XML dos not violate DocBook model constraints
10. OASIS DocBook conversion to STS - converting the DocBook XML to STS XML
    - used for OASIS/ISO simultaneous publishing (e.g. ISO/IEC 19845)
    - used for OASIS/CEFACT simultaneous publishing (e.g. BDXR XHE - Exchange Header Envelope)
11. Publishing DocBook XML to output PDF using DocBook stylesheets through intermediate XSL-FO
    - used for OASIS specifications and committee notes
    - used for raw DocBook publications
12. Publishing DocBook XML to output HTML using DocBook stylesheets with images referenced and not embedded
    - used for OASIS specifications and committee notes
    - used for raw DocBook publications
13. Legacy ISO-STS inputs are transformed to NISO-STS inputs when necessary

XSL-FO augmented with foreign namespaces for XHTML

```
<xs:template>
  <para>Attribution for quoted content</para>
  <xs:param name="c:f"><para>Dynamic formatting properties.</para></xs:param>
</xs:template>
<xsl:template match="attrib" priority="1">
  <xsl:param name="c:f" as="document-node()+" tunnel="yes"/>
  <block x:h="div*{name(.)}">
      <xsl:copy-of select="$c:f/c:blockLevel/@*,$c:f/c:attrib/@*"/>
      <xsl:call-template name="c:checkIdAttr"/>
      <!--prefix might be defined-->
      <xsl:copy-of select="($c:f/c:attrib/c:prefix)[last()]/node()"/>
      <xsl:apply-templates/>
  </block>
</xsl:template>

<xs:template>
  <para>Text in which white-space and new-lines are respected.</para>
  <xs:param name="c:f"><para>Dynamic formatting properties.</para></xs:param>
</xs:template>
<xsl:template match="preformat | code" priority="1">
  <xsl:param name="c:f" as="document-node()+" tunnel="yes"/>
  <block linefeed-treatment="preserve" white-space-collapse="false"
      text-align="start" white-space-treatment="preserve"
      x:h="div*{name(.)}">
      <xsl:copy-of select="$c:f/c:monospace/@*,$c:f/c:blockLevel/@*"/>
      <xsl:attribute name="font-size"
                  select="concat(c:shrink80lineText(.),'%')"/>
      <xsl:call-template name="c:checkIdAttr"/>
      <xsl:apply-templates/>
  </block>
</xsl:template>
```

XSL-FO augmented with foreign namespaces for XHTML

```
<xsl:variable name="c:tableWrapSpecificUseHeaderDef">
  <!--pull out of the table-wrap-foot all with numeric specific-use-->
  <xsl:for-each select="table-wrap-foot/
                        p[@specific-use castable as xsd:integer]">
    <block>
      <inline x:h="span*{name(../..)}-table-wrap-header">
        <xsl:call-template name="c:exposeElementAndAttributesInline">
          <xsl:with-param name="c:node" select=".."/>
        </xsl:call-template>
        <xsl:call-template name="c:exposeElementAndAttributesInline"/>
        <xsl:apply-templates select="."/>
      </inline>
    </block>
  </xsl:for-each>
</xsl:variable>
<xsl:variable name="c:tableWrapSpecificUseHeaderUse">
  <x:onlyXHTML>
    <xsl:copy-of select="$c:tableWrapSpecificUseHeaderDef"/>
  </x:onlyXHTML>
  <block>
    <retrieve-table-marker retrieve-boundary-within-table="{
     if( xsd:boolean(($c:f/c:tableWrapSpecificUseHeader/@c:repeat)[last()]) )
     then 'table' else 'page'}"
    retrieve-class-name="_page-break-specific-use-{generate-id($c:here)}"/>
  </block>
</xsl:variable>
...
     <xsl:if test="$c:titleBeforeFlag">
       <table-row>
         <table-cell>
           <xsl:copy-of select="$c:tableWrapSpecificUseHeaderUse"/>
         </table-cell>
       </table-row>
      </xsl:if>
```

AHRTS automates comparison of changed stylesheets

Over 110 entry points available in REST call to Réalta server

- e.g. `NSAI-xml2pdf` (Ireland), `NORSOK-xml2pdf-publisher` (Norway), `NORSOK-xml2pdfhtml-editor` (Norway), SNV-xml2pdfhtml (Switzerland), etc.

A change to the core stylesheets might impact results for every entry point

Regression scripts available both locally for developers and remotely on the server

The name of the directory, input XML, and output PDF all are the same as the entry point

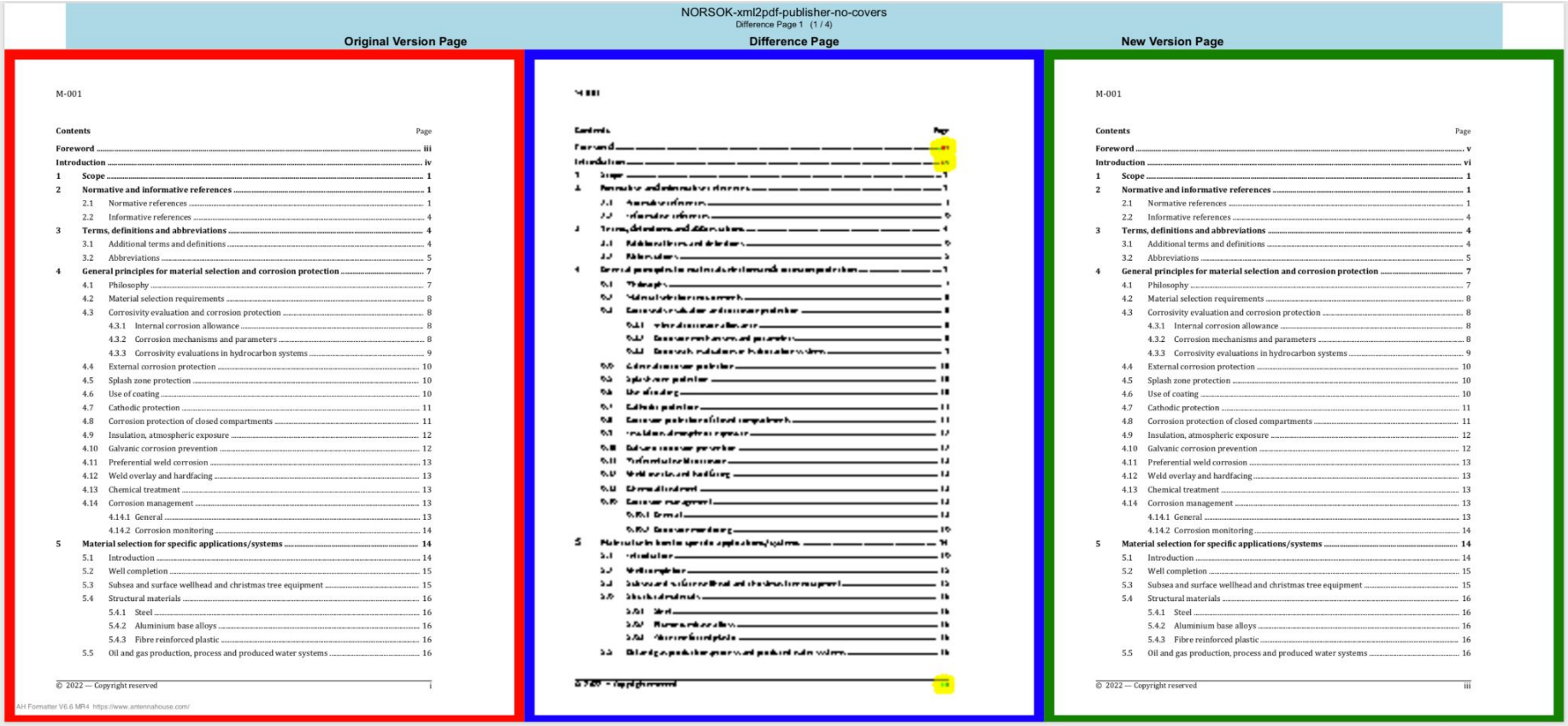
- e.g. `NSAI-xml2pdf/NSAI-xml2pdf.xml` and `NSAI-xml2pdf/NSAI-xml2pdf.pdf`

The developer environment copies the output PDF to a `results/` directory

AHRTS invoked on the command line to compare old and new `results/` directories

AHRTS can be used interactively when debugging the changes to a single entry point

AHRTS automates comparison of changed stylesheets



Using XSL-FO and AHRTS at RealtaOnline.com

XML Prague 2022 - XSO-FO, CSS, and Paged Output Meetup

Using XSL-FO and AHRTS at RealtaOnline.com

G. Ken Holman

Note: some of the images in this presentation are excerpted from the free XSL-FO 1.1 textbook in PDF that can be downloaded in full at no charge from http://www.CraneSoftwrights.com/training/#pfux