DEMOCRATIZING KNOWLEDGE

through

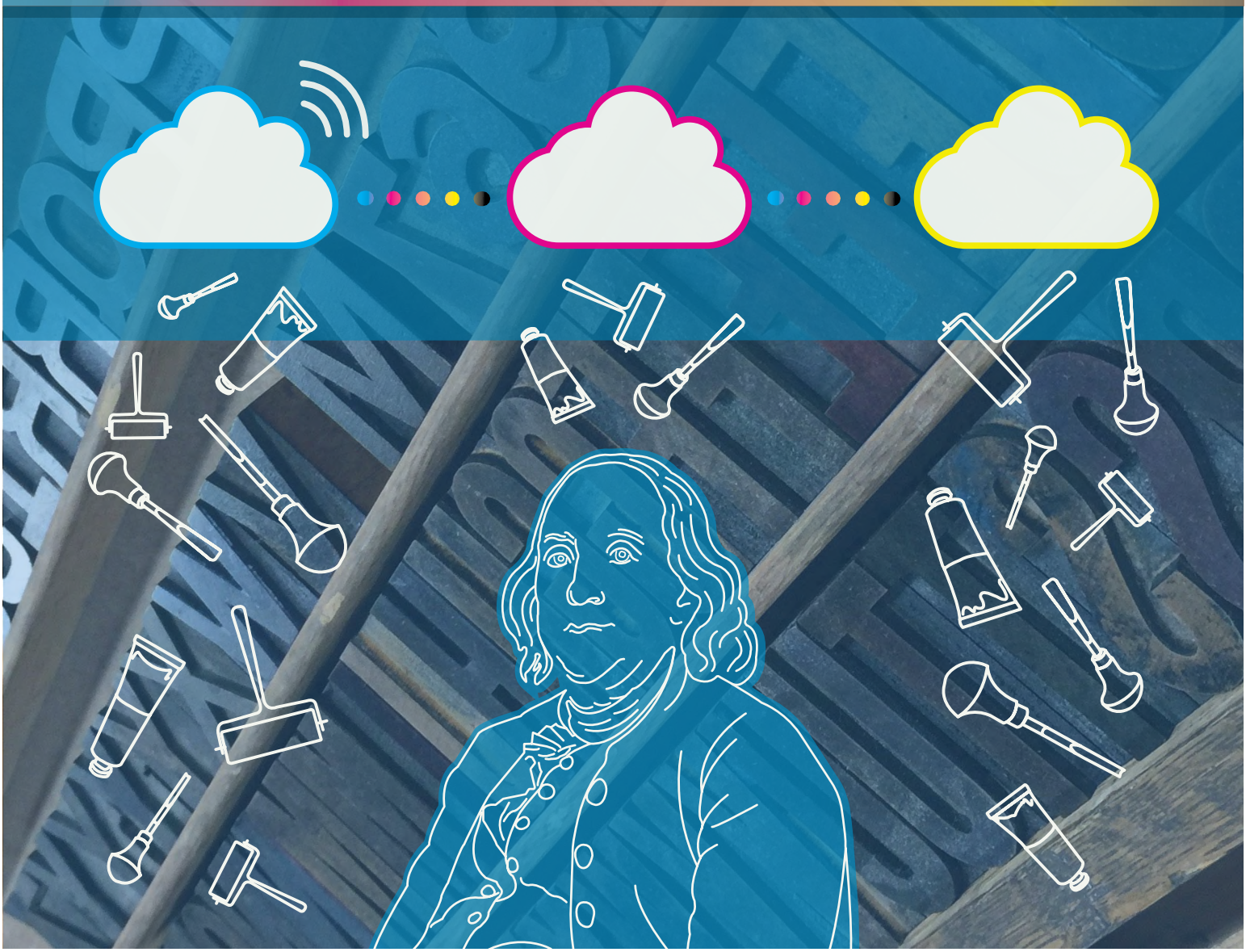
PRINT

Visual Communications Journal

Fall 2016, Volume 52, Number 2

Teaching XML: Lesson Content for Graphic Communications Education

DANIEL G. WILSON, D.I.T.



GRAPHIC COMMUNICATIONS EDUCATION ASSOCIATION

Visual Communications JOURNAL

Volume 52 Number 2

FALL 2016

Acknowledgements

Editor

Dan Wilson, Illinois State University

Editorial Review Board

Cynthia Carlton-Thompson, North Carolina A&T State University
Bob Chung, Rochester Institute of Technology
Christopher Lantz, Western Illinois University
Devang Mehta, North Carolina A&T State University
Tom Schildgen, Arizona State University
Mark Snyder, Millersville University
James Tenorio, University of Wisconsin–Stout
Renmei Xu, Ball State University

Cover Design

Ben Alberti, Western Technical College
Instructor, Barbara Fischer

Page Design, Layout, and Prepress

Janet Oglesby and Can Le

Printing, Bindery, and Distribution

Harold Halliday, University of Houston
University of Houston Printing and Postal Services

About the Journal

The *Visual Communications Journal* serves as the official journal of the Graphic Communications Education Association, and provides a professional communicative link for educators and industry personnel associated with design, presentation, management, and reproduction of graphic forms of communication. Manuscripts submitted for publication are subject to peer review. The views and opinions expressed herein are those of authors and do not necessarily reflect the policy or the views of the GCEA.

Article Submission

Please follow the guidelines provided at the back of this *Journal*.

Membership and Subscription Information

Information about membership in the Association or subscription to the *Journal* should be directed to the GCEA First Vice-President.

Reference Sources

The *Visual Communications Journal* can be found on EBSCOHost databases.
ISSN: Print: 0507-1658 Web: 2155-2428

Teaching XML: Lesson Content for Graphic Communications Education

by Daniel G. Wilson, D.I.T. • Illinois State University

Introduction

The Extensible Markup Language, or XML, is a standard for encoding many different forms of data. XML is used in a surprising number of ways in the graphic communications industry, so much so that basic instruction in the XML language and its applications within the industry may need to be a standard component of any comprehensive graphic communications curriculum. Examples of where XML is used in graphic communications include automating the transformation of unformatted publication content into various formats, such as ePubs (an e-book format), PDFs, and Webpages. Page layout applications like Adobe InDesign have extensive tools to work with XML data. XML is also used to encode and exchange data about print production processes within Management Information Systems (MIS). Though not visible to the user, recent versions of Microsoft Office files are encoded in XML (thus the "x" in the extension, .docx).

The topic of XML can be taught at many levels of depth. This article suggests to high school and college-level Graphic Communications teachers a framework for teaching the fundamentals of XML concepts and skills. Presented is sample lesson content in XML. This includes an explanation of how XML code is structured, how the components of an XML application work, and step-by-step instructions on using an XML editor and Adobe InDesign with XML data.

Overview of XML

The Extensible Markup Languages (XML) is an extension of the Standardised Generalised Markup Language (SGML) created by Charles Goldfarb, along with Ed Mosher and Ray Lorie in the 1970s while working at IBM (Anderson, 2004). Today, XML is a data exchange format for documents that can be read by both machines (computers) and humans (Walsh, 1998). XML is an open standard, defined by the World Wide Web Consortium (Quin, 2015). There are three components to an XML application: (1) data, (2) structure, and (3) presentation. Each of these components exist as it's own XML document, but all three XML documents work together to form an application. Each of these components will be explained in further detail below.

Data: XML Application Component #1

Tags are used to structure the data in XML, and the data take the form of elements and attributes. There are rules for how to structure elements and attributes. Figure 1 is a simple XML document of a directory.

An example of a simple XML data file

```
1 . <?xml version="1.0"?>
2.      <Employee>
3.           <Title level = "3">Chief Engineer</Title>
4.           <First_Name>Ronda</First_Name>
5.           <Last_Name>Grand</Last_Name>
6.           <Phone_1>(345) 345-6789</Phone_1>
7.           <Email>rgrand@graymatter.com</Email>
8       </Employee>
```

Figure 1

Line 1 of figure 1 shows that this is an XML document conforming to "version 1.0". Line 2, (<Employee>), is the root tag. Each XML document needs to have a root tag. By way of example, HTML documents have <html> as the root tag. "Title" is an element of the XML document. Other elements are First_Name, Last_Name, etc. Line 3 shows the element "Title" along with an attribute (level=3). The attribute provides a way to add characteristics to elements. In this case the "Chief Engineer" is a "level 3" employee. Another basic rule on structuring XML data is to include both opening and closing tags. For example, </First_Name> is a closing tag for the opening tag <First_Name>.

Structure: XML Application Component #2

The second component of an XML application is the Document Type Declaration (DTD), which defines the structure. Essentially, the DTD specifies which elements and attributes are permissible within the XML document. Also, the DTD defines the structure of content, including the proper nesting of elements, whether text characters or non-text can occur, and other similar rules. HTML similarly requires DTDs. In the case of HTML, web browser software applications have to read and render various versions of HTML. The HTML date describing the web page starts with a tag indicating which DTD the HTML document conforms. For example, <!DOCTYPE html>

indicates the *HTML 5* DTD. The web browser reads this first line of HTML code and then displays the *HTML 5* content accordingly. Figure 2 is an example of a simple DTD that might go with the XML data provided in the previous section.

An example of a simple Document Type Declaration

```
1.  <?xml version="1.0"?><!DOCTYPE Directory >
2.      <!ELEMENT Title (#PCDATA)>
3.      <!ELEMENT First_Name (#PCDATA)>
4.      <!ELEMENT Last_Name (#PCDATA)>
5.      <!ELEMENT Phone_1 (#PCDATA)>
6.      <!ELEMENT Email (#PCDATA)>
7.      <!ELEMENT Employee (Title, First_Name, Last_Name,
        Phone_1, Email)>
8.          <!ATTLIST Title
9.                  level CDATA #REQUIRED> ]>
```

Figure 2

In the DTD example document, line 1 includes the name of the document type declaration, this one being named "Directory." Starting on line 2, all of the allowable elements within the XML application are included, each beginning with !ELEMENT. The (#PCDATA) indicates that standard text characters can be used within the element name. The root for the XML is defined on line 7, along with an ordered list of elements in parentheses. This provides the rules for the order in which the elements may appear. Finally, line 8 & 9 indicate the "Title" element must include an attribute called "level."

Presentation*:* XML Application Component #3

"All of the semantics of an XML document will either be defined by the applications that process them or by stylesheets" (Walsh, 1998). XSL is a transformation language that allows XML documents to be transformed into a graphic design format. There are variations of the XSL transformation languages, such as XSLT and XSL-FO. Essentially, the XML data can be transformed into many visual formats, including PDF, ePubs (for electronic book), and HTML for websites. Figure 3 is an example of a small portion of an XSL document that could be used to transform the data for the directory example in the previous section into HTML for display by a browser.

On line 1, the element called "Title" is invoked. On line 2, the html tag, <h3>, is applied to the content within the

An example of a portion of an Extensible Style Language file

Given this portion of the XML data:

```
<Title> Chief Engineer </Title>
```

This portion of the XSL document:

```
1 <xsl: template match='Title'>
2  <h3><xsl:apply-templates/></h3>
3 </xsl:template>"
```

Would render the text into this:

```
<h3>Chief Engineer</h3>"
```

The text would be displayed by a Web browser like this:

Chief Engineer"

Figure 3

"Title" tag. Finally, the HTML file would be written by an XML processor, and then the resulting tagged html content could be rendered by a web browser.

Applications of XML in Graphic Communications

XML can be utilized in a variety of ways. Such utilizations can be found in web development as well as in print and digital production workflows. The following sections will ilustrate XML use in publications, asset management, and print production.

XML for Publications

Publishers use XML to encode content within a Content Management System. This XML tagged content can then be transformed into HTML for display on a website with the use of an XSL document. That same content might also be transformed with another XSL file into an ePub file for display on a mobile device. For print applications, another XSL file could be used to transform the same content into PDF in preparation for printing. In this way, a single source of content can be transformed into many graphic forms through automation.

There are many XML format standards for publications. The DocBook standard is used for encoding technical documentation in XML, the ePub standard is used to encode digital books in XML, and XHTML is a version of HTML that conforms to XML rules (Quin, 2015).

As previously mentioned, XSL-FO, an acronym for XSL Formatting Objects, is another transformation language variation of XSL. An XSL-FO file contains the objects that make up a document, such as "characters, blocks of text, images, tables, borders, (and) master pages" (Deach,

2002). This type of file can be helpful in the print production of mass-produced documents. Some documents, such as magazines and newspapers, are best suited for human/manual graphic design and modification. However, some mass-produced documents including financial-planning guides, owner and maintenance manuals, and legal agreements and contracts can be automatically formatted and constructed with XSL-FO to yield PDF files ready for print (Deach, 2002).

XML in Print Production

XML is also often used as the foundation for encoding print production workflow data in a print environment. For example, the Job Definition Format (JDF), managed by the International Cooperation for the Integration of Processes in Prepress, Press and Postpress Organization (CIP4) is an XML format that allows for communication between devices and Management Information Systems (MIS) in a print workflow. Thus, the process of "gathering, reporting, and controlling data" for each job has the potential to achieve its goals with maximum efficiency. Such information can include XML tags for page size, binding, trapping, imposition, costing data, and contact information (Waldman, 2003).

XML in Asset Management

XML is also used as the foundation for metadata in graphic communications. Metadata can be defined as "data that describes data." In the case of photographs, metadata includes characteristics about the the photo, such as the f/stop and shutter speed used, who holds the copyright, and what white balance settings were used during shooting. The metadata standard for photos is IPTC (stands for International Press Telecommunications Council) (IPTC, 2016), which is used by camera manufacturers and photo editing applications. The IPTC metadata is embedded in the photo file, and then can be read by applications such as Adobe Photoshop or by Content Management Systems (CMS) for publishing.

Adobe has developed it's own metadata platform based on XML called the Extensible Metadata Platform (XMP). Every file worked with in Adobe Illustrator, InDesign, Photoshop, Premier, etc., can carry with it embedded metadata (figure 4). Because the metadata has its basis in XML, the metadata platforms are easily cross-system compatible to allow for automation in publishing. Each Adobe software product includes an XMP dialog box where metadata can be viewed and edited.

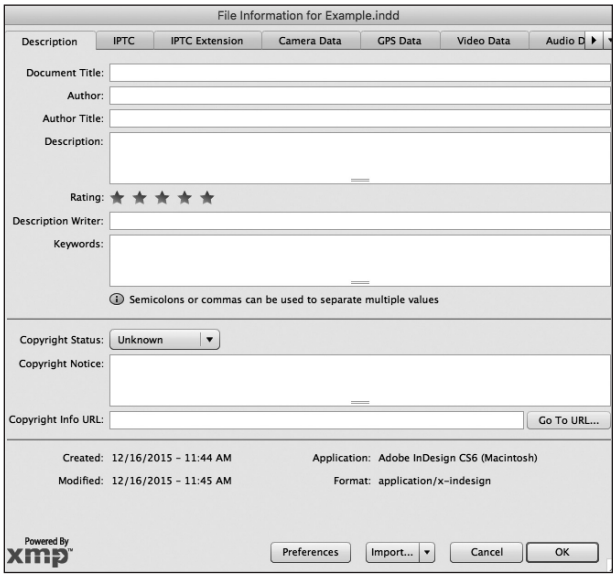


Figure 4

Additionally, metadata standards allow Content Management Systems to automate the publication and distribution of content. For example, PRISM is a magazine industry standard for XML formatted metadata that can be used to build efficient, multi-channel publishing solutions. Idealliance, the graphic communications industry association for digital content, maintains the PRISM metadata standards. With PRISM a publisher can create, manage, aggregate, produce, distribute and reuse content through XML (PRISM, 2016).

Introductory XML Lesson for Students

XML can be integrated into the Graphic Communications curriculum by presenting the aforementioned concepts of XML. However, teaching the key concepts of XML can be challenging to those not accustomed to coding. Provided here is an introductory assignment that has been used by the authors to teach graphic communications students. *Note: These project instructions would require some experimentation and modification by the instructor before undertaking with students.*

The project allows students to work with XML code in a variety of ways. This assignment involves students in (1) editing an XML data document, (2) transforming the XML into HTML via an XSL document for viewing in a web browser, and then (3) working with the same XML data in Adobe InDesign to yield a print-ready document. *Note: The hands-on lesson would typically take about three hours of lab sessions to demonstrate and complete. The*

student would be expected to have some knowledge of HTML and CSS. The learning objectives the student would meet include:

1. Use an XML editor to compose XML data and transform the data into an HTML document using an XSL style sheet.
2. Use knowledge of HTML to analyze and edit an XSL style sheet to achieve new HTML design elements.
3. Setup an Adobe InDesign document with XML data using the "structure" and "tags" panels.

XML into HTML using an XML editor

There are many free *HTML editors* that can be used to write XML data, but *XML editors* are more functional: they include additional style sheet rendering engines that can output PDF files or HTML files from XML data with the use of an XSL file. Note that the step-by-step instructions provided below assume that the student already has a basic knowledge of HTML and also that these instructions include reference to an XML editor called "EditiX (http://www.editix.com/)." However, any XML editor can be substituted.

Figure 5 displays a block of XML code for a movie review publication. Students could be given the premade XML file and then be asked to enter movie review textual information using an XML editor. Next, the student would use a provided XSL file (shown in figure 6) to transform the

XML Code for the Movie Review

```
<?xml version="1.0" encoding="UTF-8"?>
<movie_reviews>
<review>
    <title></title>
    <teaser> </teaser>
    <cast>
        < person ></ person >
        < person ></ person >
        (etc.)
    </cast>
    <director></director>
    <duration></duration>
    <genre></genre>
    <year></year>
    <body> </body>
    <rating></rating>
</review>
<review>
    <title></title>
    <teaser> </teaser>
    <cast>
        < person ></ person >
        < person ></ person >
        (etc.)
    </cast>
    <director></director>
    <duration></duration>
    <genre></genre>
  <year></year>
    <body> </body>
    <rating></rating>
  </review>
  </movie_reviews>
```

Figure 5

XSL content for transforming tagged content into HTML

```
<?xml version="1.0"?> <xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/review">
    <html>
      <head>
        <font face="Arial" size="2"/>
      </head>
        <body>
          <xsl:apply-templates select="title"/>
          (<xsl:apply-templates select="year"/>) <br/>
          <xsl:apply-templates select="teaser"/> <p/>
          <xsl:apply-templates select="cast"/> <br/>
          <xsl:apply-templates select="director"/> <br/>
          <xsl:apply-templates select="duration"/> <br/>
          <xsl:apply-templates select="rating"/> <p>
          <xsl:apply-templates select="body"/> </p>
        </body>
      </html>
  </xsl:template>
<xsl:template match="title"> <h1> <xsl:value-of select="."/> </
h1> </xsl:template>
<xsl:template match="teaser"> <xsl:value-of select="."/> </
xsl:template>
<xsl:template match="director"> <b>Director: </b> <xsl:value-of
select="."/> </xsl:template>
<xsl:template match="duration"> <b>Duration: </b> <xsl:value-of
select="."/> minutes </xsl:template>
<xsl:template match="rating"> <b>Our rating: </b> <xsl:value-of
select="."/> </xsl:template>
<xsl:template match="cast"> <b>Cast: </b> <xsl:apply-tem-
plates/> </xsl:template>
```

```
<xsl:template match="person[position() != last()]"> <xsl:value-of select="." />, </xsl:template>
<xsl:template match="person[position() = (last()-1)]"> <xsl:value-of select="." /> </xsl:template>
<xsl:template match="person[position() = last()]"> and <xsl:value-of select="." /> </xsl:template>
<xsl:template match="body"> <xsl:apply-templates/> </xsl:template>
<xsl:template match="body//title"> <i><xsl:value-of select="."/></i> </xsl:template>
<xsl:template match="body//person"> <b><xsl:value-of select="." /></b> </xsl:template>
</xsl:stylesheet>
```

Figure 6

Step-by-step instructions for transforming the XML into HTML

Follow these steps to import your XML document into an XML editor (editix2015) and apply the supplied XSL (moviereviewstyle.xsl) to transform the code into HTML.

a. Launch editix2015.

b. Go to File> Open. Navigate to your XML file (your review encoded as XML).

c. Go to XSLT/XQuery:>Transform using XSLT.

d. Next to XSLT document, click the folder icon (on the far right) and navigate to the *moviereviewstyle.xsl* file. Click OK. The "*result.html*" file will be generated and saved back into your source folder. Open it in a browser and check it out.

e. Pretty plain, huh? Figure out how to stylize at least two html tags in the XSL file, and then output a better designed version. You can incorporate the syntax for any CSS in-line style (just Google "in-line CSS examples" for ideas).

Figure 7

Figure 8

XML movie review into an html document for display in a web browser.

The step-by-step instructions provided in figure 7 challenge the student to make modifications to the XSL file to achieve new design results. This requires the student to do some analysis and experimentation.

Import XML into a formatted document using Adobe InDesign

The next portion of the assignment involves importing the XML data into Adobe InDesign and then structuring the XML tagged data to yield an automated layout. In this way, students are introduced to how XML structured content can be used to efficiently automate document production with a page layout program. Adobe InDesign has in-depth information online about using with XML (Adobe, 2015). For this portion of the lesson, it would be best if students have a basic level of skill using Adobe InDesign. Note that the instructor should make a simple InDesign template to share for the assignment (similar to figure 8).

Next, the student can follow the instructions in figure 9 to build the automated XML document. Note that these

Step-by-step instructions for using XML with Adobe InDesign

A template for importing XML data (see figure 7) would need to be developed in InDesign to automate production of a dynamic movie review book.

a. Open the InDesign template called "MovieReviewTemplate.indt" in InDesign (figure 7).

b. Open the Tags panel in InDesign: Windows> Utilities> Tags

c. Load your XML tags (elements): In the Tags panel contextual menu> Load Tags. Navigate to your XML file (the one that contains both movie reviews and a "movie_reviews" root) and choose to load.

d. Open the structure pane (it may already be open). View> Structure> Show Structure. Click to highlight the "root" in the structure pane. Then, click on the root of your XML file, in this case, "movie_reviews" (the name in the structure pane will change from "root" to the name of your XML file root).

e. Next, highlight the text box that will contain the XML content and click the "review" tag. Next, using the text tool, you will be highlighting each text element and tagging them (but only the elements within the page that will be replaced with XML encoded content). Note: These must appear in the document in the same order that they are in the XML code. A structure will

appear in the structure pane that should match the structure of your XML file. You may want to open the XML file in an editor just to double check the order.

f. Next, you will copy the text boxes from page 1 of your InDesign document to page 2. Notice that a new review structure appears in the Structure Pane.

g. Now, flow your .XML data into the template: File> Import XML. Make sure the "Do not import contents of whitespace-only elements" and "create link" are checked.

h. Save as the new document.

Figure 9

instructions assume the use of Adobe CS6 or CC 2015. Newer or older versions may require some reworking of the instructions, but the basic concepts are the same.

Conclusion

Though there is no data to verify the notion, anecdotal evidence suggests that few graphic communications-oriented programs at the high school and college level have even basic instruction in XML. Since XML has become integrated into the management of graphic information content, students would be well served by developing fundamental knowledge of the technology**.** The lesson described in this article will provide students with a foundation in XML. Insights into how XML is structured, how the components of an XML application work, and how to use an XML editor and Adobe InDesign with XML data form the basis for a working knowledge of XML. Along with this fundamental knowledge, XML-related applications such as Digital Asset Management (DAM), database management, and digital publishing can be taught and understood on a deeper level. Whether in web production, print production, or management of graphic media, XML plays a big part and is likely to grow even more significant in years to come.

References

Adobe InDesign Help / Work with XML. Retrieved on December 12, 2015 from https://helpx.adobe.com/indesign/using/xml.html

Anderson, T. (2004) Introducing XML. Retrieved on November 15, 2015 from http://www.itwriting.com/xmlintro.php

Deach, S. (2002). What is XSL-FO and when should I use it? *The Seybold Report*, *2*(17), 1–8.

Extensible Markup Language (XML). (2016, January 16). Retrieved from https://www.w3.org/XML/

Hammond, R. (2006). XSLT: The genie inside the bottle. *Online*, *30*(3), 29–31.

IPTC is the Global Standards Body of the News Media. (2016, January 16). Retrieved from https://iptc.org/about-iptc/

Quin, L. Extensible Markup Launguage. Retrieved on December 1, 2015 from http://www.w3.org/XML/.

PRISM Metadata Initiative: Authoring Articles in PRISM XML. (2016, January 16). Retrieved from http://www.idealliance.org/specifications/prism-metadata-initiative

Waldman, Harry. (2003). JDF-A Dynamics Leap Forward. Printing Impressions Magazine. 1–4.

Walsh, N. (1998). A technical introduction to XML. *World Wide Web Journal,* 2–49. Retrieved from http://nwalsh.com/docs/articles/xml/

Wilcockson, David. (2014). XML Workflow for Digital Content. *Presentation*. Retrieved from http://www.librios.com/

Manuscript Form and Style

- Prepare manuscripts according to the APA style, including the reference list.
- List your name and address on the first page only. Article text should begin on the second page.
- Provide a short biography for yourself that can be used if the article is accepted for publication.
- All articles must be submitted in electronic form on a CD-ROM or as an email attachment.
- Submit a Microsoft Word document, maximum of 10 pages (excluding figures, tables, illustrations, and photos). Do not submit documents created in page-layout programs.
- Word documents must have been proofread and be correct.
- Call out the approximate location of all tables and figures in the text. Use the default style "Normal" on these callouts. The call-outs will be removed by the designer.
- Use the default Word styles only. Our designer has set up the page layout program styles to correspond to those style names.
  - Heading 1
  - Heading 2
  - Heading 3
  - Normal

Graphics

- Be sure that submitted tables and other artwork are absolutely necessary for the article.
- Write a caption for each graphic, include captions in a list at the end of your Word document.
- Electronic artwork is preferred and should be in PDF or TIFF format.
- Send all artwork files and hard copies of these files with your submission.

Tables

- Set up tables in separate documents, one document for each table.
- Do not attempt to make it "pretty." Use the default Word style "Normal" for all table text. Do not use any other formatting.
- Do not use hard returns inside the table ("enter" or "return").
- Get the correct information into the correct cell and leave the formatting to the designer.
- Tables will be formatted by the designer to fit in one column (3.1667" wide) or across two columns (6.5" wide).

Artwork

- Scan photographs at 300 ppi resolution.
- Scan line drawings at 800 ppi resolution.
- Screen captures should be as large as possible.
- Graphics should be sized to fit in either one column or across two columns.
  - One column is 3.1667" wide, two columns are 6.5" wide.
  - Graphics may be larger than these dimensions, but must not be smaller.

Manuscript Guidelines

Eligibility for Publication

Members of the Graphic Communications Education Association, or students of GCEA members, may publish in the *Visual Communications Journal*.

Audience

Write articles for educators, students, graduates, industry representatives, and others interested in graphic arts, graphic communications, graphic design, commercial art, communications technology, visual communications, printing, photography, desktop publishing, or media arts. Present implications for the audience in the article.

Types of Articles

The *Visual Communications Journal* accepts four levels of articles for publication:

1. Edited articles are accepted or rejected by the editor. The editor makes changes to the article as necessary to improve readability and/or grammar. These articles are not submitted to a panel of jurors. The decision of the editor is final.
2. Juried articles are submitted to the editor and are distributed to jurors for acceptance/rejection. Juried articles are typically reviews of the literature, state-of-the-art technical articles, and other nonempirical papers. Jurors make comments to the author, and the author makes required changes. The decision of the jurors is final.
3. Refereed articles are submitted to the editor and are distributed to jurors for acceptance/rejection. Refereed articles are original empirical research. Jurors make comments to the author and the author makes required changes. The decision of the jurors is final.
4. Student articles are submitted by GCEA members and are accepted/rejected by the editor. These articles are not submitted to a panel of jurors. The editor's decision is final. Please be aware that poorly written student papers will be rejected or returned for editing.

Submittal of Manuscripts

All manuscripts must be received by the editor no later than December 15th to be considered for the spring *Journal* or by June 15th to be considered for the fall *Journal*. Include digital copies of all text and figures. Prepare text and artwork according to the instructions given in these guidelines. Be sure to include your name, mailing address, e-mail address, and daytime phone number with your materials. E-mail all materials to the editor (address shown below).

Acceptance and Publication

If your article is accepted for publication, you will be notified by e-mail. The *Visual Communications Journal* is published and distributed twice a year, in the spring and in the fall. Printed copies are mailed to GCEA members. A PDF version of the *Journal* is published online at www.GCEAonline.org.

Notice

Articles submitted to the *Journal* cannot be submitted to other publications while under review. Articles published in other copyrighted publications may not be submitted to the *Journal*, and articles published by the *Journal* may not be published in other publications without written permission of the *Journal.*

Submit All Articles and Correspondence to:
Dan Wilson, dan.wilson@illinoisstate.edu
or check www.GCEAonline.org for contact information for the GCEA First Vice-President.

See following page for style guidelines