XSL-FO
Understand and Use

XML Processing for PDF and Print

Manfred Krüger

Manfred Krüger

XSL-FO
Understand and Use

XML Processing for PDF and Print

MID/Information Logistics

MID/Information Logistics · D-69221 Dossenheim · Friedrichstraße 43A
Phone: +49-(0)6221 868 9040 · Fax: +49-(0)6221 868 9039
Web: www.mid-heidelberg.de · E-Mail: krueger@mid-heidelberg.de

ISBN PDF-E-Book: 978-3-9822359-9-8
ISBN Print: 978-3-9822359-8-1

Editing: Edwina Pfendbach
Design and production: Manfred Krüger, Dossenheim
Print: MM-Druck, Heidelberg

Table of Contents

Preface

Fifteen years have passed since the publication of the first edition of this textbook[1] in German language, six years since the publication of the second edition[2]. The XSL technology of formatting (XSL-FO) has not changed during this time. Only the XSL transformation technology (XSLT) has evolved, making more and more data suitable for formatting processing with XSL-FO.

The knowledge of how to apply the XSL-FO technology has developed considerably during this period. This increased application knowledge and the added typographical features have been and will continue to be significantly enhanced by the continuous further development of a formatter product, the XSL Formatter from the Japanese software company Antenna House.

The German language of this textbook limited its worldwide use. This limitation is intended to eliminate this English language edition.

I have adapted the chapter which is dedicated to the handling of Antenna House's XSL Formatter to the current software version 7.x.

The concept of the book has remained the same: The first part is intended to awaken an understanding of the XSL-FO technology even among people who have had little or no contact with XML in the past, but who are professionally involved in the design and processing of documentation or publications.

The second part consists of two prototypical developments for forms and book-like documents, which can be easily adapted and extended to meet individual application requirements based on the growing understanding of this technology.

1 Manfred Krüger: „XSL-FO verstehen und anwenden“, dpunkt.verlag 2006.
2 Manfred Krüger: „XSL-FO verstehen und anwenden“, MedienEdition Welsch 2014.

The third part contains the documentation of the XML document structure used in the second part, which is modeled to a minimal extent and oriented on the popular DocBook document type definition. Systematically ordered font character tables are the final section of this part.

The XSL-FO reference originally contained in the third part forms a separate publication[3]. This reference includes the standard constructs of the XSL Recommendation V. 1.1 as well as all proprietary extensions of Antenna House's formatter product. Here, in this publication, you will find – where necessary and useful – links to this XSL-FO reference, which can be activated by a click when using the PDF version and allow the parallel handling of textbook and reference.

At the beginning I pointed out the increase of knowledge in handling and using the XSL technology in the last two decades. This increase in knowledge has been accompanied by a steadily growing scope of practical application in many industries, including the publishing industry. However, the practical spread of the XSL-FO concept throughout the world did not correspond to the public perception of those involved in automated document processing in these industries.

This may also be related to the fact that application development is often not sufficiently professional – as I know from some insights into XSL-FO applications available to me. The shortcomings of these applications often lie in the fact that the application developers lack either the basic understanding of this technology or of the design or typographical rules, or both. With this publication, I am trying to increase the knowledge in both areas and at least stimulate a professional discussion on these topics.

The English version of this book has been translated from the original German edition using DeepL. With the help of the DeepL-API this was done completely automated while preserving the XML

[3] Manfred Krüger: „XSL-FO Vollständige Referenz mit den Erweiterungen des Antenna House XSL Formatter“, free download: `https://www.antennahouse.com/xsl-fo-vollstandige-referenz-by-manfred-kruger-free-download`

structure of the data! Parts that did not need to be translated, such as the code examples, were excluded from the translation. However, this was not the end of the translation work.

In contrary, it was just beginning. Having now used the interactive DeepL app, my editor Edwina Pfendbach and I went through the entire translation sentence by sentence. In the process, it became apparent that particularly the deficiencies in the German text led to translations that in some cases distorted the meaning. I thank Edwina Pfendbach for her patience and hope that the result will be beneficial to the users of this book.

Manfred Krüger
Dossenheim, February 2021

I. Fundamentals

1 XSL – What? What for? Why?

1.1 XSL – What is Behind it?

The eXtensible Stylesheet Language (XSL) is the concept for the transformation and formatting of XML data or documents. Developed within the World-Wide-Web consortium, it was first published in version 1.0 in 2001, last as version 1.1 on 5 December 2006 (`www.w3.org/TR/xsl11`).

In the XSL concept, three sub-concepts work together to ensure that the intended objectives are achieved and can be implemented in a practical manner:

- A concept for structuring the processing specifications and transforming the output data into a form that is as easy as possible to handle for the subsequent processing systems. This sub-concept is the eXtensible Stylesheet Language Transformations, first published in version 1.0 on November 16, 1999, last as version 3.0 on June 8, 2017 (`www.w3.org/TR/2017/REC-xslt-30-20170608/`).
  In the context of this book on XSL-FO, only XSLT version 1.0 is referred to. The extensions in versions 2 and 3 are not required for a basic understanding of the XSL-FO stylesheet concept.
- A concept for accessing the structures in the XML source data in order to assign the processing specifications to the data. This sub-concept is the XML **Path** Language (XPath), first published in version 1.0 on 16 November 1999, last as a recommendation on 8 April 2014 in version 3.0 (`www.w3.org/TR/xpath-30`).
  In the context of this book on XSL-FO, only XPath version 1.0 is referred to. The corrections and extensions in versions 2 and 3 are not required for a basic understanding of the XSL-FO stylesheet concept.

- ❑ A concept for processing in automated typesetting and pagination systems for print output or interactive handling. This sub-concept, known as XSL-FO (eXtensible Stylesheet Language Formatting Objects), is specified in the eXtensible Stylesheet Language (XSL) mentioned above.

The XSL concept is XML-based throughout, i.e. XSL stylesheets are well-formed XML documents.

Admittedly, the positioning of XSL as an overarching concept and at the same time as a partial concept is somewhat confusing and may also give the impression that the concept as a whole is inconsistent. The former is correct, but the latter leads to a thorough misjudgment of the usability and performance of the concept.

1.2 XSL – What is it Suitable for?

In general, the aim of processing XML data with XSL is to present the XML data without limiting the form of presentation or the medium in which it is displayed. The use of XML data for processing with XSL can be very diverse.

- ❑ Transformation of the source data into other XML structures,
- ❑ Extraction of data from the source documents and generation of new compilations (e.g. directories, registers),
- ❑ Reformatting of output data for processing in non-XML systems (e.g. computer typesetting or word processing systems),
- ❑ Presentation in web browsers (with HTML) and finally the
- ❑ Processing in XML-based typesetting and pagination systems which, by applying a sub-concept of XSL – that is XSL-FO.

Cf. ch. 3 According to the topic of this book, only the application of the sub-concept XSL-FO will be discussed in more detail here, and XPath and XSLT only as far as absolutely necessary.

XSL-FO is used for applications that are commonly referred to as typesetting and pagination or more generally as document formatting.

Cf. ch. 7 The output is a sequence of pages in fixed dimensions (width and height). Accordingly, the core feature of XSL-FO technology is page and document design (e.g. header, footer, margins, multiple columns, placement of content outside the text flow, etc.), line breaks (with hyphenation), filling the type area with vertical wedges (to achieve pages of equal height), column or page breaks (What must be held together in a column/page? Where in the text flow can column/page breaks be made?) and page order/page sequence (page templates, left/right pages).

An important limitation is that XSL-FO technology can only be used for automated typesetting and pagination. This results from the process approach described in ch. 2 and illustrated in an application example.

Automated typesetting, as opposed to interactive typesetting and pagination, is particularly suitable for processing large volumes (legal publications, technical documentation), for processing spontaneously compiled documents (print-on-demand), for achieving a consistent layout and – last but not least – for reducing typesetting costs.

SGML users Note for SGML users: SGML documents (conforming to ISO standard ISO 8879:1986) can be processed with XSL if they have been transformed into well-formed XML documents. For example, `sx` can be used, which is part of James Clark's SGML system `SP` and can be obtained free of charge from him (`www.jclark.com/sp/`).

1.3 XSL – Why Should it be Preferred to Other Concepts?

In the discussion about the development of typesetting and pagination systems, which has been going on for about four decades, the aspect of automated versus interactive page design has always been in the foreground. At least implicitly, it has been judged and claimed that high-quality typesetting (typography) could not be achieved without interactive intervention in individual pages. Automated typesetting systems had the odour that they would only produce typographically simple typesetting results of inferior quality. Until today, automated typesetting applications have only been considered justified if large volumes have to be processed for reasons of cost and time, and impoverished design can therefore be tolerated.

TEX That this need not be the case is shown by the publicly available and free typesetting system TEX (Donald E. Knuth: The TeXbook. Reading 1984), which really cannot be accused of neglecting typographic aspects.

DSSSL With XSL, however, automated typesetting production gains a potential that has not been available anywhere before in a conceptually integrated and practicable way: the division of the typesetting process into two process steps according to the model of the SGML-based processing concept DSSSL (Document Style Semantics and Specification Language – ISO/IEC 10179:1996). In the first step, the source data is transformed according to the typesetting and make-up requirements and the desired layout specifications are assigned to the objects to be formatted. In the second step, the actual typesetting and pagination process is then mastered, in which the formatting tool, the XSL formatter, can limit itself to page layout and pagination. This two-step process has become practicable by the innovative XSL stylesheet technology, in which both steps are specified according to the data structure of the source data. The combined specification for these two process steps using processors with very different functionality (structure processing in the first step – page layout in the second step) is what is really new about XSL.

Cf. ch. 2 With this division into two processing steps, considerable performance advances and expansions have become possible compared to conventional technologies, both in the area of structural processing and in the area of page design, which are also implemented by the sub-concepts XSLT and XSL-FO. It should be kept in mind that these performance enhancements in page design are only visible in the typesetting results if the XSL formatter implements the corresponding XSL-FO features in the layout.

However, the quality of the typesetting result is also influenced by features that XSL-FO does not specify, but which represent the properties of the XSL formatter, e.g. line breaks in justified typesetting with the smallest possible word spacing and optimized hyphenation. The XSL formatter used for this book has significant advantages over its XSL-FO competitors precisely in these proprietary features.

A problem from the past remains: If the XSL stylesheets are written without typographic expertise, the typesetting results will continue to be unsatisfactory from a design point of view. But even if experienced typographers continue to convert their designs, which have been tested in interactive typesetting, into XSL stylesheets, the quality that one is accustomed to from interactive typesetting is still far from being achieved. For automated typesetting, all possible content environments in the stylesheet must be considered in advance and robustly specified for aesthetically pleasing typesetting. In other words, the rules for good typography in automated typesetting are different from those in interactive typesetting.

This book has been written for all those who want to apply their typography knowledge in stylesheet development and is provided with many explanations and concrete syntax examples.

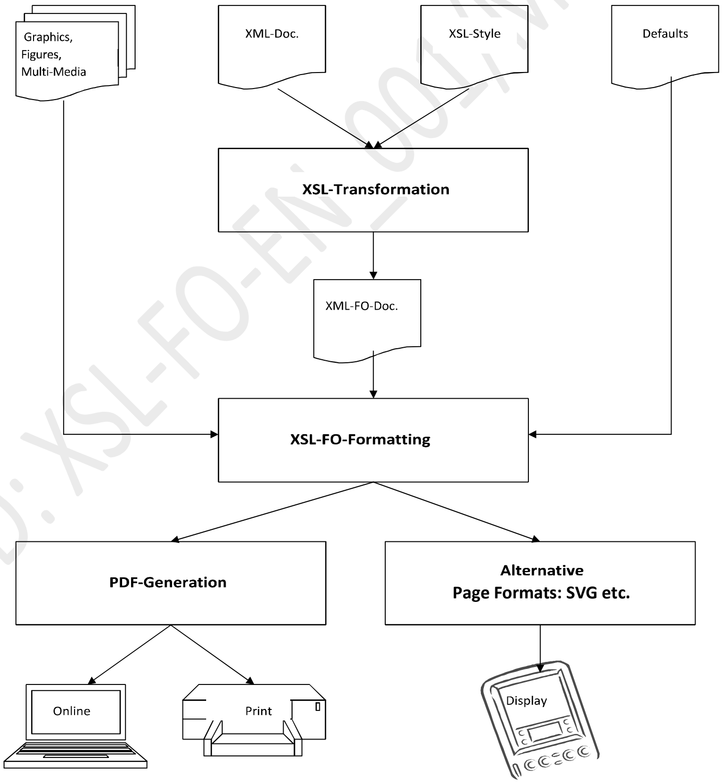
In addition, the syntax of stylesheets is that of normal XML documents, which can best be created and edited with XML-supporting editor tools under syntax control for XSLT and XSL-FO conformity. I myself use the Oxygen XML Editor from Syncro Soft SRL for creating and editing my XSL stylesheets.

Last but not least, anyone who masters the writing and editing of XSL stylesheets for page design thus has a technology that can be used very widely without significant changes, e.g. for the transformation of XML data to HTML (the first process step of structure processing is pretty much the same, only HTML structures take the place of the XSL-FO constructs). Also the transformation from one XML structure to another should not be difficult after these experiences.

2 XSL-FO – The Production Process

How does the processing of XML-structured documents work with the concept of XSL? This will be described in the following, illustrated by the following graphic and demonstrated with an example. This example shows the characteristic features of processing with XSL-FO. Therefore, it is a bit more comprehensive than the usual "Hello World" example.

Fig. 2-1
XSL-FO-Processing



Basically: The processing of an entire XML document takes place in several steps without any possibility of feedback, i.e. without partial repetition of individual process steps or interactive, corrective interventions, i.e. completely automated.

2.1 The XML source document and the XSL stylesheet

XML source documents can be DTD- or Schema-controlled, but also simply well-formed.

This is followed by the well-formed XML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<Book>
   <Frontmatter>
      <Title-page>XSL-FO – Understanding and Using</Title-page>
      <Article>
         <Title>Preface</Title>
         <P>This is the content of the preface.</P>
      </Article>
   </Frontmatter>
   <Part>
      <Title>Fundamentals</Title>
      <Article>
         <Title>Introduction to XSL-FO</Title>
         <P>This is the content of the introduction.</P>
      </Article>
      <Article>
         <Title>The Language of XSL-FO</Title>
         <P>This is the description of the language.</P>
      </Article>
   </Part>
   <Part>
      <Title>Application</Title>
      <Article>
         <Title>Page Structure and Sequence</Title>
         <P>
            This is how you describe the page structure and the page
sequence.
         </P>
      </Article>
```