Extensible Markup Stylesheet Formatting Objects (XSL-FO)

Asst. Prof. Dr. Kanda Runapongsa (krunapon@kku.ac.th)
Dept. of Computer Engineering
Khon Kaen University

1

Overview

- What is XSL-FO?
- Value of XSL-FO
- XSL-FO Tool: Apache FOP
- What XSL-FO Specifies

2

What is XSL-FO?

- Extensible Stylesheet Language-Formatting Objects
  - A way to make print directly from XML
  - A page description language (that particular software can read)
  - XSL-FO rendering software
    - Takes XML as input
    - Gives PDF, Postscript (RTF, MIF, etc.) as output
  - Described in a W3C recommendation called XSL (Extensible Stylesheet Language)

3

Design Goals for XSL Formatting

- Allow designers to express how structured (XML) content should be presented
  - In print (pamphlet, magazine, bound volume, poster…)
  - On screen
  - In other media such as audio or braille

(without being tied to a particular application/vendor)

4

The Dream Behind XSL-FO

- XML separates specification of format from content
  - XML will specify the content
  (list, product name, paragraph, surname)
  - XSL-FO will specify the layout, pagination, styles
- Goal: high-quality, high-volume, content-driven publishing
  - Batch processing on one file or many
  - Formatting is content-driven (from the XML tags)
- Definition of layout and styles
  - Not bound to any platform
  - Not proprietary to any software

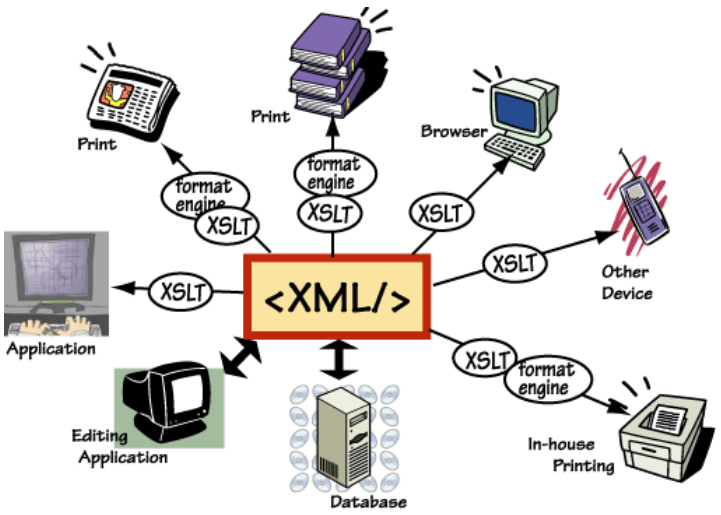
5

How XSL-FO Files Get Created

- You could write XSL-FO code by hand but
  - It was not designed for that
  - Nobody does
- You could use a GUI tool to
  - Drag and drop your wishes
  - Write XSL-FO behind the scenes
- But XSL-FO is typically made using XSLT
  - Which is an XML transformer
  - Changes your XML tags to
  - XSL-FO tags with your content inside them

6

XSL-FO Creates Print (and Web) Output



7

XSL-FO Supports Big Batch Production

- Write instructions once, run on many files
- Hands-free page make-up
- Consistency of generated content
- Format complex documents in seconds
- Run unattended to make consecutive output files
- Save typesetter time on routine tasks

8

XSL-FO is an XML Vocabulary (Tag Set)

- XSL-FO vocabulary is a set of tags that describe
  - The layout geometry of a page

  (into which you pour content)
  - A set of formatting objects
    - That say how to put content on the pages
    - That describe how the document should be rendered

<fo:block font-family="Helvetica">

9

XSL-FO is Another Way to Display XML

- XSL-FO is a vocabulary of tags into which XML can be translated
- XSL-FO software (a rendering engine) makes the display from those tags
- We know the vocabulary to make the print effect we want
  - <fo:block>…</fo:block>
  - <fo:inline font-style="bold">Wow!</fo:inline>
  - <fo:inline font-style="italic>…</fo:inline>

(XSL-FO can describe more complex, sophisticated page layouts than HTML)

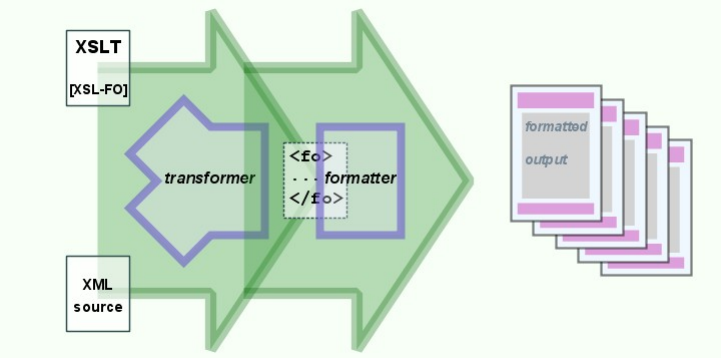
10

So an XSL-FO Document is

- An XML document
- Tagged in the XSL-FO tag set
- That contains your text and graphic content wrapped in formatting object tags

11

How XSL-FO Formatting Works



12

The Complete XSL-FO Picture

- In XSL-FO processing
  - Tags and content in an XML document
  - Are transformed
    - from their XML tags
    - into XSL-FO tags
      - Which describe the styling
      - And hold the content
  - XSL-FO rendering engine
    - understand these tags
    - makes pages from them (PDF, Postscript, RTF, etc.)
  - And printed/displayed using a display engine (like Adobe Acrobat for the PDF or Word for the RTF)

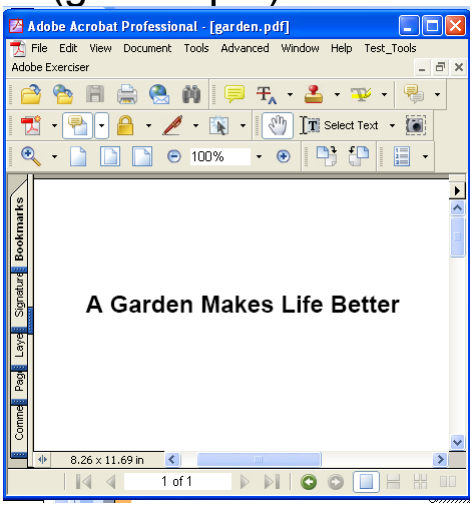
13

XML Input File & PDF Output

- XML Input (garden.xml)

```
<?xml
  version="1.0"?>
<chapter>
  <title>A Garden
  Makes Life
  Better</title>
</chapter>
```

- PDF Output (garden.pdf)



XSL + XML → XSL-FO (1/2)

- File garden-fo.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   xmlns:fo="http://www.w3.org/1999/XSL/Format"
   version='1.0'>
   <xsl:output method="xml"/>
   <xsl:template match="/">
      <fo:root>
         <fo:layout-master-set>
            <fo:simple-page-master master-
            name="simple" page-height="29.7cm"
            page-width="21cm" margin-
            left="2.5cm" margin-right="2.5cm">
               <fo:region-body margin-top="3cm"/>
            </fo:simple-page-master>
         </fo:layout-master-set>
```

15

XSL + XML → XSL-FO (2/2)

```
         <fo:page-sequence master-reference="simple">
            <fo:flow flow-name="xsl-region-body">
               <xsl:apply-templates/>
            </fo:flow>
         </fo:page-sequence>
      </fo:root>
   </xsl:template>
   <xsl:template match="chapter/title">
      <fo:block  font-weight="bold" font-size="18pt">
         <xsl:apply-templates/>
      </fo:block>
   </xsl:template>
</xsl:stylesheet>
```

```
C:\>xalan.bat –in garden.xml –xsl garden-fo.xsl –out gardent.fo
```

16

The Formatting Object File

- File garden.fo

```
<?xml version="1.0" encoding="UTF-8"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
    <fo:layout-master-set>
        <fo:simple-page-master margin-right="2.5cm"
        margin-left="2.5cm" page-width="21cm" page-
        height="29.7cm" master-name="simple">
            <fo:region-body margin-top="3cm"/>
        </fo:simple-page-master>
    </fo:layout-master-set>
    <fo:page-sequence master-reference="simple">
        <fo:flow flow-name="xsl-region-body">
            <fo:block font-size="18pt" font-weight="bold">A
                Garden Makes Life Better
            </fo:block>
        </fo:flow>
    </fo:page-sequence>
</fo:root>
```

17

Using Apache FOP

- FOP: Formatting Object Processor
  - A print formatter for XSL formatting objects.
- It can be used to render an XML file containing XSL formatting objects into a page layout.
- The main target is PDF but other rendering targets are supported, such as AWT, PCL, text and direct printing.
- Command Line Example
  - fop.bat -fo garden.fo -pdf garden.pdf

18

Role of the XSL-FO Engine

- Understands the XSL-FO vocabulary

  <fo:block>

  <fo:wrapper>

  <fo:wrapper font-style="italic">
- Produces display for paper
- An XSL-FO programmer needs to know which XSL-FO tags to use to make the pages he/she wants

19

Features of XSL-FO Formatting

- XSL-FO
  - Is a typesetting application
  - Has a complex, extensible Page Model (multicolumn)
  - Has sophisticated text typography (hanging indents, multi-font)
- XSL-FO usually created by transformation
- Formatting is based on XML tagging
- Internationalization designed in

20

XSL-FO Deal with Typesetting Complexity

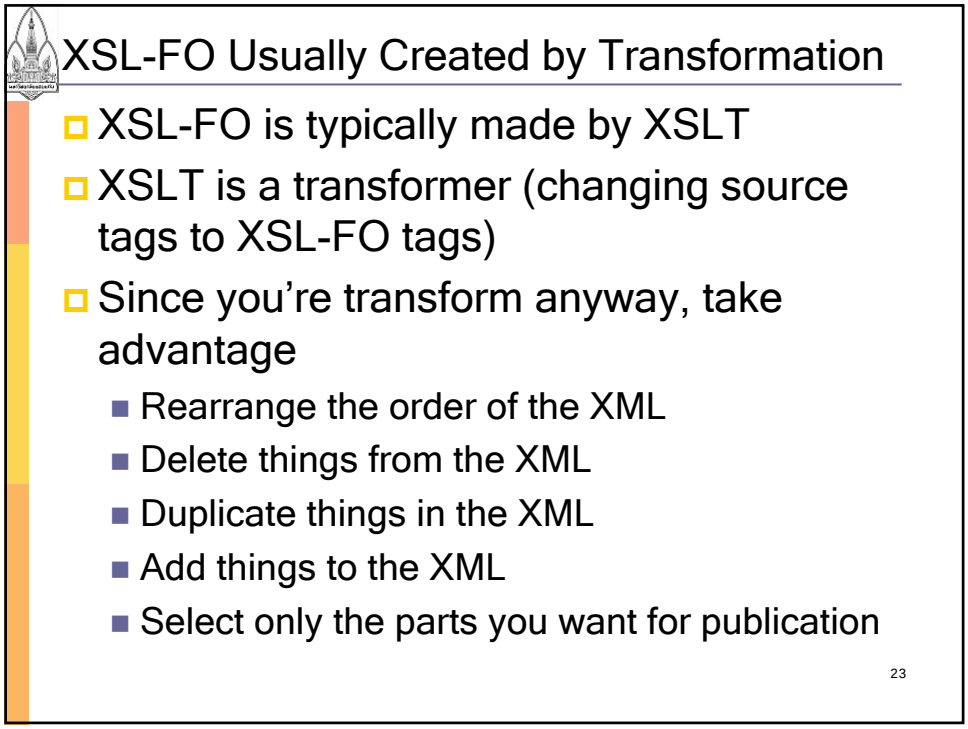
- Multiple columns
- Endnotes and footnotes
- Block-level behaviors (leading, margins, padding, space before/after)
- Text-level behaviors (font family, style, weight, size, color)
- Hyphenation and justification
- Marginalia
- Rules and leaders
- Page markers and links
- Autogenerated cross-references, Tables of Contents, indexes

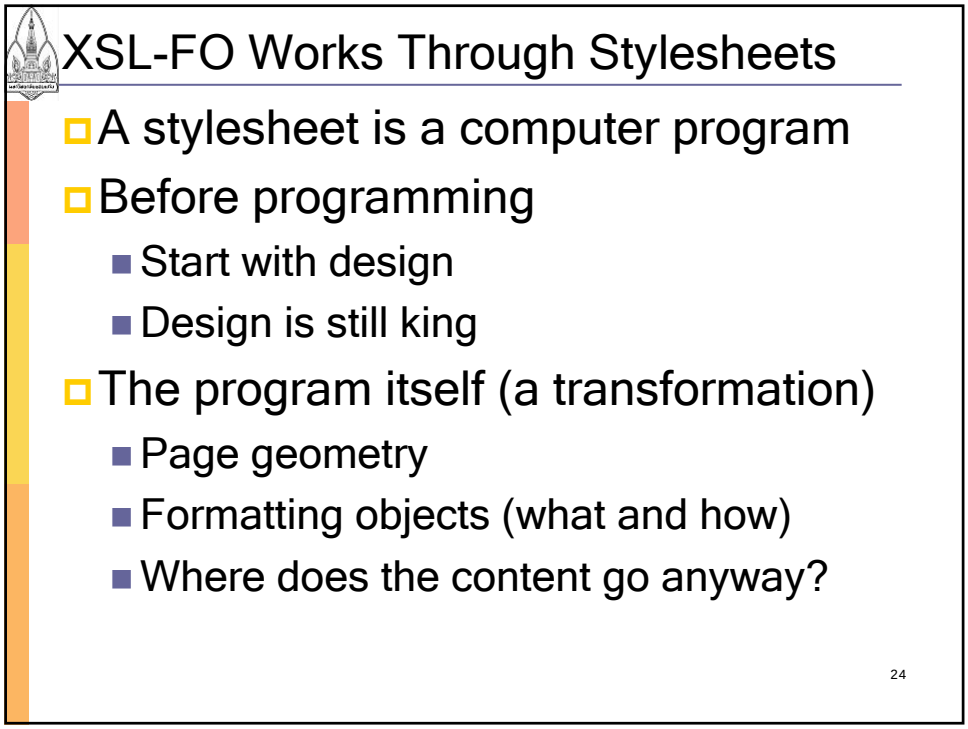
21

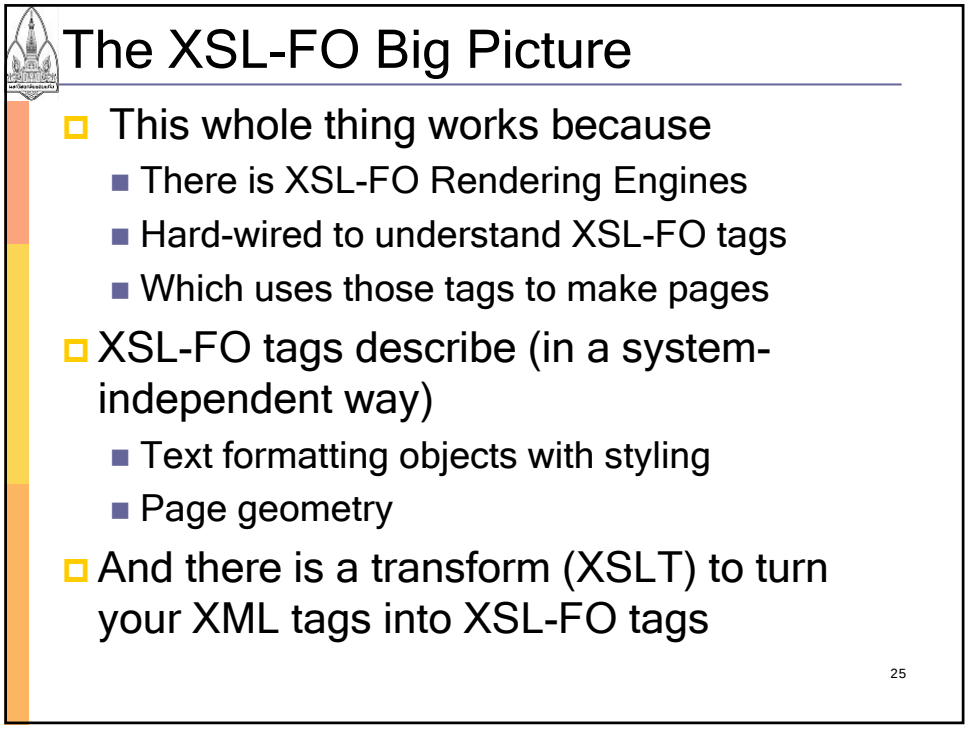
XSL-FO's Page Model

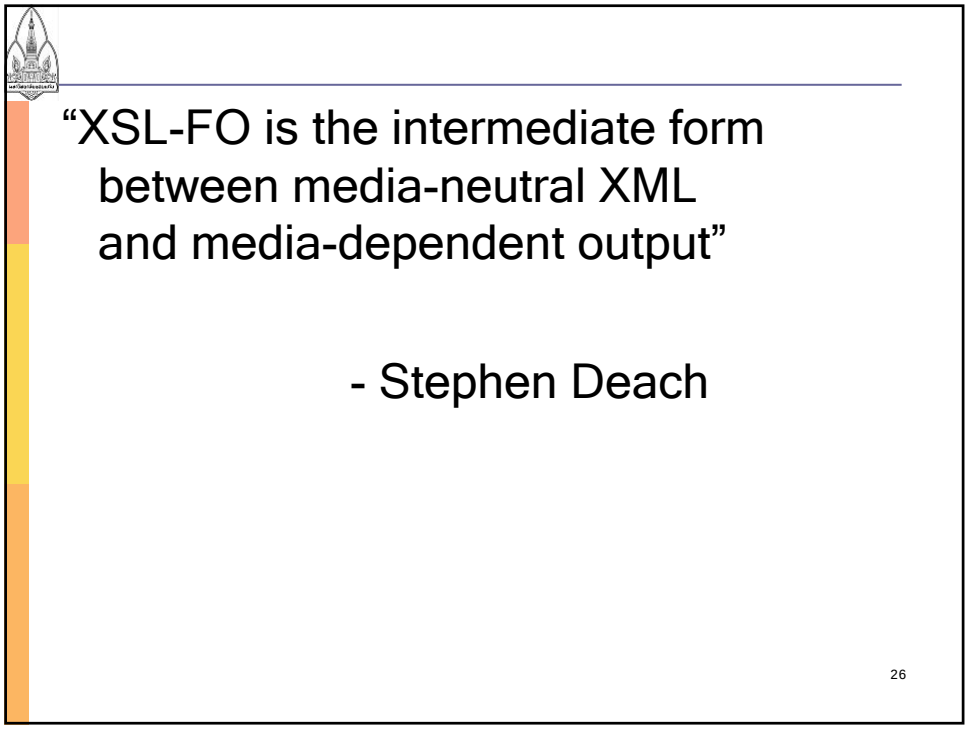
- XSL-FO describes page layout
  - page size
  - margins and columns and gutters
  - headers and footers
  - side-bars, etc
- Can describe sequences of different page layout templates
  - First page followed by later pages
  - Recto/verso alternating

22


XSL-FO Usually Created by Transformation
XSL-FO is typically made by XSLT
XSLT is a transformer (changing source tags to XSL-FO tags)
Since you're transform anyway, take advantage
Rearrange the order of the XML
Delete things from the XML
Duplicate things in the XML
Add things to the XML
Select only the parts you want for publication
23


XSL-FO Works Through Stylesheets
A stylesheet is a computer program
Before programming
Start with design
Design is still king
The program itself (a transformation)
Page geometry
Formatting objects (what and how)
Where does the content go anyway?
24


The XSL-FO Big Picture
This whole thing works because
There is XSL-FO Rendering Engines
Hard-wired to understand XSL-FO tags
Which uses those tags to make pages
XSL-FO tags describe (in a system-independent way)
Text formatting objects with styling
Page geometry
And there is a transform (XSLT) to turn your XML tags into XSL-FO tags
25


"XSL-FO is the intermediate form between media-neutral XML and media-dependent output"
- Stephen Deach
26

All XSL-FO Programs Start with Page Design

- Page layout(s) must be designed
  - First page/recto/verso pages
  - Headers and footers
  - Whitespace and margins
- Text typography must be designed
  - Fonts
  - Text size and leading
  - Title design and indents
- There is still a people process

27

Components of an XSL-FO Document (1/2)

- An XSL-FO document has two major parts
- Both parts are enclosed in a top-level element <fo:root>
- Part one contains page descriptions (layouts and page masters)
  - General layout of each potential page
  - Instructions to the rendering engine on which page templates to use when
  - Inside element <fo:layout-master-set>

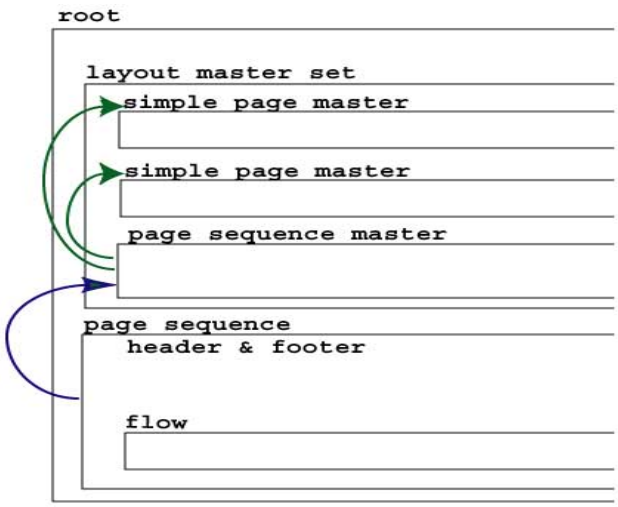
28

Components of an XSL-FO Document (2/2)

- Part two contains page sequences (content flows)
  - Assign content to pages
  - Assign formatting styles (properties) to content
  - Inside one or more <fo:page-sequence> elements

29

Organization of a Typical XSL-FO Document



30

XSL-FO Document Structure

```
<?xml version="1.0"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">

   <fo:layout-master-set>
      <fo:simple-page-master master-name="A4">
            <!-- Page template goes here -->
      </fo:simple-page-master>
   </fo:layout-master-set>

   <fo:page-sequence master-reference="A4">
      <!-- Page content goes here -->
   </fo:page-sequence>

</fo:root>
```

31

Page Descriptions (Layouts)

- Describe dimensions and aspects of one page
  - In print, this is a single print page (8.5 by 11, A4, etc.)
  - For web design these can be BIG pages with scrollable regions (HTML-style page)
  - Provides
    - A name, so we can send content to that named page-design
    - Page width
    - Page margins
    - Dimensions of page regions
- Inside element <fo:simple-page-master>

32

Multiple Page Masters

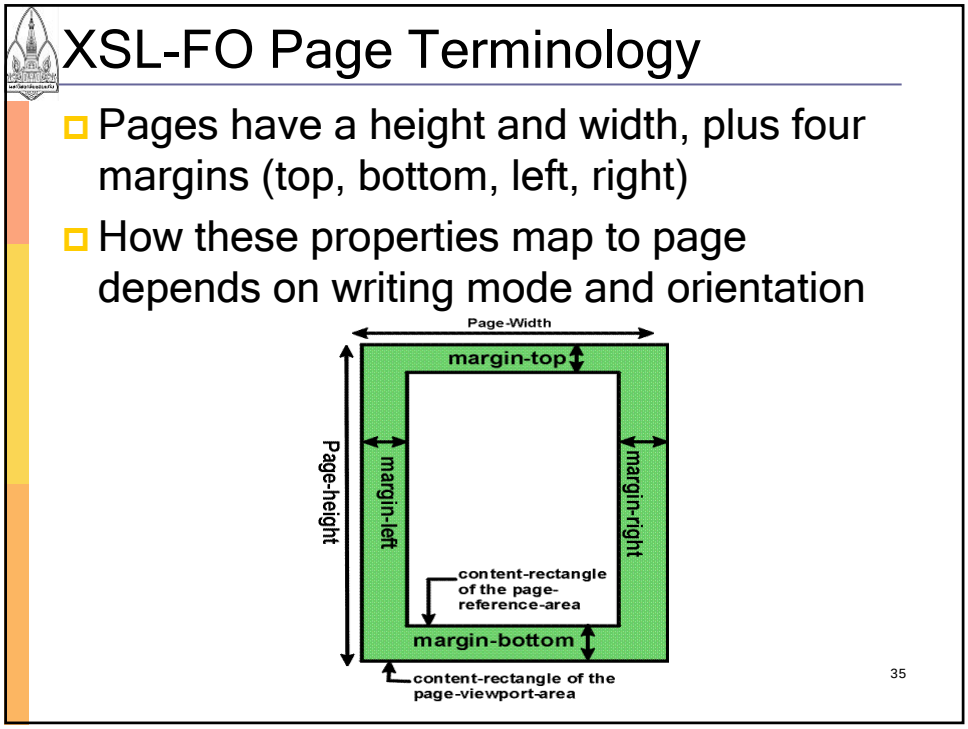
- You can have more than one <fo:simple-page-master>
  - With different identifiers
  - Useful if you have regular sequences of page layouts
  - E.g., chapter first page followed by alternative left and right hand pages

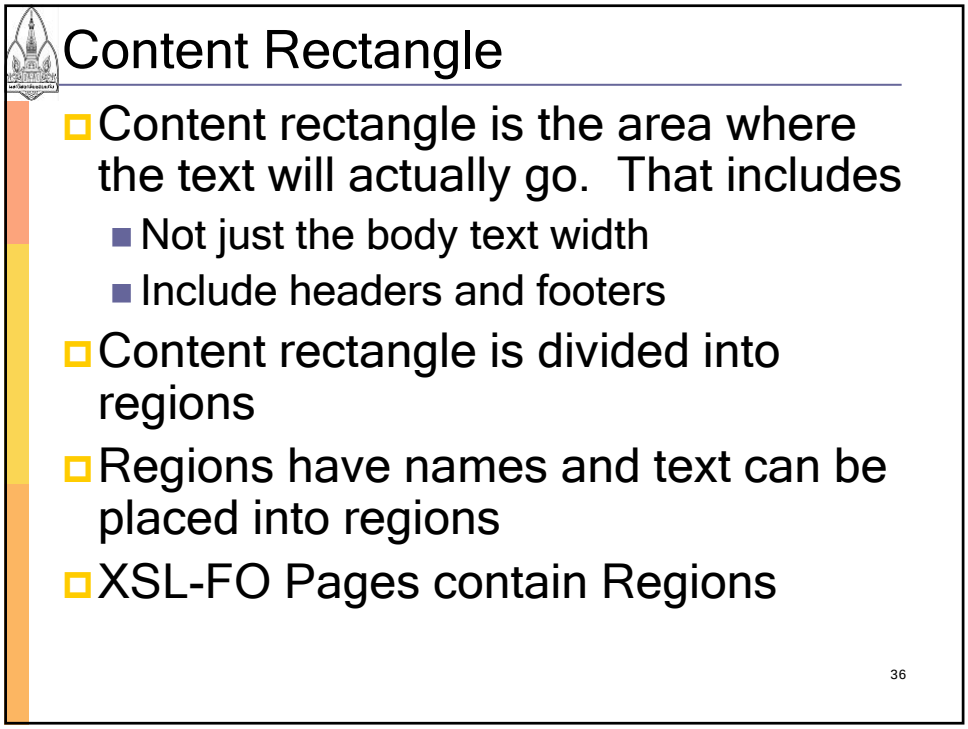
33

XSL-FO Areas

- The XSL formatting model defines a number of rectangular areas (boxes) to display output
  - Pages
  - Regions
  - Block areas
  - Line areas
  - Inline areas

34


XSL-FO Page Terminology
Pages have a height and width, plus four margins (top, bottom, left, right)
How these properties map to page depends on writing mode and orientation
Page-Width
margin-top
Page-height
margin-left
margin-right
content-rectangle of the page-reference-area
margin-bottom
content-rectangle of the page-viewport-area
35


Content Rectangle
Content rectangle is the area where the text will actually go. That includes
Not just the body text width
Include headers and footers
Content rectangle is divided into regions
Regions have names and text can be placed into regions
XSL-FO Pages contain Regions
36

Geometry of Regions Inside the Content Rectangle

- Content area contains five **region-viewport-areas**
- These are set *inside* the page in the space between the page margins
- Provide space for
  - any content flows (such as paragraphs) or
  - **static content** (such as headers or footers)

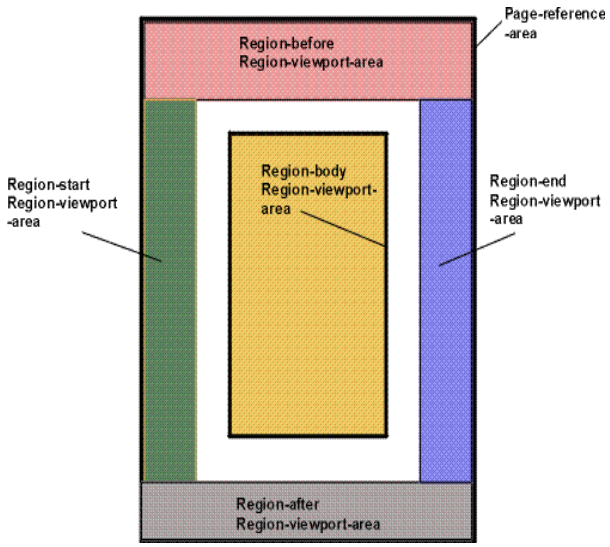
37

XSL-FO Regions

- Each XSL-FO Page contains a number of Regions
  - Region-body (the body of the page)
  - Region-before (the header of the page)
  - Region-after (the footer of the page)
  - Region-start (the left sidebar)
  - Region-end (the right sidebar)

38

Content Rectangle Regions



39

Technical Note: Regions

- Regions can overlap, based on
  - Size they are defined to be (extent)
  - Margin of neighboring region
  - Amount of overlap = extent – margin
- Whether text will overflow or be scrolled, clipped, etc., can also be connected through properties
- The region-body region (but not other regions) may contain columns

40

XSL-FO Block Areas

- XSL-FO Block areas define small block elements (the ones that normally start with a new line) like paragraphs, tables and lists
- XSL-FO Block areas can contain other Block areas, but most often they contain Line areas

41

XSL-FO Line Areas and Inline Areas

- XSL-FO Line Areas
  - XSL-FO Line areas define text lines inside Block areas
  - XSL-FO areas contain Inline areas
- XSL-FO Inline Areas
  - XSL-FO Inline areas define text inside Lines (bullets, single character, graphics, and more)

42

Example Types of Formatting Object

- Inline-level FOs
- Block-level FOs
- FOs for Lists
- FOs for Tables
- Pages and Layout Objects
- Out of Line FOs (includes links)

43

Inline-level FOs

- These things are in the same line as the things around them
- Do not add new structure or change writing direction, font families and sizes
- Sample tags
  - <fo:inline>
  - <fo:wrapper>
  - <fo:page-number>

44

Block-level FOs

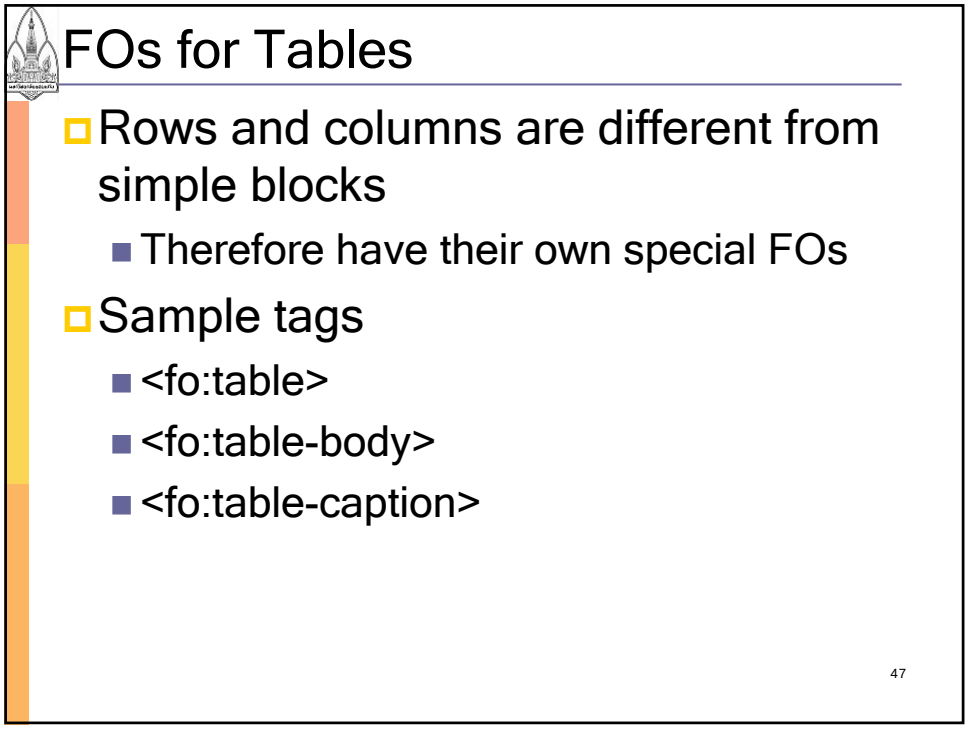
- Things that are separate from the things around them, not runin
- Blocks include paragraphs, titles, and block quotes
- Sample tags
  - <fo:block>
  - <fo:block-container>

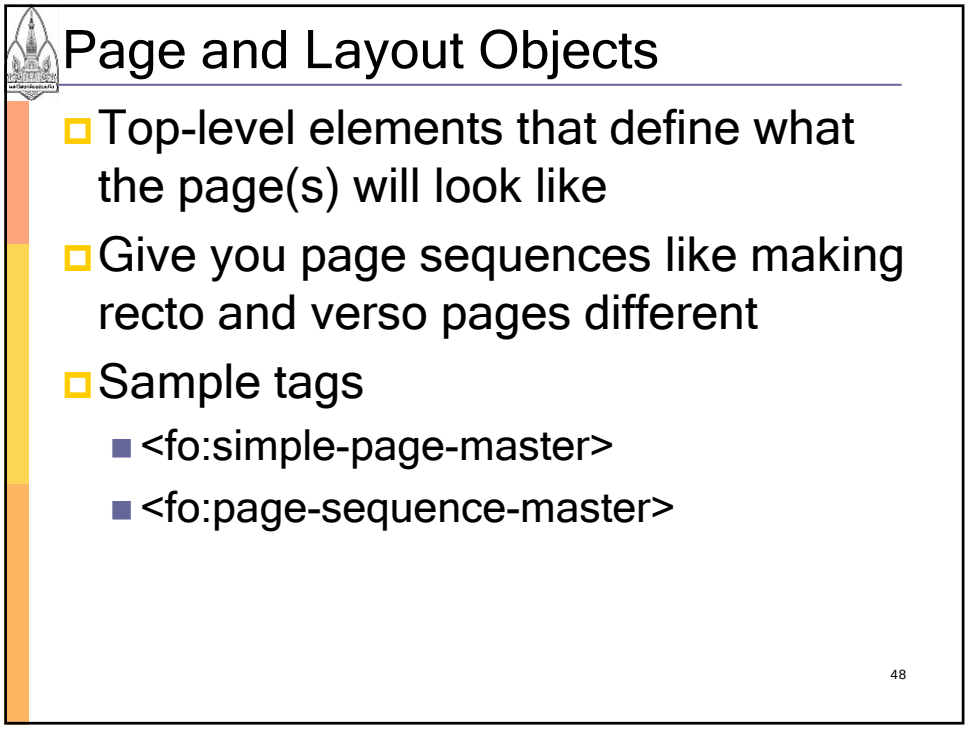
45

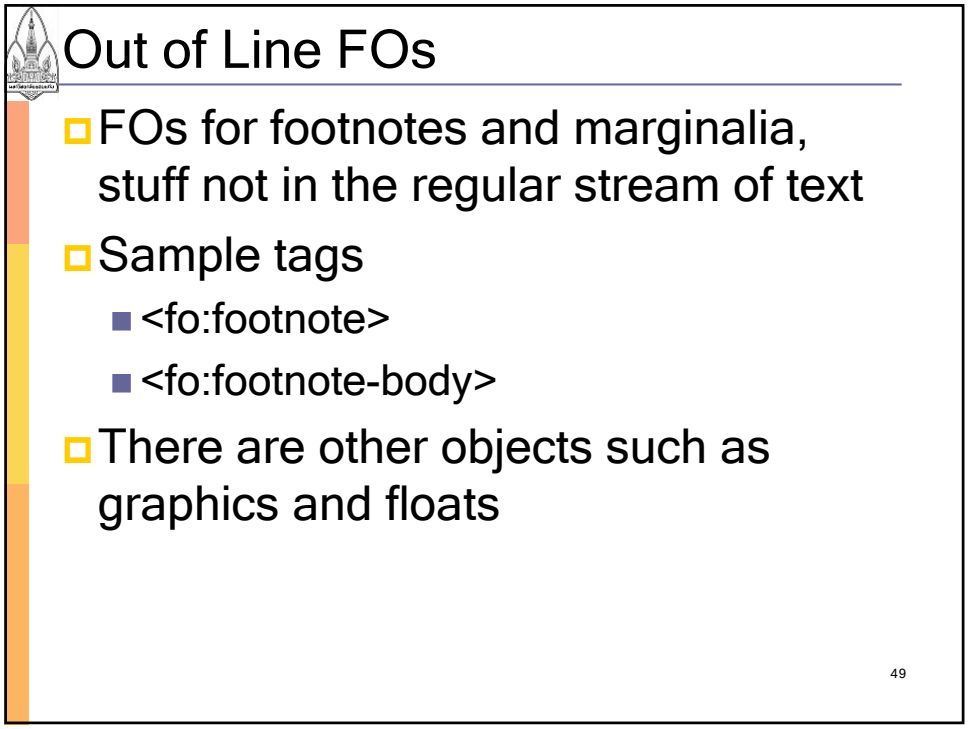
FOs for Lists

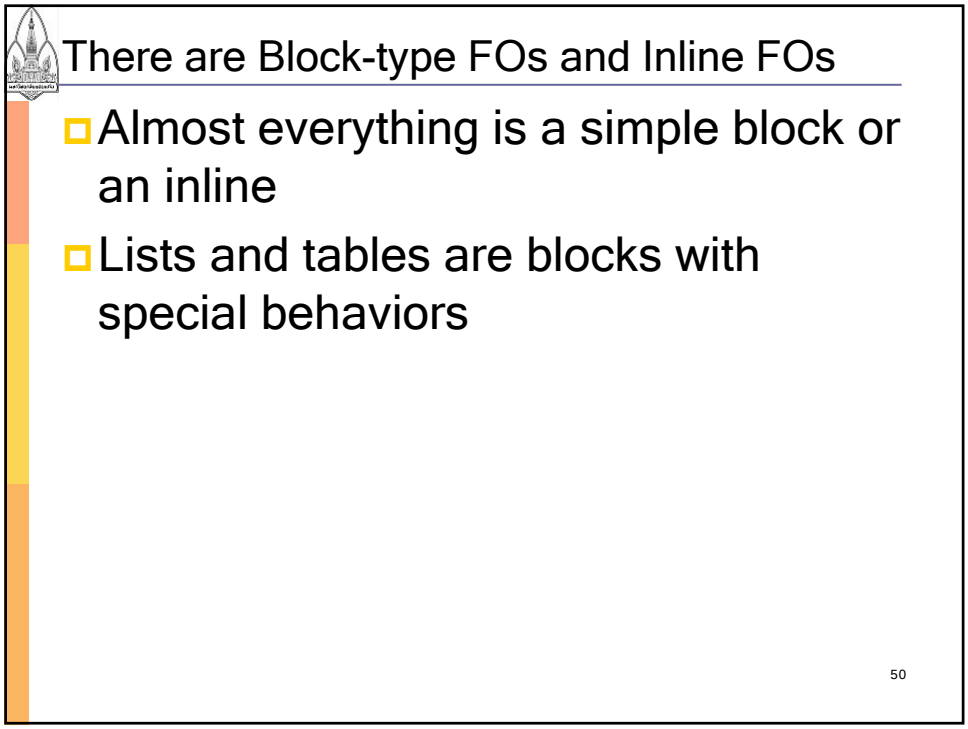
- Lists are complex block-type objects, typically composed of list items
- Lists are typically composed of
  - Some sort of prefix character (like a bullet, number)
  - Then the body of the list item
- Sample tags
  - <fo:list-block>
  - <fo:list-item>

46


FOs for Tables
Rows and columns are different from simple blocks
Therefore have their own special FOs
Sample tags
<fo:table>
<fo:table-body>
<fo:table-caption>
47


Page and Layout Objects
Top-level elements that define what the page(s) will look like
Give you page sequences like making recto and verso pages different
Sample tags
<fo:simple-page-master>
<fo:page-sequence-master>
48


Out of Line FOs
FOs for footnotes and marginalia, stuff not in the regular stream of text
Sample tags
<fo:footnote>
<fo:footnote-body>
There are other objects such as graphics and floats
49


There are Block-type FOs and Inline FOs
Almost everything is a simple block or an inline
Lists and tables are blocks with special behaviors
50

Up to This Point

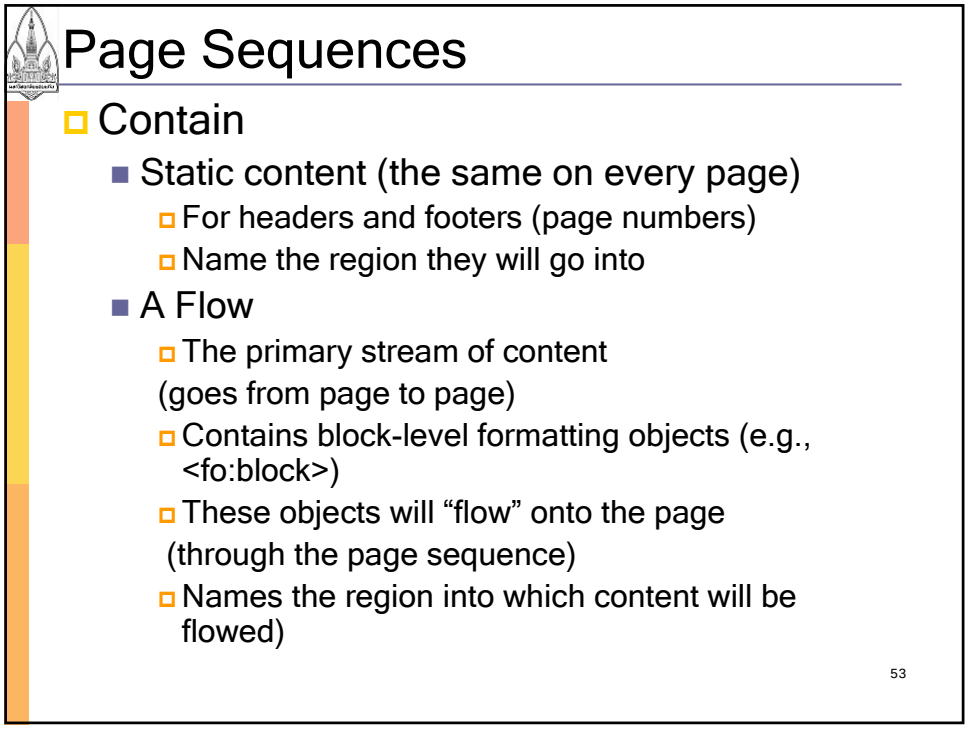
- We have described pages and pages geometry
- Put page layout is only half the story
- Where is the text, the content?
- Where does the content go?

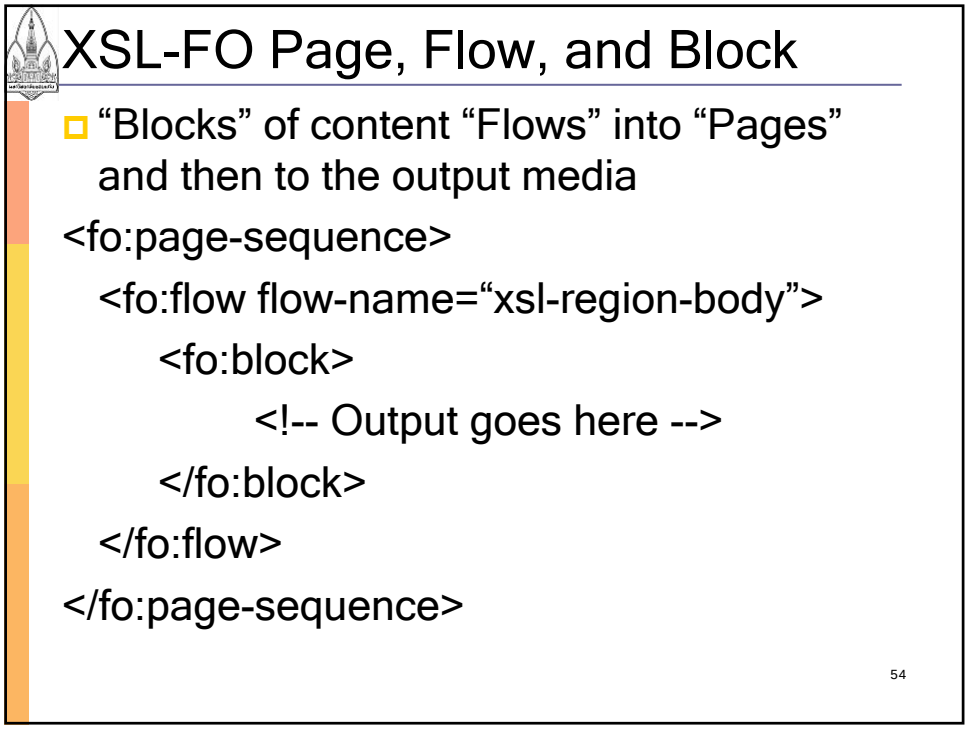
51

Where the Content Goes

- That's the second half of the stylesheet
- Content goes into Page Sequences
- <page-sequence> is a wrapper element for content
- As Dave Pawson expressed it "The page-sequence element contains the content to fill a sequence of pages"
- Page sequences contain Formatting Objects such as <fo:block>s

52


Page Sequences
Contain
Static content (the same on every page)
For headers and footers (page numbers)
Name the region they will go into
A Flow
The primary stream of content
(goes from page to page)
Contains block-level formatting objects (e.g., <fo:block>)
These objects will “flow” onto the page
(through the page sequence)
Names the region into which content will be flowed)
53

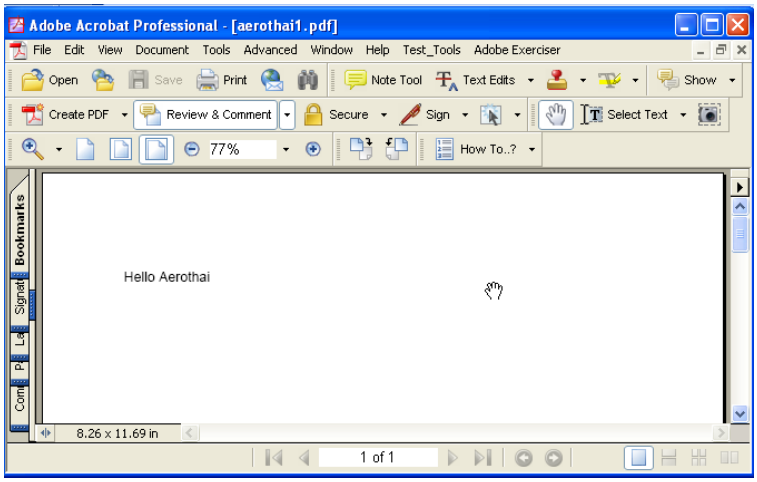

XSL-FO Page, Flow, and Block
“Blocks” of content “Flows” into “Pages” and then to the output media
<fo:page-sequence>
<fo:flow flow-name=“xsl-region-body”>
<fo:block>
<!-- Output goes here -->
</fo:block>
</fo:flow>
</fo:page-sequence>
54

XSL-FO Input1

```
<?xml version="1.0"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
   <fo:layout-master-set>
        <fo:simple-page-master margin-right="2.5cm"
        margin-left="2.5cm" page-width="21cm"
      page-height="29.7cm" master-name="A4">
                <fo:region-body margin-top="3cm"/>
           </fo:simple-page-master>
   </fo:layout-master-set>
   <fo:page-sequence master-reference="A4">
        <fo:flow flow-name="xsl-region-body">
                <fo:block>Hello Aerothai</fo:block>
        </fo:flow>
   </fo:page-sequence>
</fo:root>
```

55

XSL-FO Output1



56

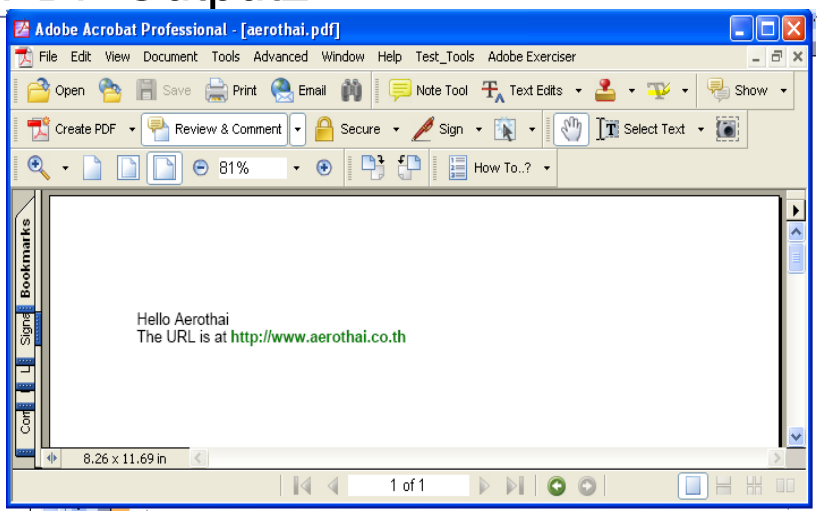
XML Input2 (aerothai.xml)

- aerothai.xml

```
<?xml version="1.0"?>
<company>
  <name>Aerothai</name>
  <url>http://www.aerothai.co.th</url>
</company>
```

57

PDF Output2



xalan.bat –in aerothai.xml –xsl aerothai-fo.xsl –out aerothai.fo

fop.bat –fo aerothai.fo –pdf aerothai.pdf

58

Stylesheet to produce XSL-FO

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

    xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <xsl:output method="xml"/>
  <xsl:template match="/">
    <fo:root
    xmlns:fo="http://www.w3.org/1999/XSL/Format">
      <fo:layout-master-set>
        <fo:simple-page-master
          margin-right="2.5cm" margin-left="2.5cm" page-width="21cm" page-height="29.7cm"
          master-name="A4">
          <fo:region-body margin-top="3cm"/>
        </fo:simple-page-master>
      </fo:layout-master-set>
```

```
      <fo:page-sequence master-reference="A4">
        <fo:flow flow-name="xsl-region-body">
          <xsl:apply-templates/>
        </fo:flow>
      </fo:page-sequence>
    </fo:root>
  </xsl:template>
  <xsl:template match="company">
    <fo:block>
      <fo:block>Hello <xsl:value-of select="name"/></fo:block>
      <fo:block>The URL is at
                                    <fo:inline font-weight="bold" color="green"><xsl:value-of select="url"/></fo:inline>
                                </fo:block>
    </fo:block>
  </xsl:template>
</xsl:stylesheet>
```

59

XSL Input2

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

    xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <xsl:output method="xml"/>
  <xsl:template match="/">
    <fo:root
    xmlns:fo="http://www.w3.org/1999/XSL/Format">
      <fo:layout-master-set>
        <fo:simple-page-master
          margin-right="2.5cm" margin-left="2.5cm" page-width="21cm" page-height="29.7cm"
          master-name="A4">
          <fo:region-body margin-top="3cm"/>
        </fo:simple-page-master>
      </fo:layout-master-set>
```

```
      <fo:page-sequence master-reference="A4">
        <fo:flow flow-name="xsl-region-body">
          <xsl:apply-templates/>
        </fo:flow>
      </fo:page-sequence>
    </fo:root>
  </xsl:template>
  <xsl:template match="company">
    <fo:block>
      <fo:block>Hello <xsl:value-of select="name"/></fo:block>
      <fo:block>The URL is at
                                    <fo:inline font-weight="bold" color="green"><xsl:value-of select="url"/></fo:inline>
                                </fo:block>
    </fo:block>
  </xsl:template>
</xsl:stylesheet>
```

60

Content Formatting Happens Inside the Flow

- Various Formatting Objects are placed inside the flow…
- Blocks are used to format simple structures
  - Paragraphs, Titles, Lines
  - Captions, Display quotes, Section heads, etc.
- Other Formatting Objects are available for more complex structures (lists, tables, footnotes, etc. are more complex)

61

Blocks Take Typographic Properties

- Blocks define the way the text will look through properties
  - Font

  font-family, font-size, font-weight
  - Color

  color, background-color
  - Spacing

  space-before, space-after
  - Line height
  - Indents and margins
- Blocks may be nested, and inherit properties from ancestors

62

Block Properties are Attributes of the <fo:block> Element

```
<fo:block
  text-align="start"
  margin-left="1.5in"
  space-after="6pt"
  line-height="18pt"
  color="red"
  font-family="sans-serif"
  font-weight="bold"
  font-size="16pt">Hello, World</fo:block>
```

63

The Actual Text goes Inside the Block

- This XML

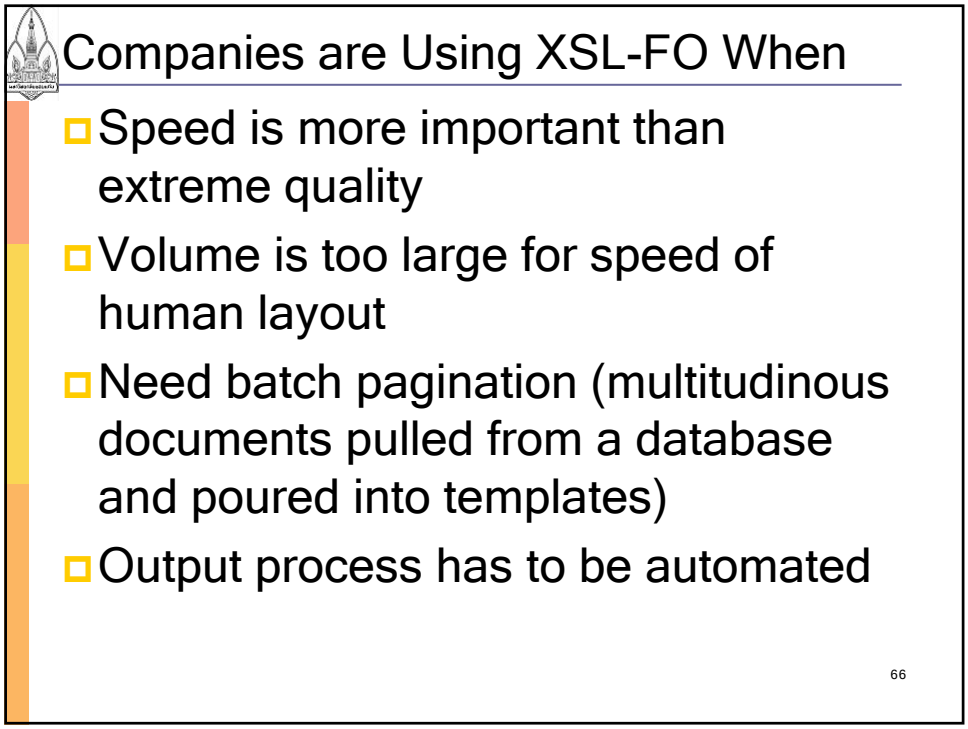
  ```
  <recipe><title>Carrots with Ginger
    Sauce</title>...
  ```

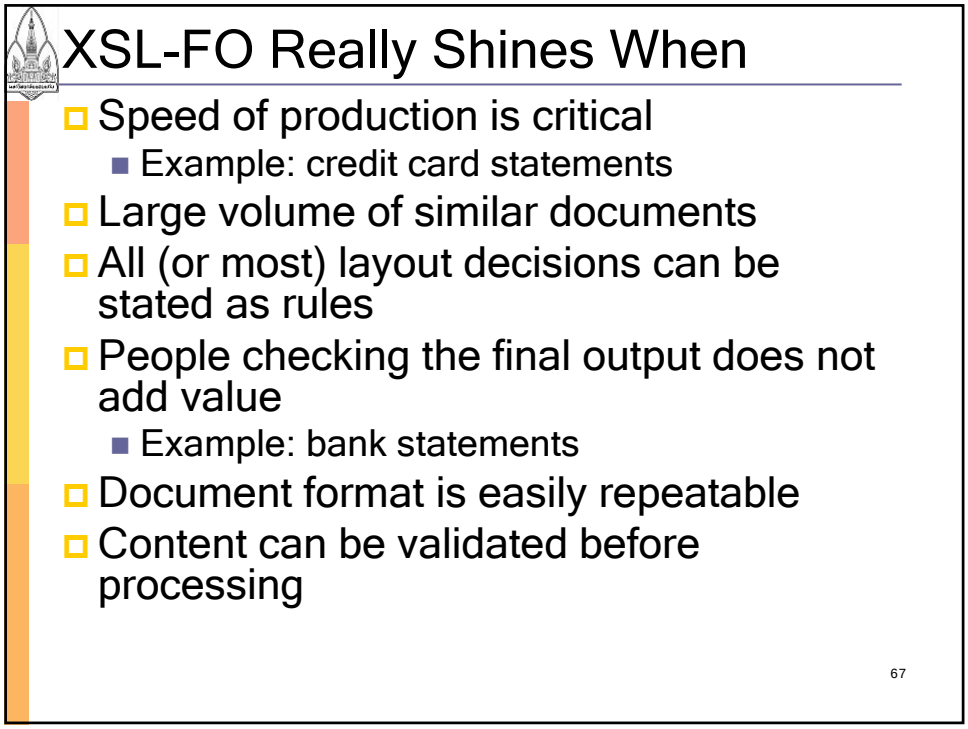
- Might be put inside a block like this

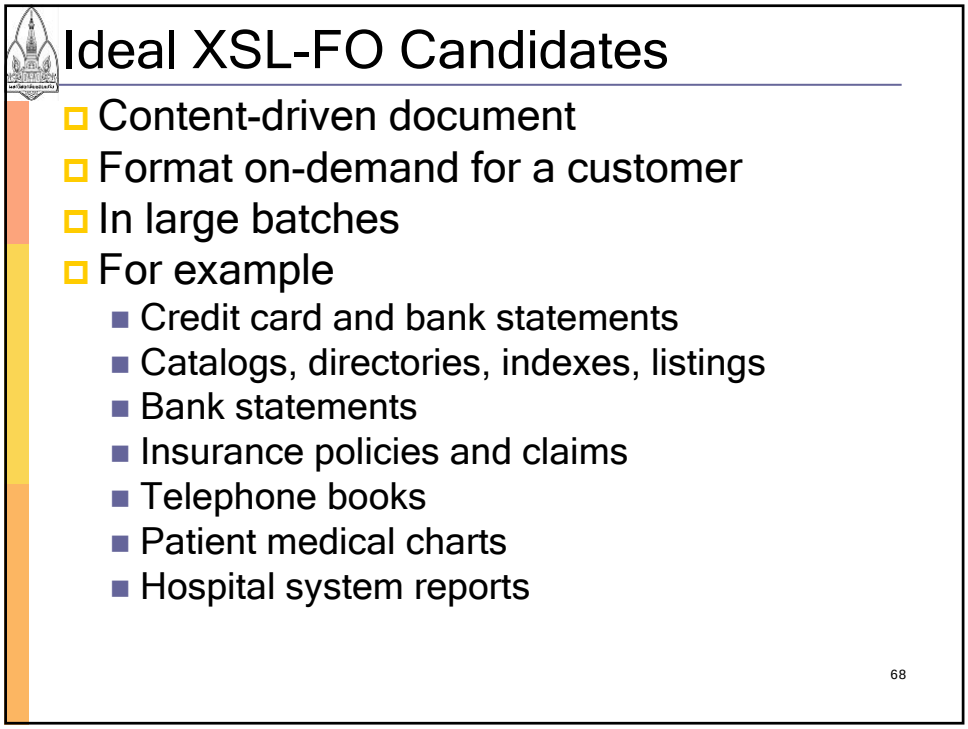
  ```
  <fo:block font-weight="bold" font-
    size="14pt">Carrots with Ginger
    Sauce</fo:block>
  ```

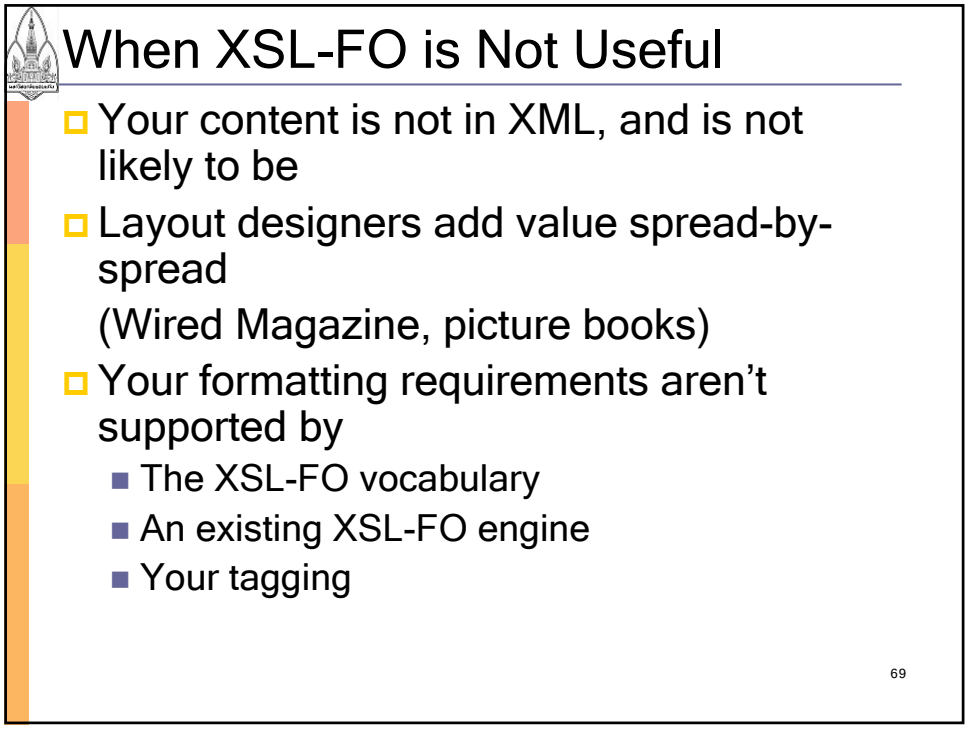
64

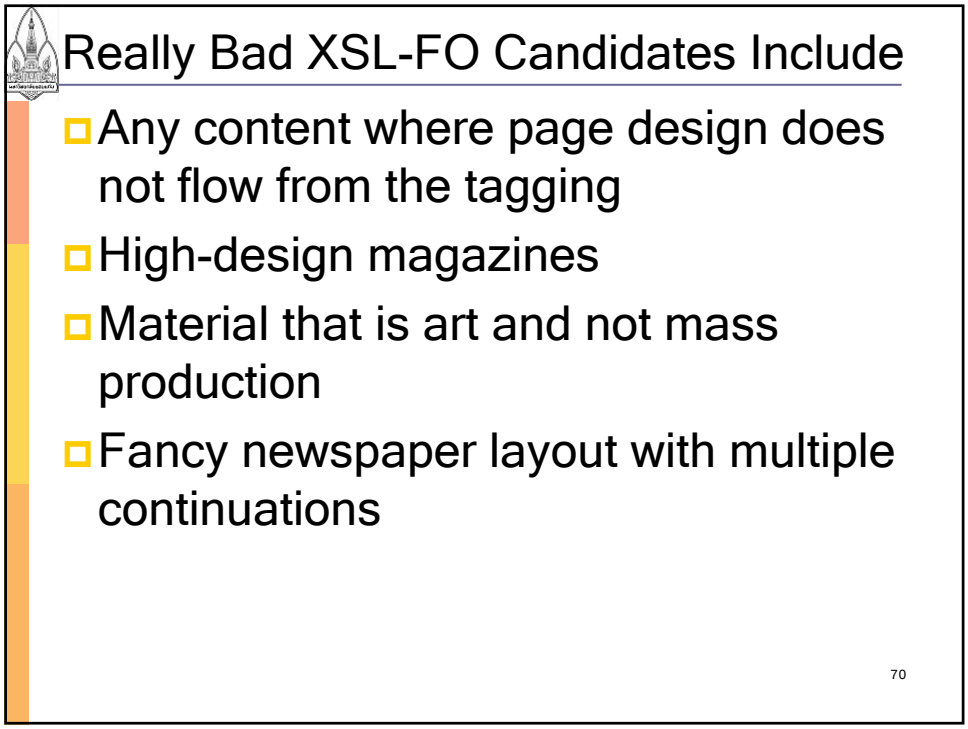

XSL-FO Software Vendors Name Their Clients
Visa International
Bank of America
Mastercard
European Customs Union
Pratt & Whitney
Nike Customer Service Center
Bell South
TransCanada Pipelines Limited
SAS
Cisco
Siemens AG
Semantec
Credit Swiss
Hewlett-Packard
SAP
Airbus
Lockheed Martin
HSBC
Network Appliances
65


Companies are Using XSL-FO When
Speed is more important than extreme quality
Volume is too large for speed of human layout
Need batch pagination (multitudinous documents pulled from a database and poured into templates)
Output process has to be automated
66


XSL-FO Really Shines When
Speed of production is critical
Example: credit card statements
Large volume of similar documents
All (or most) layout decisions can be stated as rules
People checking the final output does not add value
Example: bank statements
Document format is easily repeatable
Content can be validated before processing
67


Ideal XSL-FO Candidates
Content-driven document
Format on-demand for a customer
In large batches
For example
Credit card and bank statements
Catalogs, directories, indexes, listings
Bank statements
Insurance policies and claims
Telephone books
Patient medical charts
Hospital system reports
68


When XSL-FO is Not Useful
Your content is not in XML, and is not likely to be
Layout designers add value spread-by-spread
(Wired Magazine, picture books)
Your formatting requirements aren't supported by
The XSL-FO vocabulary
An existing XSL-FO engine
Your tagging
69


Really Bad XSL-FO Candidates Include
Any content where page design does not flow from the tagging
High-design magazines
Material that is art and not mass production
Fancy newspaper layout with multiple continuations
70

You Might Not Want XSL-FO If …

- Content is well-formed XML with no DTD or schema

  (Too much variation; too many surprises)
- Page-limited productions, where text will be altered to fit the space
- Most text changes are made after layout
- DTD or schema changes constantly

(XSL-FO stylesheet need constant modification to keep up)

- Content is highly variable in structure

71

What Need to Know to Write an XSL-FO Stylesheet

- XML source tags (what are they, what do they mean)
- Target page design (look, behavior)
- The XSL-FO vocabulary (how to make the desired effects)
- Some XSLT, maybe quite a lot (how to do the transform)
- Relation between XML source and desired effects

72

Tools Setup Procedure

- Set environment variable LOCAL_FOP_HOME to directory fop-0.20.5
- Test by echo %LOCAL_FOP_HOME%
- Add directories that has “fop.bat” and “xalan.bat” in PATH environment variable
- xalan.bat –in aerothai.xml –xsl aerothai-fo.xsl –out aerothai-fo.xml
- fop.bat –fo aerothai-fo.xml –pdf aerothai.pdf

73

References

- Deborah Aleyne Lapeyre and B. Tommie Usdin, “Introduction to XSL-FO Concepts”, http://www.mulberrytech.com/papers/Intro2XSL-FO/
- J. David Eisenberg, “Using XSL Formatting Objects”, http://www.xml.com/pub/a/2001/01/17/xsl-fo/index.html

74