WORLD WIDE WEB consortium

Publishing on the Web using XSL

Max Froumentin, W3C

Page 2002

7 February 2002

http://www.w3.org/People/maxf/talks/2002-02-XSL/talk.pdf

I. HTML and XML

1. HTML
2. XML

II. Styling and XML

3. Style Sheets
4. CSS
5. Shortcomings of CSS

III. XSLT and XSL-FO

6. The XSL Process
7. The XSL syntax
8. Two specifications
9. XSLT
10. Formatting Objects
11. Example

IV. XSL on the Web

12. Principle
13. Pre-generation
14. Dynamic generation
15. In-browser XSLT
16. Example
17. Example II (MathML)
18. The MathML Stylesheet

V. Advanced Usage

19. Conversion
20. Programming
21. Example

VI. The Future

22. XSLT 2.0

23. XSL 2.0
24. Conclusion

1. HTML and XML

HTML and XML: HTML

HTML is the language of the Web: text with a set of *tags* to mark headers, paragraphs, etc.

```
<html>
 <h1>Title of the document</h1>
 <p>This is the <i>text</i> of
  the document</p>
</html>
```

- `h1`: level 1 heading (title)
- `p`: paragraph
- `i`: italics, etc.

HTML and XML: XML

XML generalises HTML: one can define a language with any tag.

```
<conference>
 <name>Page2002</name>
 <date day="6-8" month="2" year="2002"/>
 <location>Ikebukuro</location>
 <abstract>This conference is about...</ab
</conference>
```

In order to specify how to display the document (on paper, on the Web, etc.) it is necessary to *apply style* to it.

2. Styling and XML

Styling and XML: Style Sheets

Style sheets are useful for:

- Specifying the presentation of XML/HTML documents (e.g. make all headers red)
- Separating presentation from content (separate files, simpler to maintain)
- in particular to allow users to alter the look of Web pages they receive.
- Allowing devices to render Web pages as best they can.

Two ways to create style sheets: CSS and XSL

Styling and XML: CSS

Example: press release

CSS uses a simple syntax:

```
body {
  background-color: #ddd;
  font-size: 200%;
}
a { color: green; }
h1 { color: red; }
```

Styling and XML: Shortcomings of CSS

People concerned with styling needed more:

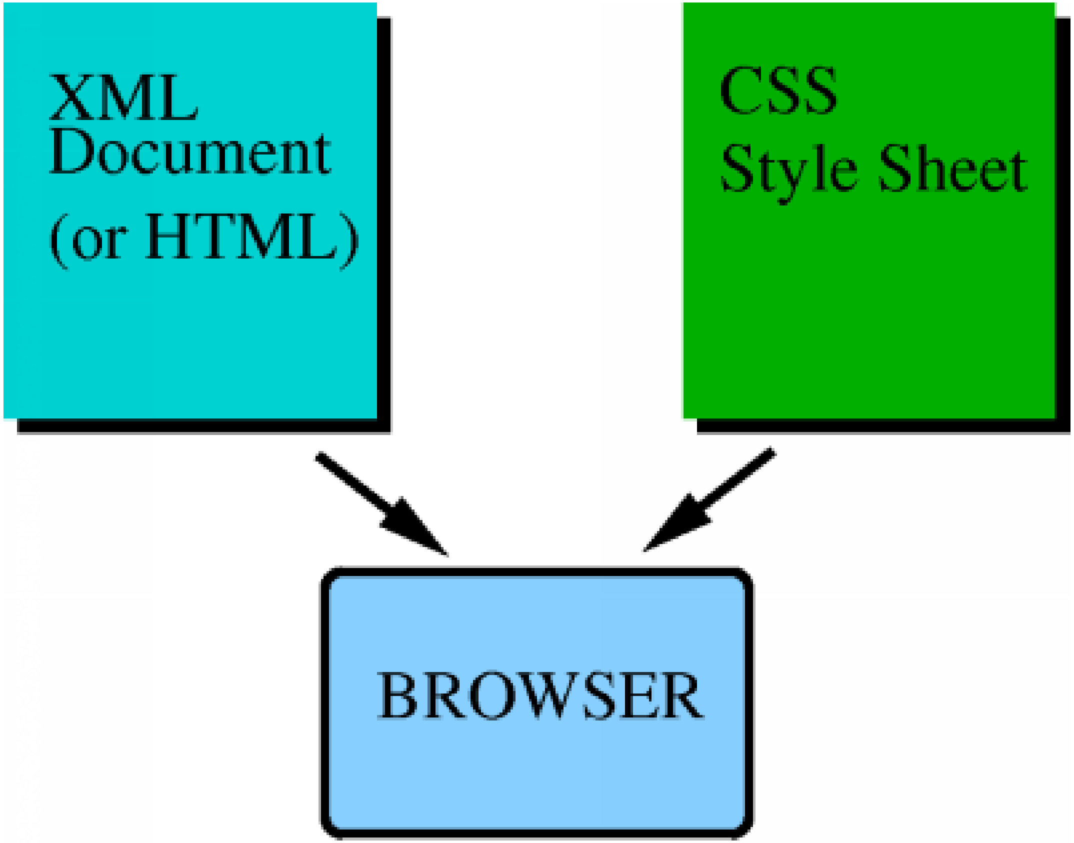
- Content generation: indexes, table of contents. More generally, allow the author to write as little as possible
- Complex formatting properties: pagination, I18N, etc. which CSS2 does not have.

XSL was designed to solve that. But it became much more...

3. XSLT and XSL-FO

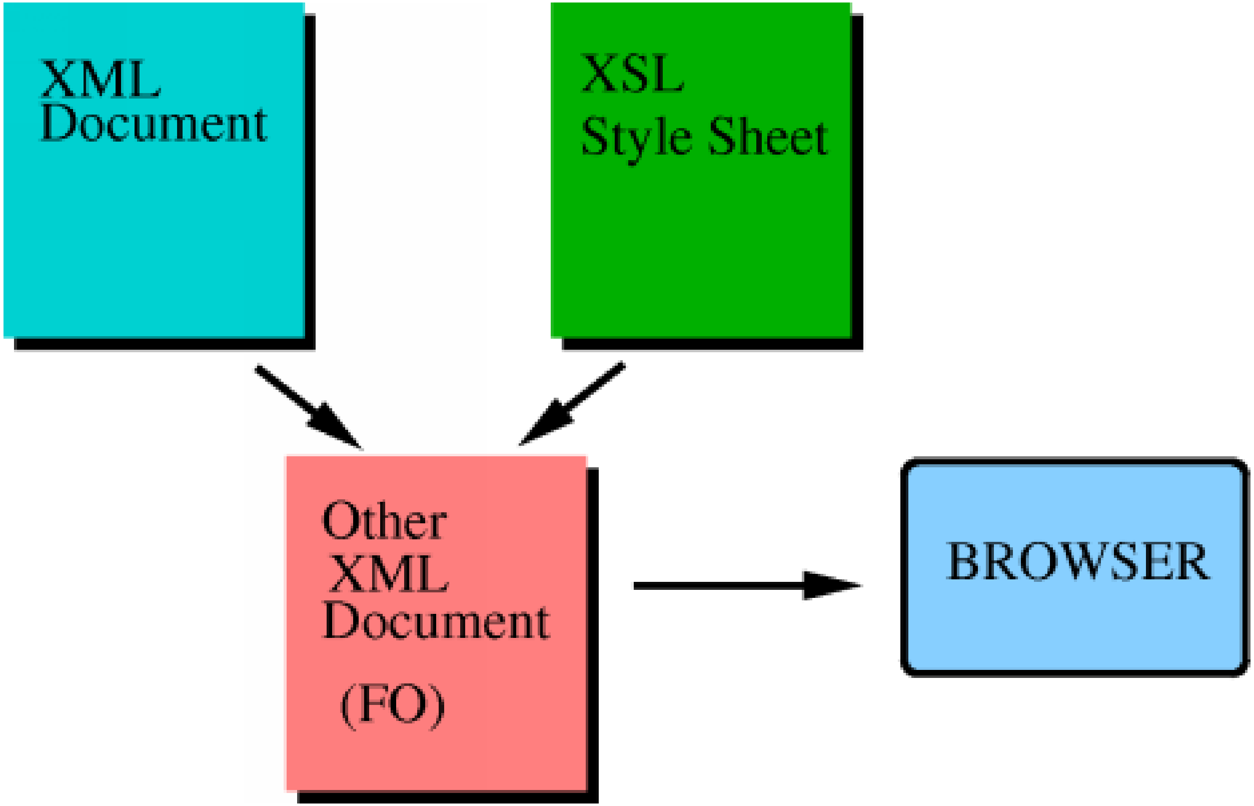
XSLT and XSL-FO: The XSL Process

Whereas CSS does this:



XSLT and XSL-FO: The XSL Process

XSL is designed to operate as:



XSLT and XSL-FO: The XSL Process

FO is an XML vocabulary, with presentation semantics. It defines:

- Tags such as `<block>`, `<inline>`, `<page-sequence>`, `<footnote>`
- Attributes such as text-indent, writing-mode, color (many borrowed from CSS)

  ```
  <block text-indent="30pt"
  writing-mode="rl-tb">
  ```
- A pagination model and area model to specify the placement on pages

XSLT and XSL-FO: The XSL syntax

A stylesheet is a set of templates:

```
<xsl:template match="body">
  <page-sequence background-color="#ddd"
                 font-size="200%">
    <xsl:apply-templates/>
  </page-sequence>
</xsl:template>
```

But there are many more constructs that CSS rules do not have, like programming language statements: loops, conditions, etc. Stylesheets can become quite complex.

XSLT and XSL-FO: Two specifications

The XSL spec is split to reflect this process:

- Transformations: anyXML-to-FO
- Rendering: how to render Formatting Objects

The Great Idea was to make the XSL transformations more general: anyXML-to-anyOtherXML and to make it to a separate specification: XSLT.

XSLT and XSL-FO: XSLT

- XSLT thus became a generic transformation language that is now very popular (in particular to transform custom XML documents into HTML for presentation in a browser).
- The rest of this talk will show examples of the use of XSLT.

XSLT and XSL-FO: Formatting Objects

- Formatting Objects serve XSL's original purpose of advanced pagination and is more and more used for technical documentation. The XSL 1.0 Recommendation defines how to render the FO vocabulary.
- We will not go deeper into FOs since they will be detailed in the next presentation, but an example follows.

XSLT and XSL-FO: Example

XSLideMaker

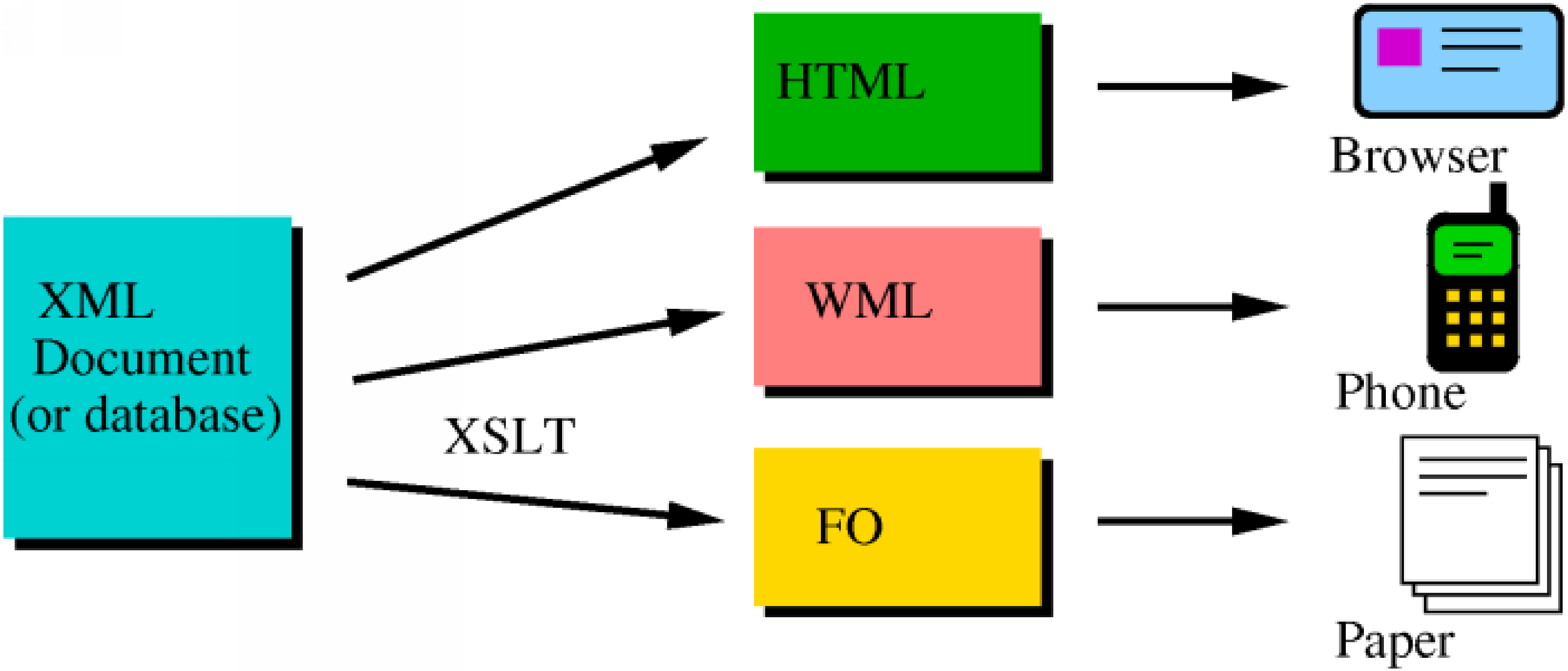
This set of slides was generated using XSL.

- source document: XHTML edited using Amaya, with figures in SVG and PNG.
- How the source document is displayed when edited is not important since an XSL stylesheet will "style" it to its final form.
- stylesheet that generates FO+SVG file, converted to PDF.

This shows a way to use XSL "off-line", without necessarily publishing them on the Web

4. XSL on the Web

XSL on the Web: Principle



With XSLT, it is possible to use XSL for publishing any kind of XML data to different sorts of devices (computers, phones, PDAs) that can display one type

XSL on the Web: Principle

of XML presentation language (HTML, WML, FO). Several scenarios exist for performing the transformation: pre-generation, on-demand transformation and browser-side.

XSL on the Web: Pre-generation

From one XML source, a set of stylesheets generate documents in various formats at different URLs:

```
http://www.example.org/news.html
http://www.example.org/news.svg
http://www.example.org/news.wml
```

Benefits:

- Standard stylesheet benefit: once stylesheets are written just the XML content needs to change
- Allows for simple client that knows only one format (SVG, WML, etc.)

XSL on the Web: Pre-generation

- Can also generate different versions for same-format, like

`http://www.example.org/news-big.html`

Example: The XSL page at W3C

- One XML source
- A set of stylesheets (XML to XHTML, XML to RSS, etc.)

The XHTML stylesheet performs complex operations: grouping, sorting, etc.

XSL on the Web: Pre-generation

Drawback of the static approach: that many useless pages could be generated. Also, different URL for version.

XSL on the Web: Dynamic generation

If client specifies a way to send its preferences, the server can generate the correct version as asked.

- Content negociation: using HTTP Accept headers, the client can specify what format it prefers

  ```
  Accept: application/svg+xhtml
  GET http://www.w3.org/news
  ```

- CC/PP: many other parameters: screen size, colour preferences, localisation, device speed, etc.

XSL on the Web: Dynamic generation

Advantages:

- one URL for every version of the page (`http://www.w3.org/news`)
- more parameters can: different styles for each format (colors, size, etc.)

Drawbacks:

- High demand on server (simultaneous requests)

XSL on the Web: In-browser XSLT

More and more Web browsers can perform XSLT transforms.

- The transformation can be done in the browser: thta is how it was meant to be from the start, but until it was actually possible (implemented), people used a few of the other methods.
- The language and stylesheets are the same as above
- Server load decreased
- Client preferences can be processed locally

XSL on the Web: In-browser XSLT

XSLT in browsers is becoming more and more popular:

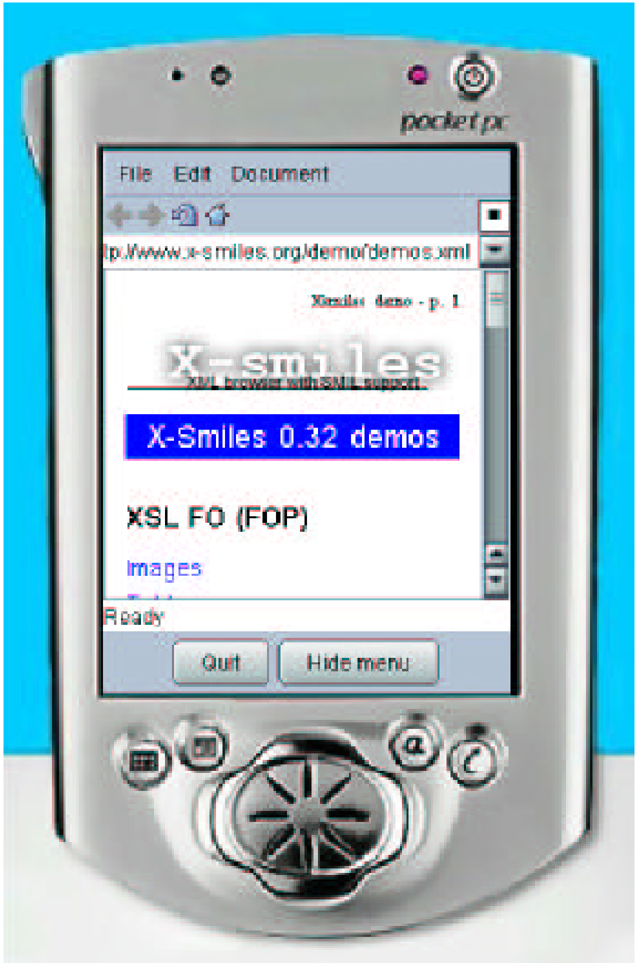
- to do XML to HTML: IE, Mozilla
- to do XML to FO: Antenna House's XML Formatter, X-smiles

XSL on the Web: Example

X-smiles: a Java Web browser for PCs and small devices.
Does in-browser XSLT and renders:

- HTML, Xforms
- Formatting Objects
- SMIL, SVG, etc.

XSL on the Web: Example



XSL on the Web: Example II (MathML)

XSLT to display formulas in many browsers, using MathML:

```
<math>
 <msup>
  <mrow>
   <mi>a</mi><mo>+</mo><mi>b</mi>
  </mrow>
  <mrow>2</mrow>
 </msup>
</math>
```

Few browsers understand MathML , but XSLT gives a solution...

XSL on the Web: The MathML Stylesheet

Using XSLT in a HTML document, one can display MathML in most popular browsers

```
<?xml-stylesheet href="mml.xsl">
<html>
 ...
 <math>...</math>
 ...
</html>
```

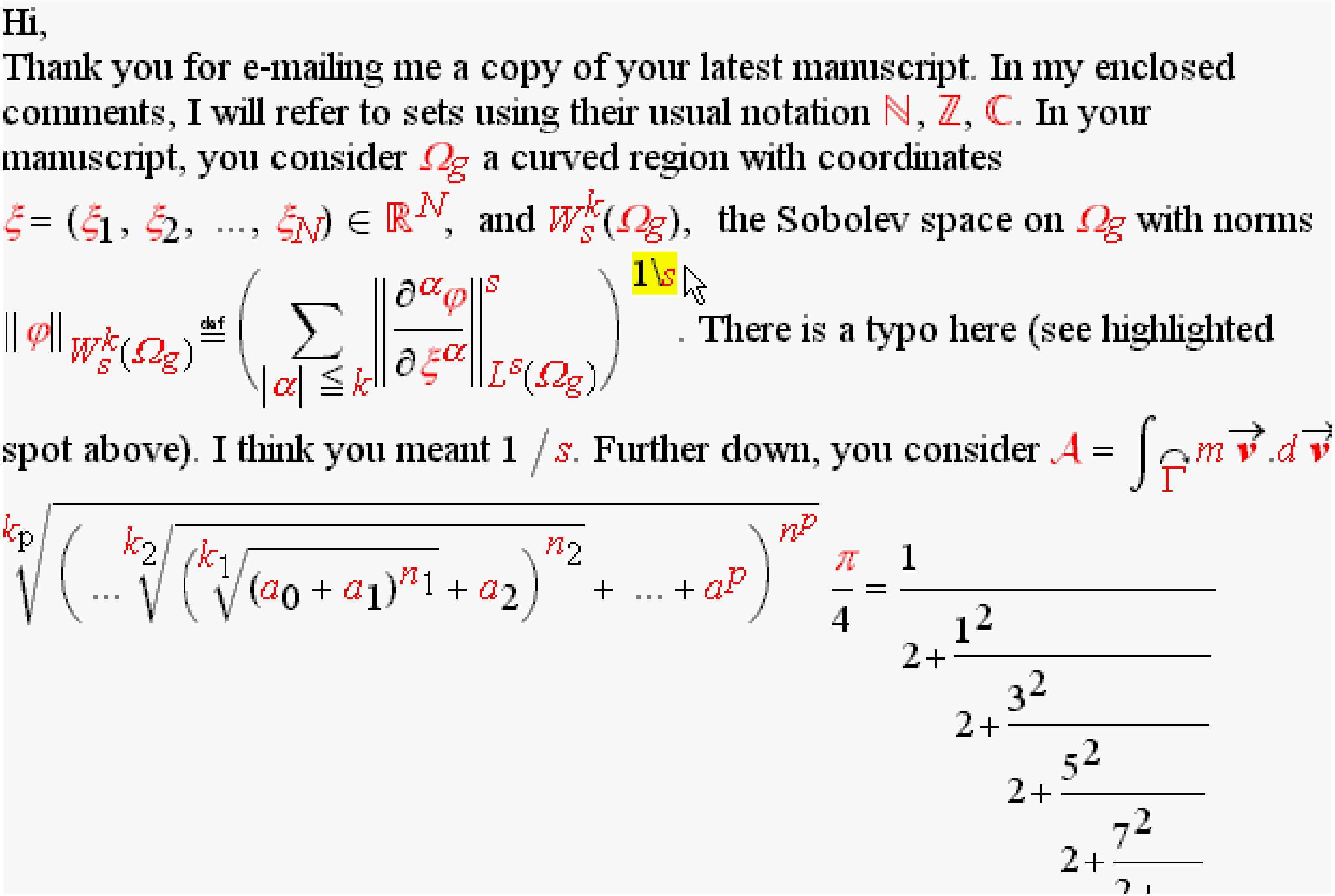
- Amaya: no XSLT, direct rendering
- Mozilla: XSLT does nothing, direct rendering

XSL on the Web: The MathML Stylesheet

- IE5.5, IE6:
  - If a MathML plug-in is installed, the stylesheet calls it (according to the user's preferences)
  - If no MathML plug-in is installed, an approximate rendering using tables and CSS is performed

This does not work with browsers that do not support XSLT or MathML.

XSL on the Web: The MathML Stylesheet



Hi,

Thank you for e-mailing me a copy of your latest manuscript. In my enclosed comments, I will refer to sets using their usual notation ℕ, ℤ, ℂ. In your manuscript, you consider Ω_g a curved region with coordinates $\xi = (\xi_1, \xi_2, \ldots, \xi_N) \in \mathbb{R}^N$, and $W_s^k(\Omega_g)$, the Sobolev space on Ω_g with norms

$$\|\varphi\|_{W_s^k(\Omega_g)} \overset{\text{def}}{=} \left(\sum_{|\alpha| \leq k} \left\| \frac{\partial^{\alpha}\varphi}{\partial \xi^{\alpha}} \right\|_{L^s(\Omega_g)}^{s} \right)^{1\backslash s}$$

. There is a typo here (see highlighted spot above). I think you meant $1\,/\,s$. Further down, you consider $A = \int_{\overset{\frown}{\Gamma}} m\,\vec{v}\,.\,d\,\vec{v}$

$$\sqrt[k_p]{\left(\ldots\sqrt[k_2]{\left(\sqrt[k_1]{(a_0+a_1)^{n_1}+a_2}\right)^{n_2}+\ \ldots+a^p}\right)^{n^p}} \qquad \frac{\pi}{4} = \cfrac{1}{2+\cfrac{1^2}{2+\cfrac{3^2}{2+\cfrac{5^2}{2+\cfrac{7^2}{2+}}}}}$$

5. Advanced Usage

Advanced Usage: Conversion

XSLT has been found useful for other uses, such as:

- MathML to SVG,
- HTML to RSS (w3.org), etc.
- XML document validation or serialisation

Advanced Usage: Programming

- Including variables in other languages: `<xsl:variable>` and `<xsl:value-of>` allow implementation of general expressions with variables in an XML file. e.g. geometric constraints in SVG.

```
<svg ...>
  <xsl:variable name="start_x" select=
  <rect x="$start_x * 2" y="40".../>
  <circle x="$startx_x + 24" y="20"...
</svg>
```

Advanced Usage: Programming

- An XML programming language?
  - XSLT was not meant to be one, so it would be simple to use.
  - But people do crazy things with it, like generating a file from no output. This is not easy (since number iterations) are hard to implement, but people do it!

Advanced Usage: Example

Chess stylesheet

- ChessGML to SVG+SMIL animation
- This is a rather complex style sheet that turns a ChessGML file to an SVG animation of the board.

ChessGML:

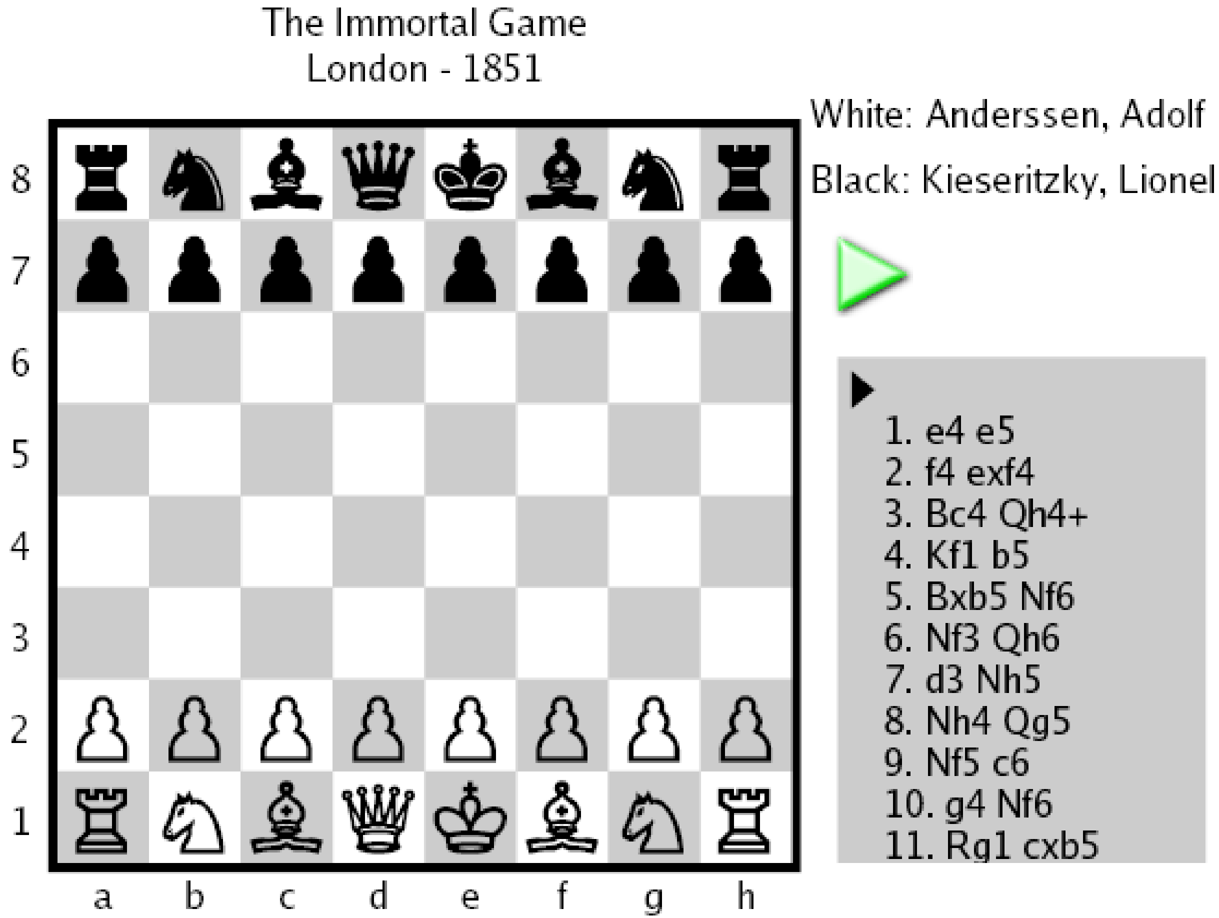
```
<mp> <!-- 4. Kf1 b5 -->
  <m c="w"><p c="w" n="k"/><e1/><f1/></m>
  <m c="b"><p c="b" n="p"/><b7/><b5/></m>
</mp>
```

Advanced Usage: Example

SVG:

```
<!-- move 4 -->
<animateTransform attributeName="transform
te" from="0,0" to="0,-4.5" accumulate="sum
="1s" restart="never" fill="freeze" xlink
<animate id="move4" begin="move3.end" xlin
buteType="XML" dur="2s" from="55" to="25"
<animate xlink:href="#F" attributeName="y"
nd" dur="2s" from="78.5" to="48.5" fill="f
```

Advanced Usage: Example



The Immortal Game
London - 1851

White: Anderssen, Adolf

Black: Kieseritzky, Lionel

1. e4 e5
2. f4 exf4
3. Bc4 Qh4+
4. Kf1 b5
5. Bxb5 Nf6
6. Nf3 Qh6
7. d3 Nh5
8. Nh4 Qg5
9. Nf5 c6
10. g4 Nf6
11. Rg1 cxb5

6. The Future

The Future: XSLT 2.0

The first Working Draft has been published:

- a more powerful language
- Designed in conjunction with XML Query
- designed for more complex use for XML databases.

The Future: XSL 2.0

There are plans for a new version of XSL with:

- More formatting objects and properties
- Generalised areas, flow control
- Compatibility with CSS3

The Future: Conclusion

- XSL is a simple yet powerful solution to do content management on the Web
- It is fully XML, allowing manipulation of all new W3C file formats: FO, SVG, MathML.
- May be confusing for programmers at first (different programming style), but worth trying!
- Very popular: many implementations and uses.