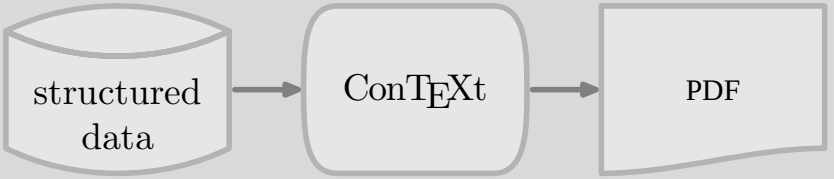
From database to presentation via XML, XSLT and ConTEXt

overview



In this session I cover three kinds of databases:

1. Going from data in XML (databases) to output.
2. Going from Comma Separated Variable files to output.
3. Going from data in relational databases to output.

The presented te$_X$niques are fairly general. TEX code is ConTEXt specific.

typesetting XML

Requirement: ConTEXt input data must be tabular (rows and columns).

Two solutions to typesetting it:

1. Turn the XML into ConTEXt commands.
2. But ConTEXt can handle the raw XML just fine. It has an embedded XML parser.

our special format

The following format is easily handled by ConTEXt or any TEX macro package that can typeset tables:

```
<rows>
  <row>
    <field>Re-introduction of ...</field>
    <field>Wlodzimierz Bzyl</field>
  </row>
  <row>
    <field>The Euromath System - ...</field>
    <field>J. Chlebíková, ...</field>
  </row>
  <row>
    <field>Instant Preview and ...</field>
    <field>Jonathan Fine</field>
  </row>
</rows>
```

turning XML into ConTEXt with XSL

```
1  <xsl:template match="/">
2  \starttext
3  <xsl:apply-templates/>
4  \stoptext
5  </xsl:template>

6  <xsl:template match="rows">
7  \starttable[|p(5cm)|p(8cm)|]
8  \HL<xsl:text/>
9  <xsl:apply-templates/>
10 \HL
11 \stoptable
12 </xsl:template>

13 <xsl:template match="row">
14 \VL <xsl:value-of select="field[2]"/>\VL <xsl:value-of select="f
15 </xsl:template>
```

typesetting XML with ConTEXt

Instead of using an XSL processor, you can immediately typeset XML straight in ConTEXt itself.

```
1  \defineXMLenvironment [rows]    \bTABLE \eTABLE
2  \defineXMLpickup      [row]     \bTR \eTR
3  \defineXMLpickup      [field]   \bTD \eTD

4  \processXMLfilegrouped {example.xml}
```

ConTEXt specific rules

Typesetting XML with ConTEXt:

1. Make sure your input data is in XML.
2. Make sure your XML is in tabular format (next topic).
3. Define mappings to the ConTEXt table, tabular or TABLE environment.
4. Use `\processXMLfilegrouped` to process your XML file.

overview
turning XML into ConTEXt with XSL
typesetting XML with ConTEXt
transforming XML to XML
typesetting CSV
typesetting SQL
close

transforming XML to XML

Usually XML isn't in the format that can be processed easily. You can use an XSLT processor to transform one format of XML into another format.

There are several types of XSLT stylesheets:

1. Fill-in-the-blanks stylesheets.
2. Navigational stylesheets.
3. Rule-based stylesheets.
4. Computational stylesheets.

to be transformed XML

```
1  <program>
2    <day weekday="Monday" date="24 September 2001">
3      <item time="9.00h"><opening/></item>
4      <item time="9.15h">
5        <presentation>
6          <author>Hans Hagen</author>
7          <title>Overview of presentations</title>
8        </presentation>
9      </item>
10     <item time="9.45h">
11       <presentation>
12         <author>Wlodzimierz Bzyl</author>
13         <title>Re-introduction of Type 3 fonts into the TeX worl
14       </presentation>
15     </item>
16     <break time="10.30h" type="coffee"/>
17     <item time="11.00u">
18       <presentation>
19         <author>Michael Guravage</author>
```

```
20          <title>Literate Programming: Not Just Another Pretty Fac
21        </presentation>
22      </item>
23      </day>
24  </program>
```

transformation

A transformation to our preferred format is:

```
1  <xsl:template match="program">
2    <rows>
3    <xsl:apply-templates select="day/item/presentation"/>
4    </rows>
5  </xsl:template>

6  <xsl:template match="presentation">
7    <row>
8      <field><xsl:value-of select="author"/></field>
9      <field><xsl:value-of select="title"/></field>
10   </row>
11 </xsl:template>
```

typesetting CSV

Some data resides in CSV files:

```
"Fred","Flintstone",40
"Wilma","Flintstone",36
"Barney","Rubble",38
```

Trick:

1. Convert it to XML first. I've written a simple Perl script to do exactly this.
2. Typeset the XML as explained.

typesetting SQL

Much of this worlds data resides in relational databases:

1. Retrieve data from it using the **select** statement. Every database has a command-line tool that can accomplish that.
2. Use this tool to output ConTEXt code,
3. Or XML.

example database

Table definition:

```
1  create table "family member" (
2    "id_family member" smallint not null primary key,
3    "surname" character varying(30) not null,
4    "family name" character varying(40) not null,
5    "age" smallint not null);
```

Get some data in it:

```
1  insert into "flintstone" ("id_flintstone", "surname", "family na
2    values (1, 'Fred', 'Flintstone', 40);

3  insert into "flintstone" ("id_flintstone", "surname", "family na
4    values (2, 'Wilma', 'Flintstone', 36);
```

creating XML from SQL

A simple ANSI SQL query to extract the data and sort it in surname is:

```
1  select surname, age
2    from flintstone
3    order by surname
```

SQL output is usually not returned in XML format, and certainly not in our example format.

I've written another Perl script to make decent XML from this.

Or simply this:

```
1  select '\VL', surname, '\VL', age, '\VL\SR'
2    from flintstone
3    order by surname
```