IBIS/XML - One step further

Atul P. Agarwal
IBIS Summit
Jan 2002
Outline

- Why IBIS/XML
- What has been demonstrated
- What needs to be done
- Issues
Why IBIS/XML

- Separation of content from presentation
- More structured representation
- New computing paradigms
  - Web Services
Presentation info in the IBIS spec

• “A line of the file may have at most 80 characters”
• “If RLC parameters are available for individual pins, they can be listed in columns 4-6 under keyword [Pin].”
• “The state line is terminated with the slash ‘/’, even if it extends over several lines to fit within the 80 character column width restriction.”
XML advantages

• Simple
• Standard
• Structured
• Textual

- The Fine print
  • XML documents can be very large
  • XML documents are difficult to read easily (need tools)
Web Services

• a new breed of Web applications
• self-contained, self-describing, modular applications
• can be published, located, and invoked across the Web.
• other applications (and other Web services) can discover and invoke the deployed service.
• built on XML/HTTP
Web Services Support Components

- **SOAP** (remote invocation)
- **UDDI** (trader, directory service)
- **WSDL** (expression of service characteristics)
- **XLANG/XAML** (transactional support for complex web transactions involving multiple web services)
- **XKMS** (XML Key Management Specification) - ongoing work by Microsoft and Verisign to support authentication and registration
Processing an XML Document

- **XML DOC**
- **DTD**
- **Style Sheet**

**Processing Agent**
- **XML Parser**
- **Parse Tree**
- **Rendering Agent**

- **3rd Party App**
- **Display Device**
IBIS/XML Flow demonstrated

- Text Ibis Model
  - Ibis C parser
  - IBIS XML Model
    - Validating Parser
      - IBIS DTD
        - Basic XSLT (Level 1)
      - XSLT processor
        - Transformed IBIS XML document
          - Parser/XSLT processor
            - Presentation XSLT (Level 2)
          - IBIS model Presentation
            - Third party application
            - Third party application
IBIS/XML for Pin

<Pin>
  <Pin_name_section>
    <Pin_name Max_len="5"> A11 </Pin_name>
  </Pin_name_section>
  <Signal_name_section>
    <Signal_name Max_len="20"> wrn </Signal_name>
  </Signal_name_section>
  <Pin_model_name_section>
    <Model_name Max_len="40"> BT2Z50CX </Model_name>
  </Pin_model_name_section>
</Pin>
XSL for Transformation 1

<xsl:template match = "Pin">
   <column width = "5">
      <xsl:value-of select = "Pin_name_section/Pin_name" />
   </column>
   <column width = "15">
      <xsl:value-of select = "Signal_name_section/Signal_name" />
   </column>
   <column width = "15">
      <xsl:apply-templates select = "Pin_model_name_section" />
   </column>
   <xsl:apply-templates select = "Pin_package_info_section" />
</xsl:template>
java org.apache.xalan.xslt.Process
   -IN Sample.xml
   -XSL sampleformat.xsl
   -OUT format.xml
## Presentation XML for a Pin

```xml
<rowdata>
  <column width="5"> A14 </column>
  <column width="15"> a[7] </column>
  <column width="15"> BT2Z50CX_PU50K </column>
  <column width="5"> 35m </column>
  <column width="5"> 3.69nH </column>
  <column width="5"> 0.48pF </column>
</rowdata>
```
XSL for Transformation 2

```xml
<xsl:template match="rowdata">
    <xsl:for-each select="column">
        <xsl:choose>
            <xsl:when test="child::keyword">
                </xsl:when>
            <xsl:when test="child::comment">
                <xsl:apply-templates select="comment" />
            </xsl:when>
            <xsl:otherwise>
                <xsl:value-of select="text()" />
                <xsl:call-template name="do-the-padding">
                    <xsl:with-param name="abc" select="text()" />
                    <xsl:with-param name="width" select="@width" />
                </xsl:call-template>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:for-each>
</xsl:template>
```
Generating the Text Format

```java
java org.apache.xalan.xslt.Process
   -IN format.xml
   -XSL ibistextformat.xsl
   -OUT sample1.ibs
```
The IBIS text format

A12 a[0] BT2Z50CX_PU50K 32m 3.45nH 0.46pF
XML for GND Clamp

<Gnd Clamp_section>
  <Comment>
    <Comment_char> | </Comment_char>
    <Comment_line> voltage I(typ) I(min) I(max) </Comment_line>
  </Comment>
  <Gnd Clamp>
    <voltage> <Number> -3.30000 </Number> </voltage>
    <typ> <Number> -11.46380A </Number> </typ>
    <min> <Number> -11.71150A </Number> </min>
    <max> <Number> 11.40800A </Number> </max>
  </Gnd Clamp>
</Gnd Clamp_section>
Elements vs Attributes

```
<Gnd_clamp_section>
  <Comment>
    <Comment_char> | </Comment_char>
    <Comment_line> voltage I(typ) I(min) I(max) </Comment_line>
  </Comment>
  <Gnd_clamp>
    <Point> voltage= -3.30000
      typ = -11.46380A
      min = -11.71150A
      max = 11.40800A
    </Point>
  </Gnd_clamp>
</Gnd_clamp_section>
```
What next?

• Standardize on the IBIS XML tags
• Standardize on the IBIS XML DTD/XSD
• Standardize on an IBIS Presentation XML
• Modify the Parser to generate XML
Issues

• XML is an evolving standard
• XSLT is complex
• DTD cannot capture the complete grammar
  - Need XSD (XML schema definition)
• Very few validating Parsers
Useful Links

• XML info
  - www.w3.org
  - www.xml.com
  - www.xslt.com

• EDA XML
  - http://www.oasis-open.org/cover/xmlAndEDA.html
  - http://www.zapthink.com/online/acronyms.htm#

• In House Info
  - http://www.eda.org/pub/ibis/xml/
  - http://www.eda.org/pub/summits/jun00/labonte.zip
Acknowledgments

• Bob Ross
  - For providing the sample IBIS models
• Pallavi Ray & Hriday Ranjan Daga
  - For doing all the hard work