XML: The Universal Publishing Format

SGML/XML Europe '98
Paris, 19 May 1998

Jon Bosak
Sun Microsystems

XML Basics

A note about this presentation

The slides presented at SGML/XML Europe '98 have been augmented here to include some comments that were made to the audience but did not appear in the slides themselves. Source files for the two "proof of concept" examples have also been provided.

Note that the online version assumes a display with a resolution of 1024 x 768. The slides should be viewable on screens with lower resolution, but they won't look as good.

What is XML?

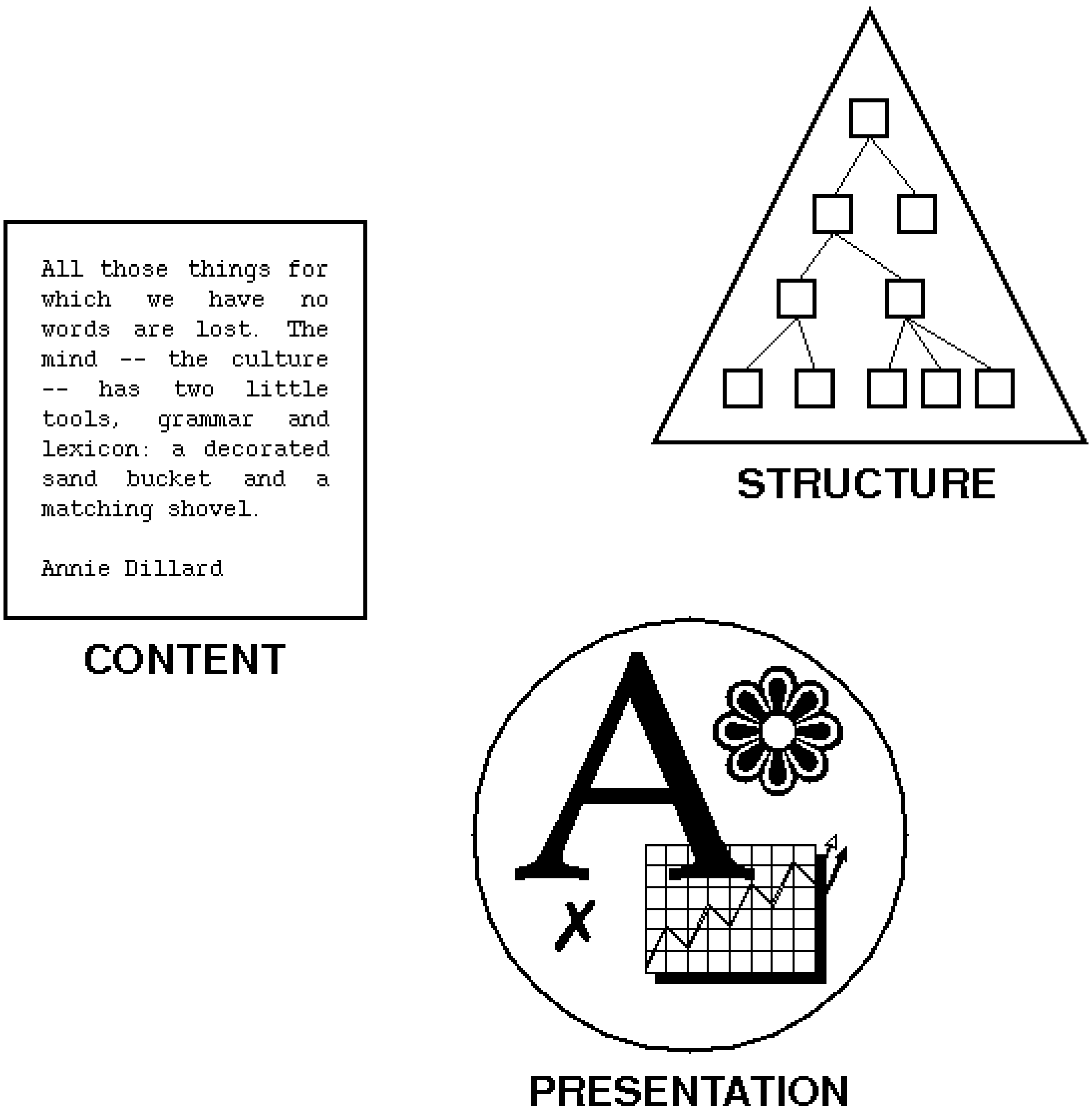
- Extensible Markup Language
- An activity of the World Wide Web Consortium (W3C) organized by Sun to put SGML on the World Wide Web
- Will create new data-centric Web applications
  - Database exchange
  - Distribution of processing to clients
  - Client-side manipulation of views into the data
  - Customization of information by intelligent agents
  - Management of document collections
- Will fundamentally change publishing on the web and then publishing in general

Who developed XML?

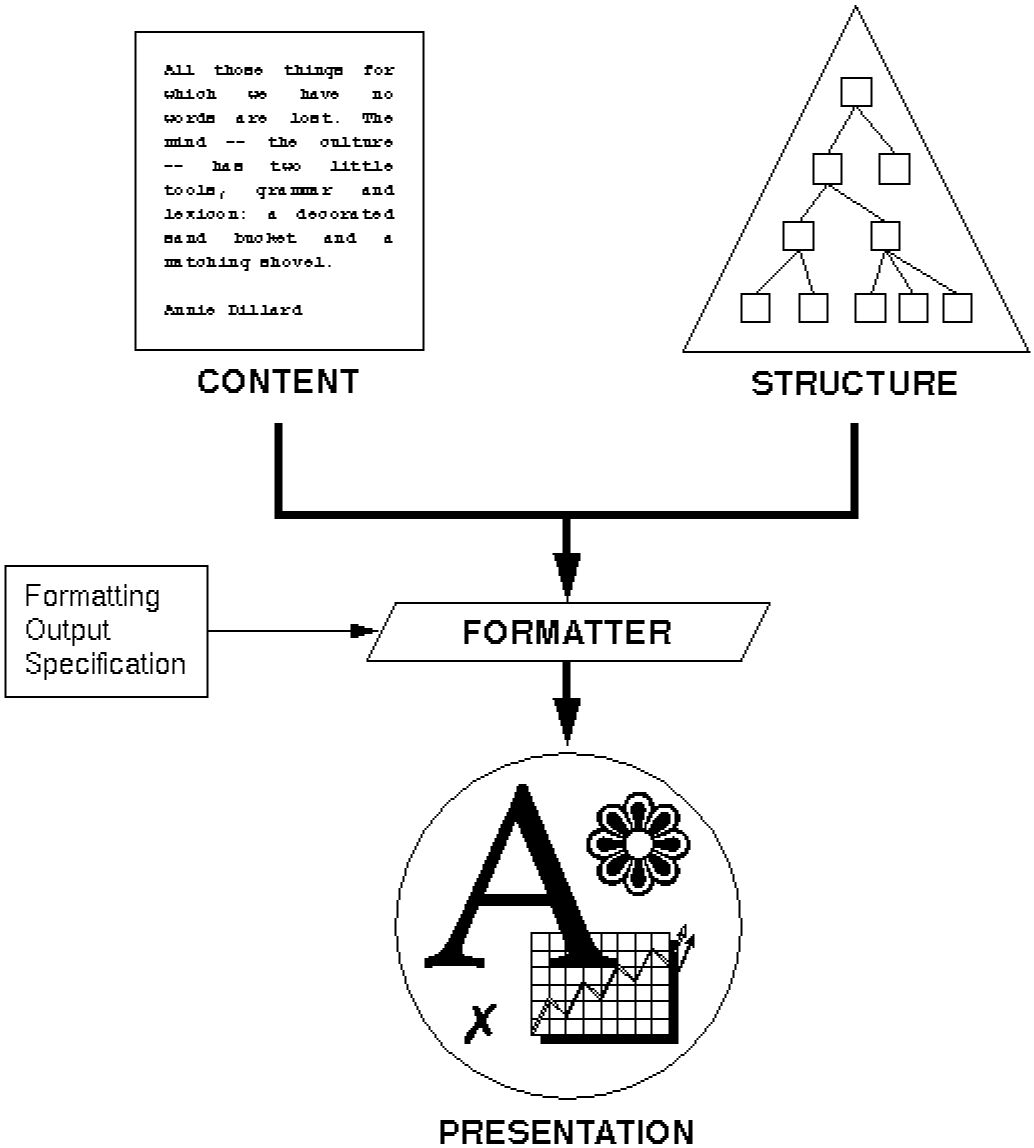
The XML specification was developed from 1996 through 1998 by a wide-ranging group of markup language experts from industry and academia.

- **Computer industry:** Sun Microsystems, Hewlett-Packard, Microsoft, Netscape, Adobe, Fuji Xerox
- **SGML vendors and system integrators:** ArborText, Inso, SoftQuad, Grif, Isogen, Texcel
- **Academic and research community:** Text Encoding Initiative (TEI), NCSA, James Clark
- **Early adopters:** DataChannel, Vignette
- **Recent additions (after XML 1.0):** IBM, Oracle, Omnimark

Document components



The SGML/XML concept



SGML and XML specify the content and structure of a document in a way that allows particular presentations to be generated as needed.

Proof of concept: this presentation

(These are links in the online version.)

- The XML source from which this presentation was produced
- The optional XML DTD used to validate the XML source
- The DSSSL style sheet for the HTML used in the online version
- The DSSSL style sheet for the RTF used in the printed version
- The Jade DSSSL engine used to produce both the HTML and RTF files
- A PostScript version of this presentation produced from Jade's RTF output

Why take this approach?

Separating content and structure from presentation and behavior makes possible

- Reusable information
- Media-independent publishing
- One-on-one marketing
- Intelligent downstream document processing
- Large-scale information management

The XML family

The XML family of languages is being created by adapting existing international publishing standards for use on the Web.

- **XML** (Extensible Markup Language): A subset of SGML (ISO 8879) designed for easy implementation
- **XLL** (Extensible Linking Language): A set of standard hypertext mechanisms based on HyTime (ISO/IEC 10744) and the Text Encoding Initiative (TEI)
- **XSL** (Extensible Stylesheet Language): A standard stylesheet language for structured information formed by subsetting DSSSL (ISO/IEC 10179), designing an alternative syntax, and incorporating key CSS concepts

XML itself

- A simplified subset of SGML (ISO 8879)
  - Very powerful
  - No limits on namespace or structural depth
  - Easy to implement
  - Small enough for Web browsers
- Internationalized from the beginning
  - Unicode for both content and markup (can mix languages)
  - XML tools **must** support both UTF-8 and UTF-16 and **can** support other encodings
- Not a language but a metalanguage
  - Designed to support the definition of an unlimited number of vertical-market languages for specific industries
  - "Write once, parse anywhere"

XML tag languages

XML allows industries to design specific tag languages to solve specific problems. Early examples:

- Chemical Markup Language (CML)
- Channel Definition Format (CDF)
- Open Financial Exchange (OFX)
- Handheld Device Markup Language (HDML)
- Resource Description Framework (RDF)
- Mathematics Markup Language (MathML)
- Precision Graphics Markup Language (PGML)

The advantage for implementors is that an unlimited number of domain-specific tag languages can all be processed by a single parser built into every Web browser.

XML in isolation

- "Syntax, not semantics"
  - Tags have no predefined meaning
  - XML by itself conveys only content and structure, not presentation or behavior
- There are important applications for XML alone: interprocess communication, object serialization, metadata, database exchange
- But associating **presentation or behavior** with XML requires **additional mechanisms**
  - Downloadable programs, applets, or scripts designed for a specific tag set (grammar)
  - Tag-sensitive components (e.g., Java beans)
  - Industry agreements on the processing of specific grammars (example of the concept: HTML)
  - Stylesheets (XSL or CSS)

Futures: XML for Data Exchange

The two XML domains

1. Data exchange
   - Data intended for consumption by machine
2. Publishing
   - Data intended for consumption by humans

What more needs to be done for data exchange?

- XML schemas
- Standard grammars (schemas and namespaces)
- Enhanced linking

XML schemas

- Enhancements to existing DTD functionality
  - Data typing
  - Inheritance
- The need for these features has been known for years
- The XML-Data submission is an interesting start but far from cooked yet
- Open questions:
  - Use instance syntax or extend existing DTD syntax?
  - Base on RDF?
- The XML WG has asked to have this item placed on its list of work so that these issues can be sorted out
- Doing a good job with schemas will take a while!

Standard grammars

- This is another old idea (standard DTDs)
- Now namespaces (vocabularies) need to be standardized, too
- Big question: are these standards controlled by user communities or by platform vendors?
- Need distributed registries beyond platform vendor control
    - Vertical industry consortia
    - Entrepreneurs
    - Neutral third parties (e.g., OASIS)

Extended linking

- Doesn't get much publicity but has important implications
- Allows new ways of associating information
- Promotes the creation of advanced information structures and site management
- Makes possible an industry devoted to knowledge management
- Will ultimately be as important as XML itself
- Keep your eye on the XLL work (XLink and XPointer)

Futures: The Larger Picture

Beyond a narrow interpretation of XML

1. XML is not just about exchanging data between machines
   - It's also about communication between humans
2. XML is not just about the web
   - It's about information in general
3. XML is not just about technology
   - It's also about the relationship between content creators and software vendors

Next step for publishing: XSL

Large-scale cross-platform Web publishing demands that XML deliver on the *display-oriented* promises:

- User-configurable views
- More powerful display-centric client-side applications
- Media-independent publishing
- In particular, printed and online deliverables from the same source
- Asian-language rendering support

XSL is intended to complete the internationalized media-independent publishing story.

Example: Database document

```
<?xml version="1.0"?>
<!DOCTYPE 書籍カタログ [
  <!ELEMENT 書籍カタログ (書籍)+ >
  <!ELEMENT 書籍 (書名,著者,出版社,(定価 | 在庫数)) >
  <!ATTLIST 書籍     xml:lang CDATA #REQUIRED >
  <!ELEMENT 書名     (#PCDATA) >
  <!ELEMENT 著者     (#PCDATA) >
  <!ELEMENT 出版社   (#PCDATA) >
  <!ELEMENT 定価     (#PCDATA) >
]>

<書籍カタログ>
  <書籍 xml:lang="JP">
    <書名>XML入門</書名>
    <著者>村田、門馬、荒井</著者>
    <出版社>日本経済新聞社</出版社>
    <定価>2800</定価>
  </書籍>
  <書籍 xml:lang="EN">
    <書名>Developing SGML DTDs</書名>
    <著者>E. Maler and J. el Andaloussi</著者>
    <出版社>Prentice Hall</出版社>
    <定価>50</定価>
  </書籍>
</書籍カタログ>
```

With stylesheet for Japanese reader

XML入門

著者　村田、門馬、荒井

出版社　日本経済新聞社

定価　2800円

Developing SGML DTDs

著者　E. Maler and J. el Andaloussi

出版社　Prentice Hall

定価　50ドル

With stylesheet for English reader

XML入門

Author: 村田、門馬、荒井

Publisher: 日本経済新聞社

Price: ¥2800

Developing SGML DTDs

Author: E. Maler and J. el Andaloussi

Publisher: Prentice Hall

Price: $50

Proof of concept: source files for the example

(These are links in the online version.)

- The UTF-16 XML source from which the different versions were produced
- The UTF-16 DSSSL style sheet used to produce the version for the reader of Japanese
- The UTF-16 DSSSL style sheet used to produce the version for the reader of English
- The Jade DSSSL engine used to produce RTF files from the source and the style sheets
- The UTF-16 RTF file for the reader of Japanese (font association done in Word 7)
- The UTF-16 RTF file for the reader of English (font association done in Word 7)

Lessons from the example

- The catalog example shows that the distinction between data exchange and publishing is ultimately an artificial one (the same source would also be used to create the printed catalog)
- The rendition in each case occurs **on the web client**
- The database owner can publish **a single data stream** to the entire world
- Consider the alternative:
    - Generation of a different HTML output stream for **every possible** user and target platform
    - Much greater load on the server
    - No user autonomy

XML inherits the social agenda of SGML

The social agenda of SGML has always been about **creator ownership of content.**

- Freedom from proprietary data formats
- Vendor neutrality
- Platform neutrality
- Language neutrality

Good news for users

XML is a big open-standards victory for users.

- Freely extensible
  - No tag name limitations
  - No language limitations
- Human-readable
  - Can maintain data using basic text tools like sed and awk
  - Perl is being optimized for XML support
- Open standard
  - In theory, XML users can't be held hostage to vendor control
- Easy to implement
  - There will be many powerful, cheap, off-the-shelf commercial XML tools
  - There is already an ever-growing set of free XML tools (almost all of them Java-based)

Economic/political implications of XML+XSL

The combination of XML and XSL can replace all existing word-processing and publishing formats.

- A single format for both print and online publishing
- A single format across different products
- A single format for all languages

What does this mean?

- Users no longer tied to a proprietary format
- A change in the relationship between software vendors and customers
- An end to domination of the market by a few big companies
- An end to domination of the market by a few big countries

What can go wrong?

The complete implementation of XML and XSL means an end to control of users through proprietary formats.

Companies that have built their business models on proprietary formats can be expected to resist this.

The most obvious ways to subvert the user-empowerment agenda of XML are:

- Control of standard schemas and namespaces
- Incomplete XSL implementation

Ways to subvert the social agenda

1. Control of standard schemas and namespaces by platform vendors
   - This is vendor dependence in another guise
   - If you propose a standard, your implementation will always be 6-12 months ahead of everyone else's
   - Platform vendors are a particular problem because they can optimize the platform for a particular grammar
2. Incomplete implementation of XSL
   - Crippled XSL limited to tagset transformations (XML to XML)
   - This implies document delivery via a single tag set (HTML)
   - In other words, a return to the status quo!
   - Solves the data exchange problem but ignores the larger publishing problem and keeps users dependent on vendors

The HTML output trap

Much can be accomplished with on-the-fly generation of HTML (see http://docs.sun.com) but it is **not** the answer for publishing in the general case. A real XML+XSL solution would provide:

- Media-independent publishing
- Independence from particular HTML implementations
- Delivery of the same source to multiple users
- Fewer round trips to the server
- Quality formatting
    - The idea that online formatting requirements are a subset of print formatting requirements is a short-term historical accident
    - In the long run, online requirements are a **superset** of print requirements

The embedded script trap

Embedding scripts or applets in documents is not a viable long-term approach, either.

- Strategies based on embedding scripts in source files don't scale; management becomes very difficult
- Documents with embedded scripts cannot be edited (so no real interoperability between editing environments)
- Therefore the final form of documents must **always** be generated rather than authored

How can we keep the XML dream alive?

- Beware of strategies that limit XML to the role of middleware
- Beware of attempts by platform vendors to dictate standard schemas (DTDs) and namespaces
- Beware of attempts by browser vendors to introduce nonstandard extensions to the XML family of languages
- Beware of attempts by browser vendors to introduce ad-hoc tags into HTML under the name of XML
- Insist on real XSL support: the ability to render formatting objects, not just HTML tags
- Support platform-independent tools vendors
- Support the only organization dedicated to interoperable document standards: OASIS (http://www.oasis-open.org)

The payoff

- Interoperability of both content and style
- Freedom from vendor control of our data
- Creator control of markup depth
- User control of views into the data
- A level playing field for independent software developers
- True international publishing across all media