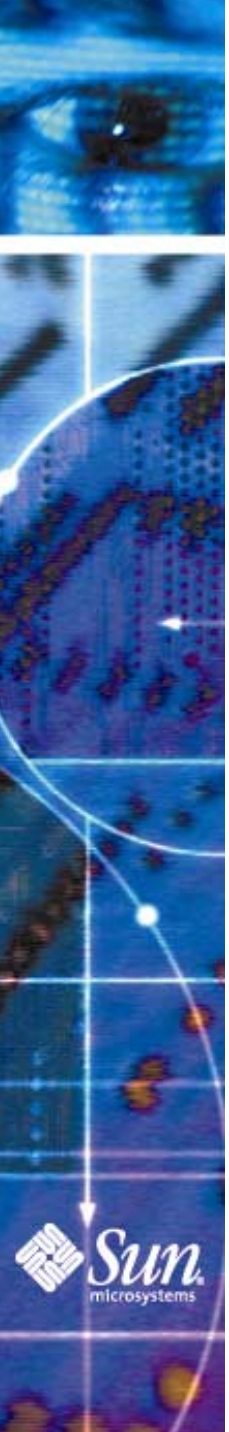
Lecture 1 (1 hour and half)

XML Overview

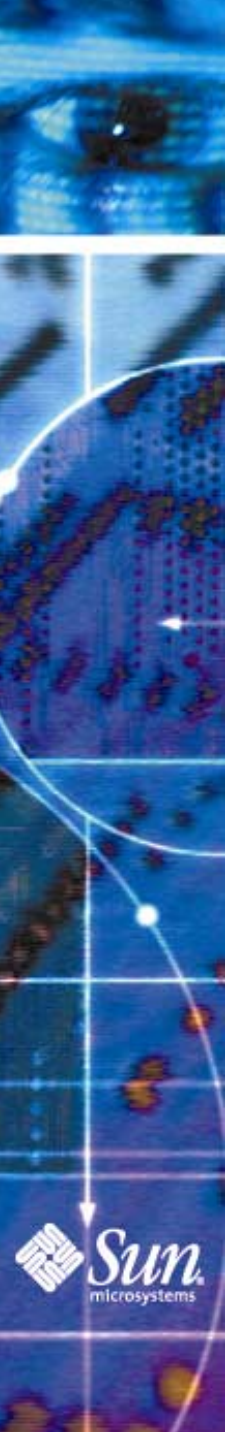


Topics

- What is XML?
- Why XML?
  - XML feautures and value-propositions
- Where does XML come from?
- Where is XML being used today?
- What do you do with XML document?
- What is going on standards front?

Objectives

- Understand fundamental concept and features of XML
- Get some perspective on how XML is being used in real-world



What is XML?

- eXtensible Markup Language
- is Meta-Markup language
- is Not just a markup language

The Data Problem

- Fundamental “issues”: How do I represent my “application” data?
  - Performance (speed/time)
  - Persistence(short/long lived)
  - Mutability
  - Composition
  - Security (encryption/identity)

Data Problem (Cont.)

- Open Information Management:
  - Interpretation
  - Presentation
  - Interoperation
  - Portability
  - Interrogation

Markup Language

- Used to markup data
  - Methodology for encoding data with some information
- Examples
  - Yellow highlighter on a string of text as emphasizer
  - Comma between pieces of data as separator

Markup Language

- Two important aspects
  - A standard for “valid markup”
    - HTML - tags
  - A standard for “what markup means”
    - HTML - tags “communicate” layout and formatting information
- Typical markup languages define a set of tags each of which has some associated meaning

Markup Language Usage Examples

- Word processing documents
- Data on the network
- Database
- Multi-media contents
- HTML

HTML

- The most popular markup language
- Defines a set of tags
- Designed for presentation for data
- HTML documents are processed by HTML processing application (Browser)

Strengths of HTML

- Easy to implement and author
  - Small number of tags
  - Simple relationship between tags
  - Syntax-checking is very forgiving
  - Limited number of formats possible
  - Viewers can be small and simple
- HTML trades power for ease of use

Weaknesses of HTML

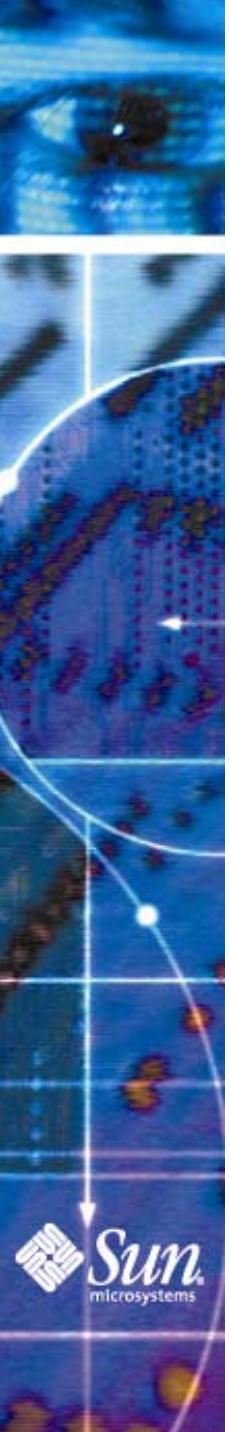
- Fixed set of tags
  - Not user extensible
    - Dependency to “markup language” definition process
  - Dependency to vendors
    - Vendor proprietary tags
    - Implementation not in sync
      - Netscape browser vs. IE
- Predefined semantics for each tag
- Predefined data structure

Weaknesses of HTML

- No formal validation
- Does not support semantic search
- Based on solely on appearance (rendering) NOT on content
- Formatting too simple
  - Limited control
- Won’t do complex documents
- Poor support for print and other media

So HTML is Not suited for handling

- Large and complex data
- Data that must be used in different ways
- Data with long life-cycle
- Data intended to drive scripts and Java applets



So XML is Born!

- Motivations
  - HTML would not work for publishing in general case
  - Web application would require a method of encoding data that could drive arbitrarily complex distributed processes
  - Without XML, HTML would be replaced by a more powerful binary and proprietary format

XML Design Goals

- XML shall be straightforwardly usable over the Internet.
- XML shall support a wide variety of applications.
- XML shall be compatible with SGML.
- It shall be easy to write programs which process XML documents.

XML Design Goals

- The number of optional features in XML is to be kept to the absolute minimum, ideally zero.
- XML documents should be human-legible and reasonably clear.
- The XML design should be prepared quickly.

XML Design Goals

- The design of XML shall be formal and concise.
- XML documents shall be easy to create.
- Terseness in XML markup is of minimal importance.

Key Features of XML

- Extensibility
- Media and Presentation independence
  - Separation of contents from presentation
- Structure
- Validation

Extensibility

- XML is Meta-markup language
- You define your own markup languages (tags) for your own problem domain
- Infinite number of tags can be defined
  - Need for domain-specific standards
  - XSLT

Extensibility

- Tags can be more than formatting
- Tags can be anything
  - Semantic data representation
  - Business rules
    - ebXML
  - Data relationship
    - EJB 2.0 Containter Managed Persistence
  - Formatting
    - XSL
  - Anything you want

Extensibility

- Many domain-specific markup languages
  - Portable data within domain-specific industry
  - Portable across the various domain
    - Healthcare and Insurance
    - Chemical and Medicine

Media (Presentation) Independence

- Clear separation between contents and presentation
- Contents of data
  - What the data is
  - Is represented by XML document
- Presentation of data
  - What the data looks like
  - Can be specified by stylesheet

Media (Presentation) Independence

- Same data can be presented to different communication medium
  - HTML browser
  - Voice device
- Same data can be presented to different format
  - Cell phone as WML
  - Desktop as HTML
- Via stylesheet

Media (Presentation) Independence

- Stylesheet
  - Instruction of how to present XML data
  - CSS
    - Tailored for HTML browser
  - XSL
    - XML based
    - General purpose
    - Work with XSLT

Separation of Contents from Presentation

- Searching and retrieving data is easy and efficient
  - Tags give search'able information
- Many applications use the same data in different ways
  - Employee data can be used by
    - Payroll application
    - Facilities application
    - Human resource application

Separation of Contents from Presentation

- Enables portability of data
- Portable over time and space

XSLT Transformation

- Example (XML -> HTML)

  XML data:

  <destination>sang.shin@sun.com</destination>

  XSLT stylesheet can say:

  - Start a new line.
  - Convert “destination” XML tag to “To:” HTML tag.
  - Display "To:" in bold, followed by a space.
  - Display your email address.

  Which produces:

  To: sang.shin@sun.com

Structure: HTML vs. XML

HTML (Automatic Presentation of Data)

```
<b> John Doe 1234  </b> // Display in bold
```

XML (Automatic Interpretation of Data)

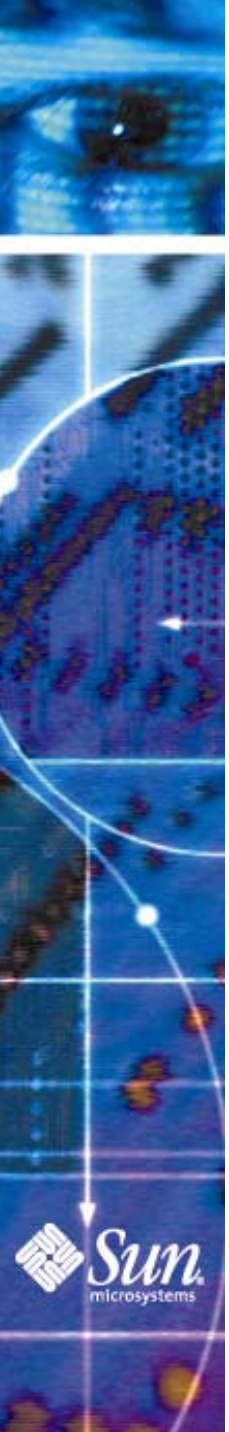
```
<Employee>
  <Name>
      <firstName> John </firstName>
      <lastName>  Doe </lastName>
  </Name>
  <EmployeeID> 1234 </EmployeeID>
 </Employee>
```

Structure

- Relationship
  - Employee is made of Name and employ-ID
  - Name is made of firstName and LastName
- Hierarchical (Tree-form)
  - Faster to access
  - Easier to rearrange
  - Can be any number of depth

Structure

- Enables to build large and complex data
- Portability of relationship and hierarchical structure



Validation

- XML data is “constrained” by a Rule (or Syntax)
  - Employee data has to have Name and Employee ID elements
  - Name has to have both firstName and LastName elements

```
<Employee>
   <Name>
        <firstName>John</firstName>
        <lastName>Doe</lastName>
   </Name>
   <EmployeeID>1234</EmployeeID>
</Employee>
```

XML Schema

- Defines Syntax
  - Structure
  - Vocabulary
- XML document + XML schema work together
  - XML document alone has to be “well-formed”
  - XML schema checks “validity”

XML Schema

- DTD (Data Type Definition) is the most popular XML schema language for now
- DTD is pretty weak schema language
  - You can't define "range check"
- There are other emerging XML schema languages
  - W3C XML schema
  - Relax

Misc. Features of XML

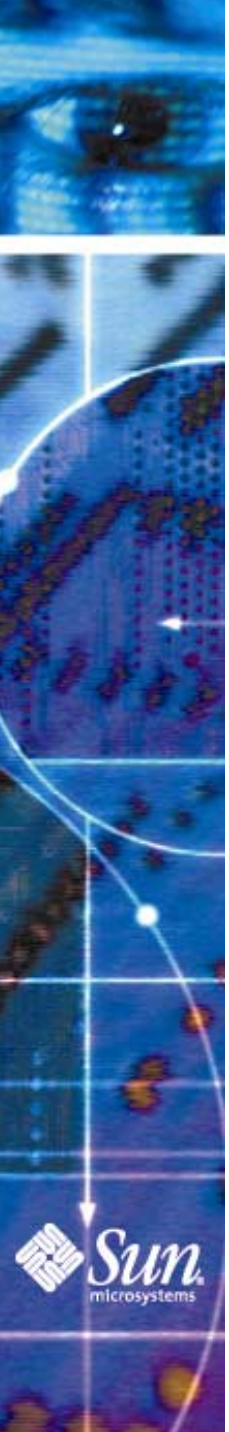
- Semantics of data
- Plain Text
- Easily Processed
- Inline usability
- Linkability
- Internationalized

Semantics of Data

- Meaning of data
- XML tags “indirectly” specifies the semantical meaning
  - does <firstName> really mean “first name”?
- Potential for divergence
  - Industry collaboration to agree upon the semantical meanings of tags
  - Need for transformation (XSLT)

Plain Text

- Can use any text-editing tool
- Easier for humans to read
  - Configuration information



Easily Processed

- Set of Well-formed rules
- Validity checking
- Ready-to-use tools
  - parsers (and validators)
  - transformers
  - browsers
  - class generators
  - IDE

Inline Usability

- Can integrate data from multiple sources
  - Can be displayed or processed as a single document
- Modularization without using Linking
- Example
  - A book made of independently written chapters
  - Same Copyright text in many books

Linkability

- Much more powerful and flexible linking capabilities than HTML's
- Bi-directional
- Mutli-directional
- W3C Xlink and Xpointer specifications

Internationalized

- XML is Unicode-based
  - You can mix languages
- Both markup and content
- XML tools must support both UTF-8 and UTF-16 encodings
- Critical for world-wide adoption of XML as universal data representation

XML History

- Emerged as a way to overcome the shortcomings of its two predecessors
  - SGML
  - HTML

XML History

- SGML
  - Powerful and extensible
    - Good for cataloging and indexing data
  - Too complex and expensive
- HTML
  - Simple and widely supported
  - Not extensible
- XML
  - “Lean and mean SGML”

XML Evolution

- Discussion got started in 1996 by Jon Bosak of Sun
- W3C-approved XML 1.0 spec in 1998
- XML-related standard activities are moving in dizzying speed
  - Horizontal
  - Vertical
  - E-Commerce

Where Does XML Get Used?

- Simple and complex data representation
- Integration of heterogeneous applications
- Portable data representation
- Displaying and publishing
- Archiving
- Data manipulation
- Business logic representation

Data representation

- XML encodes the data for a program to process
- Readable by humans
- Process'able by computers
- Complex relationship can be represented
- Internationalized
- Many 3rd-party tools
  - Editing, Syntax checking

Data representation

- Examples
  - Configuration files
    - EJB deployment descriptor
  - “make” files (Apache ANT project)

Integration of heterogeneous applications

- Typically used with Messaging system
- XML message is minimum contract for communication
  - Loosely-coupled communication
- Enables easy EAI (Enterprise Application Integration)
  - Payroll, Finance, Products
- E-Commerce
  - Supplier, distributor, manufacturer, retail

Portable data representation

- Non-proprietary
  - Application independent
  - Object-model independent
  - Language independent
  - Platform independent
  - Communication protocol independent
  - Communication media independent
- Used for means of “information exchange”

Portable data representation

- Examples
  - Purchase order, Invoice
  - Business transactional semantics
  - Patient record
  - Mathematical formula
  - Musical notation
  - Manufacturing process

Displaying and publishing

- Common data for different presentations
- Separation of contents from presentation
- Examples
  - Web information presented to different client types
  - Information rendered to different medium

Archiving

- XML data can be data storage format of choice
  - Better version management
  - Easy to process
  - Availability of read-to-use tools
    - Report generation
- Standard XML query language
- Examples
  - XML-based data repository

Data manipulation

- Many tools available for XML data manipulation
- Examples
  - Data from relational data converted to XML for easy manipulation

Business logic representation

- XML tag can be anything
- It could be business logic or action
- Basis for XML-RPC movement
- Examples
  - “Withdraw $40 from my savings account” can be represented as XML

Developer Activities on XML

- Creating XML document
  - Mostly by text-editor or WISWIG tools
  - Programmatically
- Sending and Receiving XML document
  - Over any kind of transports
    - HTTP, SMTP, FTP, ...
  - Through programming APIs
    - JMS API, JAXM API
    - Socket APIs

Sun microsystems

Developer Activities on XML

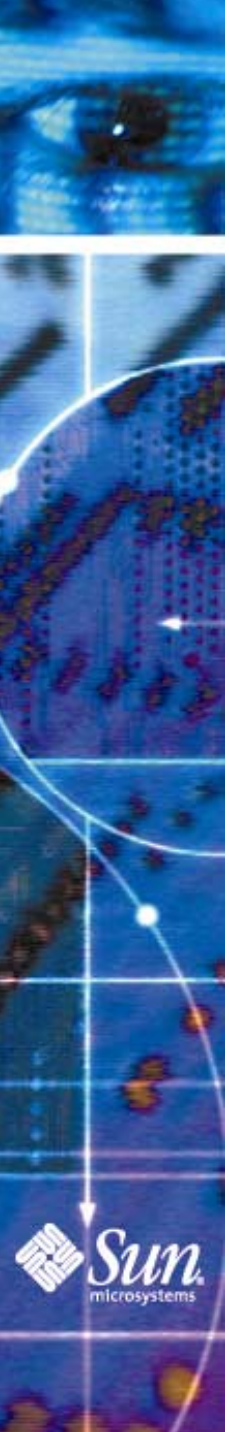
- Parsing XML document
  - Convert XML document into programming objects
- Manipulating programming objects
  - Application specific way
  - Examples
    - Display
    - Save them in database
    - Create new XML document

Developer Activities on XML

- “XML document” can be
  - URI
    - File
    - URL
  - InputStream
  - SAX Input source
  - DOM tree

Java™ Technology + XML: Symbiotic Relationship

- It's a "Match made in Heaven"
  - Java enables Portable Code
  - XML enables Portable Data
- XML tools and programs are mostly written in the Java programming language
- Better API support for Java platform than any other language



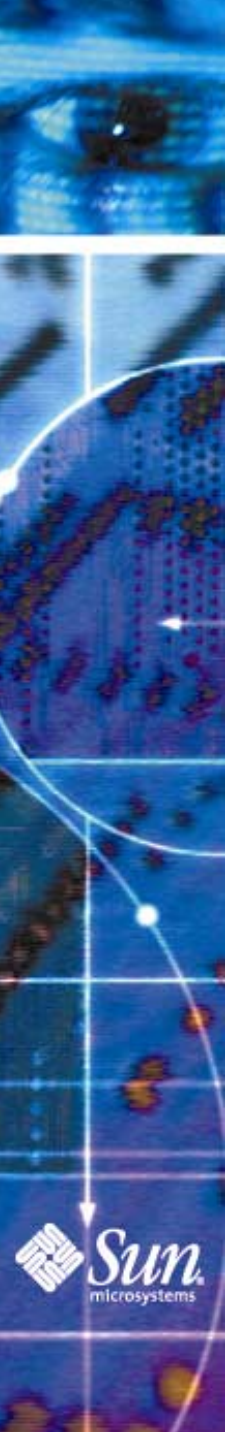
Standardization Activities

- XML core standards
  - Through Standard organisations
  - W3C, OASIS, UN/CEFACT
- XML domain-specific standards
  - Through domain-specific standard organizations
- Java technology-based APIs for XML
  - Through JCP (Java Community Process)
- E-commerce standards

Sun microsystems

XML Standards

- XML, DTD
- XSL, XSLT, XPath
- DOM, SAX
- W3C XML Schema
- Namespaces
- XLink, XPointer
- XHTML
- XQL

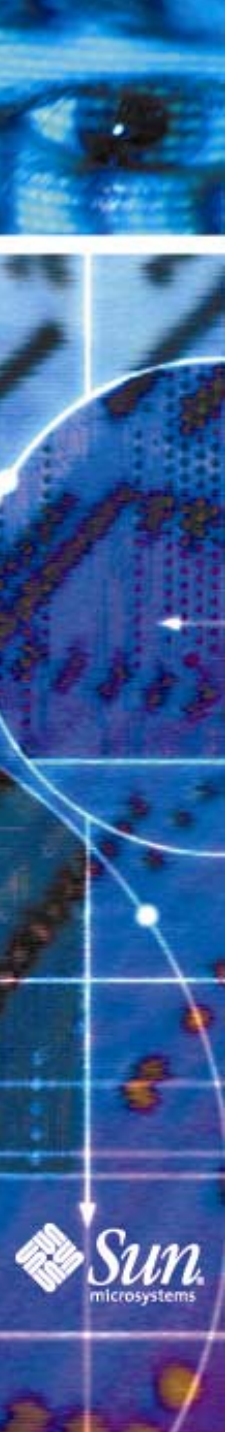


Domain-specific XML Standards

- Chemical - CML
- 2D Graphics - SVG
- Math - MathML
- Music - MusicML
- Travel -OTA
- Many more ...
  - http://xml.org/xmlorg_registry/index.shtml

Core Java APIs for XML

- JAXP: Parsing and Transforming
- JAXB: High-level XML programming
- JAXM: Messaging
- JAXR: Registry APIs
- JDOM: Java-optimized Parsing



E-Commerce Standards

- ebXML
- UDDI (Universal Description, Discovery and Integration)
- SOAP (Simple Open Access Protocol)
- W3C XP (XML Protocol)
- WSDL (Web Services Definition Lang.)
- S2ML (Security Services ML)
- XAML (Transaction Authority ML)

Summary

- XML is the next Big thing
- Is it being hype'd? Maybe yes, then again, it deserves the hype.

Key Future Predictions in summer in 1999 by Jon Bosak

- XML will be the basis for future Web standards
- XML will become universal format for data exchange in heterogenous environments
- XML will become the basis for international publishing
- XML and XSL will replace all existing word processing and desktop publishing formats

Homework of Lecture 1

- Download AMAYA browser and try to display a couple of MathML data (See the class website for details)
  - http://www.w3.org/Amaya/
- No need for submission