Java & XML for Public Access Television

Advanced Internet Issues

David Moisan

July 9th, 1999

Java

Introduction—What is Java

- Java is an object-oriented programming language and environment designed specifically for the Internet
- Java is cross-platform, running on many different machines by using a Java Virtual Machine that translates Java code to that of the machine it runs on.
- Java has networking capabilities.

What is Object-oriented Programming?

- Traditional languages like C, BASIC and Pascal are procedural
  - They describe how to do something with data, but the data & program are separate things; functions are called on data but data is separate from code

Java is Object-oriented

- But Java is *object-oriented*
  - Every piece of data in a program is an *object* that has *methods* to manipulate itself
  - Objects are arranged by *classes* of related objects that represent data of some type

Benefits of Object-Oriented Languages

- Keep data and algorithms together
- Hide details from other modules
- Confine bugs to modules—no global variables!

Java & Javascript: What's the difference?

- Javascript has many similarities to Java but was developed separately from Java
- Javascript is meant for lightweight programming inside web pages
- Javascript syntax is simpler and less restrictive than Java

Java Syntax

- Java's syntax is very similar to C.
- Standard control structures include
  - `if (i>array.length) {…some code…}`
    `…else {…more code…}`
  - `while (vector.moreelements()) {…code…}`
  - `do {} while (condition)`
  - `for (int i=1;i<10,i++) {…some code…}`
- Java has primitive variables like C: ints, floats, arrays, etc.

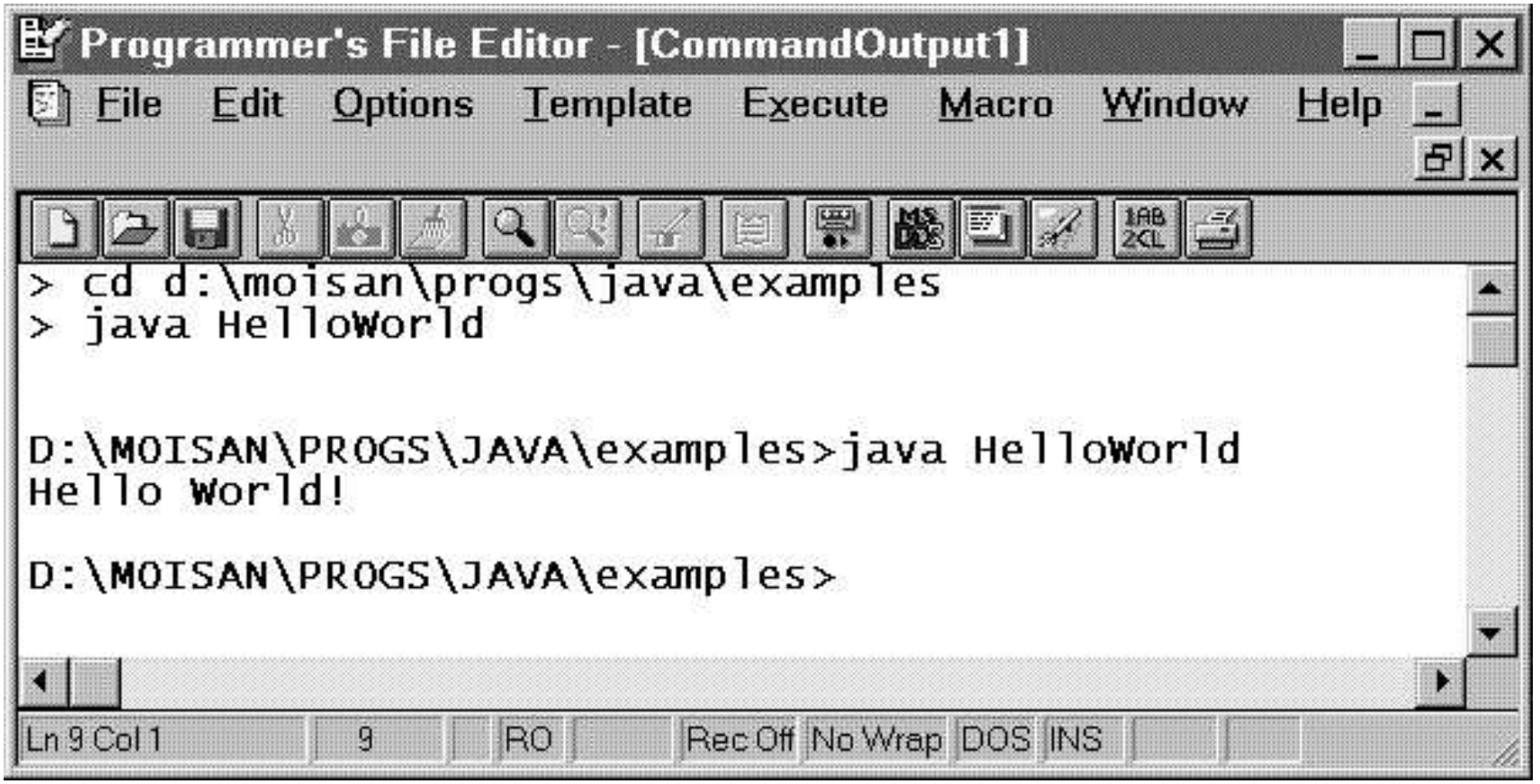
Differences between Java and C/C++

- Java is purely object oriented, unlike C++
- Java does not allow direct manipulation of pointers, reducing a major source of bugs
- Java reclaims unused memory through garbage collection, so memory allocation bugs as found in C and C++ are all but unheard of.

“HelloWorld” application

```
public class HelloWorld {
// Classic "Hello World!" program
// D. Moisan 6/20/1999
public static void main (String args[]) {
   System.out.println("Hello World!");
   System.exit(0);
   }
}
```

Running “HelloWorld”



A simple object

```
class Car {

String Make;
String Model;
String Engine;

int Wheels;
int Seats;

// This is a simple class Car with field
// variables
}
```

Fields

```
class Car {

String Make;
String Model;
String Engine;

int Wheels;
int Seats;

// Fields are variables or other objects
// in a class that hold data
}
```

Methods

```
// Methods work just like functions—they
// are code that does things
class Car {
…
public drive(String direction, speed) {
// drive this car somewhere

this.moveCar(direction, speed);
System.out.println(“Moving :” +_direction
+ “at “+ speed);
}
}
```

Creating Objects: Constructors

- Constructors are used to create and initialize objects

```
…
Car MyCar = new Car(“Chevy”, “Corvette”);
…
```

Defining Constructors

- Constructors are defined just like methods are, but with a special method of the same name as the class

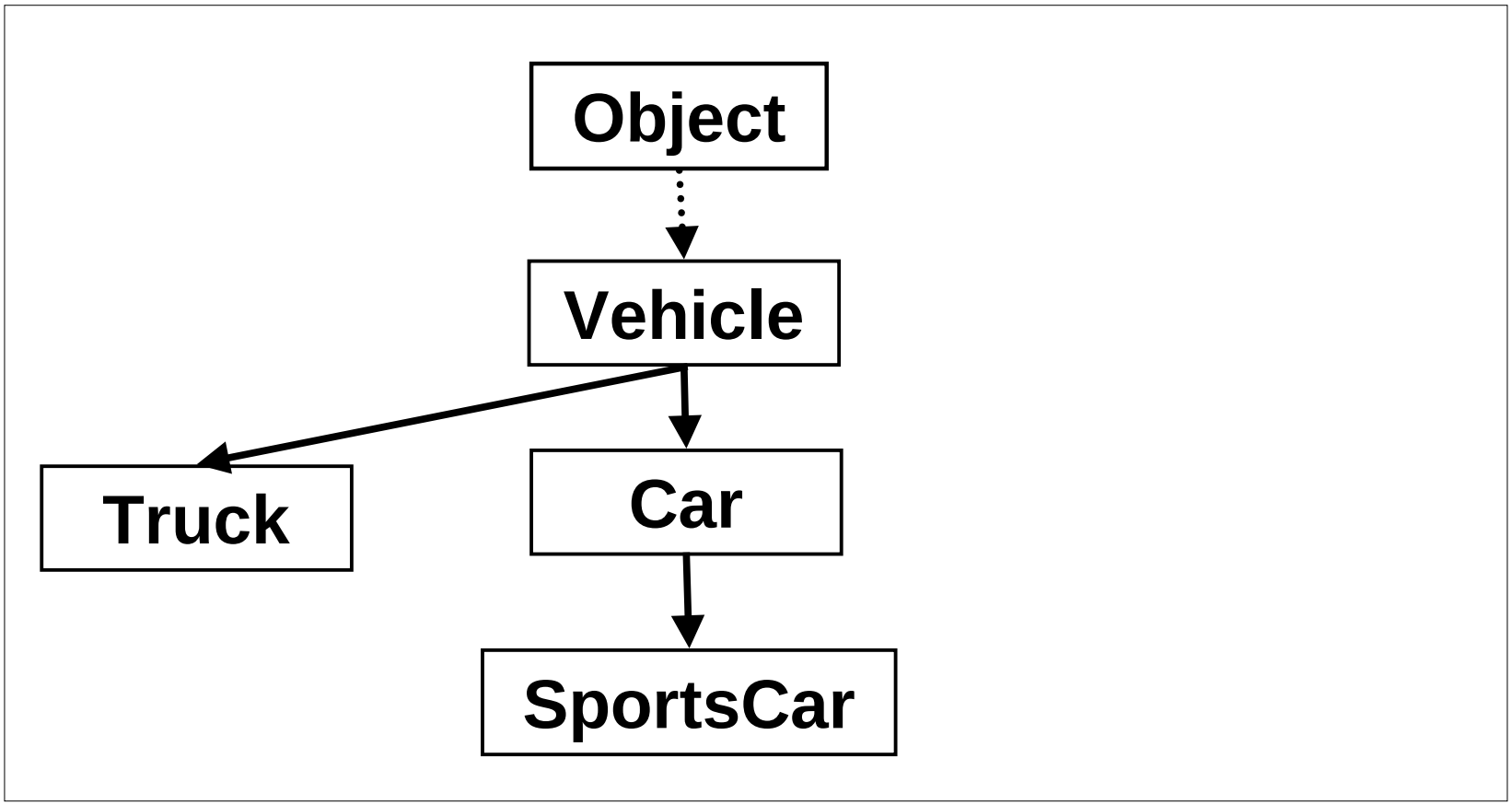
```
class Car {
…
public Car(String make, String Model) {
this.Make = make;
this.Model = model;
// this keyword refers to the current
  object
}
```

Subclasses & Inheritance

```
class SportsCar extends Car {
int CubicInchDisplacement = 350;
}

class CompactCar extends Car {
int CrushLoad = 5;
this.Make = “Kia”;
```

Inheritance Diagram



The Class Library

- Most of Java's power and capabilities come from the class library; similar to the standard library in C or C++, it defines the standard functions in Java
- Some standard class packages:
  - java.lang: Language & system features, always included
  - java.io: File and stream I/O
  - java.net: Networking
  - java.applet: Applets
  - java.awt: The Abstract Window Toolkit GUI

Applets—Java in the Browser

- An applet is a piece of program code designed to run from within a web browser
- Applets can draw graphics and interact with the user
- Applets are restricted in what they can do on the client machine—they “play in a sandbox”
- Applets can be digitally signed for greater privileges, such as in an intranet.

“HelloWorldWeb” Applet

```
import java.applet.*;
import java.awt.*;

public class HelloWorldApplet extends
  Applet {
  // This method displays the applet.
  // The Graphics class is how you do all
  drawing in Java.
  public void paint(Graphics g) {
    g.drawString("Hello World", 25, 50);
  }
}
```

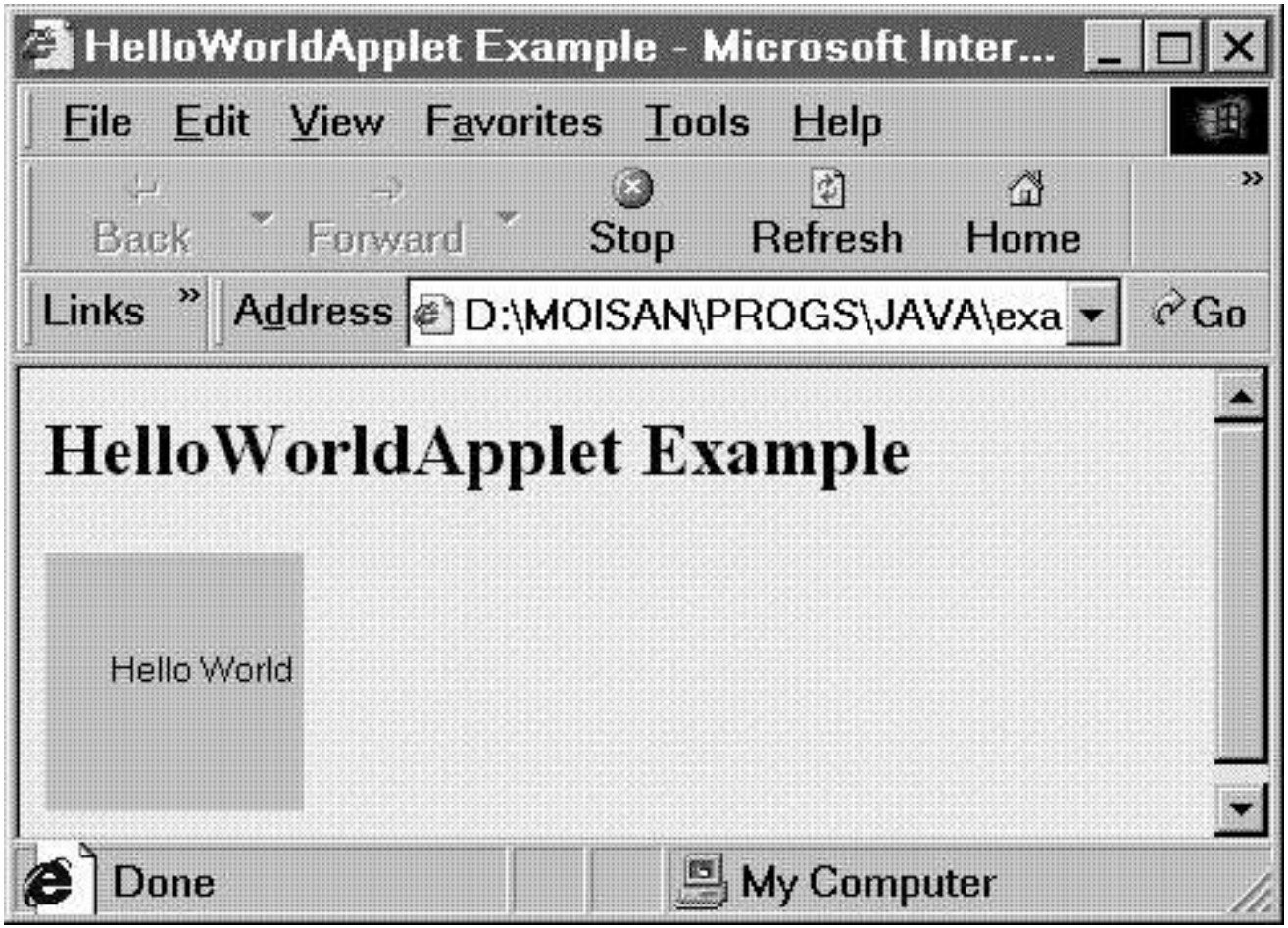
“HelloWorldApplet” HTML

◆ Applets are invoked in the web browser by the **<APPLET>** tag

```
…
<h1>HelloWorldApplet Example</h1>


<applet height="100" width="100"
  code="HelloWorldApplet.class">
Sorry, a Java-enabled browser is needed
  to view this applet
</applet>
…
```

“Hello,World” on the Browser



Differences between Applets and Applications

- Applications start running with a call to **`main()`**, the body of the program and run outside the browser
- Applets define four methods used by the browser to control applet execution, **`init()`**, **`start()`**, **`stop()`** and **`destroy()`**, and also **`paint()`** for graphics.

Other Language Features

Some other language features you should know about:

- Java Beans--code modules that can be “plugged in” to applications.
- Abstract Window Toolkit (AWT)--Java’s older GUI interface
- Swing--Java’s new GUI interface standard to replace AWT

The Various Versions of Java

- There are three versions of Sun's Java:
  - Java 1.0.2: Still seen on older browsers
  - Java 1.1: The most common version
  - Java 1.2: The newest version of Java, also known as Java2.
- Sun also provides extra API's to extend Java, most notably the Multimedia API which supports streaming media, and Swing, Java's new GUI.

Even More Java Choices

- In addition, other companies have released their own Java VM's:
  - IBM Alphaworks (a big player)
  - Microsoft
  - JVM's exist for Linux and Mac too.
- Many other companies offer Java tools and products

Developing Java: Writing & Compiling Code

- The compiler and other tools that come with the Java Development Kit (JDK) are command-line oriented.
- But there are several free graphical development packages, most notably IBM's *Visual Age for Java Basic Edition*
- One of my favorite tools for Windows is *Programmer's File Editor* (PFE).

Developing Java: Where to get tools

- Sun's website (**http://java.sun.com/**) is the canonical resource.
- Gamelan (**http://www.gamelan.com/**) has been the Java directory since the beginning
- IBM's Alphaworks site (**http://www.alphaworks.ibm.com/**) is a treasure trove of Java and XML applications, all free, many with source code!

Recommendations & More Information

- Which version of Java should you use?
  - Java 1.0.2: Obsolete; use only to support older browsers
  - Java 1.2 (Java2): “Bleeding-edge Java”; not all browsers support it yet
  - Java 1.1: Widespread support; your best choice
- Links to all Java resources in this presentation can be found at:

http://www1.shore.net/~dmoisan/comp/javaresources.html

XML

What is XML?

- XML—Extensible Markup Language—is an open, standard, format for transporting data.
- It resembles HTML, with a difference: You define your own tags to describe your data
- XML is not so much a language, but a means of defining our own markup languages for specific applications.
- It’s similar to SGML but has been simplified for the Web.

What can be represented with XML?

- Textual information, such as scripts
- Simple, flat-file, databases
- Any information that needs to be transferred across applications and platforms.

XML in the Access Center—What can it do for us?

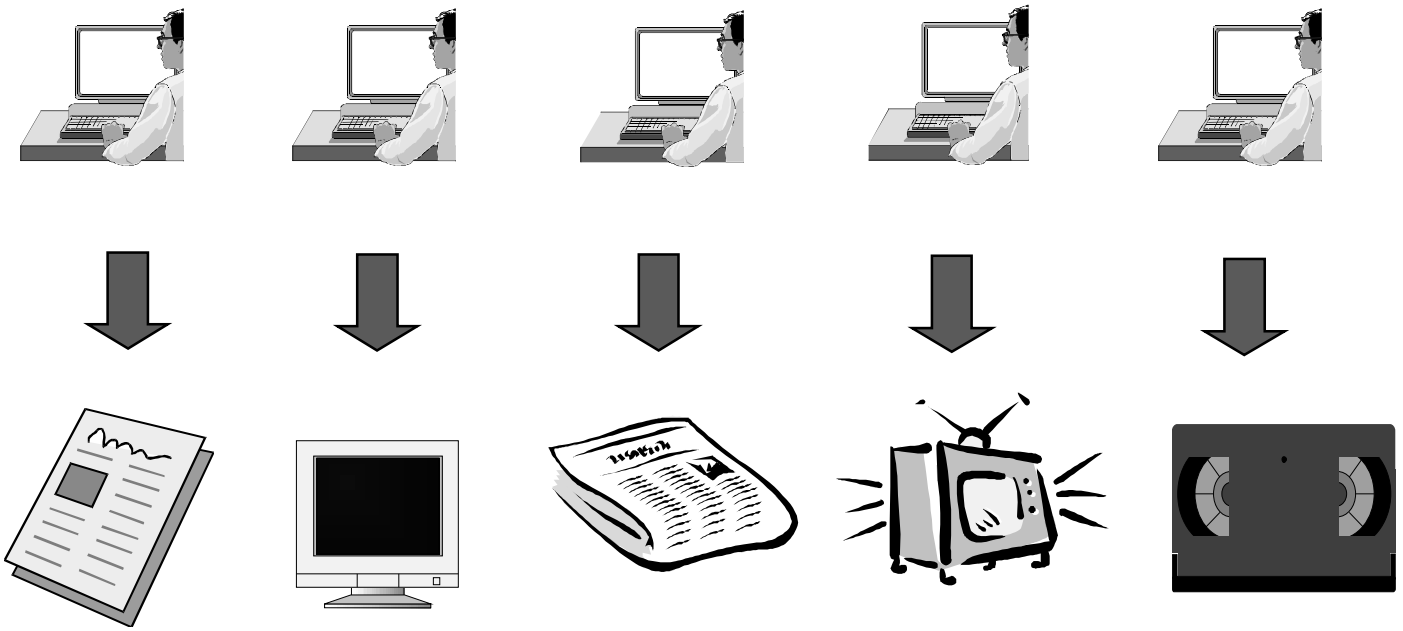
- Many access centers have to work with a limited staff & budget
- Because we are a niche market and relatively ill-funded, there is little available that we can afford.
- Many of our activities need to be tied together and tightly integrated.
- XML can help!

A typical problem: Scheduling

- At Salem Access TV, we generate weekly schedules that go on our website, to our board members, to the newspaper, to our video bulletin board and to our automation controller.
- Our program coordinator has to reenter the week's schedule as often as five times or more!

A Scheduling Problem

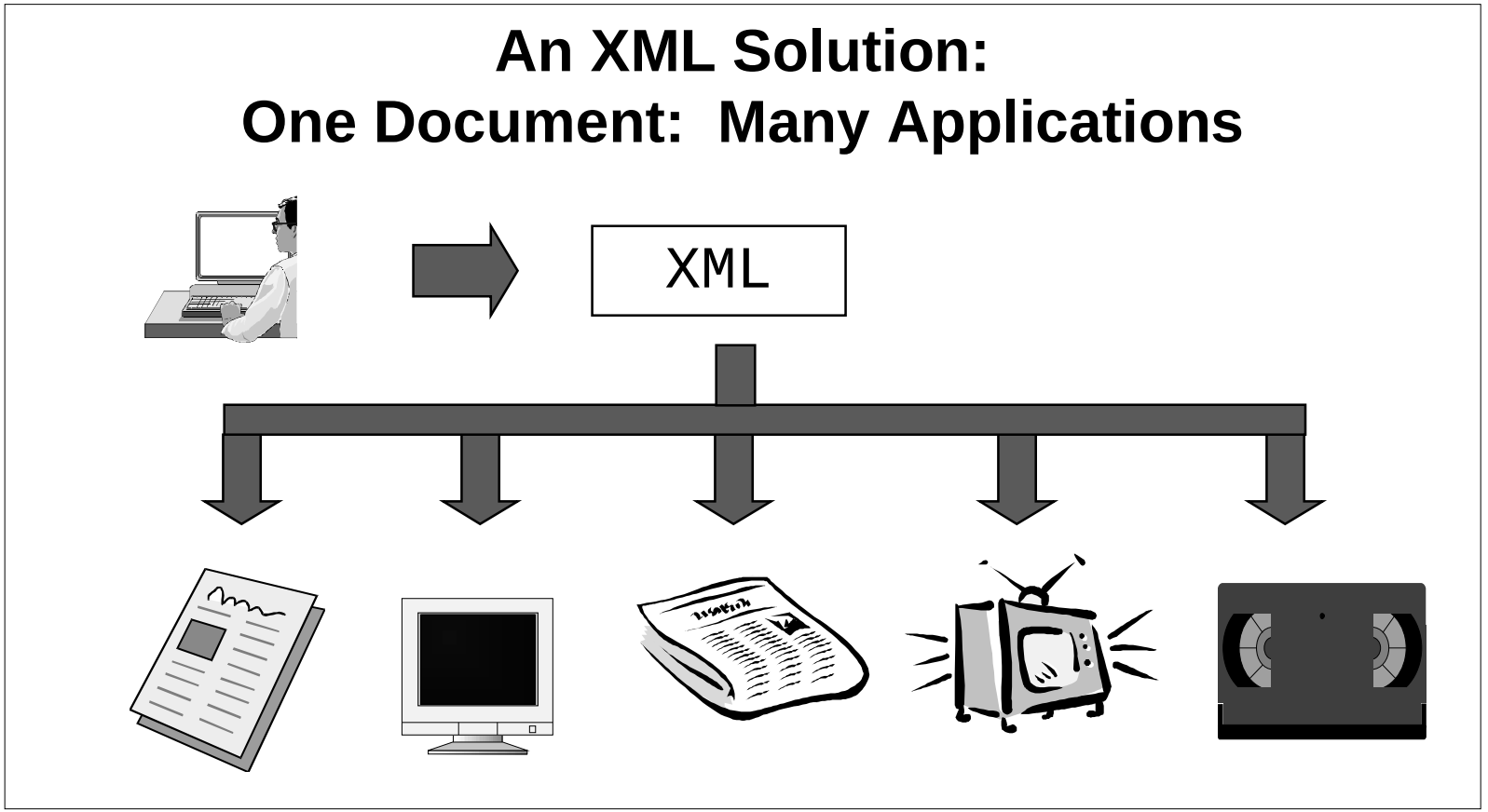
The Old Way:
Entering Schedules One at a Time



Is XML a solution?

- One solution to this problem is to define a common format for schedules that can be used across SATV in a variety of different applications
- The schedule is entered once as an XML file…
- …and is rendered by software and stylesheets to provide as many different “views” as needed.

An XML Solution



Why use XML—Why don't we use HTML?

- You can't determine what the data is in HTML—HTML has structural information but little or no semantics.
- HTML is overburdened and hard to extend in its present form
- HTML was never designed to be used by automation such as robots and agents

HTML and XML compared

Compare these two snippets of markup

Which one is easier to interpret?

First, the HTML version:

```
<br><h4>Monday, May 24th</h4>
3:00 PM: Salem High
  Basketball<strong>vs. Lynn
  Classical</strong>
```

HTML and XML compared

The XML version:

```
<DATE>Monday, May 24th, 1999</DATE>
<PROGRAMSLOT>
<TIME>3:00 PM</TIME>
<TITLE>Salem High Basketball</TITLE>
<DESCRIPTION>vs. Lynn Classical
</DESCRIPTION>
</PROGRAMSLOT>
```

A First Document: TVSCHEDULE.XML

```
<?xml version="1.0"?>

<TVSCHEDULE NAME="Salem Access Television">
<CHANNEL CHAN="3">

<BANNER>Channel 3 Program Schedule</BANNER>

<DAY>
<DATE>Monday, May 24th</DATE>
<PROGRAMSLOT>
<TIME>3:00 PM</TIME>
<TITLE>Salem High Basketball</TITLE>
<DESCRIPTION>vs. Lynn Classical</DESCRIPTION>
</PROGRAMSLOT>...
```

XML requirements

- HTML's requirements for markup are rather loose, allowing end tags to be omitted (such as **</p>**)
- To make XML easier to parse, markup standards have to be strictly enforced
- Namely, all XML documents must be well-formed or valid
- Also, unlike HTML, whitespace is preserved!

Well-Formed XML

- All XML documents must start with this declaration:

```
<?xml version="1.0"?>
```

- All tags must be closed:

```
<DESCRIPTION>SHS vs.
  Brookline</DESCRIPTION>
```

- Empty tags are written like this:

```
<br />
```

- Tags are case-sensitive!

Valid XML

- XML documents can also be valid
- To be valid, XML documents must have a document type definition (DTD), and they must conform to the rules in that DTD.
- Valid XML has a declaration like this:

```
<!DOCTYPE TVSCHEDULE SYSTEM "tvschedule.dtd”>
```

- Valid XML documents must also be well-formed

Document Type Definitions (DTD)'s

- DTD's define the content model of an XML file, namely, which tags are valid in a given context
- If you've ever validated HTML for your website, you probably know about DTD's
- DTD writing can be difficult, but there are online DTD generators that work from your XML files that will give you a useable DTD.

TVSCHEDULE.DTD

```
<!ELEMENT TVSCHEDULE (CHANNEL+)>
<!ATTLIST TVSCHEDULE
   NAME CDATA #REQUIRED>
<!ELEMENT CHANNEL (BANNER, DAY+)>
<!ATTLIST CHANNEL
   CHAN CDATA #REQUIRED>
<!ELEMENT BANNER (#PCDATA)>
<!ELEMENT
   DAY ((DATE, HOLIDAY) | (DATE, PROGRAMSLOT+))+>
<!ELEMENT HOLIDAY (#PCDATA)>
<!ELEMENT DATE (#PCDATA)>
```

TVSCHEDULE.DTD Continued

```
<!ELEMENT PROGRAMSLOT (TIME, TITLE,
  DESCRIPTION?)>
<!ATTLIST PROGRAMSLOT
      VTR  CDATA #IMPLIED>
<!ELEMENT TIME (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ATTLIST TITLE RATING CDATA #IMPLIED
                LANGUAGE CDATA #IMPLIED
                LIVE CDATA #IMPLIED
                NEW CDATA #IMPLIED>
<!ELEMENT DESCRIPTION (#PCDATA)>
```

Special Characters: Entities

- As with HTML, there are special characters that can't be used in text:

  <, **>**, **‘**, **“**, **&**

- These characters are represented by entities
  - < and > are represented as **`&lt`**; and **`>`**, just as in HTML
  - &, “ and ‘ are represented by **`&amp`**;, **`&quot`**; and **`&apos`**;
- XML lets you define your own entities, such as **`&satv`**; for “Salem Access Television”

```
<!ENTITY satv “Salem Access Television”>
```

Attributes

- You know attributes from HTML: They modify information in a tag, such as:

```
<IMG SRC="ball.gif" LENGTH="100"
  WIDTH="100">
```

- XML allows attributes as well, like

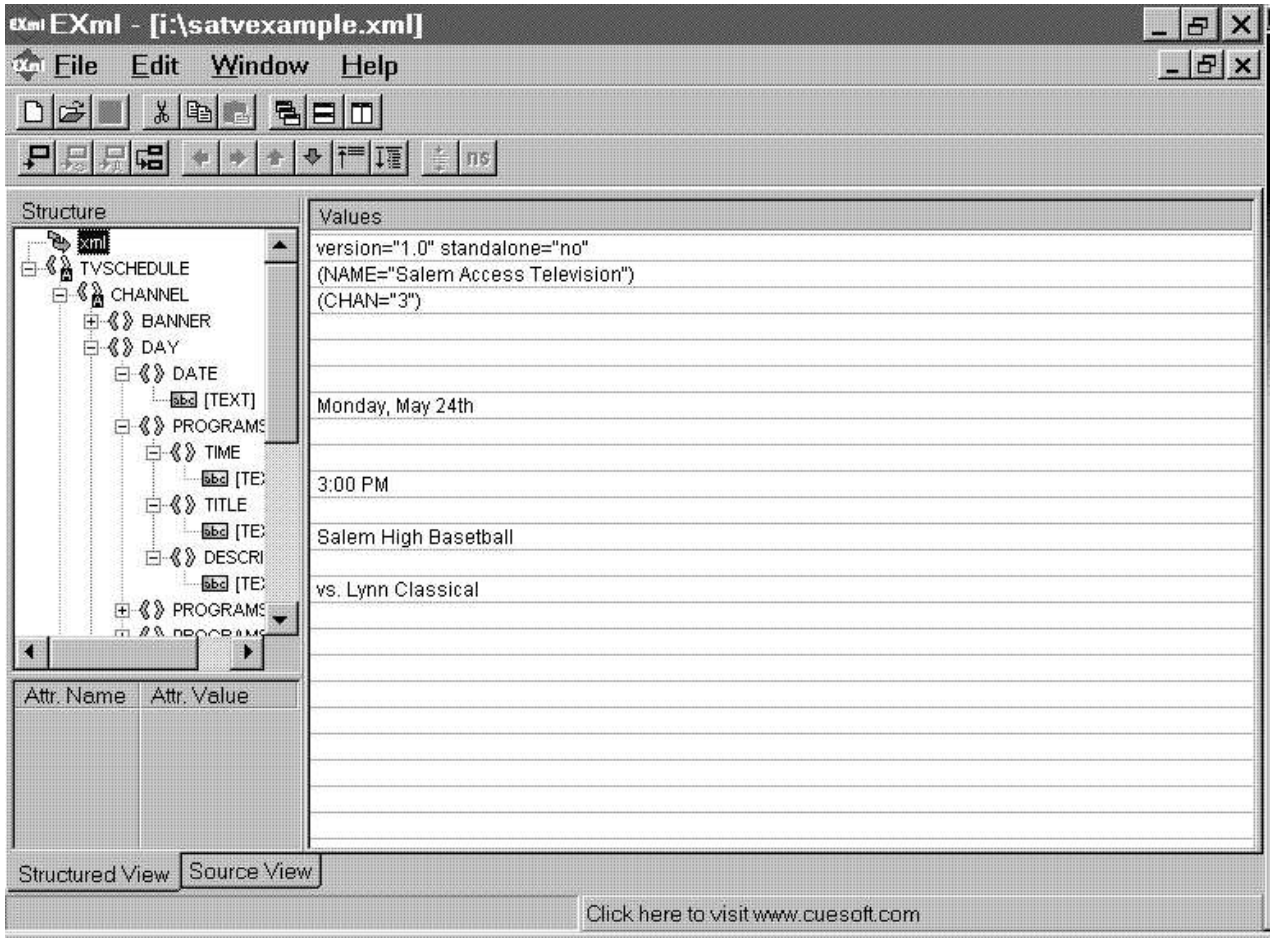
```
<TVSCHEDULE NAME="Salem Access
  Television">
```

- In XML, all attributes must be in quotes.

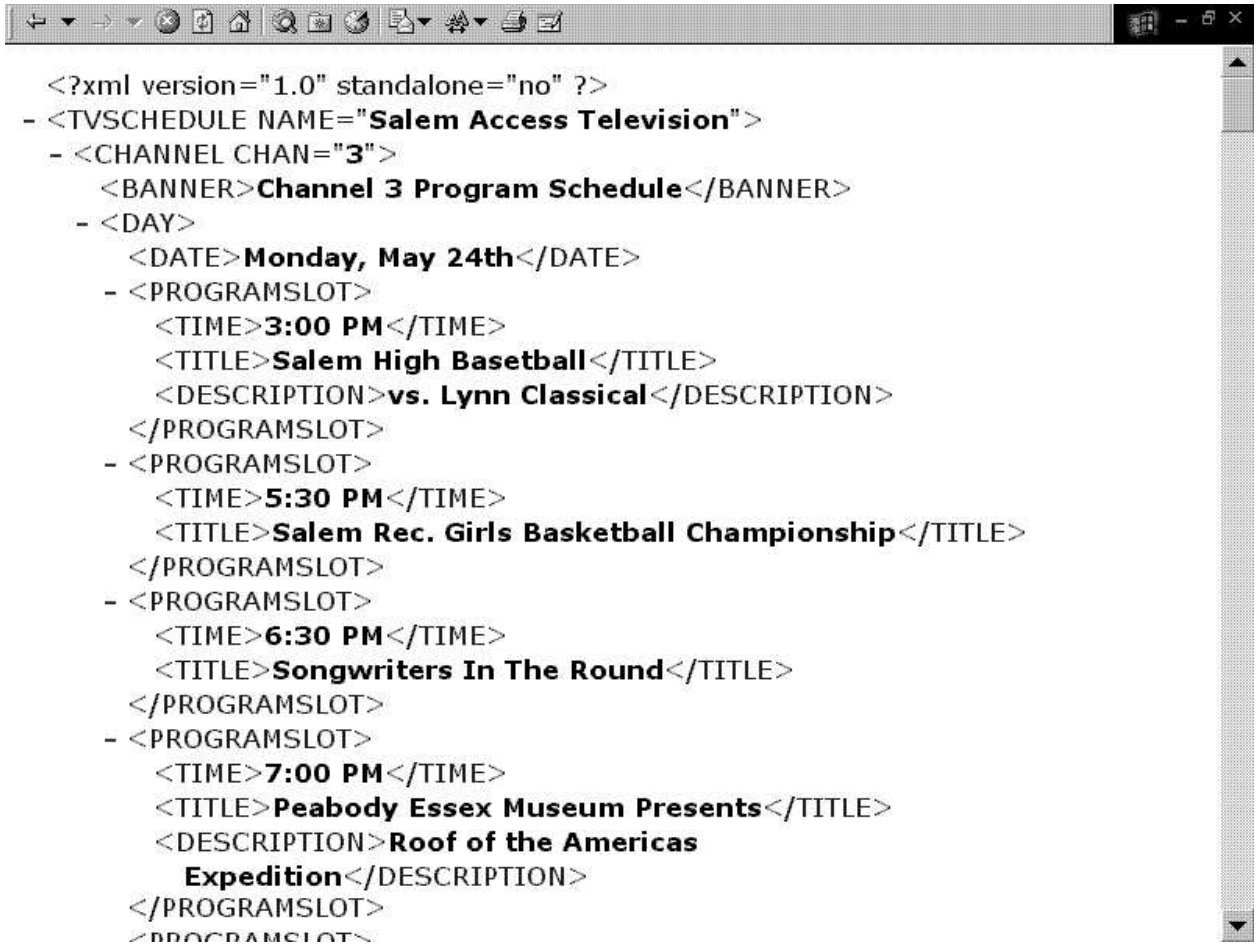
Editing XML

- Editors you may hear about include *EXML*, *XMLE*, *XML Spy* and *psgml* for Emacs.
- Most XML editors now available are primitive and cumbersome to use.
- You can, of course, use Notepad or any plain text editor for XML just as with HTML
- You most likely will use a front end to convert your application’s data to XML

Editing XML Example: EXML



Viewing XML in Internet Explorer 5



```
<?xml version="1.0" standalone="no" ?>
- <TVSCHEDULE NAME="Salem Access Television">
  - <CHANNEL CHAN="3">
      <BANNER>Channel 3 Program Schedule</BANNER>
    - <DAY>
        <DATE>Monday, May 24th</DATE>
      - <PROGRAMSLOT>
          <TIME>3:00 PM</TIME>
          <TITLE>Salem High Basetball</TITLE>
          <DESCRIPTION>vs. Lynn Classical</DESCRIPTION>
        </PROGRAMSLOT>
      - <PROGRAMSLOT>
          <TIME>5:30 PM</TIME>
          <TITLE>Salem Rec. Girls Basketball Championship</TITLE>
        </PROGRAMSLOT>
      - <PROGRAMSLOT>
          <TIME>6:30 PM</TIME>
          <TITLE>Songwriters In The Round</TITLE>
        </PROGRAMSLOT>
      - <PROGRAMSLOT>
          <TIME>7:00 PM</TIME>
          <TITLE>Peabody Essex Museum Presents</TITLE>
          <DESCRIPTION>Roof of the Americas
            Expedition</DESCRIPTION>
        </PROGRAMSLOT>
```

XSL—A Transformation and Styling language

- XSL is a transformation language that works by pattern matching
- A pattern can be an element name, or wildcard expression
- When patterns are matched, tags and text are output according to markup.

TVSCHEDULEHTML.XSL

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="DAY/PROGRAMSLOT">
<!—Output time and program title —>
<xsl:value-of select="TIME"/>: <xsl:value-of select="TITLE"/>
<!— Include optional DESCRIPTION and RATING items —>
<xsl:if test="DESCRIPTION">
<em><strong><xsl:value-of
   select="DESCRIPTION"/></strong></em></xsl:if>
<br />
<xsl:if test="TITLE/@RATING">
<em>Rating: <xsl:value-of select="TITLE/@RATING"/></em><br />
</xsl:if>
</xsl:template>
```

How XSL works

- The template markup matches PROGRAMSLOT tags that occur inside DAY tags

```
<xsl:template match="DAY/PROGRAMSLOT">
```

- Value-of tags return the contents of a tag, in this case, the NAME attribute inside TVSCHEDULE

```
<xsl:value-of select="/TVSCHEDULE/@NAME"/>
```

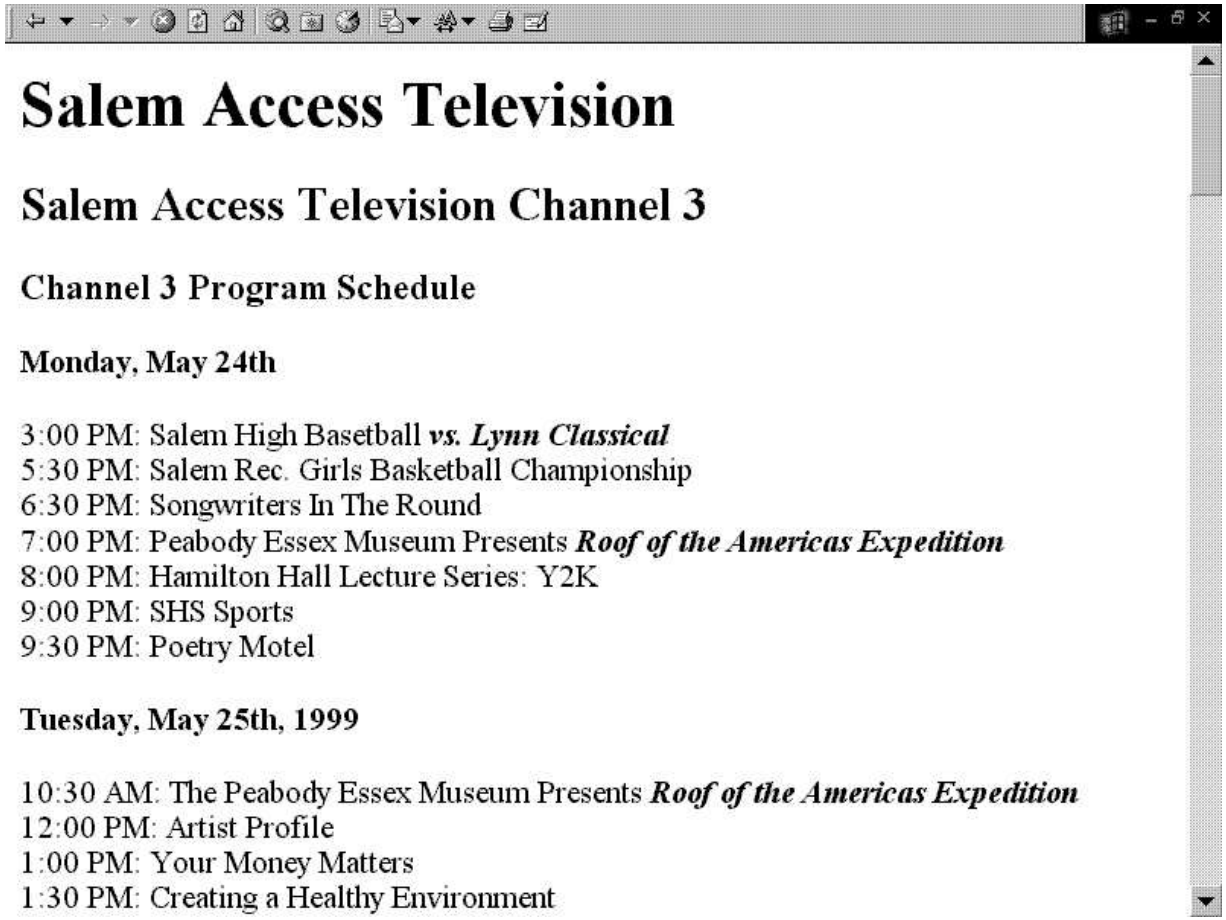
- The If tag renders its contents if a condition is met:

```
<xsl:if test="DESCRIPTION">
```

- XML documents link to their XSL stylesheets with:

```
<?xml-stylesheet type="text/xsl"
href="tvschedulehtml.xsl"?>
```

TVSCHEDULE styled with XSL



Salem Access Television

Salem Access Television Channel 3

Channel 3 Program Schedule

Monday, May 24th

3:00 PM: Salem High Basetball ***vs. Lynn Classical***
5:30 PM: Salem Rec. Girls Basketball Championship
6:30 PM: Songwriters In The Round
7:00 PM: Peabody Essex Museum Presents ***Roof of the Americas Expedition***
8:00 PM: Hamilton Hall Lecture Series: Y2K
9:00 PM: SHS Sports
9:30 PM: Poetry Motel

Tuesday, May 25th, 1999

10:30 AM: The Peabody Essex Museum Presents ***Roof of the Americas Expedition***
12:00 PM: Artist Profile
1:00 PM: Your Money Matters
1:30 PM: Creating a Healthy Environment

Some XSL Processors:

- *XT* — James Clark's XSL processor, in Java
- *LotusXSL* — Java-based XSL processor, with source
- All of these processors will run on Windows, Mac & Linux

Things to keep in mind about XSL

- XSL is a moving target—there have been numerous changes since it was first proposed two years ago and it's not done yet.
- IE 5 supports the December '98 XSL draft but not the current April '99 draft
- You may need two different XSL stylesheets for IE5 and any other processor, until a service pack is released

XHTML: Your first taste of XML

- XHTML 1.0 is a version of HTML 4.0 rewritten in terms of XML instead of SGML, as with present HTML
- Works just like HTML
- Some syntax has been changed to make it XML-compatible.

Why use XHTML?

- XHTML is readable with all current browsers (with the exception of some older Mac browsers)
- XHTML is made for extensibility in mind
- XHTML can take advantage of the many tools now available for XML

How do I use XHTML?

- Your code should be clean HTML. The messier it is, the harder it will be to convert.
- Closing tags—like **`</p>`** and **`</li>`** are mandatory.
- Tags are now in lower-case!
- Unlike HTML, whitespace is important.
- The conversion tool *Tidy* from the W3C can automatically convert your pages to XHTML and fix any style problems

XML For Programmers

- XML development software exists for many major languages, including Perl, Python, C++, Visual Basic and others.
- But almost all development in XML first starts with Java, with more XML tools available in this language than any other.

XML Parsers

- A parser is a program that will read a file and extract information that can be understood by humans, or other software
- For XML, there are two different kinds of parsers: SAX parsers and DOM parsers
- Each does the same thing, but both work differently.
- Both SAX and DOM have become de facto standards

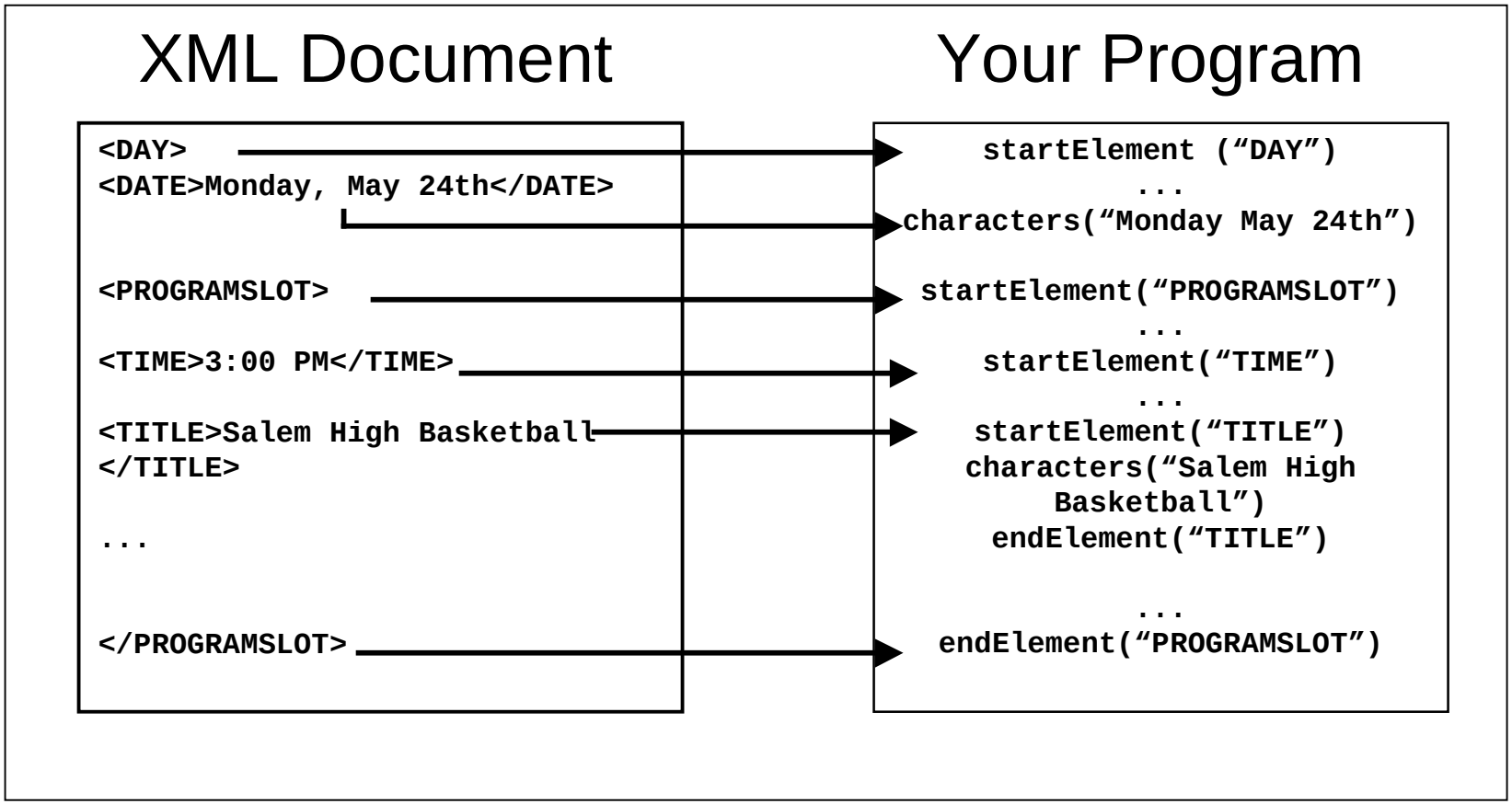
SAX: Event-driven XML parser API

- SAX: Simple API for XML
- Designed by David Megginson of Microstar
- SAX is implemented by *Ælfred*, *XML4J*, and *DCXJP.*
- SAX is designed to be compact and easily implemented.

How SAX works

- SAX is event-driven
- SAX reads your input XML document and sends events to your application that correspond to opening elements, text, whitespace and closing elements, among others

SAX Diagram


XML Document
Your Program
<DAY>
<DATE>Monday, May 24th</DATE>
<PROGRAMSLOT>
<TIME>3:00 PM</TIME>
<TITLE>Salem High Basketball
</TITLE>
...
</PROGRAMSLOT>
startElement (“DAY”)
...
characters(“Monday May 24th”)
startElement(“PROGRAMSLOT”)
...
startElement(“TIME”)
...
startElement(“TITLE”)
characters(“Salem High Basketball”)
endElement(“TITLE”)
...
endElement(“PROGRAMSLOT”)

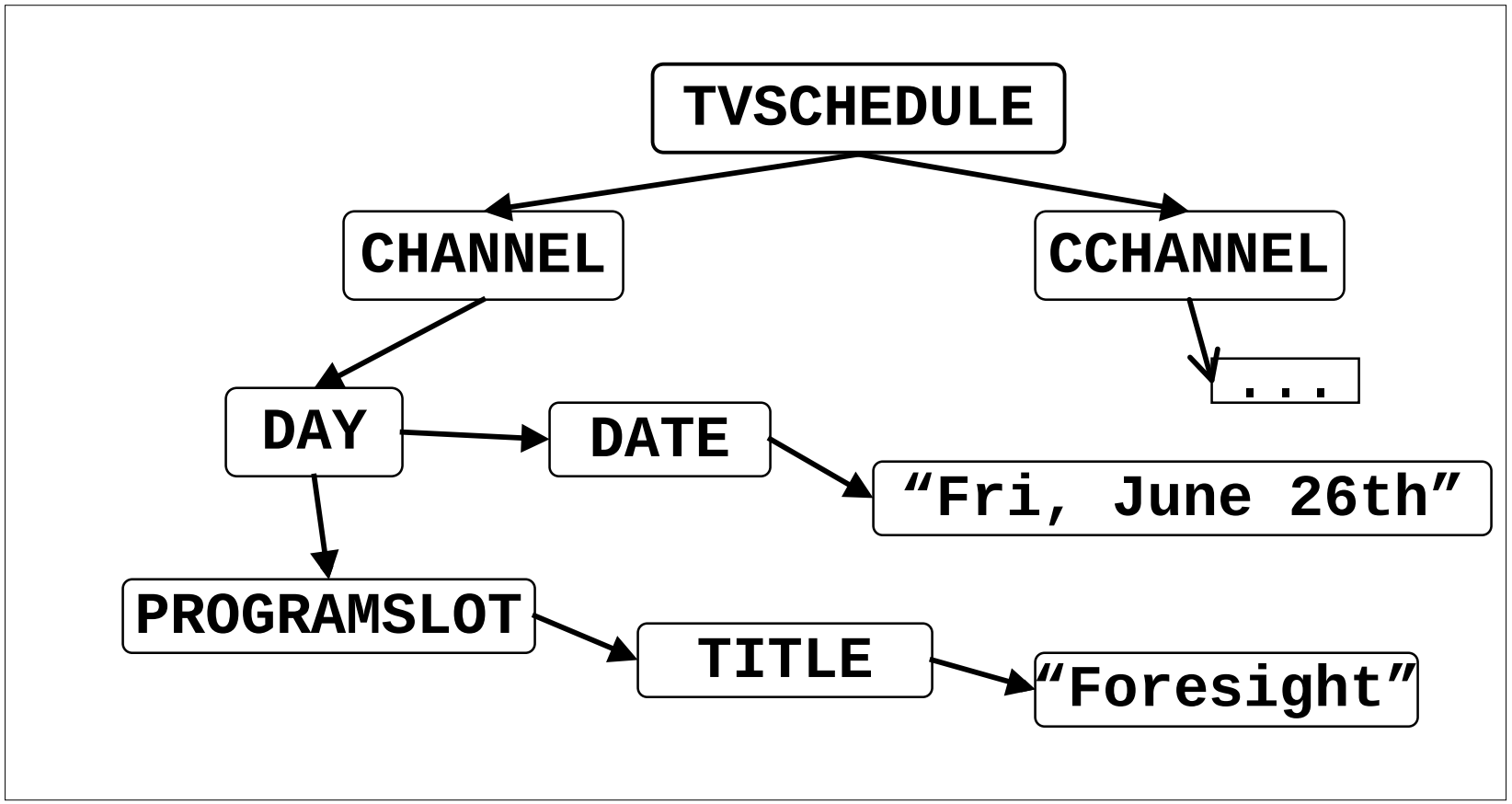
Using SAX to read XML

- To use SAX, you create an instance of a parser object that's pointed to your document and to your app.
- While SAX reads the document, it calls these methods in your app as it goes :
  - startElement: Start tag
  - endElement: End tag
  - characters: Text data
- Error handling is supported, too

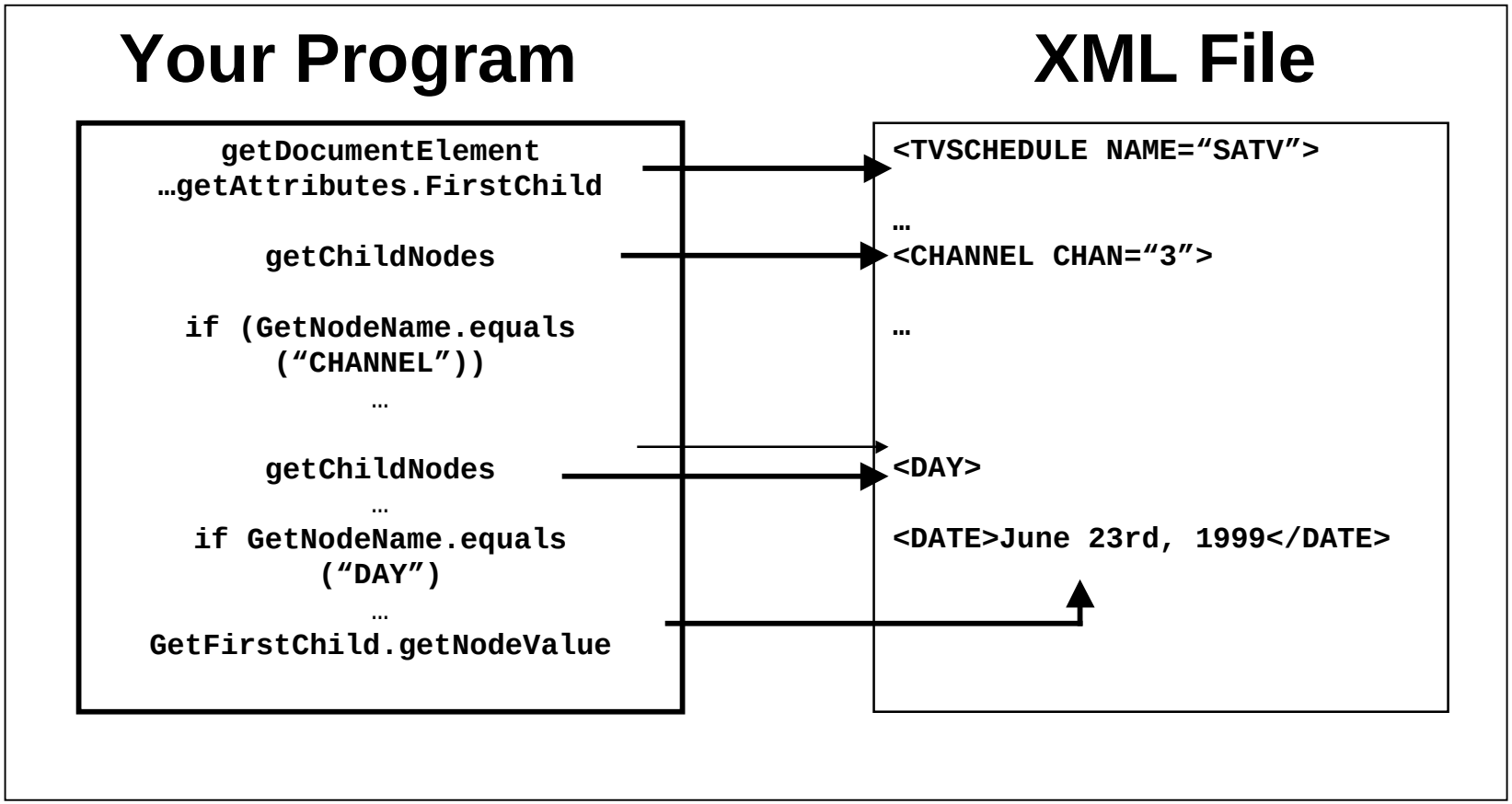
The Document Object Model: A W3C Standard

- The Document Object Model (DOM) is a standard API, from the W3C (the World Wide Web Consortium), for programs that need to manipulate HTML and XML documents.
- Unlike SAX, the DOM builds a tree of nodes, corresponding to elements, attributes and text.
- The DOM is now standard in many browsers, and supported by IBM's *XML4J* and Datachannel's *DCXJP.*

A DOM Tree


TVSCHEDULE
CHANNEL
CCHANNEL
...
DAY
DATE
“Fri, June 26th”
PROGRAMSLOT
TITLE
“Foresight”

DOM Diagram


Your Program
XML File
getDocumentElement
…getAttributes.FirstChild
getChildNodes
if (GetNodeName.equals
("CHANNEL"))
…
getChildNodes
…
if GetNodeName.equals
("DAY")
…
GetFirstChild.getNodeValue
<TVSCHEDULE NAME="SATV">
…
<CHANNEL CHAN="3">
…
<DAY>
<DATE>June 23rd, 1999</DATE>

Using the DOM to read XML

- To use the DOM, you first create an instance of the document.
- You then use method calls to traverse the document tree, for example:
  - getDocumentElement gets the root
  - getChildNodes gets all child nodes
  - getAttributes gets attributes
  - getNodeName gets the name of an element
  - getNodeValue gets the content of an element

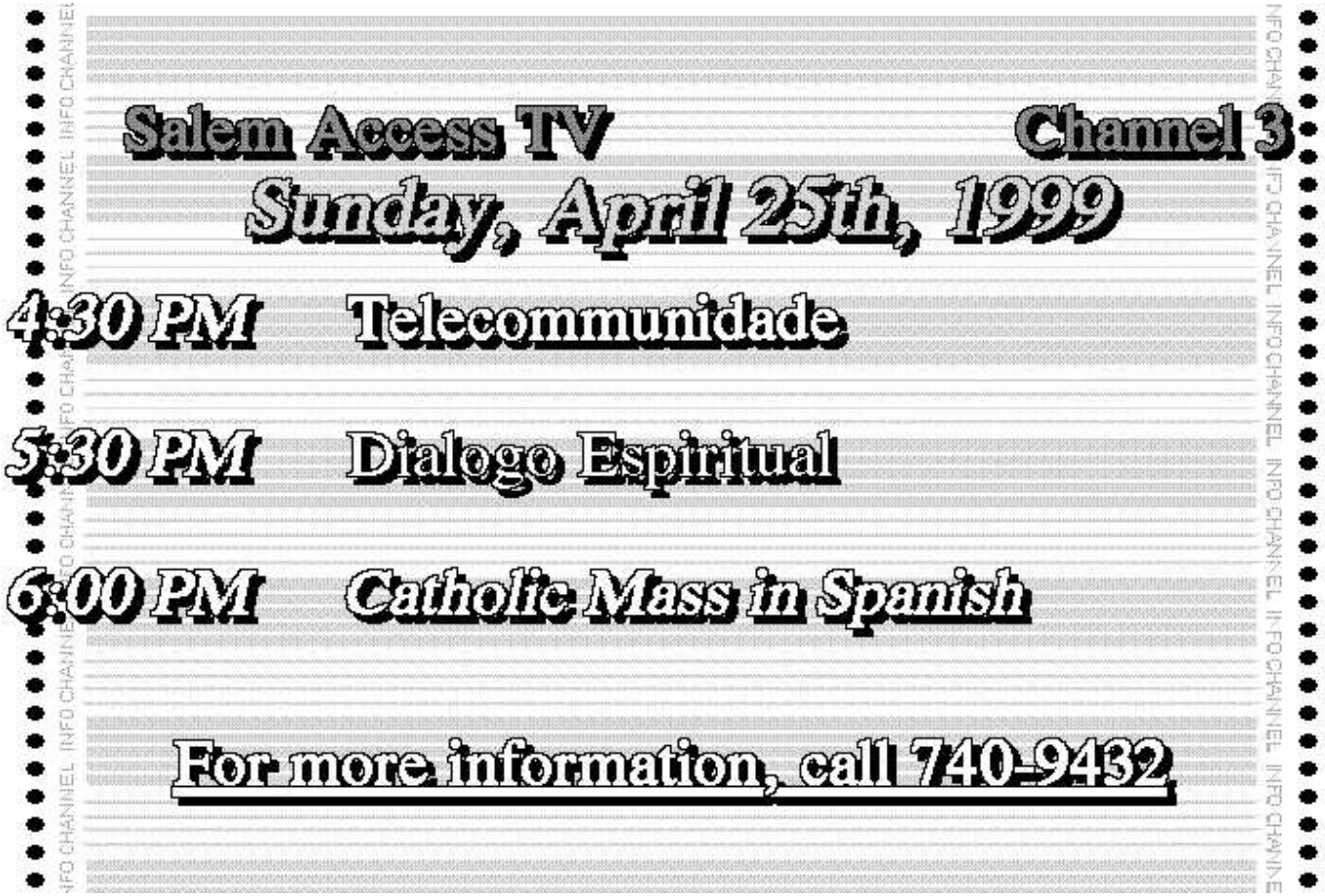
Some XML Parsers in Java

- *XML4J* — Parser from IBM's Alphaworks division. Supports both SAX and DOM
- *Ælfred* — Compact parser that supports SAX; ideal for Java applets
- *DCXJP* — Datachannel's XML parser. Supports both SAX and DOM and is used in Internet Explorer 5. Also supports XSL
- Sun is working on XML support for Java

A Example Application: ScXML

- *ScXML* is a Java application that reads the TVSCHEDULE file described earlier and converts it into Scala Lingua™ script code to display on SATV's bulletin board
- *ScXML* is template based so that staff can design pages right on the Scala, without knowing Java.

ScXML Screenshot



XML Standards: What's next?

- XSL has been split up into two different standards, XSLT, for transformation and FO (flow objects) for formatting. Expect changes.
- DOM and SAX are firm standards; both will be updated by the end of the year.
- XHTML will likely be a standard by the end of the year.
- There are many other XML features not mentioned here, including Xlink/Xpointer, that will make document handling and linking much easier.

SMIL—A Hidden Ally for Public Access

- SMIL — Synchronized Multimedia Integration Language is an XML-based language that's used to integrate multimedia, audio & video presentations.
- Supported by Real Networks G2 Player
- It supports closed captioning and ancillary text data as well
- This is a **VERY** important development for public access

Conclusion: What you need to know to use XML at your center

- Survey data in your organization
- What information needs to be shared?
- XML will not replace databases, but it can make it easier for applications to use databases.
- XML is at its best when it is used to transfer data to many different apps.
- Consider SMIL—it could be very important if you plan on streaming video.

Resources and Further Information

- The XML for Video website has links to all resources mentioned in this presentation:

  http://www1.shore.net/~dmoisan/comp/videoxml.html

- This site will be updated regularly with new software and documents
- An email list will be up and running shortly after the convention; watch for details

Acknowledgements & Credits

- Some Java examples are courtesy of *Java in A Nutshell 2nd Edition*, David Flanagan, O'Reilly & Associates, 1998 (**http://www.ora.com**)
- People who helped: Rick Hayes of Miami Valley Cable Council, Jennifer Krebs, of the City of Enumclaw, WA, & Jen Casco of Salem Access TV
- Presentation equipment courtesy MVCC & Meeting Points, Inc.