

XML(eXtensible Markup Language)

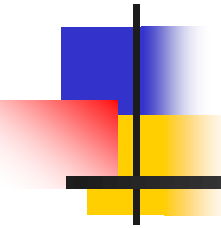
배재대학교 컴퓨터공학과
정 회 경

hkjung@mail.pcu.ac.kr

<http://mie.pcu.ac.kr>

Index

- W3C Activities
- XML Overview
- DOM/SAX
- Style
 - XSL
 - XSLT/XPath
- Schema
- Namespace
- Application
 - MathML, SVG, VoiceXML, P3P, WML, XrML



W3C Activities



Architecture

■ DOM

- platform and language neutral interface
- allows programs and scripts to dynamically access and update the content, structure and style of Web documents.
- DOM Level 1 - 1 Oct 1998
- DOM Level 2 - 13 Nov 2000
- DOM Level 3 - 9 April 2002 (WD)

■ Jigsaw

- W3C's leading-edge Web server platform
- providing an open source HTTP/1.1 implementation
- 8 April 2002 released Jigsaw 2.2.1

■ URI

- may refer to documents, resources, services, people, and, indirectly, to anything
- Web's naming and addressing technology
- URIs, URLs, and URNs 1.0 - 21 September 2001(Note)

Architecture

■ XML

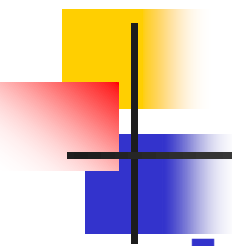
- XML 1.0 – 10 Feb 1998
- XML 1.1 – 25 April 2002 (WD)

■ Web Services

- builds a set of technologies that allow application-to-application interactions on the Web
 - XML-based protocol for communication, a description language for describing interfaces to services
- formed by expanding the former XML Protocol Activity in January 2002
- Web Services Architecture Requirements - 29 April 2002(WD)
- Web Services Description Language (WSDL)Version 1.2 - 1 July 2002(editor copy)

Document Formats

- Amaya
 - W3C's own versatile editor/browser
 - Amaya 6.28 - July 2002
- Graphics
 - Portable Network Graphics (PNG)-1 Oct 1996
 - widely supported replacement for GIF
 - WebCGM -21 January 1999
 - for industrial technical drawings
 - ***Scalable Vector Graphics (SVG)***
 - language for describing two-dimensional vector and mixed vector/raster graphics in XML.
 - SVG 1.1 – 30 April 2002
- HTML
 - HTML 4.1 – 24 Dec 1999



Document Formats

■ Internationalization

- Modern business, research, and interpersonal communication is conducted in many writing systems and languages
- Internationalization Activity tries to make sure that Web technology meets the needs of the global community
- Character Model for the World Wide Web 1.0 - 30 April 2002(WD)
- Unicode in XML and other Markup Languages - 18 February 2002(Note)

■ *Math*

- primary focus is the MathML language
- which is intended to facilitate the use and re-use of mathematical and scientific content on the Web, and for other applications
- MathML 2.0 – 21 Feb 2001

Document Formats

■ Style

- describe how documents are presented on screens, in print, or perhaps how they are pronounced
- CSS Level 1 - 17 Dec 1997
- CSS Level 2 - 12 May 1998
- XSL 1.0 – 15 Oct 2001
- XSLT 2.0 – 14 Feb 2001(WD)
- Xpath 2.0 – 14 Feb 2001(WD)

■ Device Independence

- significant efforts to integrate Web technologies into various devices
- Web services are becoming accessible from a wide range of devices including cellular phones, TV, digital cameras and in-car computers
- working to ensure seamless Web access with various kinds of devices
- CC/PP Structure – 15 March 2001
- CC/PP Requirement and Architecture – 21 July 2001
- CC/PP Terminology and Abbreviations– 21 July 2001

Interaction

■ Multimodal Interaction

- developing markup specifications for synchronization across multiple modalities and devices
- Multimodal Interaction Activity is extending the Web user interface
- Current Patent Practice - 24 January 2002(Note)

■ Synchronized Multimedia

- focuses on the design of a language for scheduling multimedia presentations where audio, video, text and graphics are combined in real-time
- SMIL 2.0 – 7 Aug 2001

■ Voice Browser

- expand access to the Web to allow people to interact with Web sites via spoken commands, and listening to prerecorded speech, music and synthetic speech
- ***VoiceXML 2.0 – 24 April 2002(WD)***
- Semantic Interpretation for Speech Recognition - 16 November 2001(WD)
- Speech Synthesis Markup Language Specification - 5 April 2002(WD)
- Speech Recognition Grammar Specification 1.0 - 26 June 2002(CR)

Technology and Society

■ XML Encryption

- Encryption renders data (plain-text) confidential (cipher-text) such that it can be safely stored or transmitted and only the intended recipients can restore the data to its original form
- XML Encryption Requirements - 04 March 2002(Note)
- XML Encryption Syntax and Processing - 04 March 2002(CR)
- Decryption Transform for XML Signature - 04 March 2002(CR)

■ XML Key Management

- developing an XML application that allows a simple client to obtain key information (values, certificates, management or trust data) from a Web service
- XML Key Management Specification (XKMS 2.0) - 18 March 2002(WD)

Technology and Society

■ XML Signature

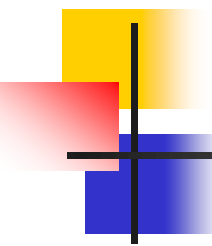
- Digital signatures provide integrity, signature assurance and non-repudiatability over Web data
- XML-Signature Requirements –14 Oct 1999(WD)
- XML-Signature Syntax and Processing - 12 February 2002
- Exclusive XML Canonicalization 1.0 - 24 May 2002(PR)

■ Privacy

- Web is being used for business and other applications where sensitive and personal information may be exchanged via the Internet
- **P3P 1.0 - 16 April 2002**
- A P3P Preference Exchange Language 1.0 (APPEL1.0) - 15 April 2002(WD)

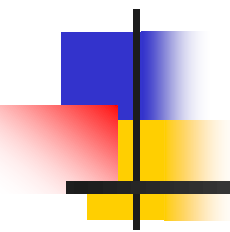
■ Semantic Web

- the idea of data on the Web defined and linked in a way that it can be used by machines for automation, integration and reuse
- RDF Model and Syntax Specification - 22 February 1999
- RDF Vocabulary Description Language 1.0: RDF Schema - 30 April 2002(WD)

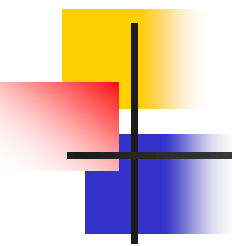


Web Accessibility Initiative

- WAI International Program Office
 - enables partnering and coordination among the many stakeholders in Web accessibility: industry, disability organizations, government, and research organizations
- WAI Technical Activity
 - conducts WAI technical work including review of technologies under development by W3C Working Groups
 - Web Content Accessibility Guidelines 1.0 - 5 May 1999
 - Web Content Accessibility Guidelines 2.0 - 24 August 2001(WD)



XML Overview



What is XML?

- A syntax for “encoding” text-based data (words, phrases, numbers, ...)
- A text-based syntax. XML is written using printable characters (no explicit binary data)
- Extensible. XML lets you define your own tags (essentially data types), within the constraints of the syntax rules
- Universal format. The syntax rules ensure that all XML processing software **MUST** identically handle a given piece of XML data.

Why XML?

- Simple (like HTML)
 - But not quite so simple
 - Strict syntax rules, to eliminate syntax errors
 - syntax defines structure (hierarchically), and names structural parts (element names) -- it is self-describing data
- Extensible (unlike HTML; vocabulary is not fixed)
 - Can create your own language of tags/elements
 - Strict syntax ensures that custom tags can be reliably processed
- Designed for a distributed environment (like HTML)
 - Can have data all over the place: can retrieve and use it reliably
- Can mix different data types together (unlike HTML)
 - Can mix one set of tags with another set: resulting data can still be reliably processed

XML Components

■ Elements

- delimited by angle brackets
- identify the nature of the content they surround
- Some elements may be empty : `<tag-name/>`
- it begins with a start-tag, `<element>`, and ends with an end-tag, `</element>`

■ Attributes

- name-value pairs that occur inside tags after the element name
- all attribute values must be quoted
- example : `<div class="preface">`
- attribute types
 - CDATA, ID, IDREF or IDREFS, ENTITY or ENTITIES, MTOKEN or NMTOKENS, a list of names
- default values
 - #REQUIRED, #IMPLIED, "value", #FIXED "value"

XML Components

■ Entity References

■ entities :

- used to represent special characters
- used refer to often repeated or varying text
- must have a unique name
- internal entities, external entities, parameter entities

■ Entity references

- begin with the ampersand and end with a semicolon
- "AT & T, Inc." : AT **&** T, Inc.

■ Character references

- decimal references: `℞`
- hexadecimal references: `℞`

XML Components

■ Comments

- begin with `<!--` and end with `-->`
- can contain any data except the literal string `--`
- are not part of the textual content of an XML document
 - An XML processor is not required to pass them along to an application

■ Processing Instructions

- provide information to an application
- `<?name pidata?>`
- name : PI target, identifies the PI to the application
- pidata : optional
- PI names beginning with XML are reserved for XML standardization
- example : `<?XML version= "1.0" encoding='UTF-8'?>`

XML Components

- CDATA Sections

- CDATA section instructs the parser to ignore most markup characters
- example : source code listing

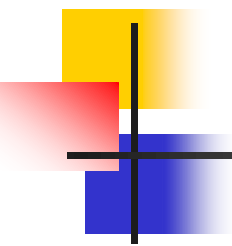
```
<![CDATA[
```

```
*p = &q;
```

```
b = (i <= 3);
```

```
]]>
```

- all character data is passed directly to the application
- comments are not recognized in a CDATA section
 - If present, the literal text `<!--comment-->` will be passed directly to the application



XML Document

- Well-formed Documents
 - document instance must conform to the grammar of XML documents
 - looks right, but logical structure is not validated
 - need to have a single outermost element (root, document element)

- Valid Documents
 - contains a proper document type declaration
 - the document obeys the constraints of that declaration

A Well-formed XML Document Ex.

**XML
Declaration**
("this is XML")

Binary encoding used in file

```
<?xml version="1.0" encoding="iso-8859-1"?>
<partorders
  xmlns="http://myco.org/Spec/partorders">
  <order ref="x23-2112-2342"
    date="25aug1999-12:34:23h">
    <desc> Gold sprockel grommets,
           with matching hamster
    </desc>
    <part number="23-23221-a12" />
    <quantity units="gross"> 12 </quantity>
    <deliveryDate date="27aug1999-12:00h" />
  </order>
  <order ref="x23-2112-2342"
    date="25aug1999-12:34:23h">
    . . . Order something else . . .
  </order>
</partorders>
```

A Valid XML Document Ex.

attribute of this
quantity element

tags
element
`<?xml version="1.0" encoding="iso-8859-1"?>`
`<DOCTYPE partorders SYSTEM "partorder.dtd">`

`<partorders`

`xmlns="http://myco.org/Spec/partorders" >`

`<order ref="x23-2112-2342"`

`date="25aug1999-12:34:23h">`

`<desc>` Gold sprockel grommets,
with matching hamster

`</desc>`

`<part number="23-23221-a12" />`

`<quantity units="gross"> 12 </quantity>`

`<deliveryDate date="27aug1999-12:00h" />`

`</order>`

`<order ref="x23-2112-2342"`

`date="25aug1999-12:34:23h">`

. . . Order something else . . .

`</order>`

`</partorders>`

Hierarchical, structured information

Defining Specific Language

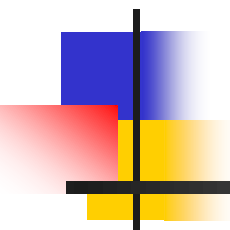
- Two ways of doing so:
 - XML Document Type Declaration (DTD) -- Part of core XML spec.
 - XML Schema -- New XML specification (2001), stronger constraints on XML documents.
- Adding dialect specifications implies two classes of XML data:
 - Well-formed
 - An XML document that is syntactically correct
 - Valid
 - An XML document that is both well-formed and consistent with a specific DTD (or Schema)
- Most current dialects defined using DTDs.
- Schemas often used for type validation.

XML Processing

- DOM
 - An object-oriented interface
 - Status
 - DOM Level 1 - 1 Oct 1998
 - DOM Level 2 - 13 Nov 2000
 - DOM Level 3 - 9 April 2002 (WD)
- SAX: Simple API for XML
 - <http://www.megginson.com/SAX/index.html>
 - An event-based interface

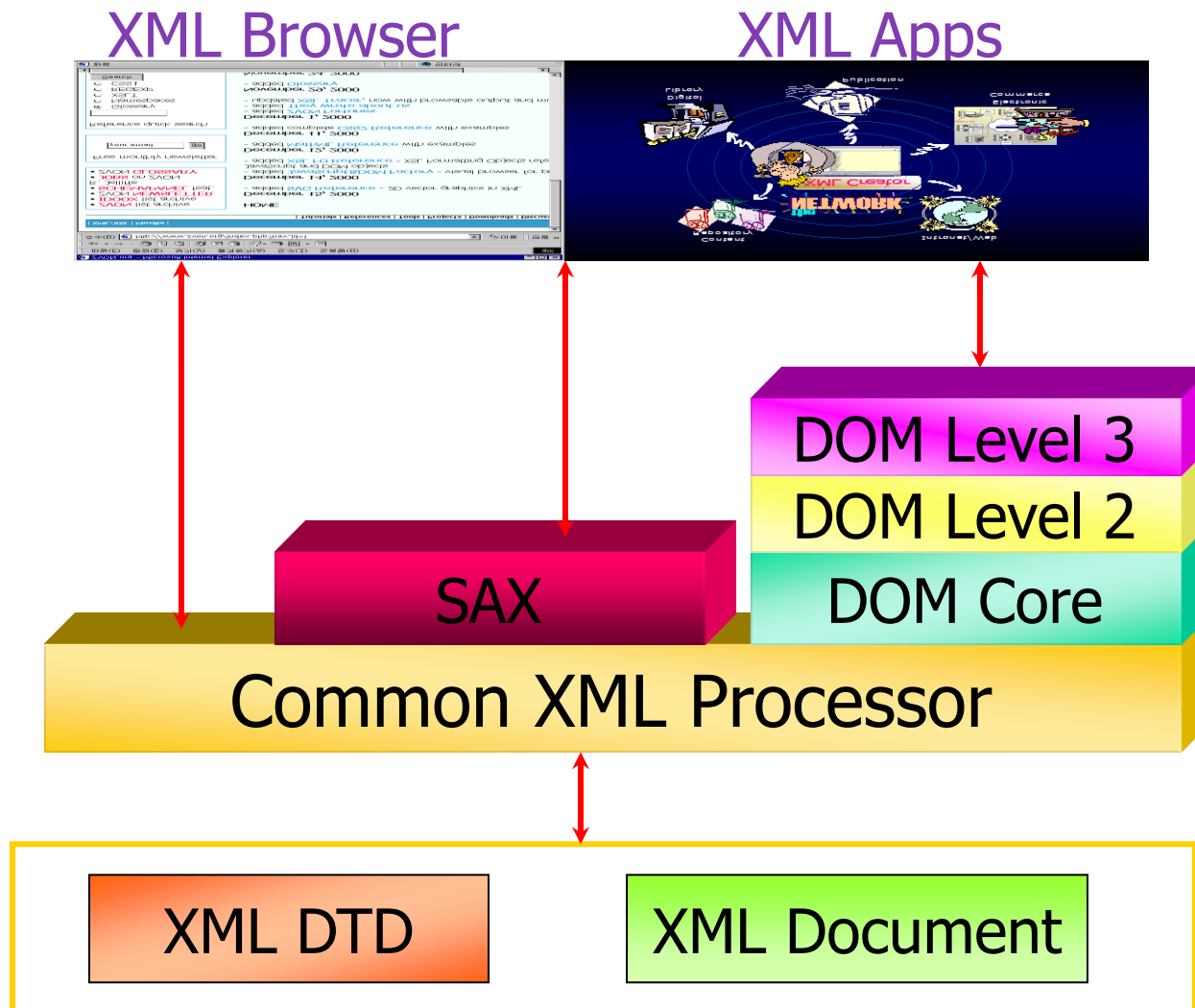
XML Processing

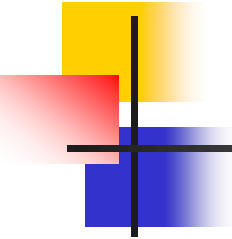
- JDOM: Java Document Object Model
 - <http://www.jdom.org>
 - A Java-specific object-oriented interface
 - Advantages
 - Very useful for dynamic modification of the tree
 - Useful for querying
 - Much nicer Object Oriented programming interface than DOM
 - Disadvantages
 - Can be slow (make that tree...), and can take up lots of memory
 - New, and not entirely cooked (but close)
 - Only works with Java, and not (yet) part of Core Java standard



DOM/SAX

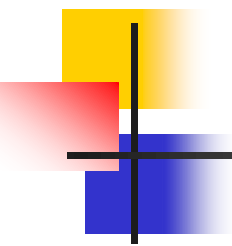
XML API role





XML API

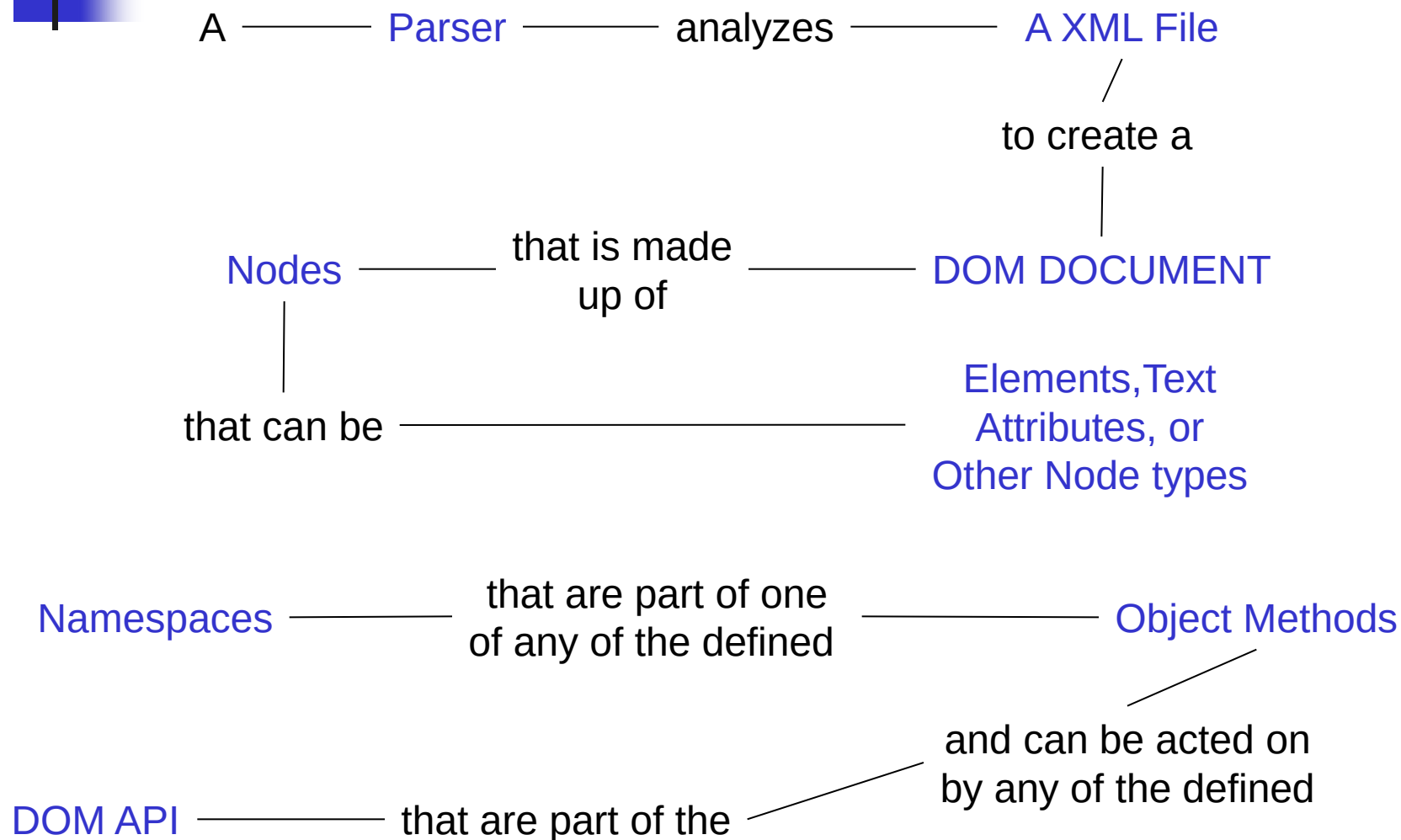
- Two types: tree-based and event-based
 - A **tree-based** API (eg., **DOM**) compiles an XML document into an internal tree structure
 - Uses much physical memory
 - An **event-based** API (eg., **SAX**) reports parsing events (such as the start and end of elements) directly to the application



What's DOM

- platform and language neutral interface
- allows programs and scripts to dynamically access and update the content, structure and style of Web documents.
- An object-oriented interface
- Status
 - DOM Level 1 - 1 Oct 1998
 - DOM Level 2 - 13 Nov 2000
 - DOM Level 3 - 9 April 2002 (WD)

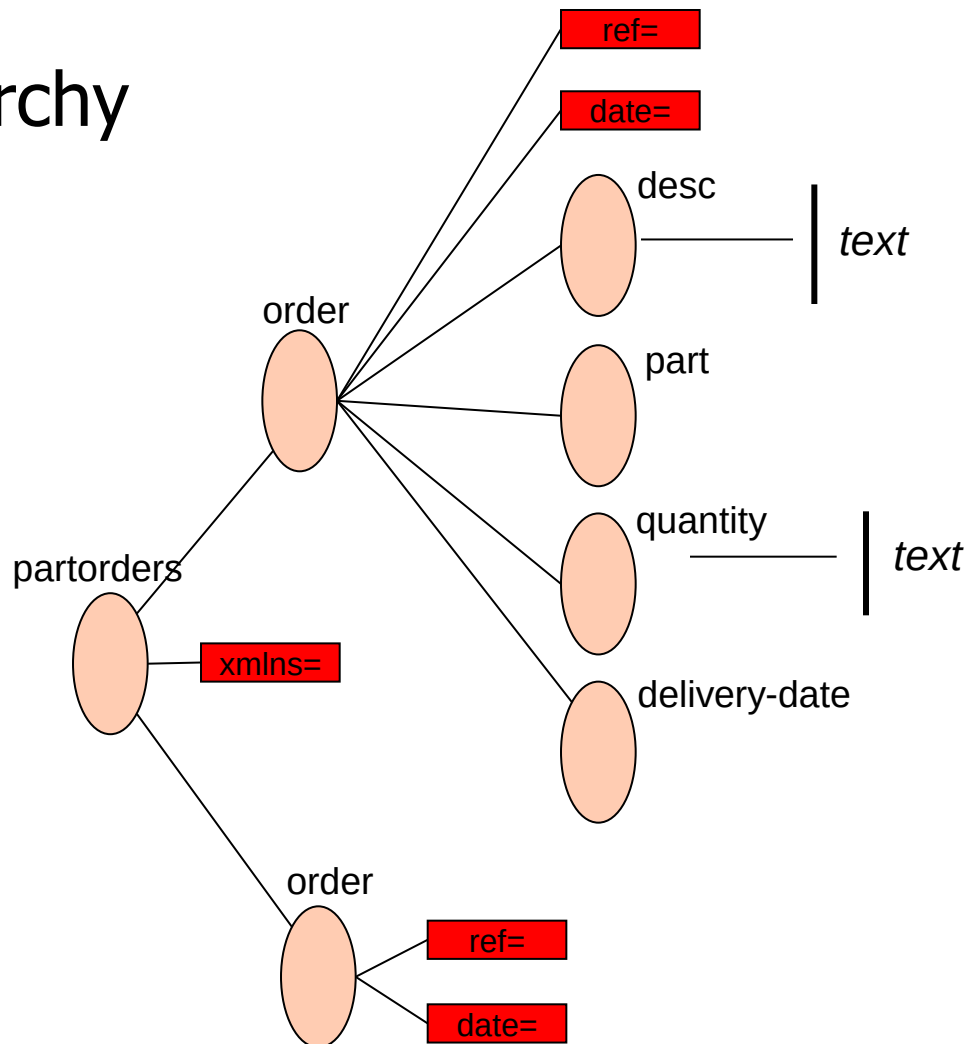
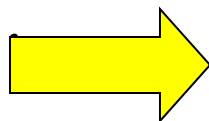
DOM roadmap



DOM Data Model

■ Creating the hierarchy

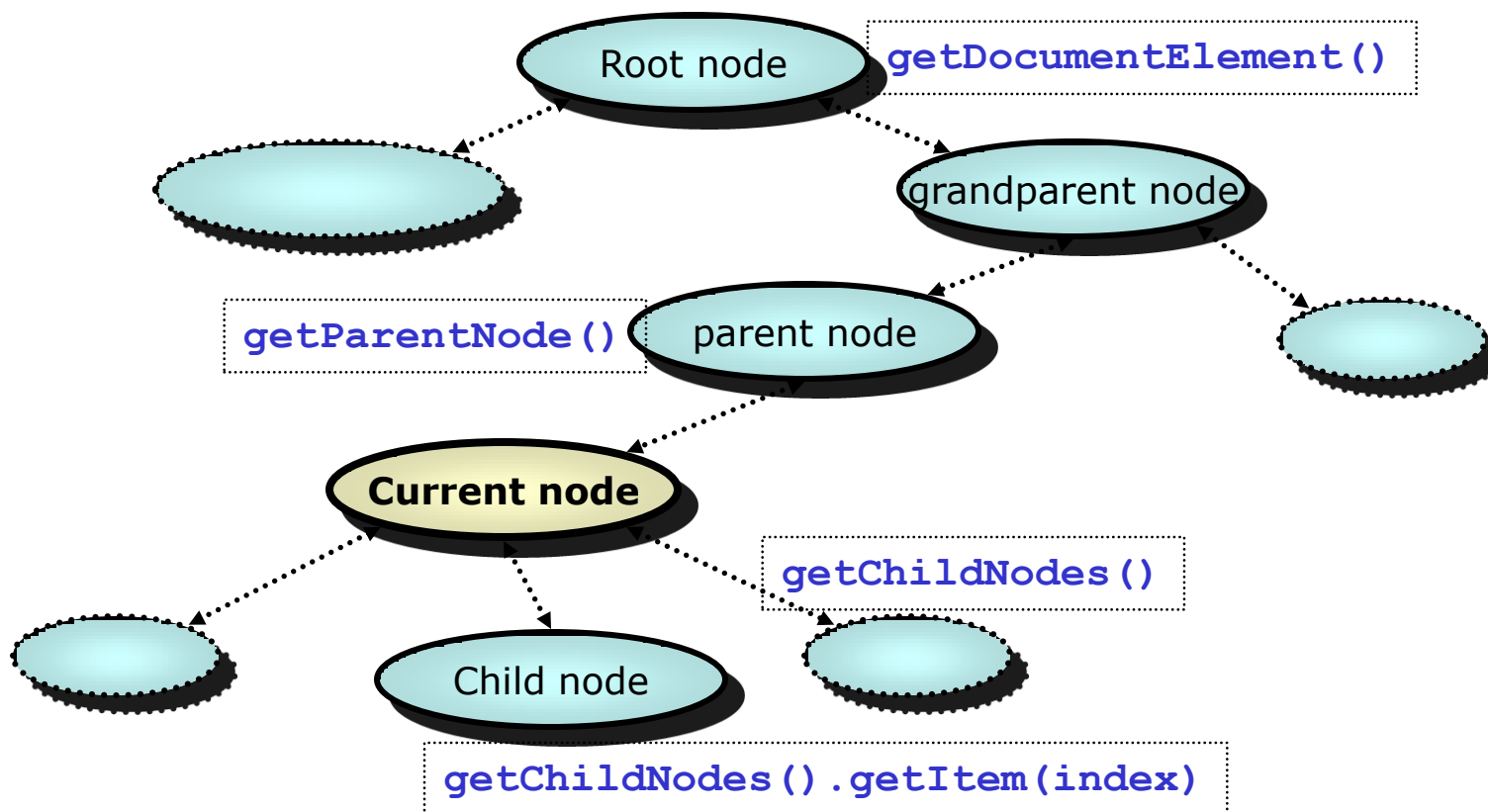
```
<partorders  
  xmlns="...">  
  <order date="..."  
    ref="...">  
    <desc> ..text.  
  </desc>  
  <part />  
  <quantity />  
  <delivery-date />  
</order>  
<order ref=".." .../>  
</partorders>
```



DOM properties

async	nextSibling	paserError
attribute	nodeName	previousSibling
childNodes	nodeType	readyState
doctype	nodeValue	url
documentElement	ondataavailable	validateOnParse
firstChild	onreadystatechange	xml
Implementation	ownerDocument	
lastChild	parentNode	

DOM Node & API Relation



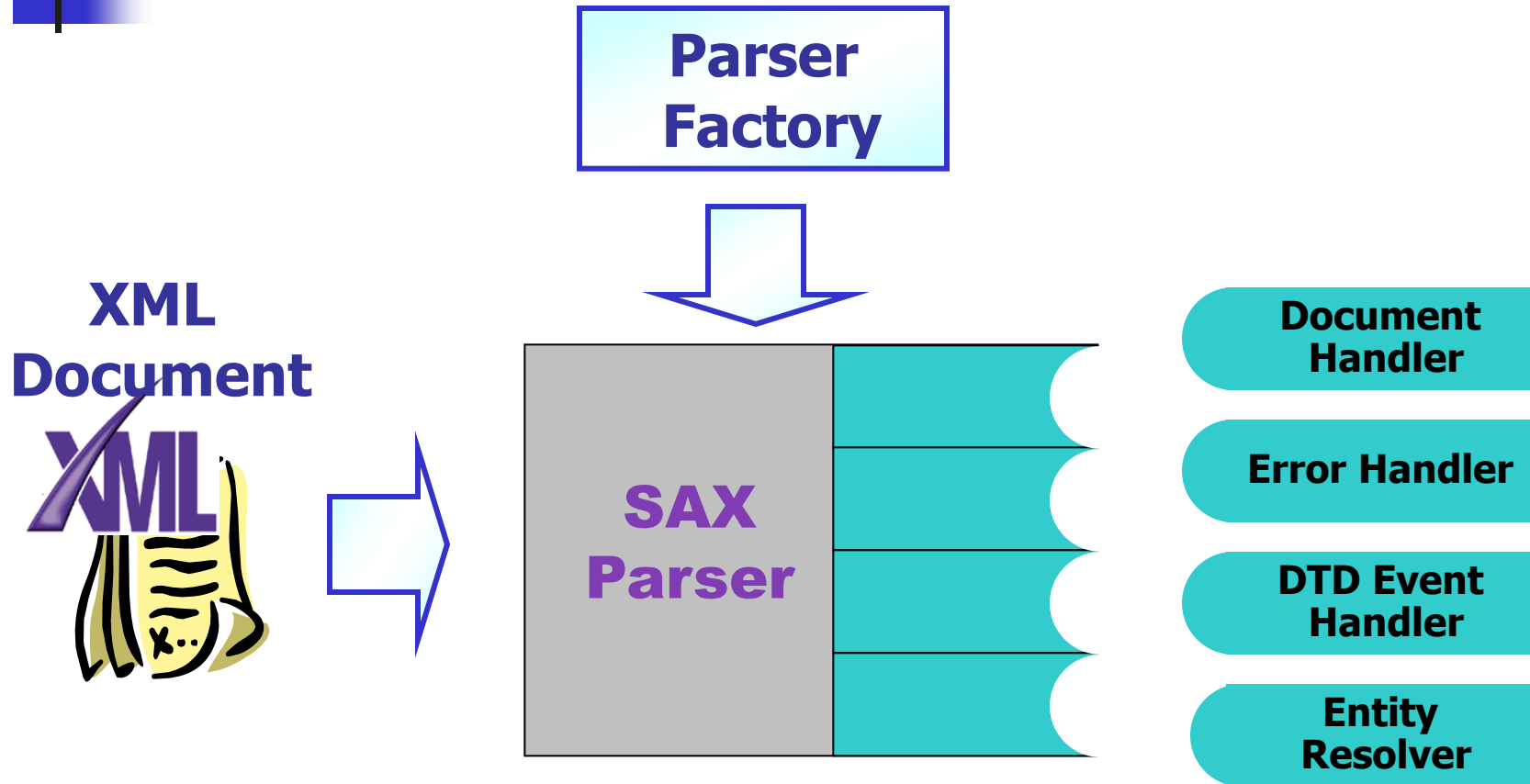
What is SAX?

- Is a standard API for event-driven processing of XML data
 - Allowing parsers to deliver information to applications in digestible chunks
- Developed by members of the XML-DEV mailing list
- First version of spec released in May 1998
- Large number of SAX compliant parsers developed in a number of languages
 - Most are freely available
 - Java ones are usually more mature and fastest to implement new specifications

SAX mechanism

- A SAX parser generates events
 - At the start and end of a document
 - At the start and end of an element
 - When it finds characters inside an element
 - Upon encountering errors
 - Upon encountering negligible whitespace
 - and at several other points
- Uses callback mechanism to notify application
- Write Java code that handles each event, and user decides what to do with the information got from the parser

SAX Structure



SAX interface

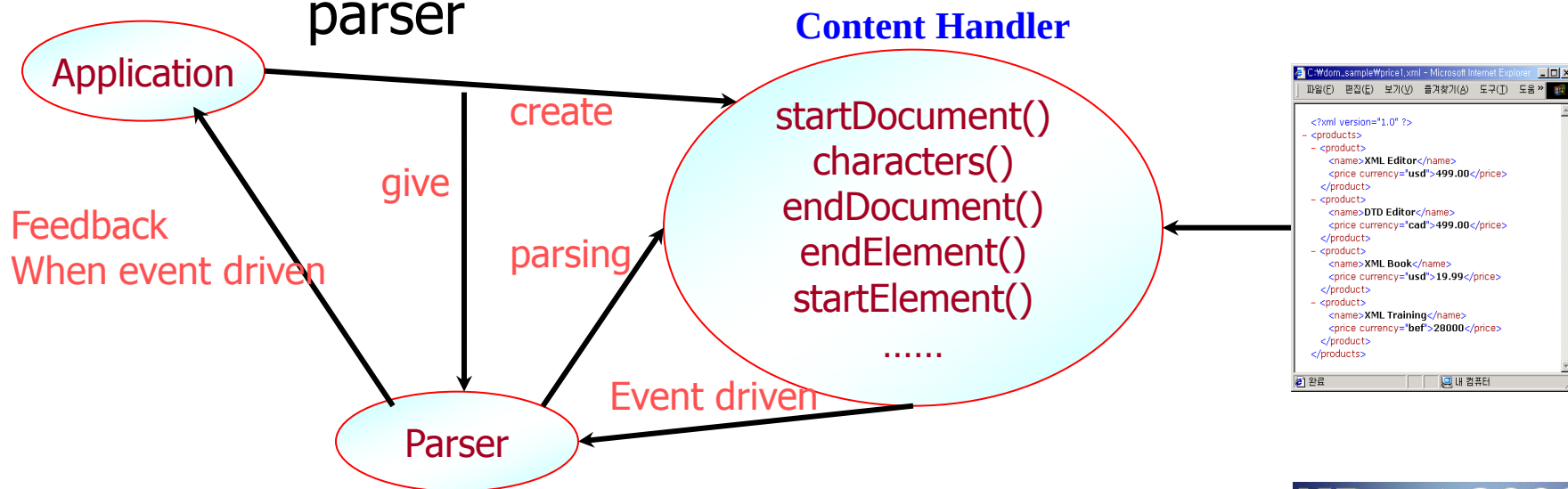
■ The Parser

- The parser developer creates a class that actually parses the XML document or data stream
- The parser reads the XML source data
- Stops reading when encounters a meaningful object
- Sends the information to the main application by calling an appropriate method
- Waits for this method to return before continuing

SAX interface

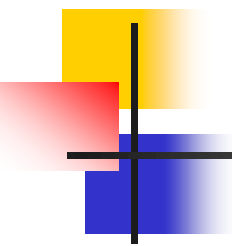
■ The Document Handler

- Interface to receive basic markup events from the parser
- the application developer must create a class that implements the DocumentHandler interface
- Usually implemented by class that activates the parser



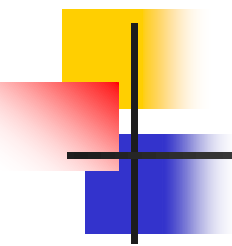
SAX interface

- The AttributeList
 - Have a wrapper object for all attribute details
 - Cannot distinguish attributes defined explicitly from those specified in DTD
- The ErrorHandler
 - When the application needs to be informed of warnings and errors
- The Locator
 - Associates a SAX event with a document location
 - can tell the entity, line number and character number of the warning or error
- The DTD Handler
 - Provides callback methods to receive notification of DTD events
 - A mechanism to inform an application about any binary entity that the parser encounters



SAX interface

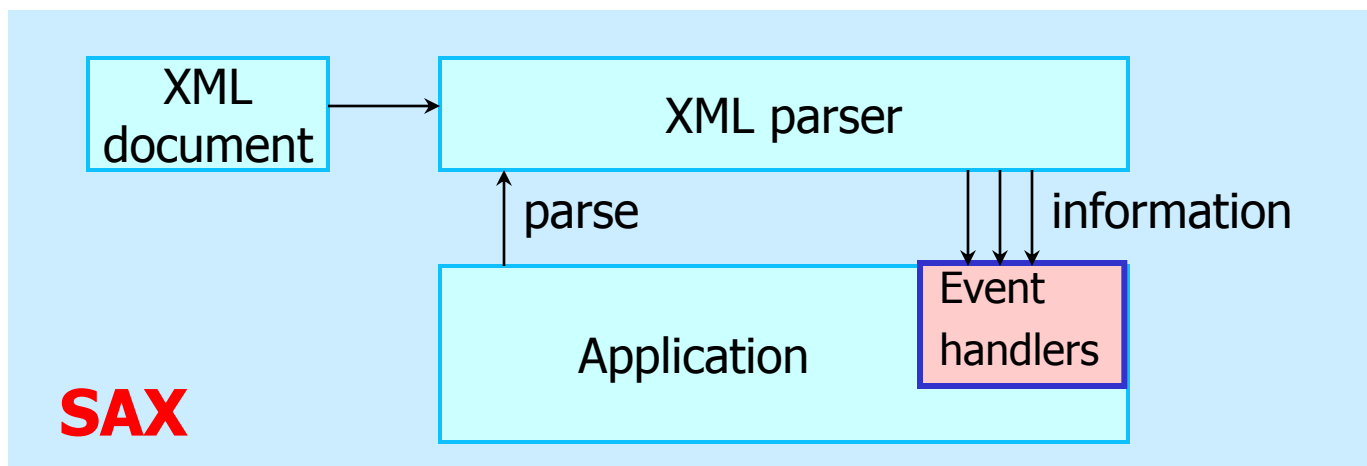
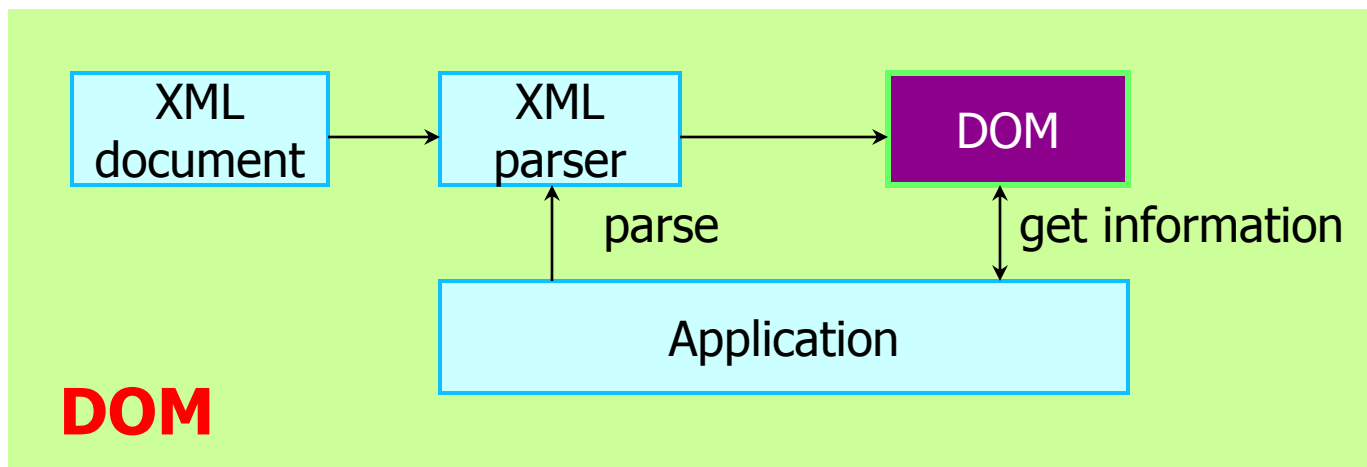
- The EntityResolver
 - The application is not aware of the physical structure of the XML data
 - The parser contains an entity manager that hides the complexity from the application which sees the data as a single stream
 - Can intercept references to entities by implementing the **EntityResolver** interface



SAX 2.0

- Supports for namespaces, for filter chains, and for querying and setting features and properties in the parser
- Public-domain software from grass-roots effort (XML-DEV mailing list), not owned by anyone, nor associated with any consortium
- Supported/to be supported by most major parsers

DOM vs. SAX



Advantages/ Disadvantages

■ DOM

■ Advantages

- Very useful for dynamic modification of, access to the tree
- Useful for querying (I.e. looking for data) that depends on the tree structure
`[element.childNodes("2").getAttributeValue("boobie")]`
- Same interface for many programming languages (C++, Java, ...)

■ Disadvantages

- Can be slow (needs to produce the tree), and can take up lots of memory
- DOM programming interface is a bit awkward, not terribly object oriented

Advantages/ Disadvantages

■ SAX

■ Advantages

- Simple to use
- Very fast
- Low memory use (doesn't read an XML document entirely into memory)

■ Disadvantages

- Not doing very much for you -- you have to do everything yourself
- Not useful if you have to dynamically modify the document once it's in memory (since you'll have to do all the work to put it in memory yourself!)

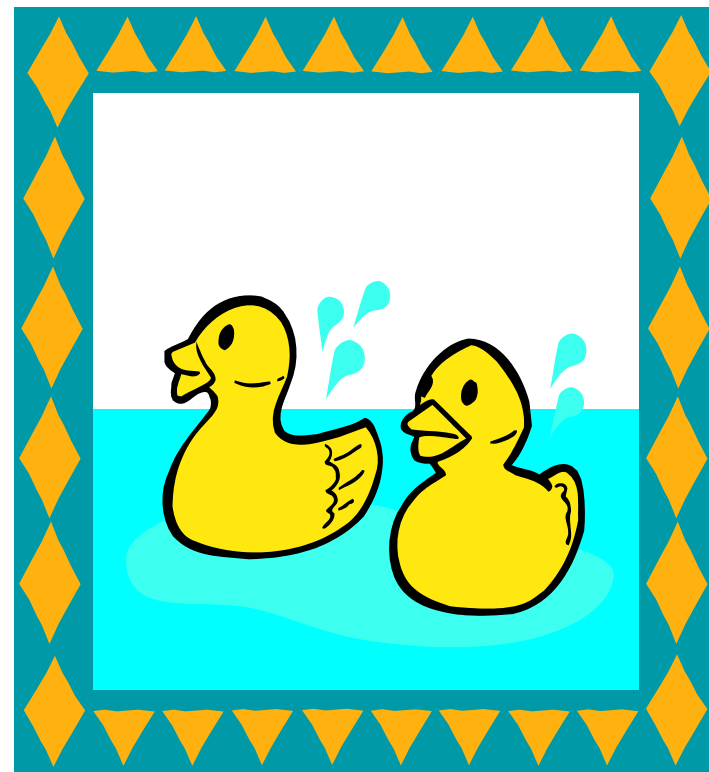


Using CSS with XML

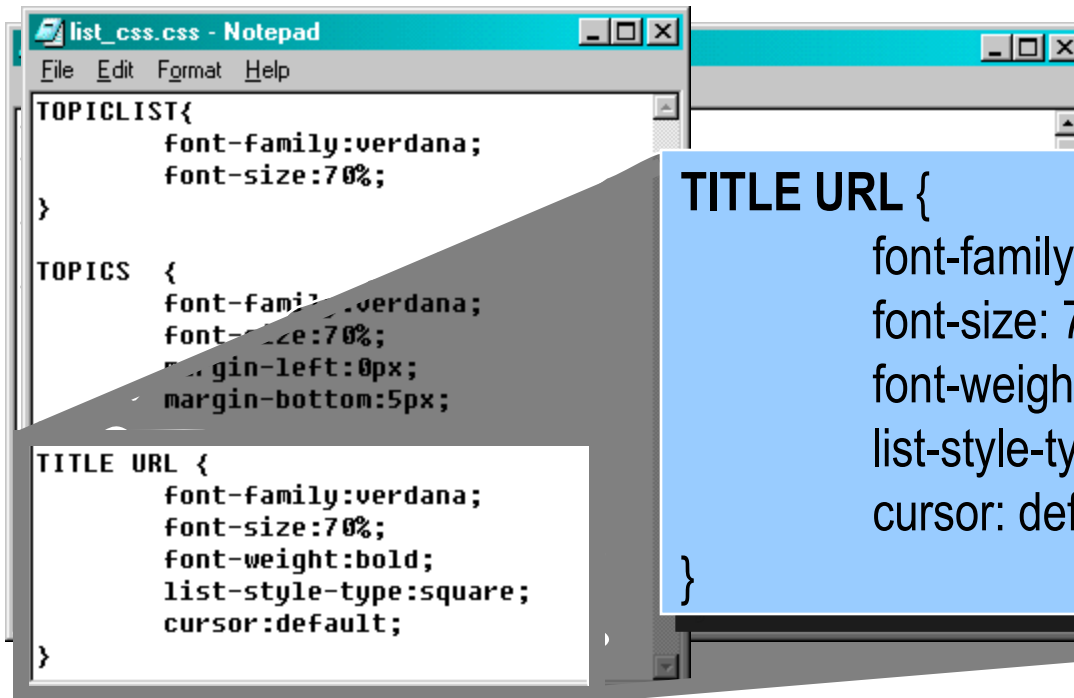
- Why use CSS with XML?
 - Easy to learn and implement
 - Already at work in the Web community
 - Works with HTML
- CSS does have a downside
 - Can't generate text
 - Can't grab an item from one place and use it again in another place
 - Isn't programming language
 - Use a simple box-oriented formatting model

Immediate Solutions

- Duck to water
CSS to XML
- Selectors and
property definitions
- Show you everything
you need to know to
create CSS style
rules and style
sheets and link them
to existing
documents



Example



```

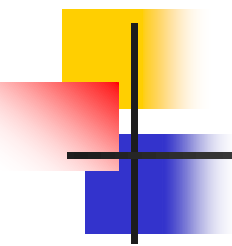
TITLE URL {
    font-family: verdana;
    font-size: 70%;
    font-weight: bold;
    list-style-type: square;
    cursor: default;
}
  
```

```

TITLE {
    margin: 10% 10% 10% 10%
}
  
```

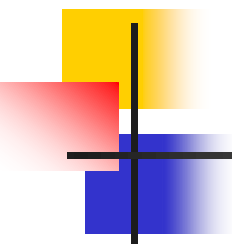
```

TITLE {
    margin: 10% 10%
}
  
```

What is XSL?

- XML provide information about the structure of a document
- XSL create style sheets for XML document
 - Formatting information
 - How to format the element, contents for viewing by user (ex, web browser)
- The following design
 - Over the internet, XML syntax, subset of DSSSL
 - Mapping of CSS into XSL, ease to create

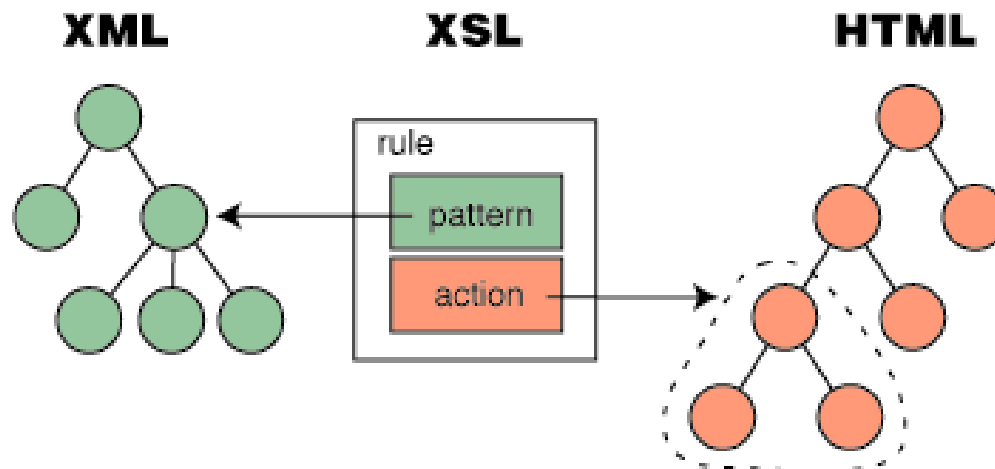


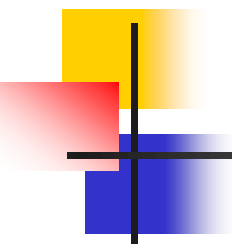
Why Stylesheet?

- Reuse
 - the same content can look different in different contexts
- Multiple output formats
 - different media (paper, online), different sizes (manuals, reports), different classes of output devices (workstations, hand-held devices)
- Styles tailored to the reader's preference
 - print size, color, simplified layout for audio readers
- Standardized styles
 - corporate stylesheets can be applied to the content at any time
- Freedom from style issues for content authors
 - technical writers needn't be concerned with layout issues because the correct style can be applied later

A construction rule

- A construction rule describes how a particular element is to be transformed into displayable output
- The construction rule consists of two parts:
 - a pattern identifying a type of XML source element
 - an action describing what to do with elements that match this pattern.





A construction rule

- Patterns
 - Choose which elements to apply which style rules to.
 - Have a target-element (matched in document).
 - Supply a type attribute (matched in element).
- Actions.
 - Objects to create in the output tree
 - Directly create flow objects, add literal text

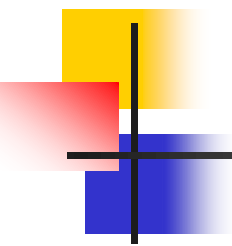
Defining XSL constructs

- XSL documents
 - An XML document
 - Must adhere to all of XML's syntactical rule
 - Using a validating XML Parser

```
<XSL>
  <IMPORT HREF="/style/main-style.xsl" />
  <HEADER>
    Header text to be affected by the header macro
  </HEADER>
  ...
</XSL>
```

```
<XSL>...</XSL>
<IMPORT>
<DEFINE-MACRO>
<DEFINE-SCRIPT
<ID> and <CLASS>
```

```
<?XML-STYLE SHEET HREF="/style/main-style.xsl" TYPE="TEXT/XSL"?>
```



Xpath

■ Define

- As XML Path Language (Xpath), with XML separate language spec.
- Question notation that do XML documents' elements and text to do filtering with addressing .
- Declarative expression to be not procedural notation

Xpath's action principle

```
<booklist>  
  <book>  
    <title>Xpath</title>  
    <writer>HK Jubg</writer>  
    <publish>MIE</publish>  
  </book>  
</booklist>
```

```
<xsl:template match = "/booklist/book/title">
```

Document Root

booklist

book

title

Xpath

writer

HK Jung

publish

MIE

Location Expression

- / : Child Operator of present node return.
- // : In Document to Recursive Descent Operator all relevant node returns.
- . :current context
- * :Regardless of element name by Wildcard all node conversions
- @ : Attribute's attribute node return.
- @* : Wildcard that replace all attribute nodes of node.
- : : As Namespace Operator, Namespace's Prefix return.
- () : Group Operator
- [] : filters

Finding Node

- General method to get node that want at node
- expression : "/"
- Appoint position using relative pass from present node using "/" that is child operator mainly.
- Form : <xsl:template match="00/00/00">
- Finding **writer** node

```
<xsl:template match ="booklist">  
  <xsl:value-of select ="../book/writer"/>  
</xsl:template>
```

Finding Attribute

- Expression that find Attribute at node
- Expression : "@"
- Form : <xsl:template
match="/@attribute_name">
- Finding books attribute

```
<xsl:template match ="book">  
  <xsl:value-of select ="/@catalogue"/>  
</xsl:template>
```

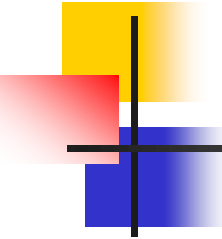
Recursive Descent Operator

- In element name regardless of Context Node's position path appointment
- expression : "//"
- "/" Operator replaces node that escape relative route and have name of appropriate element been in what position of present document.

```
<xsl:template match ="booklist">  
  <xsl:value-of select ="//title"/>  
</xsl:template>
```

Boolean, Compression Expression

- **and** : Logic operator and
- **or** : Logic operator or
- **Not()** : Logic operator conversion
- **=** : Equal operator
- **!=** : Not equal
- **<** : Comparison operator
- **>** : Comparison operator
- **>=** : Comparison operator
- **|** : Set of two nodes (Union)

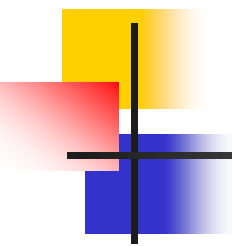


Axes

- Child::Node
 - All elements that is children of node selection
- descendant-or-self::Node or descendant::Node
 - All descendant node of context node and all <Descendant> elements of context node selection
- attribute::attribute_name
 - Attribute of node selection
- parent::Node, ancestor::Node
 - parent node of node, ancestor node of node selection
- following::Node following-sibling::Node
 - All nodes that come in front of node(only sibling node)

Node Function

- Name()
 - When wish to know name of node
- Node()
 - Is function that is not used hardly to replace node confidence.
- Processing-instruction()
 - Processing instruction return
- Text()
 - PCDATA of node confidence return .Children's PCDATA does not replace.
- Comment()
 - Comment return



Location Function

- Position()
 - Function that replace position in document of node from node-set that get.
- Last()
 - Replace last node most node-set.
- count()
 - Replace number of replaced node-set.

Number / Boolean Function

- number()
 - Function that change by number style that text style is PCDATA.
- Boolean()
 - number and 0, false. If number is value except 0, true.
 - string and length is longer than 0, true
 - node-set and is not empty, true
- not()
 - In case confirm reverse value of replaced value

XSL Transformations Concept

- 7 Type of Tree
 - **Root** : Root of tree(Include the Root element, PI, child element)
 - **Elements** : Include the root, child, descendant
 - **text** : Character Data(Character in CDATA paragraph inside <(>))
 - **Attributes** : Set of attribute node
 - **Namespaces** : Thing to divide element or attribute
 - Have URI value.
 - **Processing Instruction** : Description for instruction
 - **Comments** : node to explanation

XSLT example

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

```
<?xml-stylesheet type="text/xsl" href="14-5.xsl"?>
```

PI

```
<PERIODIC_TABLE>
```

```
<ATOM STATE="GAS">
```

root

```
<NAME>Hydrogen</NAME>
```

```
<SYMBOL>H</SYMBOL>
```

element

```
<ATOMIC_NUMBER>1</ATOMIC_NUMBER>
```

```
<ATOMIC_WEIGHT>1.00794</ATOMIC_WEIGHT>
```

```
<BOILING_POINT UNITS="Kelvin">20.28</BOILING_POINT>
```

```
<MELTING_POINT UNITS="Kelvin">13.81</MELTING_POINT>
```

Child node

```
<DENSITY UNITS="grams/cubic centimeter">
```

```
0.0899
```

```
</DENSITY>
```

```
</ATOM>
```

```
<ATOM STATE="GAS">
```

```
<NAME>Helium</NAME>
```

```
<SYMBOL>He</SYMBOL>
```

```
<ATOMIC_NUMBER>2</ATOMIC_NUMBER>
```

```
<ATOMIC_WEIGHT>4.0026</ATOMIC_WEIGHT>
```

```
<BOILING_POINT UNITS="Kelvin">4.216</BOILING_POINT>
```

```
<MELTING_POINT UNITS="Kelvin">0.95</MELTING_POINT>
```

```
<DENSITY UNITS="grams/cubic centimeter">
```

```
0.1785
```

```
</DENSITY>
```

```
</ATOM>
```

```
</PERIODIC_TABLE>
```

attribute

XSL templates

- Template rule is important part of XSL style sheet
- `xsl:template` element
 - Have the "match" attribute : node identification
- `xsl:apply-templates` element : Use to handle child node

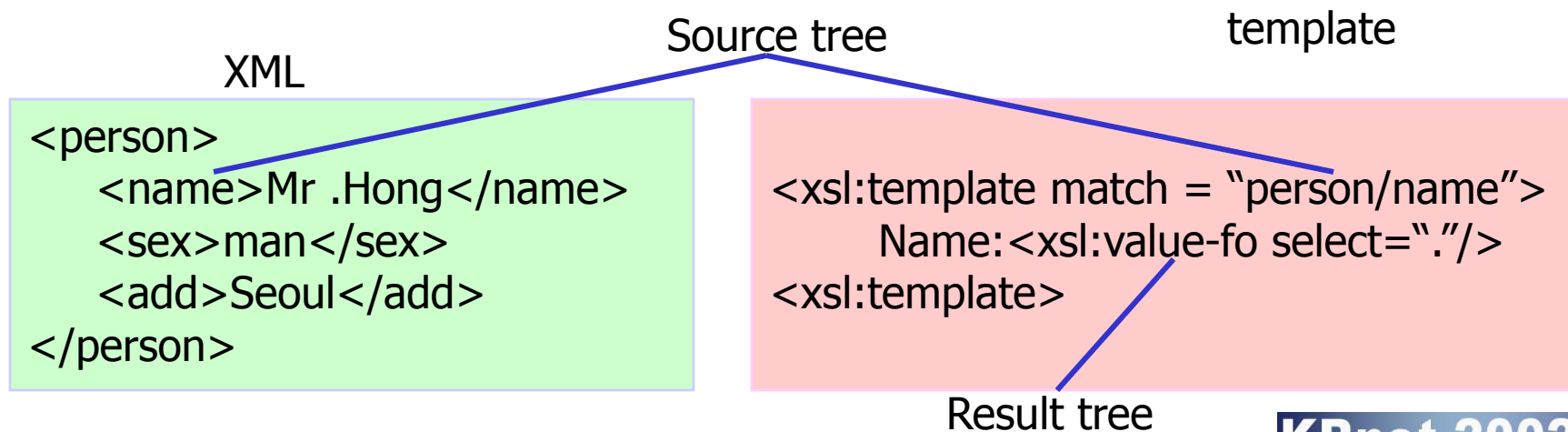
attribute

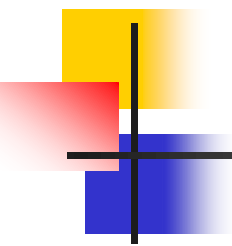
Template rule

```
<xsl:template match="atom">  
  <xsl:apply-templates select="name"/>  
</xsl:template>
```

XPath in XSLT

- Though embody XSLT, in template path appointment grammar that approach exactly in pattern
- Do so that can get correct slice of necessary information specifying address on XML Document's sections.





XSLT Patterns for matching Nodes

- Pattern : Set of condition that relate in node
- Root Node : The Root Node
- Element Names : Each name of all elements
- / : List element hierarchically.
- // : Descendants element selection
 - //from-descendants-of-self

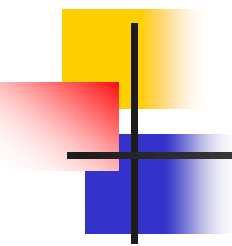
XSLT Patterns for matching Nodes

- `id()` : Unique id element value:"
 - `<xsl:template match="id('e47')">`
- `@` : Attribute selection of context node
- `comment()` : Comment node processing
- `pi()` : PI node processing
- `text()` : All text node processing
- `|` : or , Processing that several element by sequence

XSLT Expressions for Selecting Nodes

- Expression Types
 - Node Sets : List of node from input document
 - Last() : Last node in context node set
 - Position() : First position of context node list inside
 - Count(): Number of node from node set

```
<xsl:value-of select="position()"/>
```



XSLT Expressions for Selecting Nodes

- Booleans : true or false
 - Boolean Rule
 - A number is false if it's zero or NULL true
 - An empty node set is false; all other node sets are true
 - An empty result fragment is false; all other result fragments are true
 - A zero length string is false:all other strings are true
 - `<`, `>`, `<=`, `>=`, `or`, and operator use
 - Return Boolean Value.: `true()`, `false()`, `lang(code)`

XSLT Expressions for Selecting Nodes

■ Numbers

- Mathematical operator offer : +, -, *, div
- Function that number does arithmetic
 - floor() : Maximum of integer greater than parameter
 - ceiling() : Minimum of integer less than parameter
 - round() : With parameter almost similar integer
 - sum() : Total sum of node value in parameter of node set
- Binary operator
 - Mod : Remainder about that divide two numbers
 - Quo : Integer of share about that divide two numbers

```
<xsl:value-of select="round(ATOMA – ATOMB)"/>
```

XSLT Expressions for Selecting Nodes

■ Strings

- Unicode literal contiguously composition
- Rule by String() function
 - Node set is conversion by attaching node value in set
- Function
 - starts-with(main_string, prefix_string)
 - contains(containing_string, contained_string)
 - substring-before(string, marker-string)
 - substring-after(string, marker-string)
 - normalize(string)
 - translate(string, replaced_text, replacement_text)
 - concat(string1, string2, ...)
 - format-number(number, format-string, locale-string)

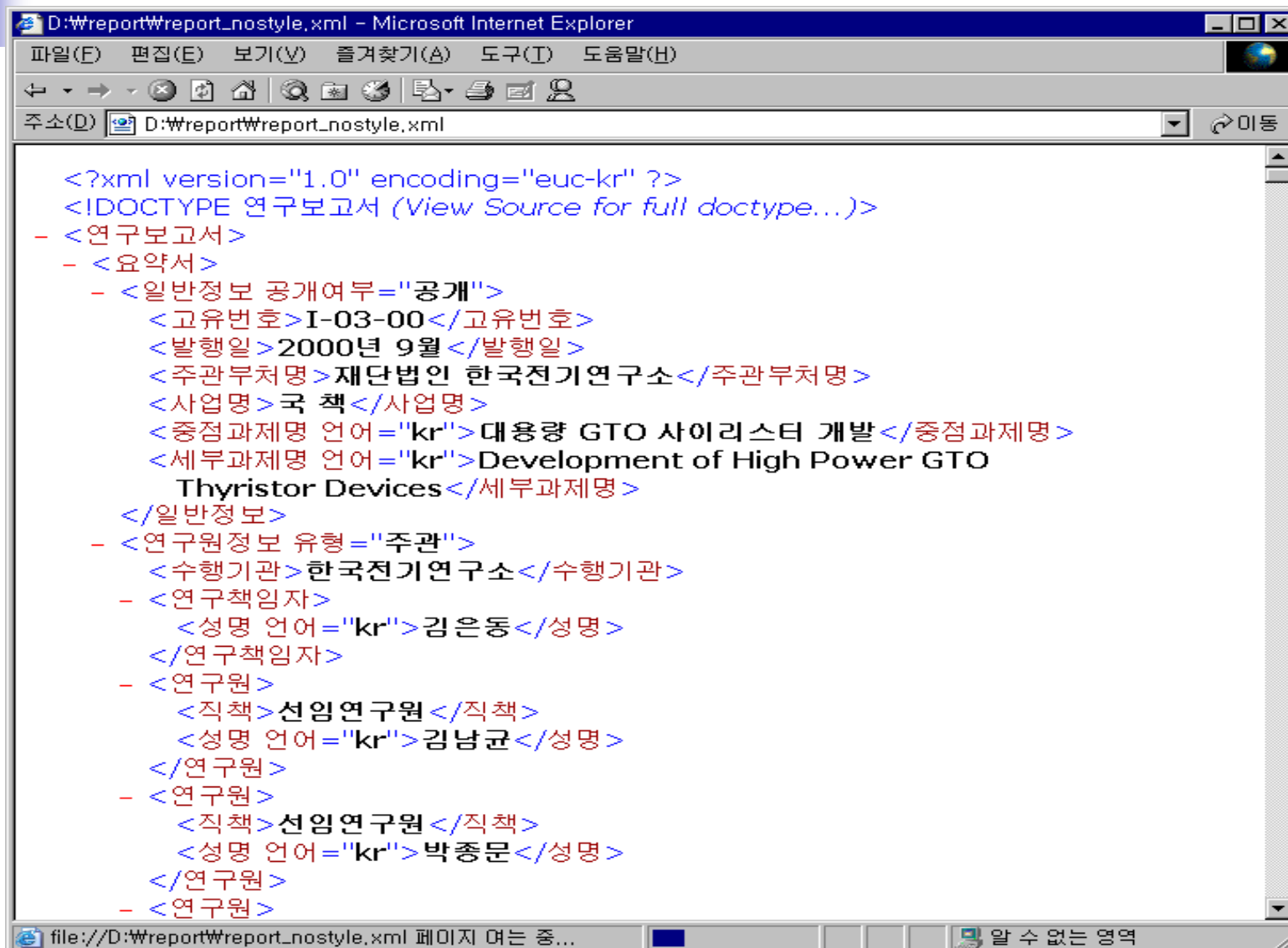
XSLT Making Choices

- two commands of conditionality processing
 - `xsl:if`
 - Simple condition command offer
 - Have one attribute
 - `xsl:choose`
 - When there is plural possibility, one selection
 - `xsl:when` child element offer
- Several StyleSheet union
 - `xsl:import`
 - inclusion with URI value of stylesheet(inclusion with other child element)
 - `xsl:include`
 - Offer with URI of other Stylesheet, top level element

report.xsl



XSL Example



```
<?xml version="1.0" encoding="euc-kr" ?>
<!DOCTYPE 연구보고서 (View Source for full doctype...)>
- <연구보고서>
- <요약서>
  - <일반정보 공개여부="공개">
    <고유번호>I-03-00</고유번호>
    <발행일>2000년 9월</발행일>
    <주관부처명>재단법인 한국전기연구소</주관부처명>
    <사업명>국 책</사업명>
    <중점과제명 언어="kr">대용량 GTO 사이리스터 개발</중점과제명>
    <세부과제명 언어="kr">Development of High Power GTO
      Thyristor Devices</세부과제명>
  </일반정보>
  - <연구원정보 유형="주관">
    <수행기관>한국전기연구소</수행기관>
  - <연구책임자>
    <성명 언어="kr">김은동</성명>
    </연구책임자>
  - <연구원>
    <직책>선임연구원</직책>
    <성명 언어="kr">김남균</성명>
    </연구원>
  - <연구원>
    <직책>선임연구원</직책>
    <성명 언어="kr">박종문</성명>
    </연구원>
  - <연구원>
```

XSL Example

C:\wstyl C:\wstyleWreport C:\wstyleWreport.xml - Microsoft Internet Explorer

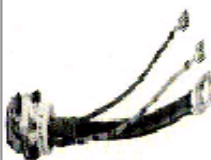
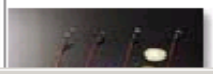
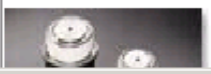
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)

VDRM
VRPM
IT(R
Rth
dV/
r
tg
tgq
Tj(M

(b) 조립구조

그림 1-16. TO-220 및 3P 플라스틱 패키지 계열의 Leadframe과 조립구조

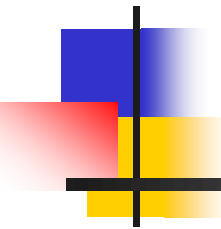
표1-10. 사이리스터의 외형구조에 의한 분류

종 류	정 의	예	
셋기동형	방열체와는 셋기동에 만들어진 나사로 죄어 부착된다.	 (a)	 (b)
평반형	방열체와는 압착하거나 납땜으로 부착하도록 한면이 평면으로 되어있다.	 (c)	 (d)
	방열체와 안착하여 부착		

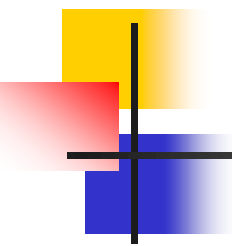
완료 완료 완료

내 컴퓨터

The 10th Korea Internet Conference & Exhibition



Schema



Limitations of XML DTD

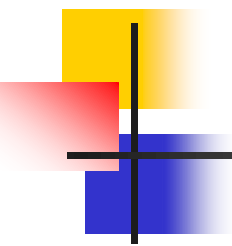
- DTD not extensible
- Only one DTD per document
- Limited support of namespaces
- Weak data typing
- No inheritance
- Document can override an external DTD
- Non-XML syntax
- No (direct) DOM support
- Limited tools

Something better?

- “Schema”, originated in database, means the organization or **structure** of a database
 - Naming of data items
 - Constraints to be applied to data (eg., data typing)
 - Relationships between data items
- **XML-Data Reduced (XDR)** - Microsoft's non-W3C-compliant implementation

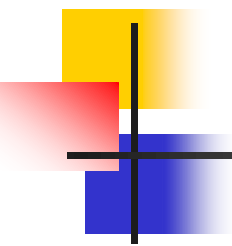
Schema Parts

- XML Schema Part 0: Primer
 - provides an easily approachable description of the XML Schema definition language
- XML Schema Part 1: Structures
 - define the nature of XML schemas and their component parts
 - provide an inventory of XML markup constructs with which to represent schemas
 - define the application of schemas to XML documents.
- XML Schema Part 2: Datatypes
 - extensible datatype system for XML which could be incorporated into XML processors



Schema vs. DTD's

- DTD is weak in data typing
 - `<quantity>hello</quantity>`
- Schemas are XML documents which can be manipulated like other XML documents
 - Valid schemas conform to DTD's
- Schemas have more detailed and robust **content models**
- Schemas are extensible
- Dynamic schemas – can be modified at runtime



XML Schema Features

- Rich datatypes
 - integer, float, date, time, boolean, ...
- User-defined types
- Extendable types
- Open, closed or refinable content models
- Grouping
- Namespace support

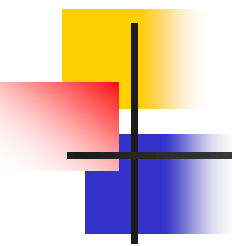
Sample XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="person">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="name"/>
        <xsd:element ref="age" minOccurs='0' maxOccurs='1'/>
        <xsd:element ref="hobby" minOccurs='1' maxOccurs='unbounded'/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="name" type="string"/>
  <xsd:element name="age" type="integer"/>
  <xsd:element name="hobby" type="string"/>
</xsd:schema>
```

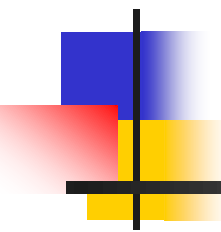
Equivalent DTD

```
<!ELEMENT person (name, age?,
                    hobby+)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT age (#PCDATA)>
<!ELEMENT hobby (#PCDATA)>
```



XML Document

```
<?xml version="1.0" encoding="UTF-8"?>
<person xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="person.xsd">
  <name>Gildong Hong</name>
  <age>30</age>
  <hobby>reading</hobby>
</person>
```



Namespace

Namespace Declarations

- Must declare preferentially to use Namespace
- Declare by Element's attribute. May declare in Root element and may do at child node
- Attribute name beginning xmlns or xmlns:.
- In case use xmlns: , name that come next is prefix and should be involved with URI

```
<x xmlns:edi="http://mie.pcu.ac.kr/namespace">
```

prefix

URI

Prefix Rule

- Is begun by character or 'underline(_) ' and string including character, number, 'dot(.)', 'hyphen(-)', 'underline(_) ' can come

right	xmlns:_my-namespace-name xmlns:한글_namespace xmlns:edi.건설 xmlns:edi.유통
wrong	xmlns: my (space) xmlns:911rescue (number) xmlns:book:kr (:) xmlns:book kr (space)

Default namespace

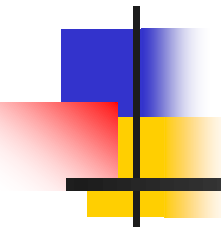
- Because there is no Prefix, is applied on child node whole without can apply namespace in particular element.
- Default namespace is not applied to attribute.

```
<?xml version="1.0"?>
<books xmlns = "http://www.book.co.kr">
  <chapter id = "chap1">
    <title> XML and Webservice</title>
    <body> Book content Good!...</body>
  </chapter>
```

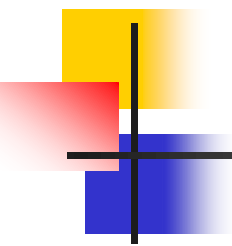
Multi namespace declarations

- Can declare compound namespace in element
- Child element applies necessary namespace selectively

```
<?xml version="1.0"?>
<bk:books xmlns:bk="urn:books"
           xmlns:isbn="urn:ISBN:0-007-23-5">
  <bt:title> XML and Webservice</bt:title>
  <isbn:no> 156739495 </isbn:no>
</bk:book>
```

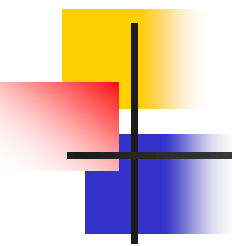


Application



Presentation

- stuff people read or look at
- Presentational Language(for people/applications)
 - MathML -- for mathematics
 - WML -- Wireless WAP-phones
 - SVG -- for graphics
 - VoiceXML -- voice interfaces
 - SMIL -- for multimedia (RealPlayer)
 - XHTML -- new HTML
- metadata
 - for describing things; for use by other software
 - RDF -- Resource Description Framework

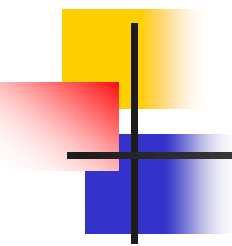


Utilities

- generic XML tools -- XSLT, Schemas,...
 - XSLT -- transform one XML document into another.
 - Schemas -- Define validation rules for a specific type of XML document
 - Can define hierarchical nesting rules for elements, allowed attributes and attribute status (like DTDs)
 - Can define stronger typing constraints on element content, attributes (e.g., Integers, Integer ranges, reals, real ranges, strings, tokens, etc.)

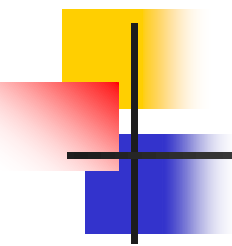
S/W development

- as a software development tool
 - ANT -- XML-based build configuration file (Java applications)
 - beanML -- language specifying composition and state of a Java-bean based application. Processing a BML script results in a running application configured as described in the script.
 - use XML as a tool for managing data inside an application
 - Extract data from databases and store in intermediate XML "object"
 - Useful for application development (sometimes!)



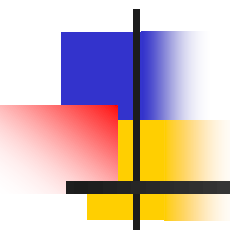
Distributed apps

- distributed application, data delivery, Web services
 - Financial information exchange
 - FpML, FinXML, OFX/IFX, FixML, GOLD, XBRL, SwiftML
 - Directory services metadata
 - dirXML, DSML (Directory Services Markup Language),
 - Other business transactions
 - FRML (first retail markup language), ebXML (generic business)
 - Control of machine-machine applications
 - XML-RPC, SOAP
 - Brokering of Web "Services"
 - Biztalk, UDDI, ebXML



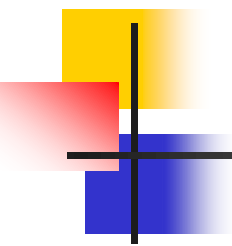
The others apps

- Privacy
 - **P3P**(The Platform for Privacy Preferences)
1.0 - 16 April 2002
 - A P3P Preference Exchange Language 1.0
(APPEL1.0) - 15 April 2002(WD)
- Digital right
 - **XrML**(eXtensible rights Markup Language)



MathML

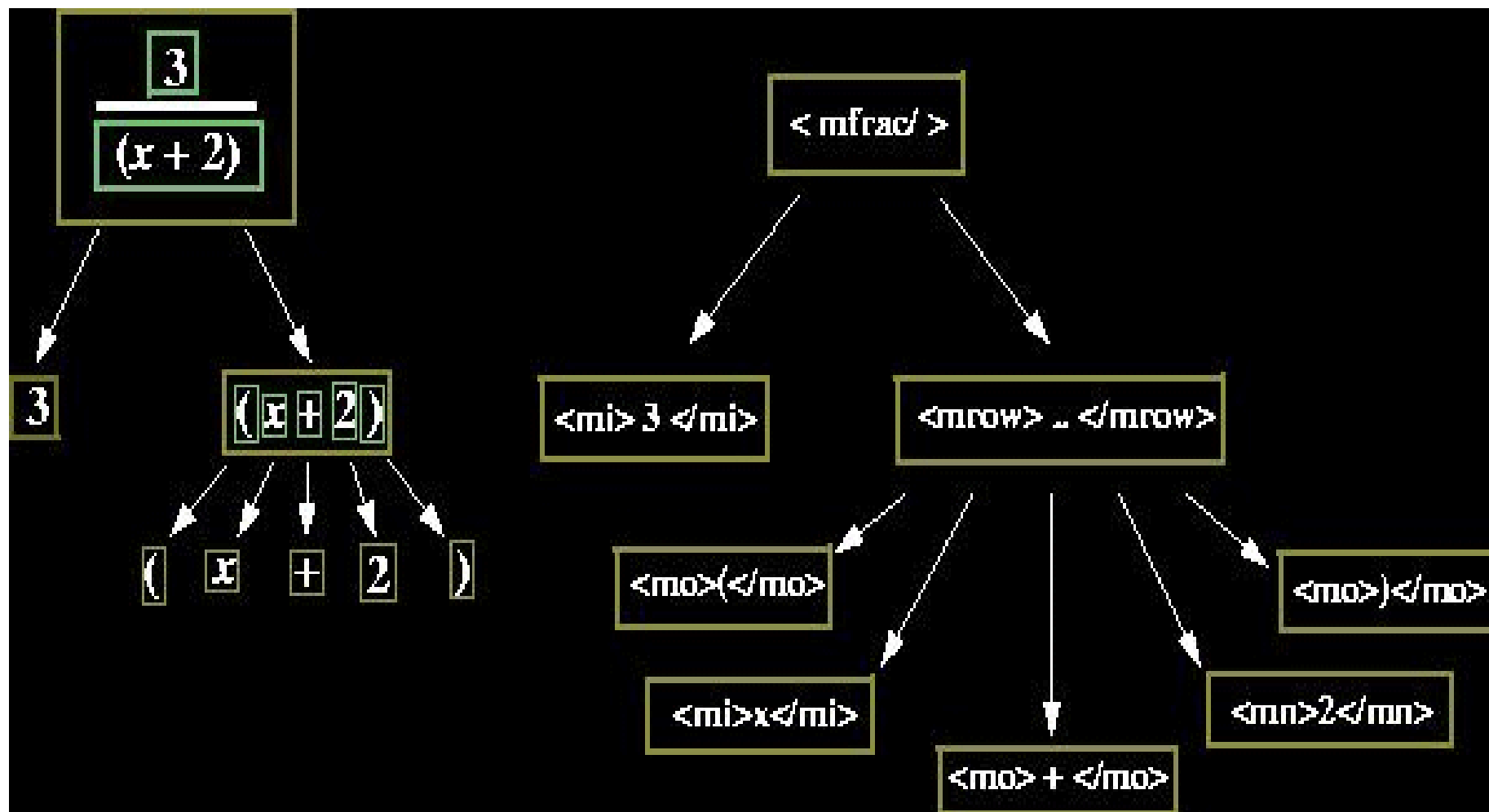
(Mathematical Markup Language)



MathML

- XML application for mathematical equations
- helps both the mathematicians and the theoretical physicists to publish and do research on the web
- Able to record both content and presentation
- Not directly supported (yet) by popular browsers
- Not meant for authoring directly
- Status
 - MathML 1.01 –7 July 1999
 - MathML 2.0 –21 Feb 2001

Math Markup Language



Why MathML? Presentation!

Fractions

$$\frac{x+1}{x+1}$$

Radicals

$$\sqrt{x+1}$$

Grouping Symbols

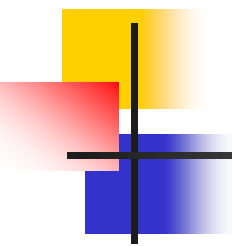
$$\sin\left(\frac{x-1}{2}\right)$$

Complex Expressions

$$\int_0^{\infty} e^{-\frac{x^2}{2}} dx = \sqrt{\frac{\pi}{2}}$$

Why MathML? Universal!

- Computer Algebra Systems
 - Mathematica, Maple, REDUCE
- TeX
 - `\begin{mathml}`
 - `\end{mathml}`
- Web Browsers
 - Netscape, MS Explorer, Amaya, Mozilla, ICE



MathML Elements

■ Presentation

$\sin(2x)$

■ Content

```
<math>  
  <apply>  
    <sin/>  
    <apply>  
      <times/>  
      <cn> 2 </cn>  
      <ci> x </ci>  
    </apply>  
  </apply>  
</math>
```

■ Interface

MathML Tags: Presentation

- Used of Attr, font,color, fontsize, fontweight, background
- Numbers, Identifier and Operators
 - ⁢(⁢), &Invisibla Comma;(⁣), ⁡(⁡)
- Display Function and Marks
 - Square root, fraction number, sup, sub
- Fence, Phantoms
 - (x2, x3)
- Tables

MathML Tags: Presentation

■ Tokens:

mn -- numerals

mi -- variables, symbol
constants, identifiers

mo -- operators, parens.

■ Grouping:

mrow mfence

■ Special Functions:

msup msqrt mfrac

```
<math>
<mrow>
<mi> sin </mi>
<mo>&ApplyFunction;</mo>
<mrow>
<mo> ( </mo>
<mn> 2 </mn>
<mo>&InvisibleTimes;</mo>
<mi> x </mi>
<mo> ) </mo>
</mrow>
</mrow>
</math>
</math>
```

MathML Tags: Content

- Prefix Notation
- `<ci>` and `<cn>` Tags
 - $x+4x$
 - x - identifier:`<ci>`, $+$ - operator, 4 - number`<cn>`
- Basic Functions
 - empty element : `<minus/>`, `<times/>`, `<divide/>`, `<power/>`, `<root/>`, `<sin/>`
- Relations and Declarations
 - `'='`, `'≠'` `<reln>`

MathML Tags: Content

- Calculus
 - differential: <diff/>, integral: <int/>
- Advanced Data Types
 - <set>, <list>, <vector>, <matrix>
- Additional Function & content Attribute
 - <log/>, <sin/>, <cos/>, <tan/>, <median/>, <var/>
 - Attribute: type, definitionURI

MathML Tags: Content

■ Tokens:

cn -- numerals

ci -- variables, symbols,
identifiers

■ Operations

apply -- begins all operators

built-in operators

<plus/>, **<times/>**,
<sin/>, **<eq/>**

```
<math>
  <apply>
    <sin/>
    <apply>
      <times/>
      <cn> 2 </cn>
      <ci> x </ci>
    </apply>
  </apply>
</math>
```

MathML Tools

■ Viewers

- Techexplorer -- browser plug-in
- WebEQ -- browser plug-in, java appl.
- Amaya
- Mozilla
- IE 5.5 Behaviors

■ Evaluation

- Mathematica (version 4.1)
 - presentation
- Maple
- REDUCE
 - presentation and content

MathML Tools

The screenshot displays a web browser window titled "Putting Math on the Web: Netscape/TeXplorer - Microsoft Internet Explorer". The address bar shows the file path: `C:\WINDOWS\Desktop\Presentation\netscape.html`. The page content is divided into two main sections.

Left Section: Examples

- Plain HTML w/ TeXexplorer
- Microsoft Word with Equation Editor
- TeX file converted with tth in DOS
- TeX DVI File
- TeX Source File with TeXexplorer
- Mathematica
- Maple
- WCU Math Forum

Right Section: Sample.nb

The right section shows a sample notebook titled "Sample.nb" with the following content:

We will describe here a method to decide how inaccurate a Riemann sum is *without first knowing the value of the integral*. For example, suppose we estimate the value of

$$\int_0^1 e^{-x^2} dx$$

otherwise known as

$$\frac{1}{2} \sqrt{\pi} \operatorname{erf}(1)$$

using a left endpoint Riemann sum with 1000 equal subintervals.

The notebook also includes the following code snippets:

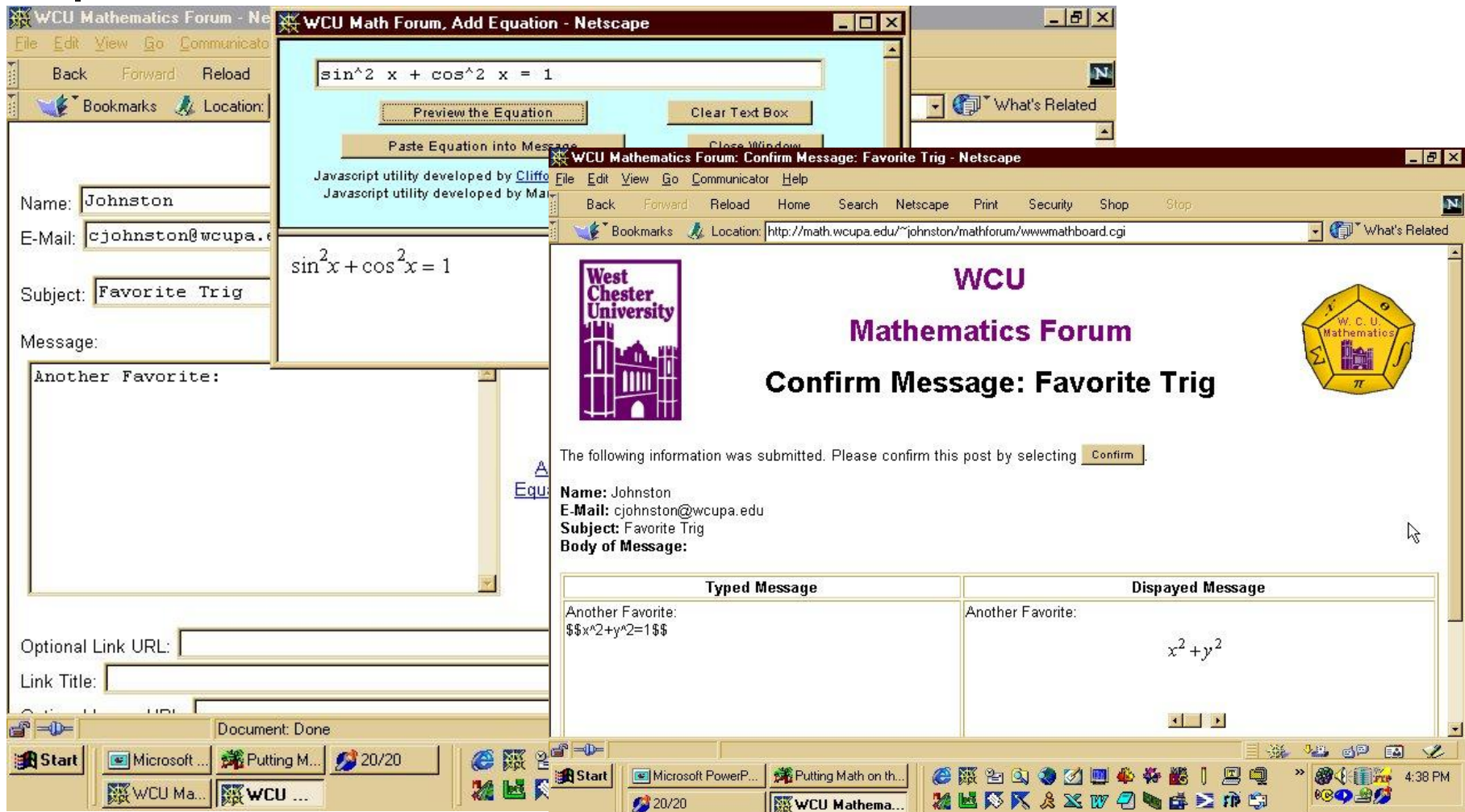
```
> 'int(e^(-x^2),x=0..1)';
```

```
> int(e^(-x^2),x=0..1);
```

MathML Tools

- Output
 - WebEQ
 - Amaya
 - EZMath
 - TtM
 - Mathematica
 - Maple
 - Suzuki Demo

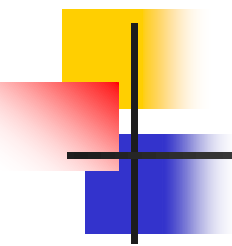
MathML Tools



The screenshot displays a Netscape browser window with the following elements:

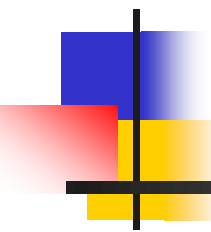
- Address Bar:** <http://math.wcupa.edu/~johnston/mathforum/wwwmathboard.cgi>
- Form Fields:**
 - Name: Johnston
 - E-Mail: cjohnston@wcupa.edu
 - Subject: Favorite Trig
 - Message: Another Favorite: $\sin^2 x + \cos^2 x = 1$
 - Optional Link URL: (empty)
 - Link Title: (empty)
- Preview Window:** Shows the rendered message: $\sin^2 x + \cos^2 x = 1$
- Confirmation Page:**
 - West Chester University Mathematics Forum**
 - Confirm Message: Favorite Trig**
 - The following information was submitted. Please confirm this post by selecting .
 - Submitted Information:
 - Name: Johnston
 - E-Mail: cjohnston@wcupa.edu
 - Subject: Favorite Trig
 - Body of Message:
 - Message Comparison Table:**

Typed Message	Displayed Message
Another Favorite: \$\$x^2+y^2=1\$\$	Another Favorite: x^2+y^2



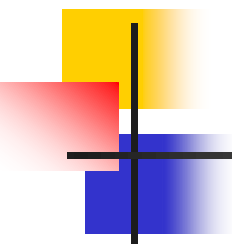
MathML Tools

- Creating Pages
 - TtM
 - LaTeX2MathML
 - WebEQ Page Wizard
 - Plain2MathML
 - WCU Math Forum
 - MakeMathML



SVG

(Scalable Vector Graphics)



Problems with Current Web Graphics

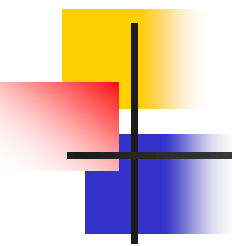
- Main formats
 - gif, png, jpeg
 - Fixed resolution images at pixel level
- Binary format
 - Contents is not searchable
 - Neither ideal for humans nor for search engines
 - No or few metadata
 - Limited animation possibilities
 - Difficult to “combine” with other content

SVG – A New Common Solution

- SVG (Scalable Vector Graphics)
 - All the major players participated
 - PGML (Precision Graphics Markup) – Adobe
 - VML (Vector Markup Language) – Microsoft
- Status
 - Under development since 1998
 - Version 1.0 Recommendation Sep, 2001
 - Version 1.1 Candidate Recommendation April 2002
 - Supported by
 - Adobe Plug-in Version 3.0 Nov 2001
 - IBM Plug-in
 - Microsoft Internet Explorer, Office XP
 - Macromedia Flash
 - Apple Quicktime, Corel

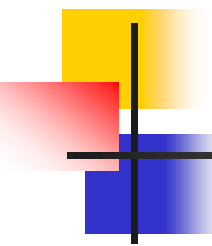
What is SVG ?

- SVG is a vector graphics format
 - Resolution independent
- Defined in XML
 - Searchable
 - Metadata can be added
 - Can be combined with other XML infrastructure
- Animation facilities
 - Borrowed from multimedia presentation community
 - Direct access to DOM(Domain Object Model) from scripts
- Hyperlinking
 - Every graphics element can be an active hyperlink



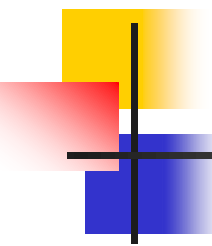
Vector Graphics Format

- Objects
 - Paths(vector graphic shapes)
 - Images
 - Text
- Operations
 - Grouping, styling
 - Transformations(nested), compositing
 - Clipping paths, alpha masks
 - Filter effects
 - Template objects
 - Declarative animations
 - Full embedded scripting



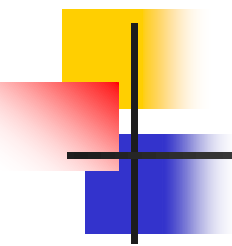
Interactive & Dynamic Objects

- Animations can be defined and triggered
 - Scripted animations
 - Access to SVG Document Object Model (DOM)
 - Rich set of event handlers
 - ECMAScript
 - JavaScript
 - Visual Basic
 - Declarative animations
 - Embed SVG animation elements



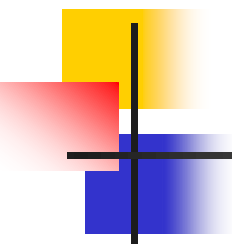
SVG features

- Plain text format
 - SVG files can be read and modified by a range of tools
 - Files are smaller and more compressible than comparable JPEG or GIF images
- Scalable
 - Unlike bitmapped GIF and JPEG formats, SVG is a vector format
 - SVG images can be printed with high quality at any resolution
 - Without “jaggies” you see when printing bitmapped images
- Zoomable
 - You can zoom in on any portion of an SVG image and not see any degradation
 - You can search for specific text strings, like city names in a map



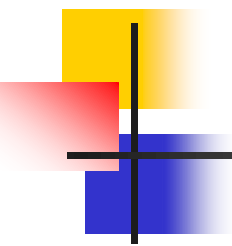
SVG features

- Scripting and animation
 - SVG enables dynamic and interactive graphics far more sophisticated than bitmapped or even Flash™ images
- Works with Java™ technology
 - SVG complements Java™ technologies' high end graphics engine
 - Java 2D API
 - Graphics2D SVG Generator
- Open standard
 - SVG is an open recommendation developed by a cross-industry consortium
 - Unlike some other graphics formats (e.g. Flash), SVG is not proprietary



Potential Impact

- Potentially huge impact
 - Common Web format
 - Common typesetting format
- Common format for
 - Web browsers
 - Typesetting tools
 - Publishing tools
- Potentially replaces
 - HTML, DHTML (all version including dynamic)
 - PostScript, PDF
 - PNG, GIF
 - RTF
 - Flash
 - Many others



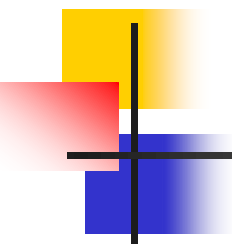
SVG advantages

- Compatibility with other mediums such as wireless devices
- Scalable Server Solutions
- Small file sizes for faster Web page downloads
- Unlimited color and font choices
- Zoomable graphics and images
- Scripting control for custom interactive events and animation
- Clean, crisp, high-resolution printing from Web browsers
- Bitmap-style filter effects for high-impact graphics
- Text-based format easily integrates with other Web technologies
- Built in International Language Support
- Reduced Maintenance Costs
- Easily Updated
- Rich Multimedia Capabilities

SVG Document structure

- Syntax
 - Hierarchical XML tags
 - Case sensitive
- Document structure
 - XML version tag
 - SVG DTD tag
 - SVG tag define size and namespace

```
<?xml version="1.0" standalone="no"?>  
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"  
    "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">  
<svg width="5cm" height="4cm"  
    xmlns="http://www.w3.org/2000/svg">  
    "graphics elements"  
</svg>
```



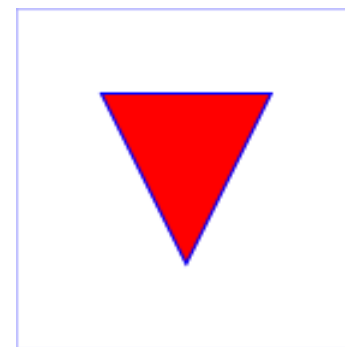
Paths

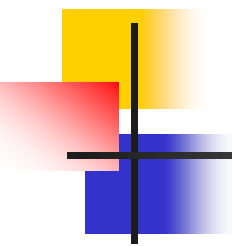
- Paths represent the outline of a shape
 - Filled, stroked, used as a clipping path
 - Any combination of the three
- Concepts
 - Current point (pen location)
 - Dragging pen
 - Open, closed paths
- Just like PostScript and PDF

Path command

- Attributes
 - d, fill, stroke, stroke-width
- M 100 100
 - Absolute move to position(100,100)
- L
 - Absolute line to position (300,100)
- Z
 - Close path
- Examples

```
<path
d="M 100 100 L 300 100 L 200 300 z"
fill="red"
stroke="blue"
stroke-width="3"/>
```



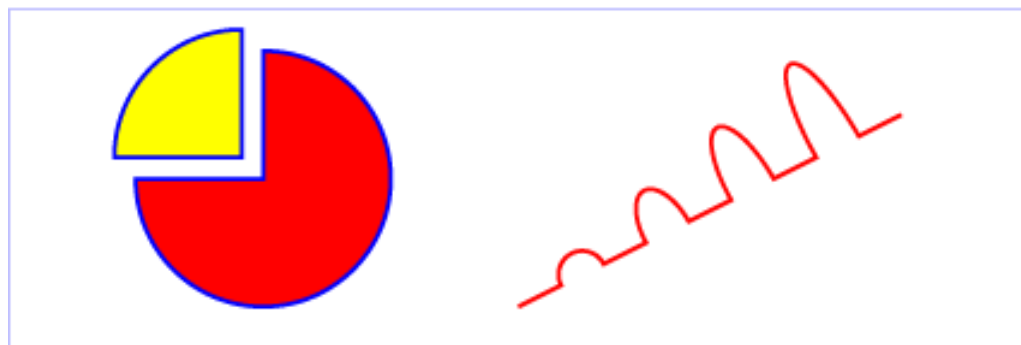


Coordinate system and units

- The initial coordinate system
 - Origin at the top/left corner
 - X-axis pointing to the right
 - Y-axis pointing down
- The initial user coordinate system has one user unit equal to the parent(implicit or explicit)user agent's "pixel"

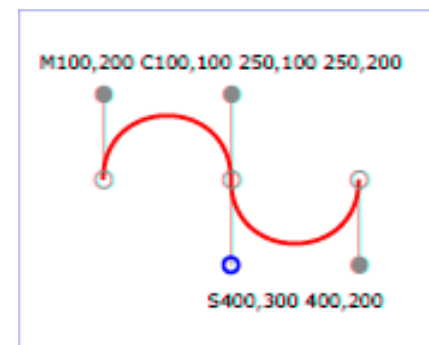
Arc example

```
<path d="M300,200
  h-150
  a150,150 0 1,0 150,-150 z"
  fill="red" stroke="blue" stroke-width="5" />
<path d="M275,175
  v-150
  a150,150 0 0,0 -150,150 z"
  fill="yellow" stroke="blue" stroke-width="5" />
<path d="M600,350
  l 50,-25 a25,25 -30 0,1 50,-25
  l 50,-25 a25,50 -30 0,1 50,-25
  l 50,-25 a25,75 -30 0,1 50,-25
  l 50,-25 a25,100 -30 0,1 50,-25
  l 50,-25"
  fill="none"
  stroke="red" stroke-width="5" />
```



Bézier curves

```
<polyline class="Connect" points="100,200 100,100" />
<polyline class="Connect" points="250,100 250,200" />
<polyline class="Connect" points="250,200 250,300" />
<polyline class="Connect" points="400,300 400,200" />
<path class="SamplePath" d="M100,200 C100,100 250,100 250,200
    S400,300 400,200" />
<circle class="EndPoint" cx="100" cy="200" r="10" />
<circle class="EndPoint" cx="250" cy="200" r="10" />
<circle class="EndPoint" cx="400" cy="200" r="10" />
<circle class="CtlPoint" cx="100" cy="100" r="10" />
<circle class="CtlPoint" cx="250" cy="100" r="10" />
<circle class="CtlPoint" cx="400" cy="300" r="10" />
<circle class="AutoCtlPoint" cx="250" cy="300" r="9" />
<text class="Label" x="25" y="70">M100,200 C100,100 250,100
250,200</text>
<text class="Label" x="325" y="350"
    style="text-anchor:middle">S400,300 400,200</text>
```



Basic shapes:rect

- Attribute

- x, y, width, height, fill, stroke, stroke-width, rx, ry

- Example

```
<rect x="100" y="100" width="400" height="200" rx="50"  
fill="green" />
```

```
<g transform="translate(700 210) rotate(-30)">
```

```
<rect x="0" y="0" width="400" height="200" rx="50"  
fill="none" stroke="purple" stroke-width="30" />
```

```
</g>
```



Basic shapes:circle

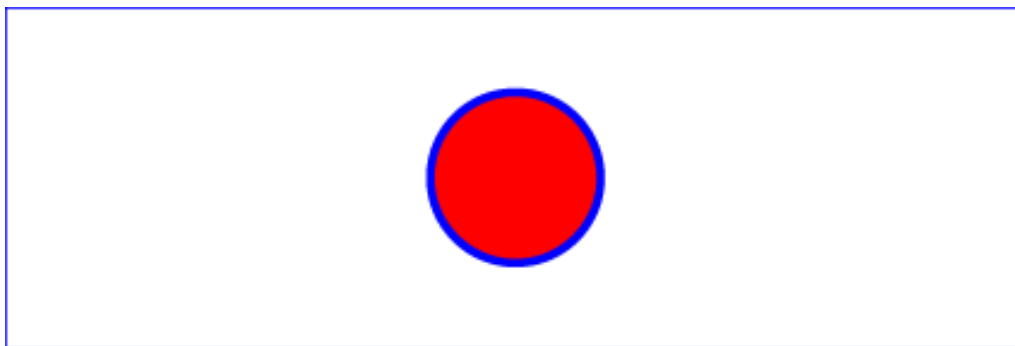
- Attribute

- cx, cy, r, fill, stroke, stroke-width

- Example

```
<rect x="1" y="1" width="1198" height="398" fill="none"  
stroke="blue" stroke-width="2"/>
```

```
<circle cx="600" cy="200" r="100" fill="red" stroke="blue"  
stroke-width="10" />
```



Basic shapes:ellipse

- Attribute

- cx, cy, rx, ry, fill, stroke, stroke-width

- Example

- `<g transform="translate(300 200)">`
`<ellipse rx="250" ry="100" fill="red">`
`</g>`
 - `<ellipse transform="`
`"translate(900 200) rotate(-30)"`
`rx="250" ry="100" fill="none"`
`stroke="blue" stroke-width="20" ,`



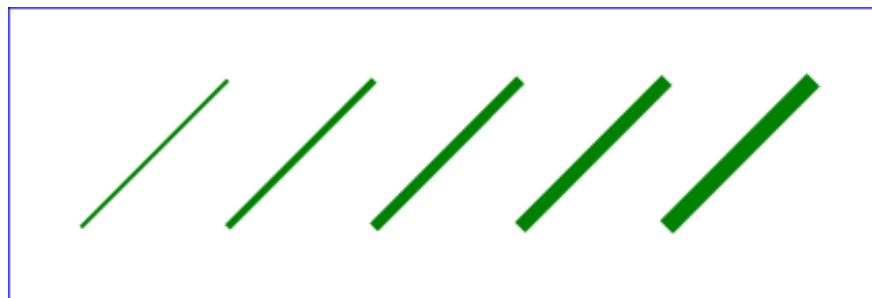
Basic shapes:line

- Attribute

- x1, y1, x2, y2,stroke-width

- Example

- ```
<g stroke="green" >
 <line x1="100" y1="300" x2="300" y2="100" stroke-width="5" />
 <line x1="300" y1="300" x2="500" y2="100" stroke-width="10" />
 <line x1="500" y1="300" x2="700" y2="100" stroke-width="15" />
 <line x1="700" y1="300" x2="900" y2="100" stroke-width="20" />
 <line x1="900" y1="300" x2="1100" y2="100" stroke-width="25" />
</g>
```



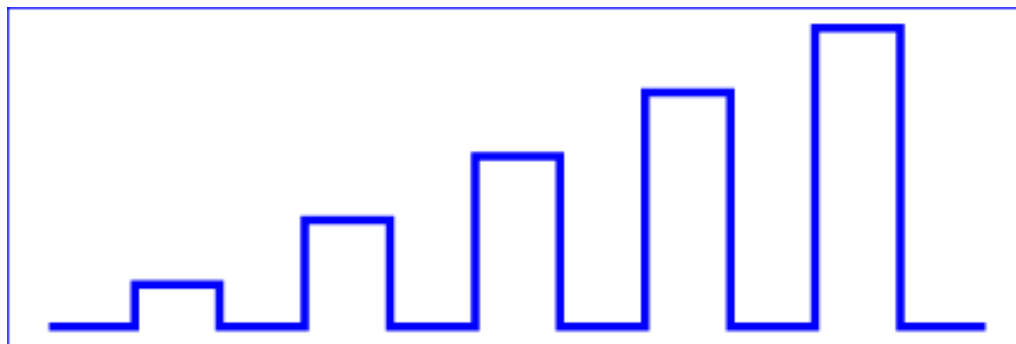
# Basic shapes:polyline

- Attribute

- points, fill, stroke, stroke-width

- Example

- `<polyline fill="none" stroke="blue" stroke-width="10" points="50,375 150,375 150,325 250,325 250,375 350,375 350,250 450,250 450,375 550,375 550,175 650,175 650,375 750,375 750,100 850,100 850,375 950,375 950,25 1050,25 1050,375 1150,375" />`



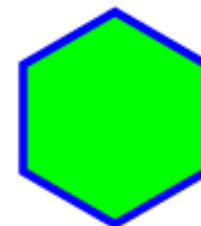
# Basic shapes: polygon

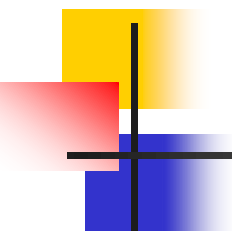
- Attribute

- points, fill, stroke, stroke-width

- Example

- `<polygon fill="red" stroke="blue" stroke-width="10" points="350,75 379,161 469,161 397,215 423,301 350,250 277,301 303,215 231,161 321,161" />`
- `<polygon fill="lime" stroke="blue" stroke-width="10" points="850,75 958,137.5 958,262.5 850,325 742,262.6 742,137.5" />`



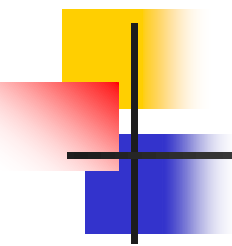


# Text

---

- A string of text at a given coordinate
- Tspan define a subspan for changing style
- Just like paths and shapes, text is subject to
  - Coordinate system transformations
  - Painting
  - Clipping
  - Masking





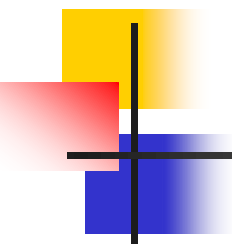
# Text

---

- Attribute
  - x, y, font-family, font, font-weight, fill
- Example
  - `<text x="250" y="150"`  
font-family="Verdana" font-size="55" fill="blue" >  
Welcome! KRnet 2002  
</text>



Welcome! KRnet 2002



# Text:tspan

---

- Within a 'text' element
- can be adjusted with absolute or relative coordinate values
  - Text
  - font properties
  - current text position

# Text:tspan

- Attribute

dx, dy, font-family, font, font-weight, fill

- Example

```
<g font-family="Verdana" font-size="45" >
```

```
 <text x="200" y="150" fill="blue" >
```

You are

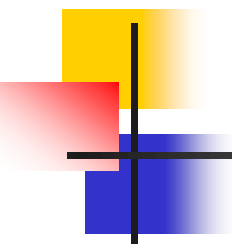
```
 <tspan font-weight="bold" fill="red" >not</tspan>
```

a banana.

```
 </text>
```

```
</g>
```

You are **not** a banana. But you **are** a peach! Cute and fuzzy



# Text decoration

---

- describes decorations that are added to the text of an element.
- Some viewers(Conforming SVG Viewers) are not required to support the blink value.
  - Value: none | [ underline || overline || line-through || blink ] | inherit
  - Initial: none
  - Applies to: text content elements

# Text decoration

- Example

```
<g font-size="60" fill="blue" stroke="red" stroke-width="1" >
<text x="100" y="75">Normal text</text>
<text x="100" y="165" text-decoration="line-through" >Text with line-
through</text>
<text x="100" y="255" text-decoration="underline" >Underlined text</text>
<text x="100" y="345" text-decoration="underline" >
<tspan>One </tspan>
<tspan fill="yellow" stroke="purple" >word </tspan>
<tspan fill="yellow" stroke="black" >has </tspan>
<tspan fill="yellow" stroke="darkgreen" text-decoration="underline" >different
</tspan>
<tspan fill="yellow" stroke="darkgreen" text-decoration="underline" >
</text>
</g>
```

Normal text

~~Text with line-through~~

Underlined text

One word has different underlining

# Text on a path

- textPath element within text to place text along a path
- Example

```
<defs>
```

```
<path id="MyPath" d="M 100 200 C 200 100 300 0 400 100 C
500 200 600 300 700 200 C 800 100 900 100 900 100" />
```

```
</defs>
```

```
<use xlink:href="#MyPath" fill="none" stroke="red" />
```

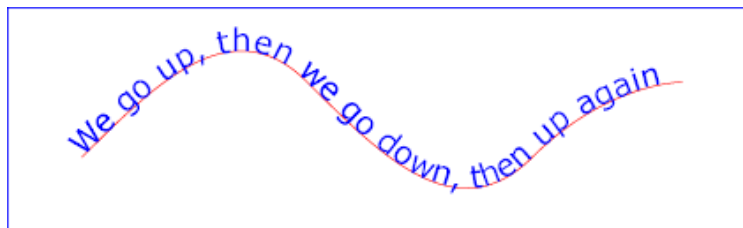
```
<text font-family="Verdana" font-size="42.5" fill="blue" >
```

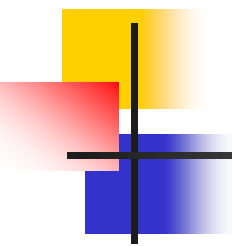
```
<textPath xlink:href="#MyPath">
```

We go up, then we go down, then up again

```
</textPath>
```

```
</text>
```





# Scripting:Script

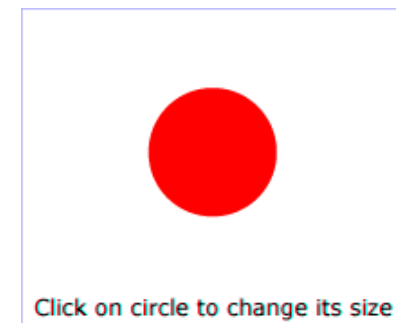
---

- A '**script**' element is equivalent to the 'script' element in HTML
- the place for scripts (e.g., ECMAScript). Any functions defined within any 'script' element have a "global" scope across the entire current document.

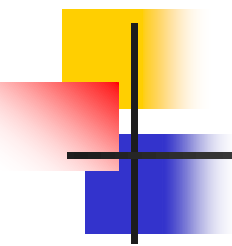
# Scripting:Script

- Exmample

```
<script type="text/ecmascript"> <![CDATA[
 function circle_click(evt)
 {
 var circle = evt.target;
 var currentRadius = circle.getAttribute("r");
 if (currentRadius == 100)
 circle.setAttribute("r", currentRadius*2);
 else
 circle.setAttribute("r", currentRadius*0.5);
 }
]]> </script>
<circle onclick="circle_click(evt)" cx="300" cy="225" r="100"
 fill="red"/>
```



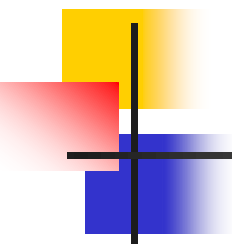




# Animation

---

- SVG supports the ability to change vector graphics over time.
- Using SVG's animation elements
  - define motion paths, fade-in or fade-out effects, and objects that grow, shrink, spin or change color.
- designed to allow future versions of SMIL



# Animation

---

- **animate** element
  - allows scalar attributes and properties to be assigned different values over time
- **animateMotion** element
  - moves an element along a motion path
- **animateColor** element
  - modifies the color value of particular attributes or properties over time
- **animateTransform** element
  - modifies one of SVG's transformation attributes over time, such as the transform attribute

# Animation example

- Example

```
<path d="M100,250 C 100,50 400,50 400,250" fill="none"
stroke="blue" stroke-width="7.06" />
```

```
<circle cx="100" cy="250" r="17.64" fill="blue" />
```

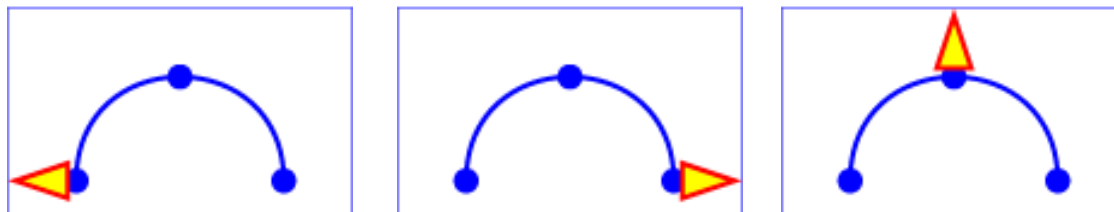
```
<circle cx="250" cy="100" r="17.64" fill="blue" />
```

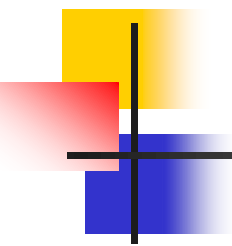
```
<circle cx="400" cy="250" r="17.64" fill="blue" />
```

```
<path d="M-25,-12.5 L25,-12.5 L 0,-87.5 z" fill="yellow" stroke="red"
stroke-width="7.06" >
```

```
<!-- Define the motion path animation -->
```

```
<animateMotion dur="6s" repeatCount="indefinite" path="M100,250 C
100,50 400,50 400,250" rotate="auto" /> </path>
```



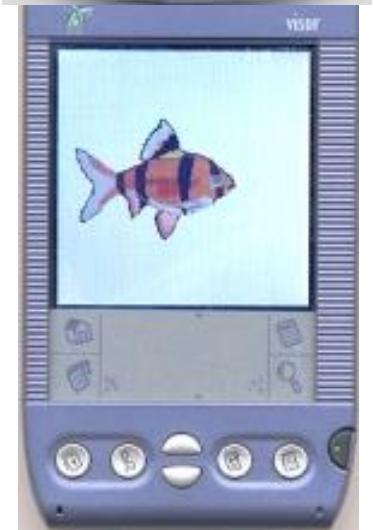


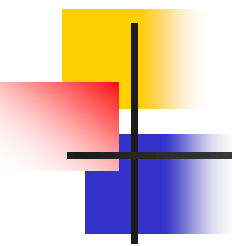
# SVG Tools

---

- SVG Viewers
  - Adobe SVG Viewer
    - Browser Plug-in
  - Apache Batik SVG browser
    - based on Java2
  - Bitflash SdVG viewer
    - subset of SVG suitable for mobile devices
  - CSIRO Pocket SVG Viewer
    - low memory footprint
    - For Pocket PC

# SVG Tools



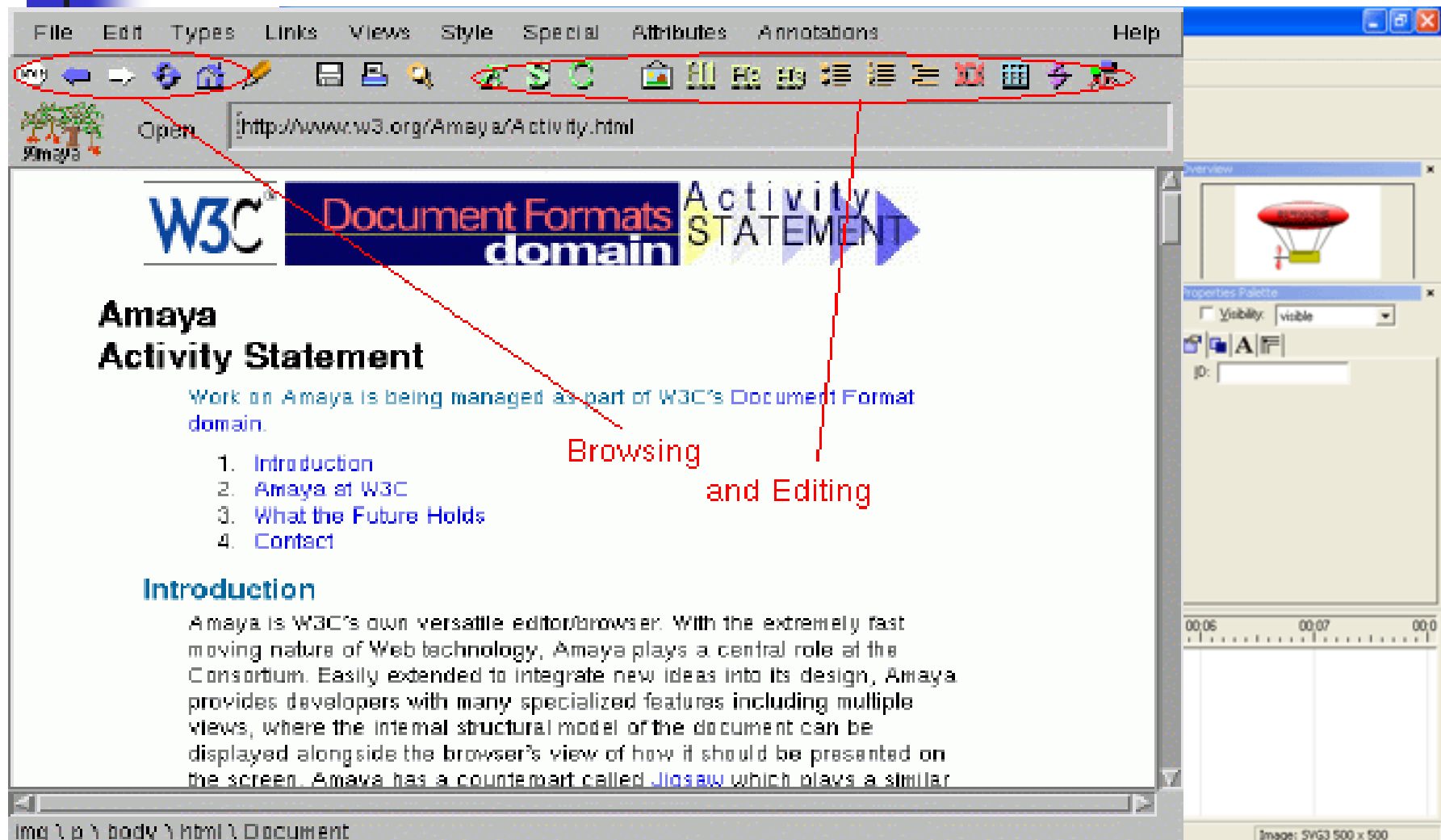


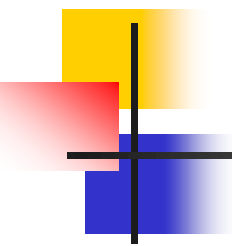
# SVG Tools

---

- SVG Editor
  - W3C Amaya
    - browser/editor for SVG and for mixed namespace XML, XHTML, SVG, and MathML
  - JASC WebDraw
    - WYSIWYG visual editing
    - both import and export of SVG,
  - various authors Sodipodi
    - small vector based drawing program
    - Both vector and bitmap objects can be arbitrarily transformed

# SVG Tools



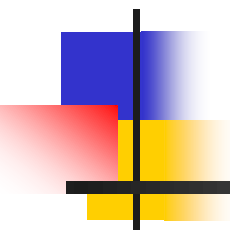


# SVG Tools

---

- SVG Convertor
  - Appligent PDFML Publisher
    - Converts from SVG to PDF
    - server-side SVG converter
  - Celinea CR2V
    - Input formats are BMP, TIFF, PNG, JPEG and GIF. CR2V
  - Gardos Software gsDXF2SVG.dll
    - ActiveX DLL that converts AutoCAD DXF files to SVG
  - Oliver Dietzel JPEG to SVG encoder
    - convert JPEG files to SVG outlines





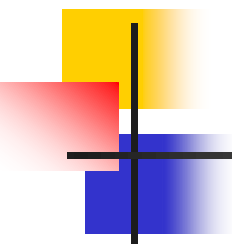
# **VoiceXML**

**(Voice Extensible Markup Language)**

---

# What is Voice XML?

- Language for specifying voice dialogs
- Extension of XML
- Designed to interact with web-based applications
- Integration of Multiple Speech and Telephony Related Technologies
  - Automated Speech Recognition (ASR)
  - Text-to-Speech Synthesis (TTS)
  - DTMF
  - Interactive Voice Response (IVR)



# Goals & Scope

---

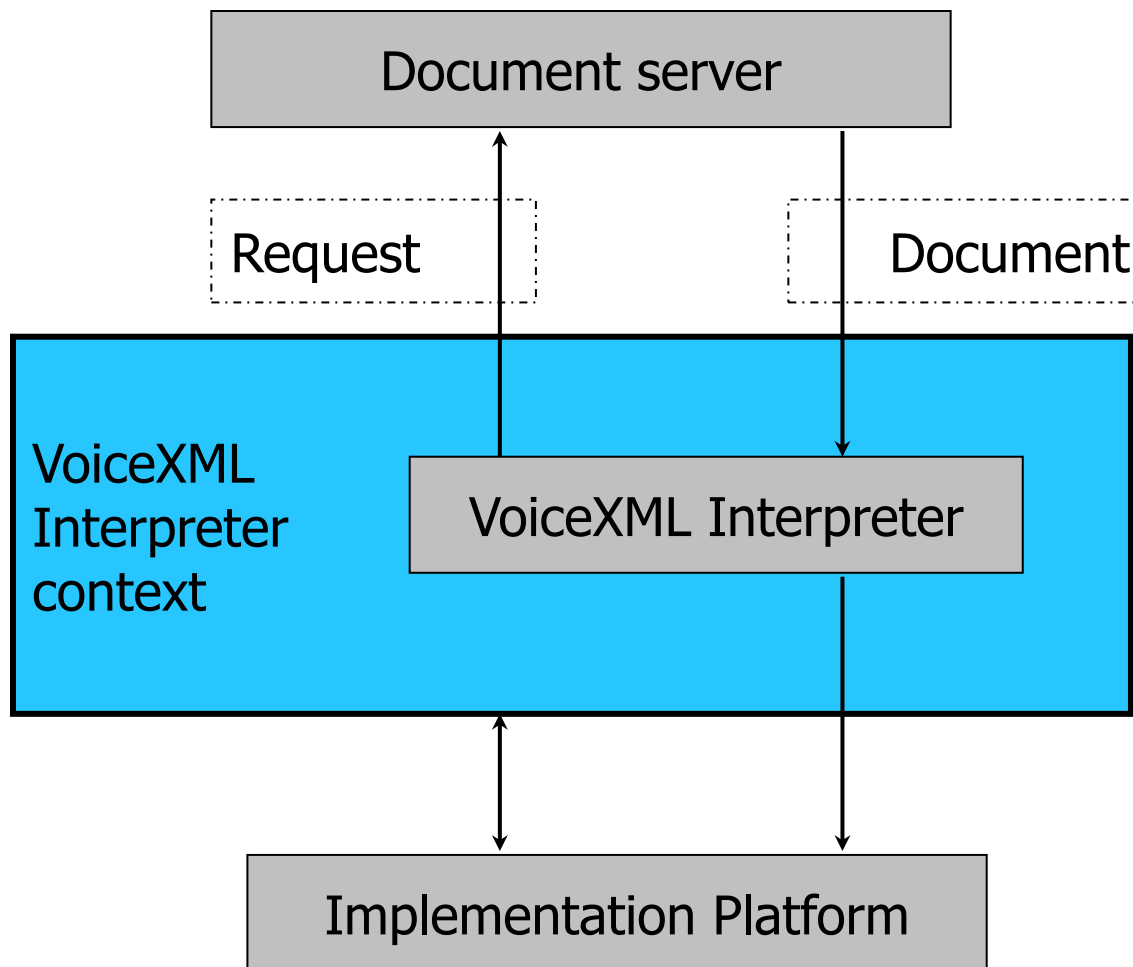
## ■ Goals

- Full power of web development and content delivery to voice response application
- Separation from low-level programming and resource management

## ■ Scope

- Synchronized speech output
- Output audio file
- Recognition of spoken, DTMF input
- Recoding of spoken input
- Control of dialog flow
- Telephony features such as call transfer and disconnect

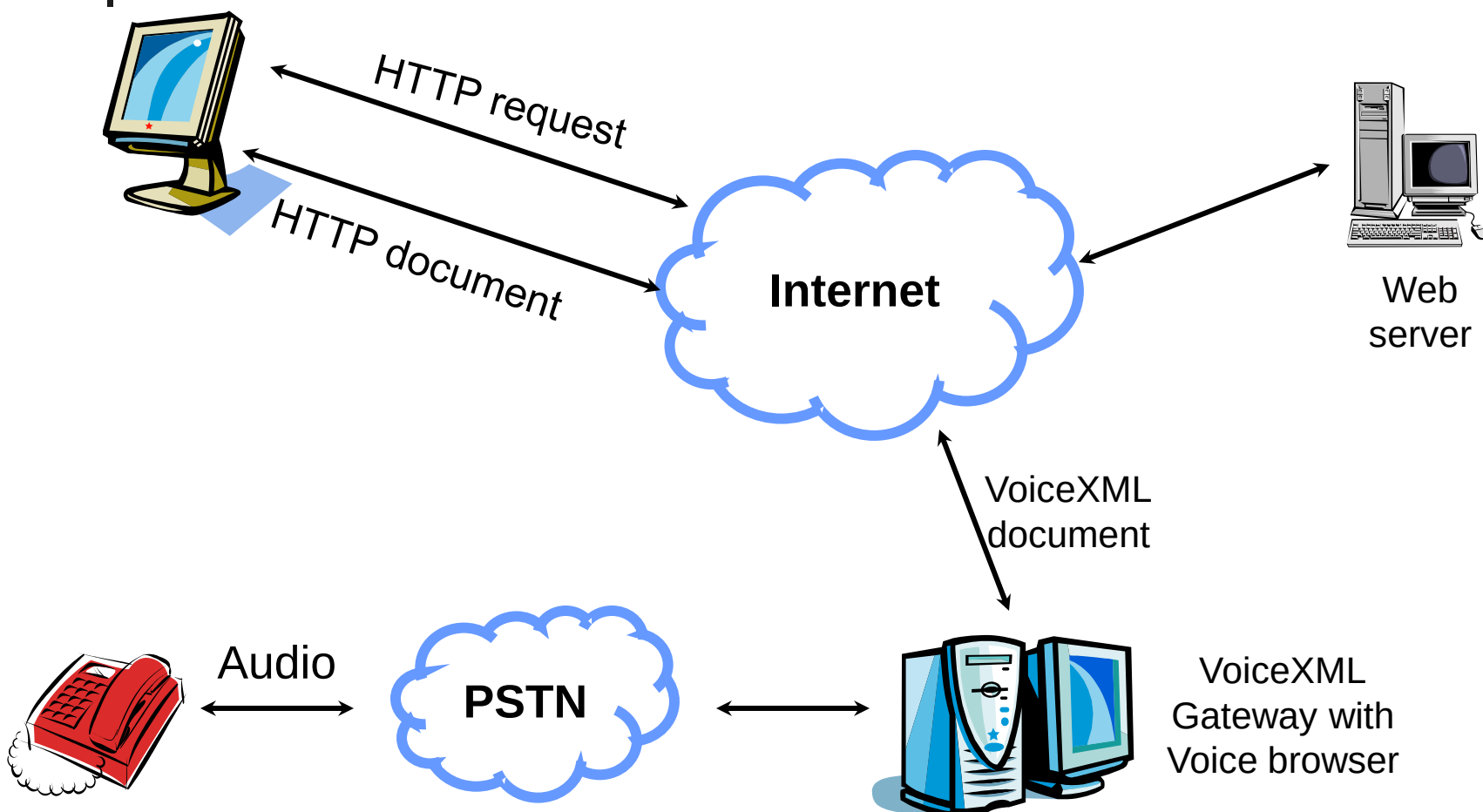
# Architecture Model

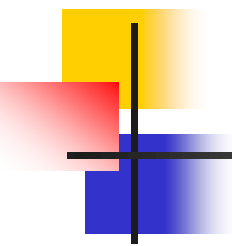


# Architecture Model

- Document server
  - Processes requests received from the VoiceXML Interpreter and responds with VoiceXML documents.
- VoiceXML Interpreter
  - Interprets the VoiceXML documents it receives from the document server.
- Implementation platform
  - Controlled by the VoiceXML Interpreter context and VoiceXML Interpreter
  - the implementation platform generates events in response to user actions and system events
  - The VoiceXML Interpreter context and VoiceXML Interpreter then handles the events.

# Gateway within network

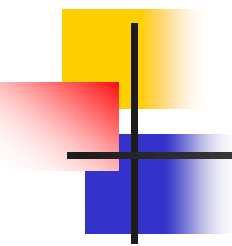




# Concept

---

- VoiceXML document
  - Conversational finite state machine
- Dialog and Subdialog
  - Form
    - Define an interaction that collections values from a set of field item variable
  - Menus
    - Present the user with choice of options
  - Subdialog
    - Mechanism fo invoking a new interaction and returning to the original form

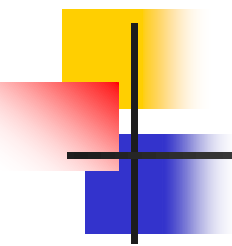


# Concept

---

- Session
  - begins once the user begins to interact with a VoiceXML document.
- Application
  - a collection of VoiceXML documents.
  - All the documents in an application share the same application root document
- Grammars
  - specifies a list of permissible vocabulary for the user
  - Each dialog has one or more speech and/or grammars associated with it.





# Concept

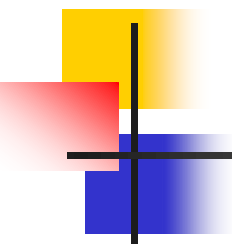
---

## ■ Events

- thrown by the VoiceXML platform for a number of reasons
- such as when a user does not respond to an input, doesn't respond correctly, requests help, etc

## ■ Link

- specifies a transition that is common to all dialogs in the scope of the link
- Support mixed initiative
- When a user input matches the link's grammar, control transfers to the link's destination



# Document structure

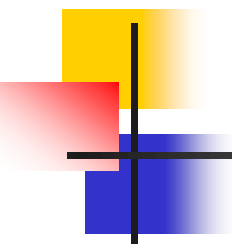
---

- Execution within document
- Execution a multi-document application
  - Root document and subdocument
  - In case where you want multiple documents to work together as one application
  - Benefits
    - Information sharing
    - Use can always interact with common forms, links, menus
- Subdialog
  - Mechanism for decomposing complex sequences of dialogs to create reusable components

# Dialogs

- Documents are composed of dialogs
- Two top-level dialog elements
  - Forms: `<form>`
  - Menus: `<menu>`
- Extra top-level elements

<code>&lt;meta&gt;</code>	Metadata items in name/value pairs
<code>&lt;var&gt;</code>	Variable declaration
<code>&lt;script&gt;</code>	ECMAScript client side scripting
<code>&lt;catch&gt;</code>	Catch an event
<code>&lt;link&gt;</code>	Transition common to all dialogs in scope

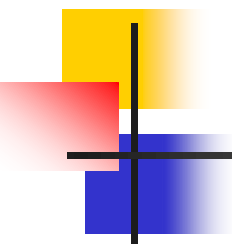


# Forms

---

- Discrete dialog elements
  - Denoted by <form> tag
  - Optional "id" attribute to specify name
  - Responsible for executing some part of the dialog
- Forms contain various elements
  - Elements perform tasks required by form
  - Several, among which those known as "form items"

```
<form id="welcome">
 ..elements..
</form>
```



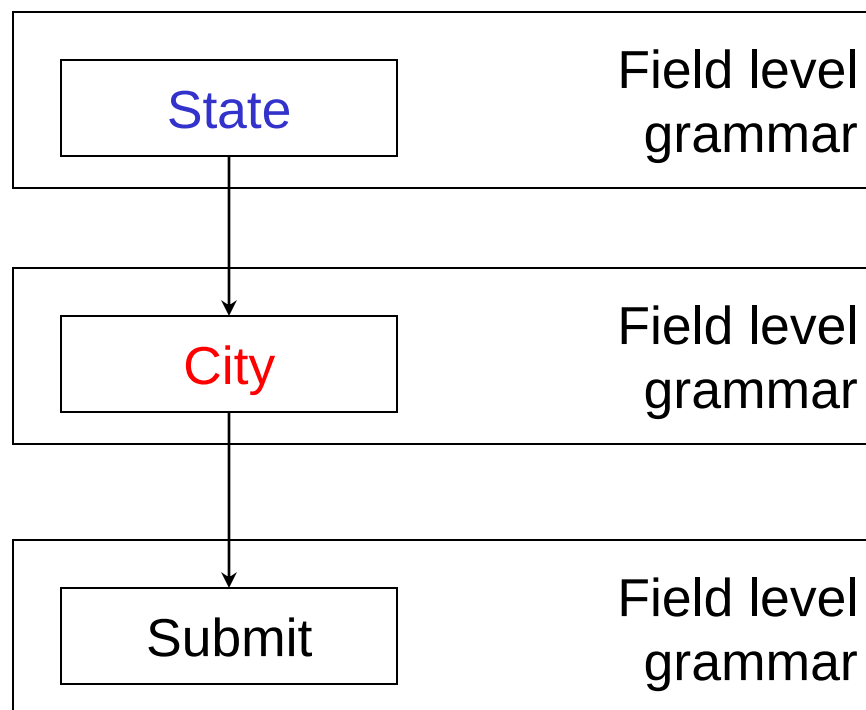
# Form Types

---

- Directed Forms
  - Form items are executed exactly once in sequential order
  - Implement a computer-directed interaction

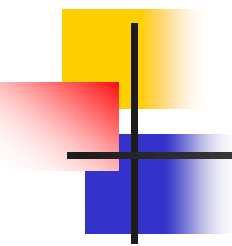
# Form Types(Directed Form)

## Form



# Form Types(Directed Form)

```
<form id="weather_info">
 <block> Welcome to the weather information service! </block>
 <field name="state">
 <prompt> What state? </prompt>
 <grammar src="state.gram" type="application/x-jsgf"/>
 <catch event="help">
 Please speak the state for which you want the weather.
 </catch>
 </field>
 <field name="city">
 <prompt> What city? </prompt>
 <grammar src="city.gram" type="application/x-jsgf"/>
 <help>
 Please speak the city for which you want the weather.
 </help>
 </field>
 <block>
 <submit next="/servlet/weather" namelist="city state"/>
 </block>
</form>
```



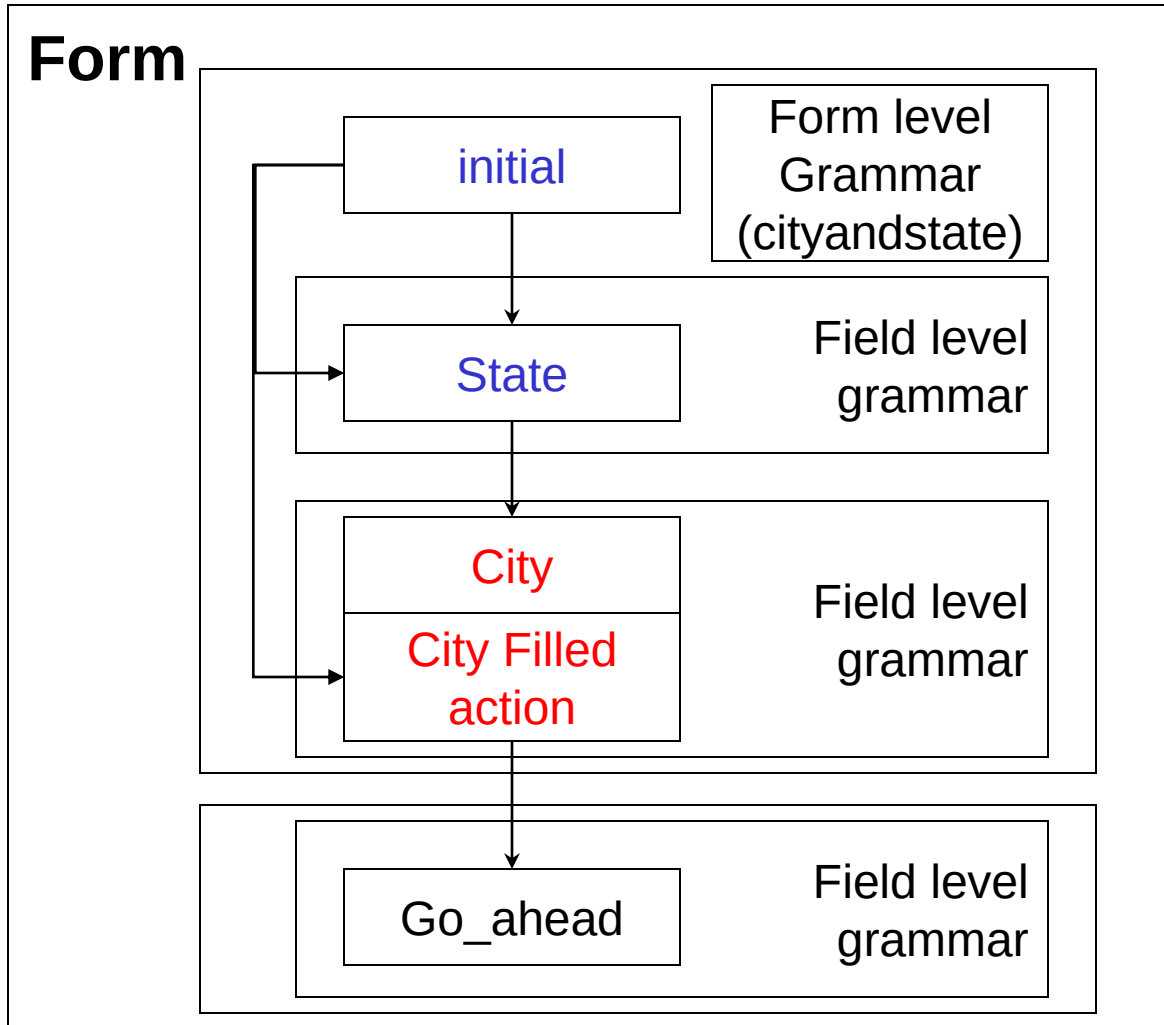
# Form Types

---

- Mixed Initiative Forms
  - Both the computer and the human direct the conversation
  - Has one or more <initial> form items
  - Has one or more **form-level grammars**
    - Fields can be filled in any order
    - More than one fields can be filled by a single utterance



# Form Types(Mixed Initiative Forms)



# Form Types(Mixed Initiative Forms)

```
<form id="weather_info">
 <grammar src="state.gram"
 type="application/x-jsgf"/>
 <block> Welcome to the weather
 information service!
</block>
 <initial name="start">
 <prompt>For what city and state do
 you want the weather? </prompt>
 <noinput>
 <reprompt/>
 <assign name="start"
expr="true"/>
 </noinput>
 </initial>
 <field name="state">
 . . . Field Elements . . .
 </field>
 <field name="city">
 . . . Field Elements . . .
 </field>
 <block>
 <submit next="/servlet/weather"
 namelist="city state" />
 </block>
</form>
```

- Welcome to the weather information service. For what city and state do you want the weather?
- (silence)
- Welcome to the weather...

# Form Types(Mixed Initiative Forms)

```
<form id="weather_info">
 . . . Previous form items . . .
 <field name="state">
 <prompt> What state? </prompt>
 <help> Please speak the state for
 which you want the weather. </help>
 </field>
 <field name="city">
 <prompt> Say the city in
 <value expr="state"/>
 for which you want the weather.
 </prompt>
 <help> Please speak the city for
 which you want the weather.
 </help>
 <filled>
 . . . Why use this? . . .
 </filled>
 </field>
 <block>
 <submit next="/servlet/weather"
 namelist="city state" />
 </block>
</form>
```

- What state?
- **Help**
- Please speak the state for which you want the weather
- **Washington**
- Say the city in Washington for which you want the weather
- **Seattle**
- (retrieve info from server)

# Form Types(Mixed Initiative Forms)

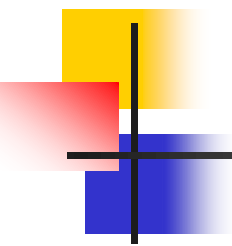
```
<form id="weather_info">
 . . . Previous form items . . .
 <initial name="start">
 <prompt> For what city and state do
 you want the weather? </prompt>
 </initial>
 . . . "state" field . . .
 <field name="city">
 <prompt> Say the city in <value
expr="state"/>
 for which you want the weather.
 </prompt>
 <filled>
 <if cond="city='LA' &&
state==undefined">
 <assign name="state"
 expr="'California'" />
 </if>
 </filled>
 </field>
 <block>
 <submit next="/servlet/weather"
 namelist="city state" />
 </block>
</form>
```

- ...For what city and state do you want the weather?
- LA
- (matches with "city" field and fills "state" field as well)

# Forms and Fields

```
<field name="op">
 <prompt>
 Choose add, subtract, multiply
or divide.
 </prompt>
 <grammar>
 [add subtract multiply divide]
 </grammar>
 <help>
 Please say what you want to do.
 <reprompt/>
</help>
<filled>
 <prompt>
 Okay, let's<value expr="op" />
 two numbers.
 </prompt>
</filled>
</field>
```

- **<filled>**
  - Action to take once the field is "filled in"
  - Implicit confirmation
- **<help>**
  - Catches the "help" event
- **<reprompt>**
  - Returns dialog to the first prompt in the field



# Features

---

- Grammars : specifies a set of utterance that a user may speak
  - Supporting any grammars
- Speech markup
  - Specifies characteristics of words
- Events handling
- Resource fetching
- Object
  - Platform dependent object call

# Application

- Voice portals

- Just like Web portals, voice portals can be used to provide personalized services to access information like stock quotes, weather, restaurant listings, news, etc.

- Location-based services

- You can receive targeted information specific to the location you are dialing from. Applications use the telephone number you are dialing from

- Voice alerts (such as for advertising)

- VoiceXML can be used to send targeted alerts to a user. The user would sign up to receive special alerts informing him of upcoming events.

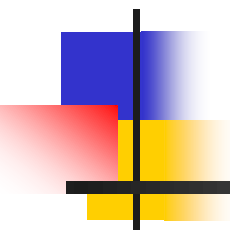
- Commerce

- VoiceXML can be used to implement applications that allow users to purchase over the phone. Because voice gives you less information than graphics, specific products that don't need a lot of description (such as tickets, CDs, office supplies, etc.) work well.

# Voice XML Tools

- BeVocal Café
  - eb-based VoiceXML development environment providing a carrier-grade VoiceXML 2.0 (and 1.0) interpreter
- Conita
  - based upon Open VXi
- Motorola has the Mobile Application Development Toolkit (MADK)
  - supports VoiceXML 1.0 (as well as WML and VoxML)
- SpeechWorks
  - VoiceXML 2.0 support to any platform





**P3P**

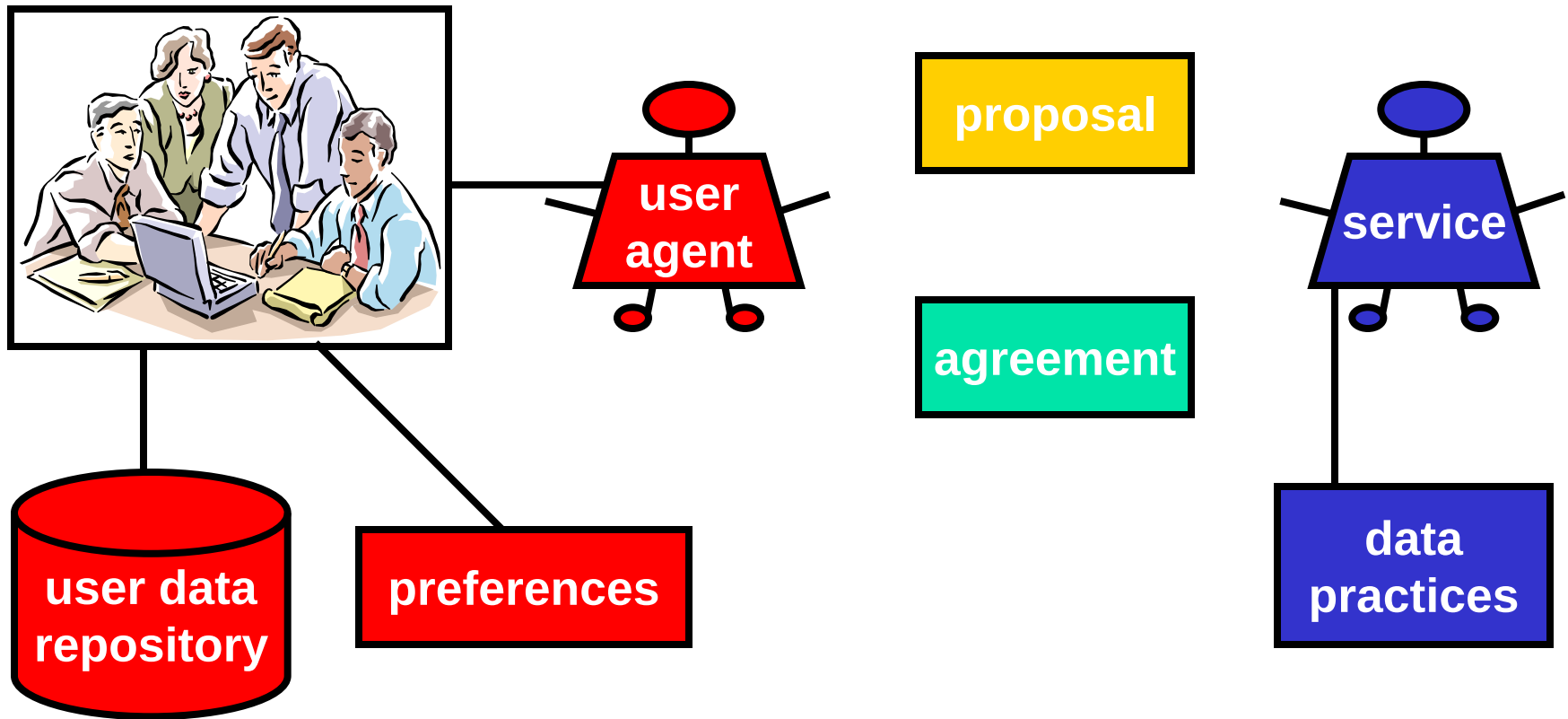
**(Platform for Privacy Preferences)**

---

# P3P Introduction

- Individual privacy protection standard technical platform that develop in W3C
- Series standard presentation about data processing that consist in website
- Information according to information open level that read and establish information about relevant web site's privacy selectively offer
- Suppose that need to supply personal information according to user's situation unlike an encryption tool that control personal information transmission in principle
- Whether must supply personal information in some time, user selects and gives help to decide.
- Participants
  - Microsoft, AT&T, Hewlett Packard, IBM, Nokia, etc.

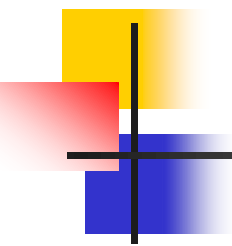
# Basic P3P Concepts



# A Simple P3P Conversation



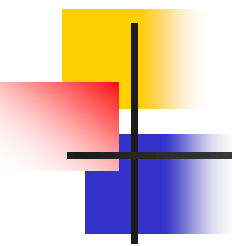
- *User agent:* Get index.html
- *Service:* Here is my P3P proposal - I collect click-stream data and computer information for web site and system administration and customization of site
- *User agent:* OK, I accept your proposal
- *Service:* Here is index.html



# P3P is a work in progress

---

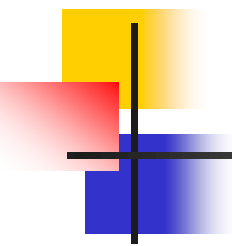
- These slides are based on the status of P3P as of May 10 2000
- Public feedback solicited November 1999 through April 2000, the P3P Last Call period
- Changes made to address feedback received during the last call period
- Additional changes may be made as a result of implementation experience
- For the latest P3P specification see <http://www.w3.org/TR/P3P/>



# Original idea behind P3P

---

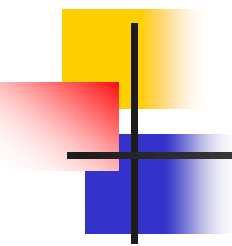
- A framework for automated privacy discussions
  - Web sites disclose their privacy practices in standard machine-readable formats
  - Web browsers automatically retrieve P3P privacy policies and compare them to users' privacy preferences
  - Sites and browsers can then negotiate about privacy terms



# P3P1.0 – A first step

---

- Offers an easy way for web sites to communicate about their privacy policies in a standard machine-readable format
  - Can be deployed using existing web servers
- This will enable the development of tools that:
  - Provide snapshots of sites' policies
  - Compare policies with user preferences
  - Alert and advise the user



# P3P 1.0 specification defines

---

- A standard schema for data a Web site may wish to collect (base data schema)
- A standard set of uses, recipients, data categories, and other privacy disclosures
- An XML format for expressing a privacy policy
- A means of associating privacy policies with Web pages or sites
- A mechanism for transporting P3P policies over HTTP



# Future versions of P3P

- Allow web sites to offer a choice of policies
  - P3P 1.0 supports only one policy per resource
- Allow for “negotiation” and explicit agreements to be reached between user agent and web site
  - P3P 1.0 policies are “take-it-or-leave-it”
- Allow for non-repudiation of agreements, signatures from third-party seal providers, etc.
  - P3P 1.0 offers no mechanism to prove that certain communication took place
- Facilitate automated data transfer
  - P3P 1.0 requires external mechanisms (e.g., automatic form-fill) to transfer data

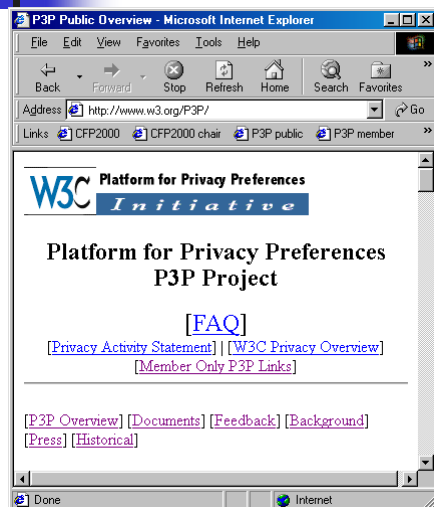
# P3P is a partial solution

- P3P1.0 helps users understand privacy policies but is not a complete solution
- Seal programs and regulations
  - help ensure that sites comply with their policies
- Anonymity tools
  - reduce the amount of information revealed while browsing
- Encryption tools
  - secure data in transit and storage
- Laws and codes of practice
  - provide a base line level for acceptable policies

# Implementing a P3P 1.0 server

- Formulate privacy policy
- Translate privacy policy into P3P format
- Place P3P policy on web site
  - One policy for entire site or multiple policies for different parts of the site
- Associate policy with web resources:
  - Place P3P policy reference file (which identifies location of relevant policy file) at well-known location on server;
  - Configure server to insert P3P header with link to P3P policy reference file; or
  - Insert link to P3P policy reference file in HTML content

# A simple HTTP transaction



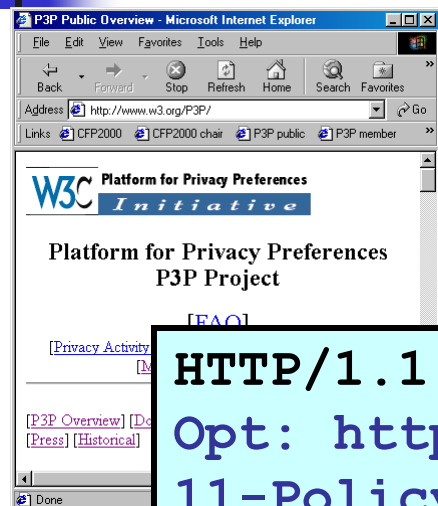
GET /x.html HTTP/1.1  
Host: foo.com  
. . . *Request web page*

HTTP/1.1 200 OK  
Content-Type: text/html  
. . . *Send web page*

**Web  
Server**

# A simple HTTP transaction

## With P3P 1.0 added



GET /x.html HTTP/1.1  
Host: foo.com  
. . . *Request web page*

Web  
Server

HTTP/1.1 200 OK  
Opt: http://www.w3.org/2000/P3Pv1/; ns=11  
11-PolicyRef: http://foo.com/p3p.xml  
Content-Type: text/html  
. . . *Send web page*

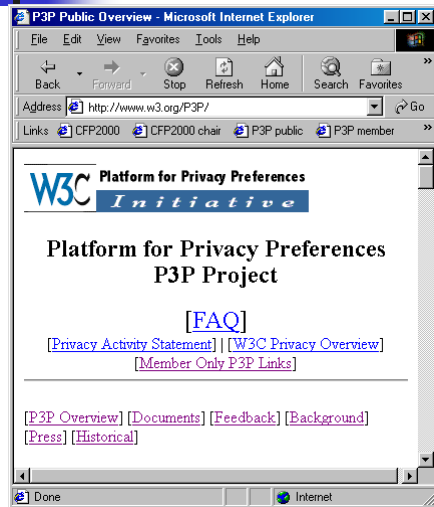
*Request Policy Reference File*

*Send Policy Reference File*

*Request P3P Policy*

*Send P3P Policy*

# Or using /p3p.xml file



GET /p3p.xml HTTP/1.1

Host: foo.com

*Request Policy Reference File*

Web  
Server

*Send Policy Reference File*

*Request P3P Policy*

*Send P3P Policy*

GET /x.html HTTP/1.1

Host: foo.com

. . . *Request web page*

HTTP/1.1 200 OK

Content-Type: text/html

. . . *Send web page*

# The P3P vocabulary

- **Who** is collecting data?
- **What data** is collected?
- For **what purpose** will data be used?
- Is there an ability to **change preferences** about (opt-in or opt-out) of some data uses?
- Who are the data **recipients** (anyone beyond the data collector)?
- To what information does the data collector provide **access**?
- What is the data **retention** policy?
- How will **disputes** about the policy be resolved?
- Where is the **human-readable privacy policy**?

# Example privacy policy

**At CatalogExample, we care about your privacy. When you come to our site to look for an item, we will only use this information to improve our site and will not store it in an identifiable way.**

**CatalogExample is a licensee of the PrivacySealExample Program. ...**

**Questions regarding this statement should be directed to:  
CatalogExample 1-248-392-6753**

**When you browse through our site we collect:**

The basic information about your computer and connection to make sure that we can get you the proper information and for security purposes

Aggregate information on what pages consumers access or visit to improve our site

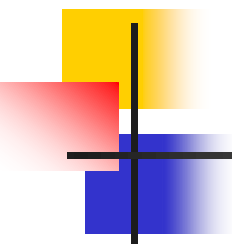
**We purge the browsing information that we collect regularly**



# P3P/XML encoding

```
<POLICY xmlns=http://www.w3.org/2000/P3Pv1
 discuri="http://www.catalog.example.com/Privacy.html">
 <ENTITY><DATA-GROUP><DATA ref="#business.name">CatalogExample</DATA>
 <DATA ref="#business.contact-info.telecom.telephonenumber.intcode">1</DATA>
 <DATA ref="#business.contact-info.telecom.telephonenumber.loccode">
 248</DATA>
 <DATA ref="#business.contact-info.telecom.telephonenumber.number">
 3926753</DATA>
 </DATA-GROUP></ENTITY>
 <DISPUTES-GROUP> <DISPUTES resolution-type="independent"
 service="http://www.PrivacySeal.example.org"
 short-description="PrivacySeal.exampleorg"
 <REMEDIES><correct/></REMEDIES>

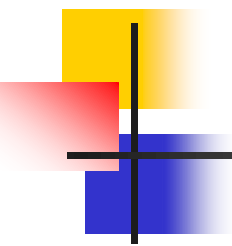
 </DISPUTES></DISPUTES-GROUP>
 <ACCESS><nonident/></ACCESS>
 <STATEMENT>
 <PURPOSE><admin/><develop/></PURPOSE>
 <RECIPIENT><ours/></RECIPIENT>
 <RETENTION><stated-purpose/></RETENTION>
 <DATA-GROUP>
 <DATA ref="#dynamic.clickstream.server"/>
 <DATA ref="#dynamic.http.useragent"/>
 </DATA-GROUP>
 </STATEMENT>
</POLICY>
```



# P3P Demo implementations

- AT&T Privacy Minder
- AT&T P3P Proposal Generator
- ENC Privacy Information Management System
- IBM P3P Parser
- Microsoft Privacy Wizard
- NEC P3P for Perl
- NCR P3P user agent demo
- W3C P3P client prototype

*<http://www.w3.org/P3P/implementations>*

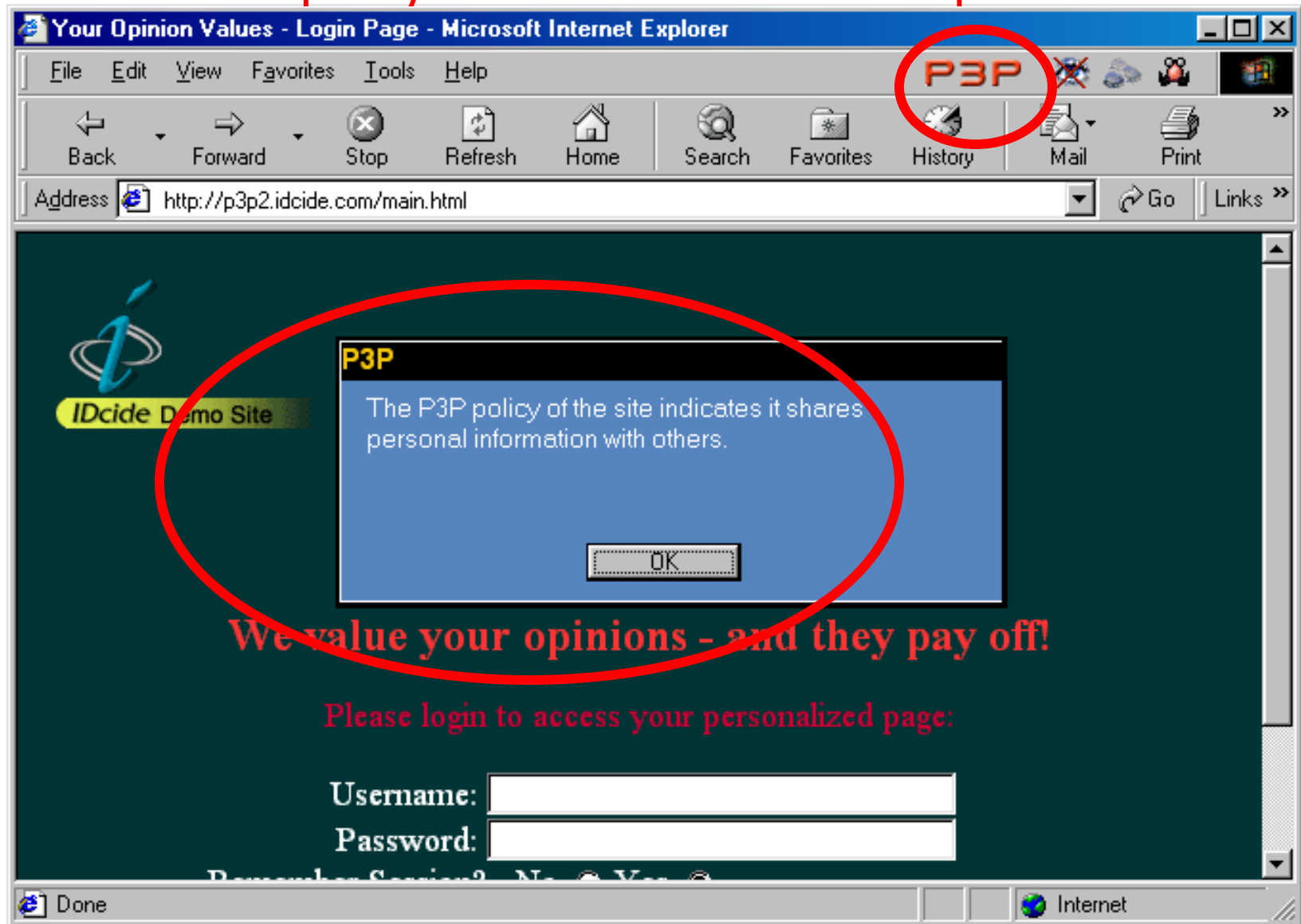


# P3P enabled web sites

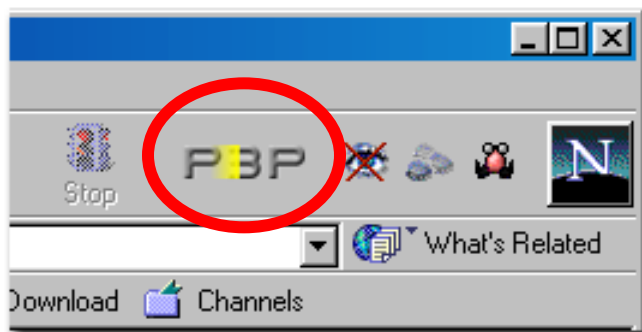
---

- [www.aol.com](http://www.aol.com)
- [www.att.com](http://www.att.com)
- [www.cdt.org](http://www.cdt.org)
- [www.engage.com](http://www.engage.com)
- [www.hp.com](http://www.hp.com)
- [www.ibm.com](http://www.ibm.com)
- [www.idcide.com](http://www.idcide.com)
- [www.microsoft.com](http://www.microsoft.com)
- [www.pg.com](http://www.pg.com)
- [www.ttuhs.edu](http://www.ttuhs.edu)
- [www.youpowered.com](http://www.youpowered.com)
- [www.vineyard.net](http://www.vineyard.net)
- [www.w3.org](http://www.w3.org)

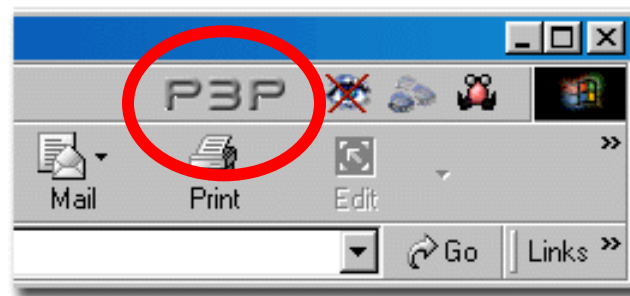
Double clicking on the P3P icon indicates where the site's policy differs from the user's preferences



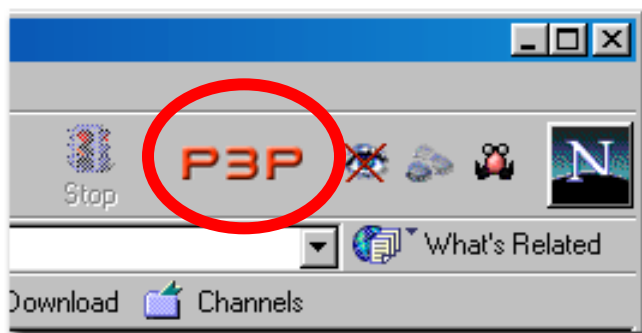
## IDcide P3P Icons



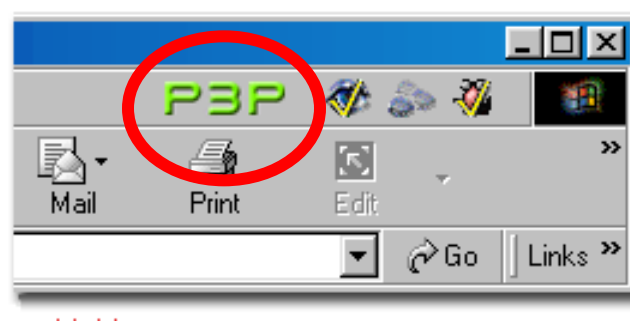
Searching for  
a P3P policy



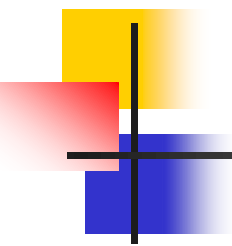
No P3P policy found



P3P policy is  
NOT acceptable



P3P policy is  
acceptable



# PrivacyBank.Com

---

- An Infomediary Service
- Does not currently use P3P but could be adapted to use P3P
- A good example of an interface that might be suitable for a P3P user agent

*<http://www.privacybank.com>*

billing - Microsoft Internet Explorer

Full Policy Details - Microsoft Internet Explorer

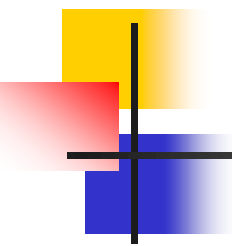
### Privacy Policy of Starbucks

Information Requested	Used For:	Required?
First Name	    	YES
Last Name	    	YES
Address 1	    	YES
Address 2	    	NO
City	    	YES
State/Province	    	YES
Zip/Postal Code	    	YES
Country	    	YES
E-mail	    	YES
Daytime Phone Number	    	YES

OK

Comments? Send mail to [comments@privacybank.com](mailto:comments@privacybank.com)

Copyright © 1998-1999 Millet Software, Inc.



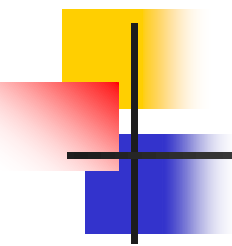
# Benefits and problems

---

## Potential problems

- Data practices may not be privacy friendly
- Individuals may lower expectations to match marketplace choices
- May lead to more data disclosure





# Benefits and problems

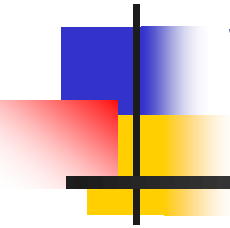
---

## Potential benefits

- Increase users' knowledge of data practices
- Help enforce privacy directives
- Create new markets; lead to better choices
- Promote opt-in business models

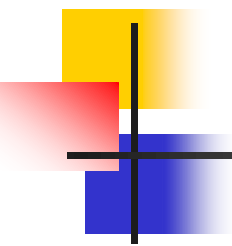
# Advantages of P3P

- *Intuitive* - promotes a seamless browsing experiences while addressing privacy concerns
- *Transparent* - makes privacy policies clear to web users
- *Global* - developed with international diversity in mind
- *End-to-End* - provides tools to more easily create policies and checks sites for privacy assurance seals
- *Expandable* - future versions could support automatic negotiation of privacy agreements and digital signature-based authentication
- *Available* - demos currently available



# **WML(Wireless Markup Language)**

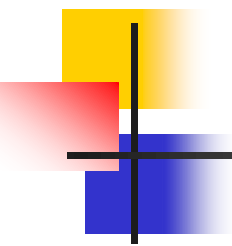
---



# WAP?

---

- An open, global standard that empowers monile users whit wireless devices to easily access and interact with information and services instantly
- A standard created to make accessing the internet as easy and convenient as using a cellular phone



# WAP Forum

---

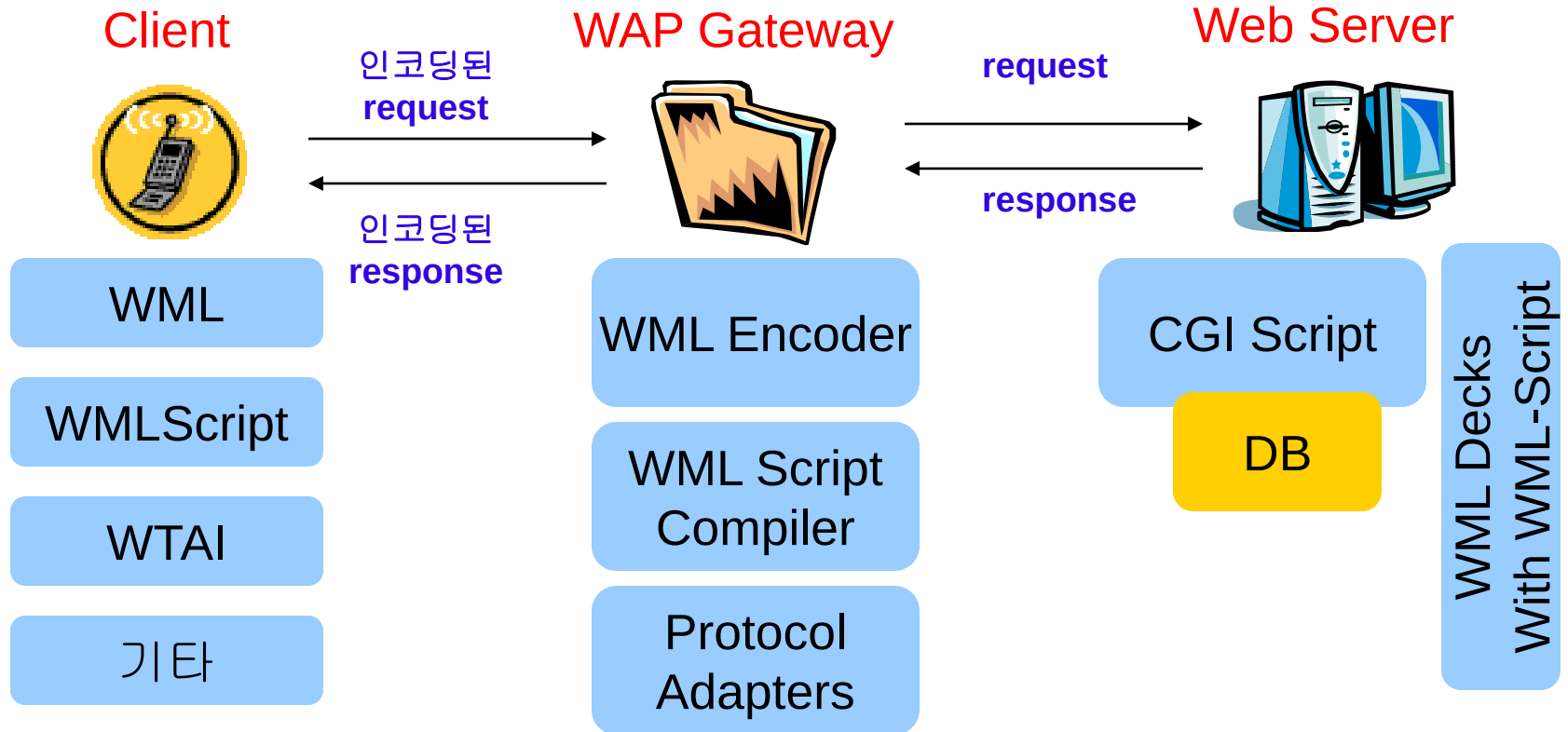
## ■ Goals

- Ensure interoperability
- Foster growth of the wireless market
- Make accessing the internet portable and convenient

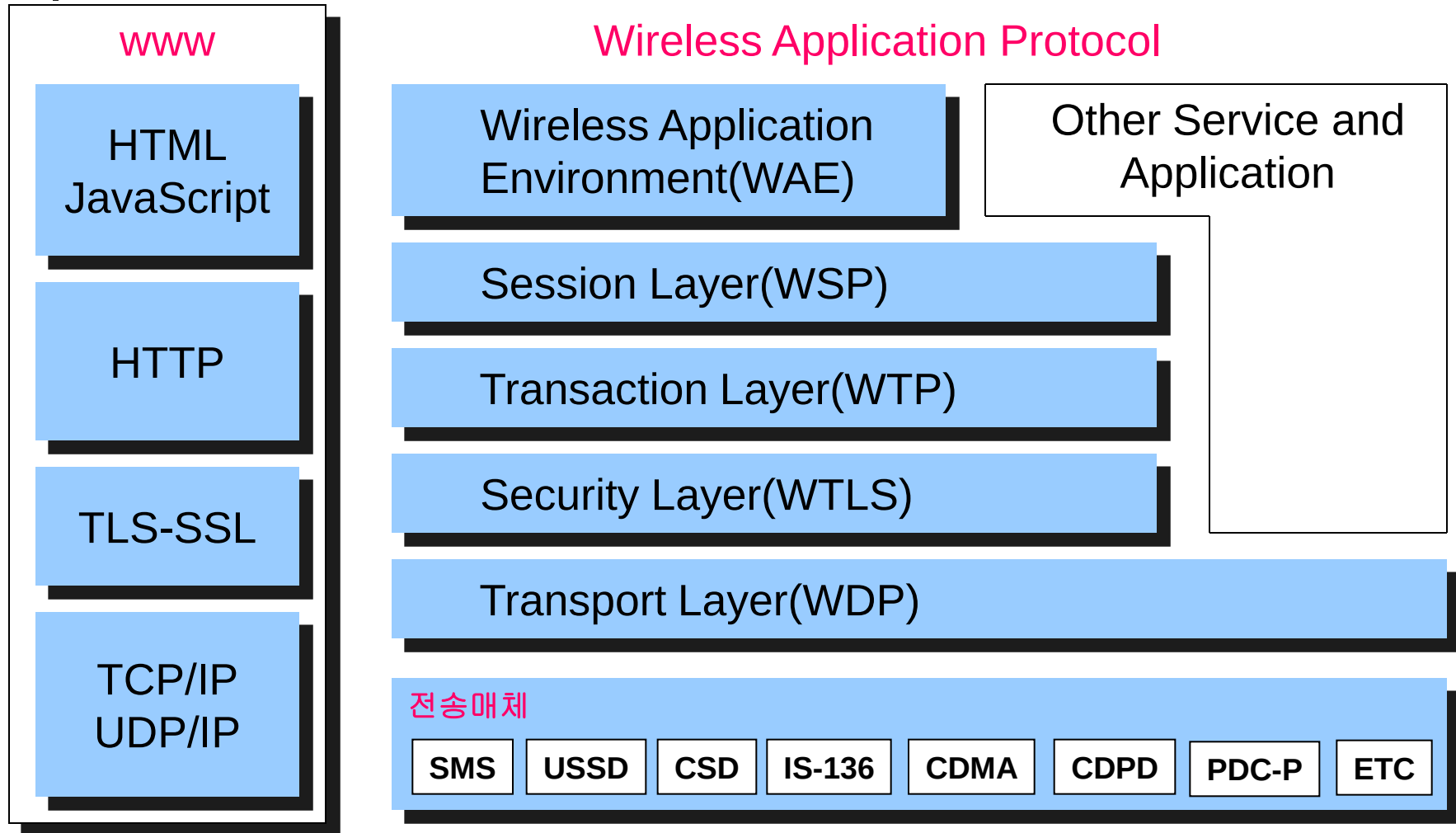
## ■ Objectives

- Create a global protocol specification to work across differing wireless network technologies
- Submit specification for adoption by appropriate industry and standard bodies
- Enable applications to scale across a variety of transport options and device types

# WAP Architecture



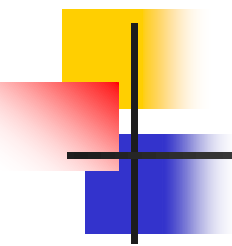
# WAP vs. WWW



# WML Scope

- Define:
  - markup language based on XML in specifying content & user interface for narrowband devices
  - ( cellular phone, pager, PDA...)
- Constraints of small narrowband devices:
  - small display & limited user input facilities
  - narrowband network connection
  - limited memory & conceptual resources





# WML Scope

---

- Functional areas:
  - text presentation & layout
  - deck/card organizational metaphor:  
a WML deck = a collection of cards
  - inter-card navigation & linking:  
using fragment anchors to identify each WML card within a WML deck
  - string parameterization & state management:  
all decks can be parameterised using state model  
more efficient use of network resources

# Decks

- The smallest unit of WML that is transmitted to a WAP device
- page information, much like Web page

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML
1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml"
>

<wml>
 <head>...head information...</head>
 <template>...template information...</template>
 <card>...card information...</card>
</wml>
```

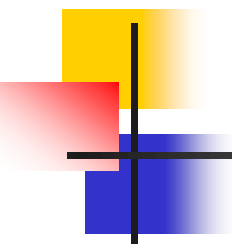
# Decks

- Document Prologue
  - xml declaration & document type declaration
  - it is an error to omit the prologue

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
 "http://www.wapforum.org/DTD/wml_1.1.xml">
```

- <head> element
  - info relating to the deck , including meta-data & access control elements

```
<head>
 <access>, <meta>
</head>
```



# Decks

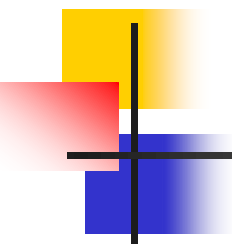
---

- **<access> element**
  - specify access control information for the entire deck
  - it is an error for a deck to contain more than one access element

<access

domain="STRING" //valid partial or full domain name  
path="STRING" //valid relative or absolute path name

>



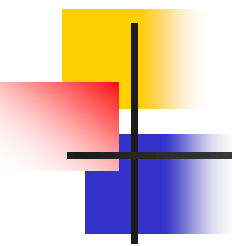
# Decks

- `<meta>` element
  - generic meta-info(property names & values) relating to the WML deck

`<meta`

```
http-equiv="STRING" //header name
name="STRING"
forua="true | false " //indicates that the
meta info should be delivered to the user
agent
content = "STRING" //header's setting
scheme = "STRING" //secondary info for
 interpreting the meta data
```

`>`



# Cards

---

- container of text & input elements

<card

id="ID"

newcontext="true | false"

onenterbackward ="URL"

onenterforward = "URL"

ontimer ="URL"

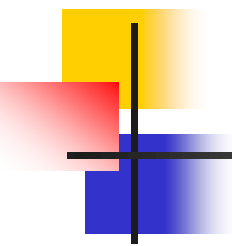
ordered="true | false "

title =VDATA

>

<onevent>,<do>,<p>,<timer>

</card>

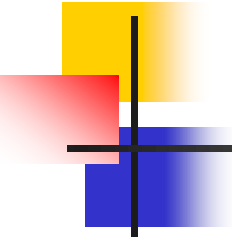


# Cards

---

## ■ Attributes

- *id* : card name
- *newcontext* : instructs the device to remove all context-specific variables, clear the history stack, and reset the device to a well-known state
- *onenterbackward* : result of *<prev>*
- *onenterforward* : result of *<go>*
- *ontimer* : to go if a *<timer>* element expires
- *ordered* : indicating that the card's content should be presented in a list format
- *title* : the label for bookmark



# Content

## ■ Styled Text

TAG	FONT CHARACTERISTIC
<b>	Bold
<big>	Large
<em>	Emphasize
<i>	Italic
<small>	Small
<strong>	Strong emphasis
<u>	Underline

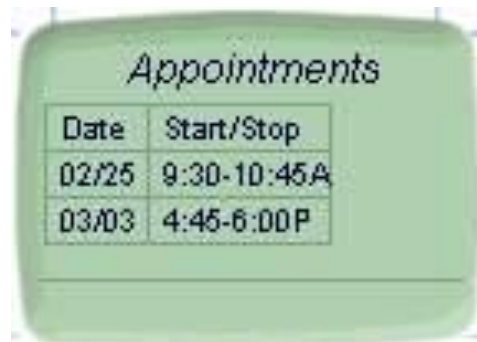


# Content

## ■ Tables

```
<table
 align="STRING" //list of alignment
 designators,"L","C","R"
 columns="NUMBER"
 title ="VDATA" >
 <tr>
 <td>

<a>,<anchor>,
,FMTEXT,,<table>,TEXT
 </td>
 </tr>
</table>
```



Date	Start/Stop
02/25	9:30-10:45A
03/03	4:45-6:00P

# Content

## ■ Images

`<image`

`align = "top | middle | bottom"`

`alt = "VDATA"`

`height = "LENGTH"`

`hspace = "LENGTH"`

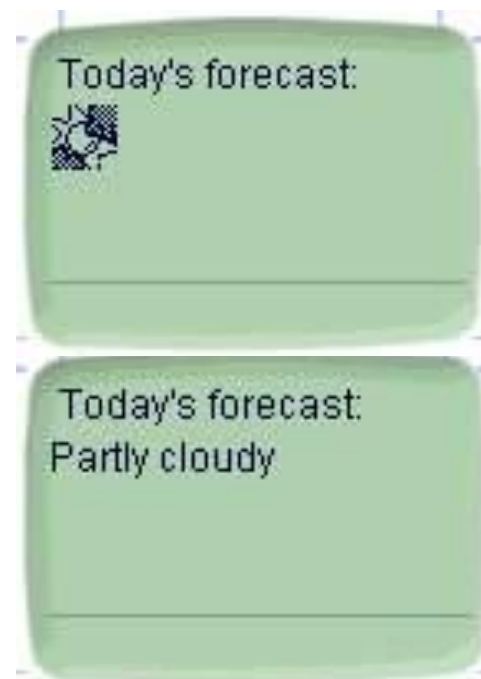
`localsrc = "VDATA"`

`src = "URL"`

`vspace = "LENGTH"`

`width = "LENGTH"`

`/>`



# Events

- `<anchor>` element

```
<anchor title = "VDATA" >

,<go>,,<prev>,<refresh>,TEXT
</anchor>
```

- a short-form syntax

```


,<go>,,<prev>,<refresh>,TEXT
</anchor>
```

# Event

## ■ Timers

- provides mechanism for invoking a task after a certain period of time
- A card can have only one timer, and a <timer> can have only one task

```
<timer
 name = "NAME"
 value = "VDATA"
>
```



# Data Entry

- `<option>` element
  - single possible choice the user can make.
  - `<select>` element must contain one or more `<option>`

`<option`

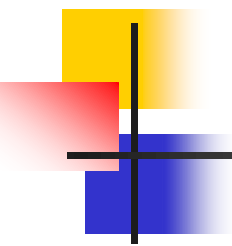
```
onpick = "URL"
title = "VDATA"
value = "VDATA"
TEXT, <onevent>
```

`</option>`

>

The image displays four mobile device screens arranged in a 2x2 grid, each showing a selection interface. Each screen has a dark blue header bar with white text, a light green body with radio button options, and a bottom bar with 'Select' and 'OK' buttons.

- Top-left screen:** Header 'Your sex: Male'. Options: ☐ Your sex: Male, ☒ Your income:, ☐ Your hobbies: ski.
- Top-right screen:** Header 'Male'. Options: ☒ Male, ☐ Female.
- Bottom-left screen:** Header '\$10-25K'. Options: ☒ \$10-25K, ☐ \$25-50K, ☐ \$50-100K, ☐ Over \$100K.
- Bottom-right screen:** Header 'Skiing'. Options: ☒ Skiing, ☐ Reading, ☐ Movies.



# Data Entry

---

- `<option>` element
  - *onpick* : A URL to execute if this `<option>` is selected or deslected ( only multiple option list )
  - *title* : the option's title, used when it is displayed
  - *value* : used when the user agent sets the select's name variable

# Data Entry

## ■ Multilevel Choices

### ■ <optgroup>element

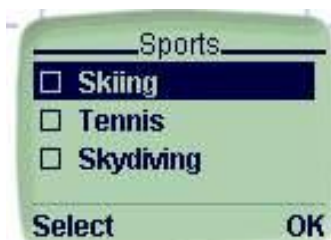
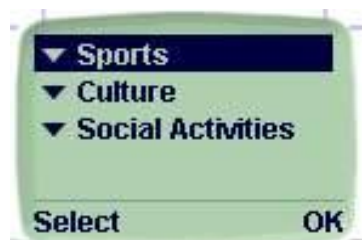
- define hierarchical relationships between <option> elements

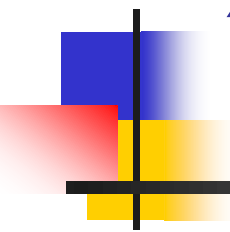
//used when displaying an <option>group

```
<optgroup title = "VDATA" >
```

```
 <optgroup>, <option>
```

```
</optgroup>
```



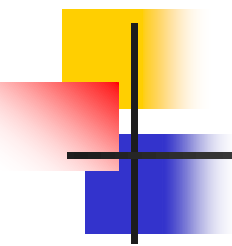


# **XrML**

**(eXtensible right Markup Language)**

---

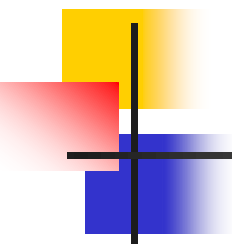




# XrML Overview

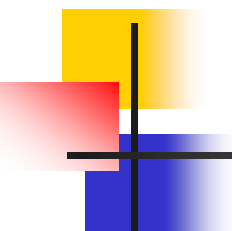
---

- XrML – eXtensible rights Markup Language
- Originates from Xerox PARC in 1994
- Digital contents to divide or use rights, involved conditions and obligations specification offer
- Contents identification, confidentiality, integrity support
- Many flexibles, customizable, extensible offer
- Data is offer in next time site : [www.xrml.org](http://www.xrml.org)
- ContentGuard's fundamental tool offer : language, parsers, interpreter



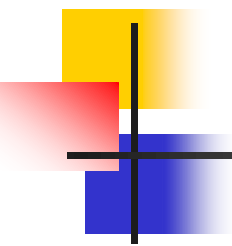
# XrML advantage

- Aid Human's Knowledge Processing:
  - Conversion of XRML (including Rule Identification Language) to XML/HTML
- Aid Agent's Knowledge Processing against Web page
  - Triggering rule-based inferences in the agents, possibly from the Workflow Management System
- Maintain Consistency between Rule Base and Web page
  - Aid the extraction(semi-automatically) of rules from XML maintaining consistency between them



# XrML 2.0 Feature

- Mathematical Precision – no ambiguity
- Expressiveness – advanced business models, life-cycle management, usage state tracking, pattern matching
- Well defined core and extensions architecture
  - Compact: Use of only those terms needed
  - Applications based on equality & pattern matching enable extensions without the need to upgrade
- Comprehensive Security
  - Entity authentication (Users, software, hardware, Digital Items, etc. )
  - Integrity and confidentiality of rights expressions
- Up-to-date Standards and Technologies



# XrML Advanced Features

---

- Mechanisms for Enhanced Expressiveness
  - Variables (via ForAll and XmlPatternAbstract)
  - Rights Grouping (via GrantGroup)
  - Delegation (via DelegationControl)
  - Meta Rights (via Issue, Obtain, and Revoke)
  - “Attribute” Certificates (via PossessProperty)

# XrML Evolution

## XrML 2.0 (11/01) ContentGuard

- ⊙ Support for More Business Models
- ⊙ Enhanced security, flexibility & extensibility

## XrML 1.2 (11/01) ContentGuard

- ⊙ Final Maintenance Release of 1.X

## XrML 1.03 (8/00) ContentGuard

- ⊙ Enhancements added to increase flexibility

## XrML 1.0 (4/00) ContentGuard

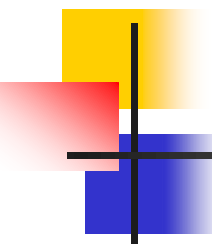
- ⊙ Conversion to XML based language
- ⊙ Additional Extensions

## DPRL 2.0 ('97-'99) Xerox

- ⊙ Enables specification of rights (fees, terms, and conditions) for digital works

## DPRL 1.0 ('94-'96) Xerox

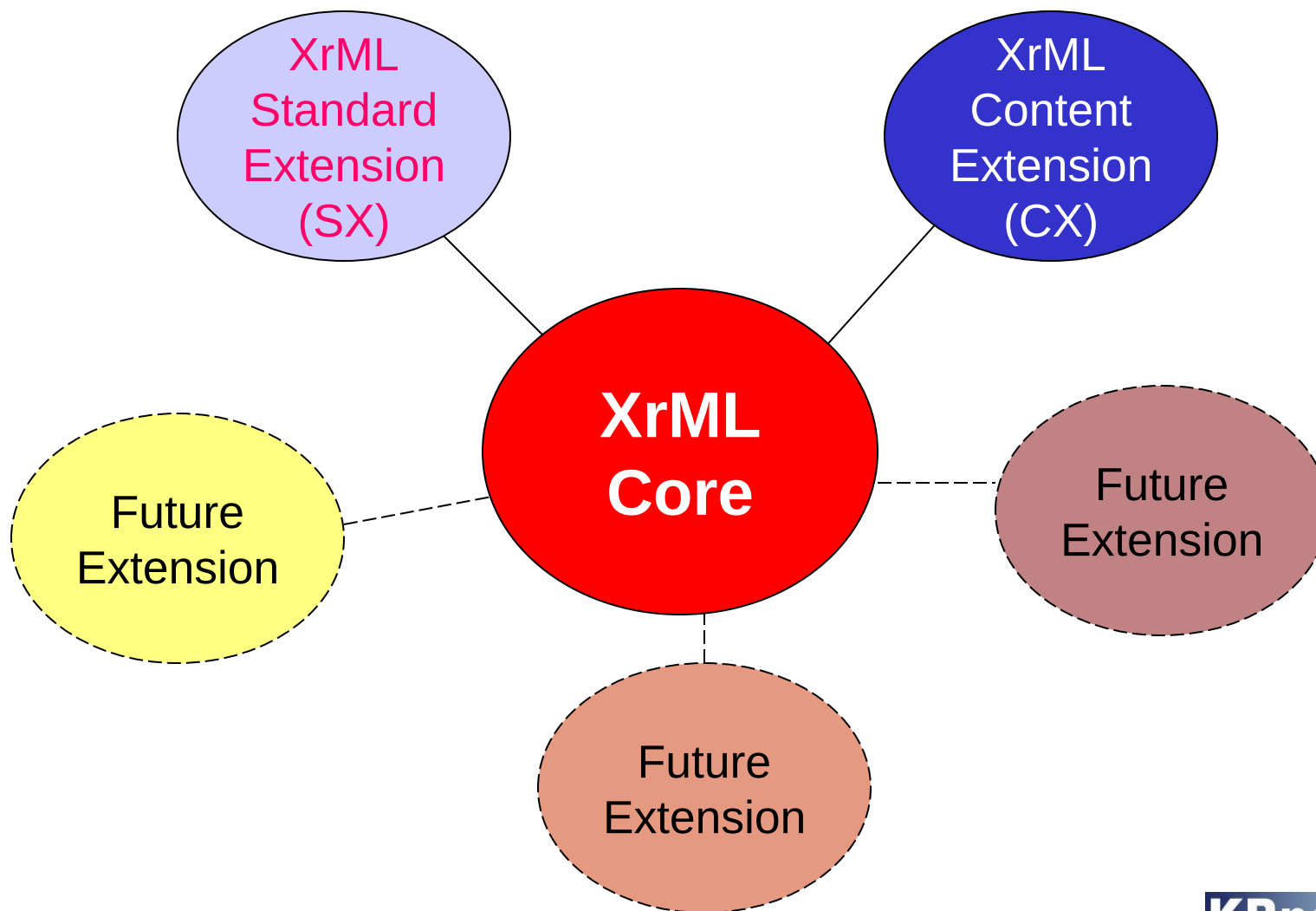
- ⊙ Focus on machine enforceable rights



# XrML Specification Configuration

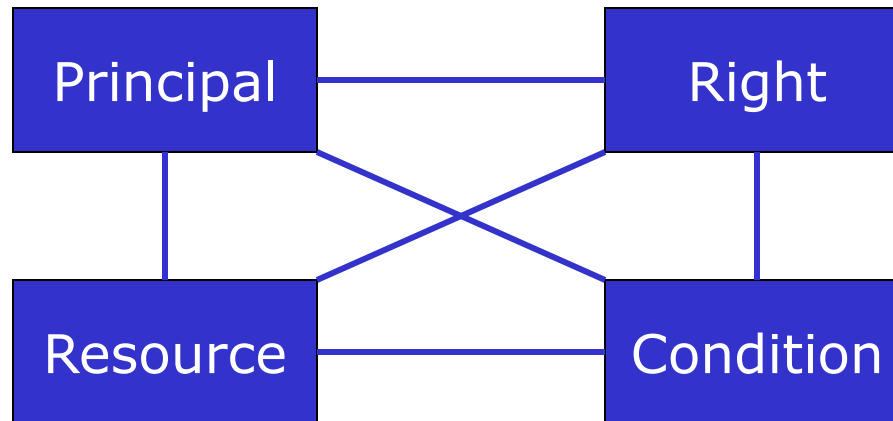
- Primer : XrML Fundamental Concept
- XrML Core Schema : Description about core of XrML design and rescue
- Standard Extension Schema : XrML standard extension, extension about common type, element that is used in XrML use scenario
- Content Extension Schema : XrML Content Extension, right, conditions, Language extension to define type, element to describe metadata
- Appendices : index, glossary, list of reference

# XrML Extensibility Architecture



# XrML base feature

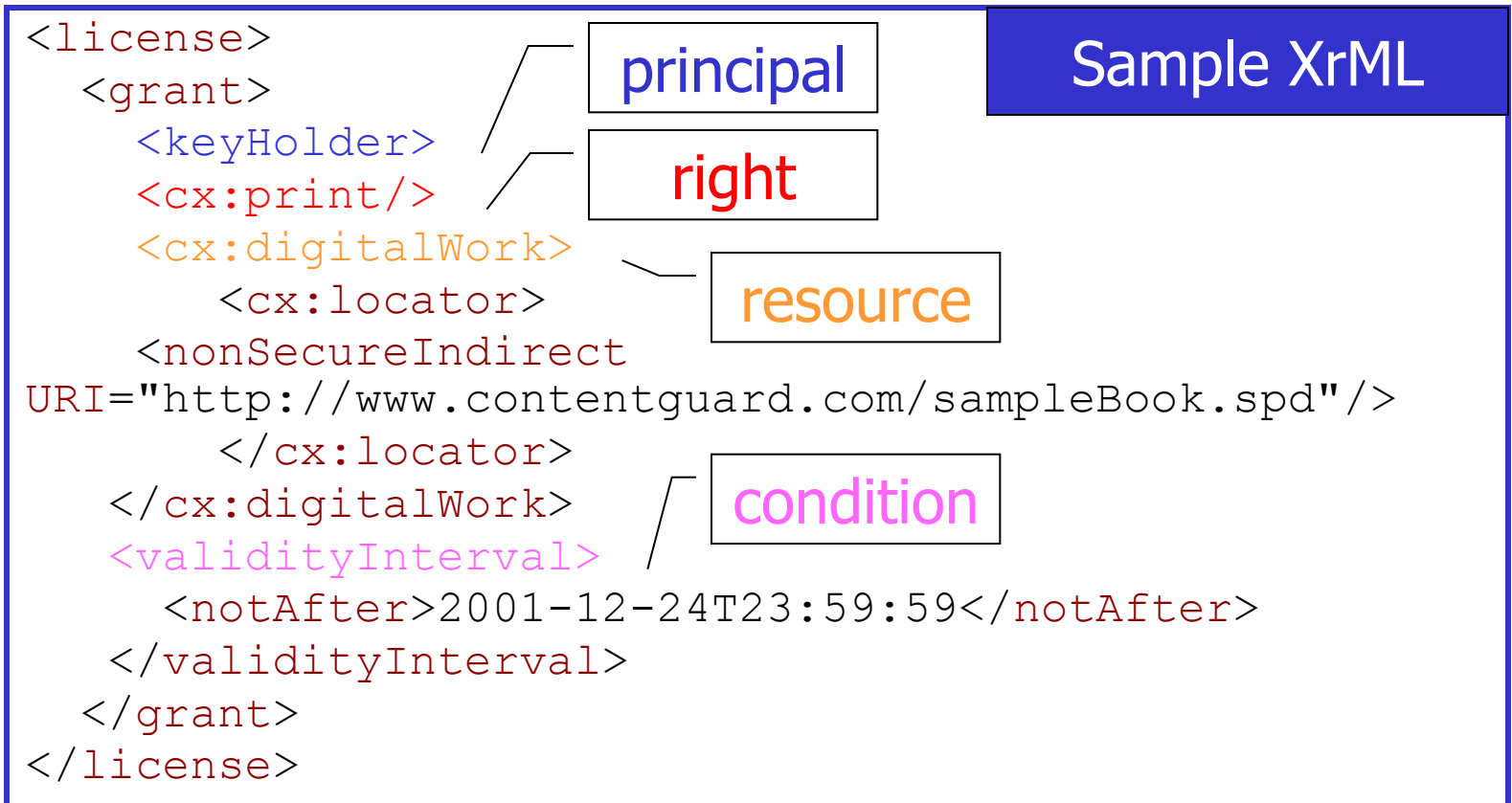
- Four Key Components
  - Principal
  - Right
  - Resource (Work, Service, Name, etc.)
  - Condition
- Granting Mechanisms
  - Grant
  - License

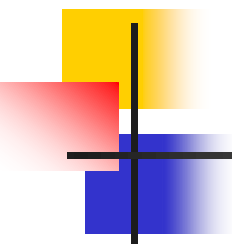




# XrML Core

- XrML core
  - License, grant, principal, right, resource, condition





---

# Q&A