

3. XML Architecture (& Components)

In order of appearance:

- XML Design Goals
- XML Component Architecture Overview
- What is XML?
- XML Parts
- DTD: Document Type Definition
- XML Schema
- XLink / XPointer: XML **Linking**
- Styling with XSL: Extensible Stylesheet Language
- XSLT (XSL Transformation)
- XPath
- RDF: Resource Description Framework
- SOAP: Simple Object Access Protocol
- DOM: Document Object Model
- WSDL
- ebXML (e-business XML)
- ADML: Architecture Description Markup Language

Alphabetically ordered:

- ADML: Architecture Description Markup Language
- DOM: Document Object Model
- DTD: Document Type Definition
- ebXML (e-business XML)
- RDF: Resource Description Framework
- SOAP: Simple Object Access Protocol
- XLink / XPointer: XML **Linking**
- XML Design Goals
- XML Parts
- XML Component Architecture Overview
- XPath
- XML Schema
- Styling with XSL: Extensible Stylesheet Language
- XSLT (XSL Transformation)
- What is XML?
- WSDL

1 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



XML Design Goals

- XML shall be straightforwardly usable over the Internet.
- XML shall support a wide variety of applications.
- XML shall be compatible with SGML.
- It shall be easy to write programs which process XML documents.
- The number of optional features in XML is to be kept to the absolute minimum, ideally zero.
- XML documents should be human-legible and reasonably clear.
- The XML design should be prepared quickly.
- The design of XML shall be formal and concise.
- XML documents shall be easy to create.
- Terseness in XML markup is of minimal importance.



2 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Typical XML Architecture Components

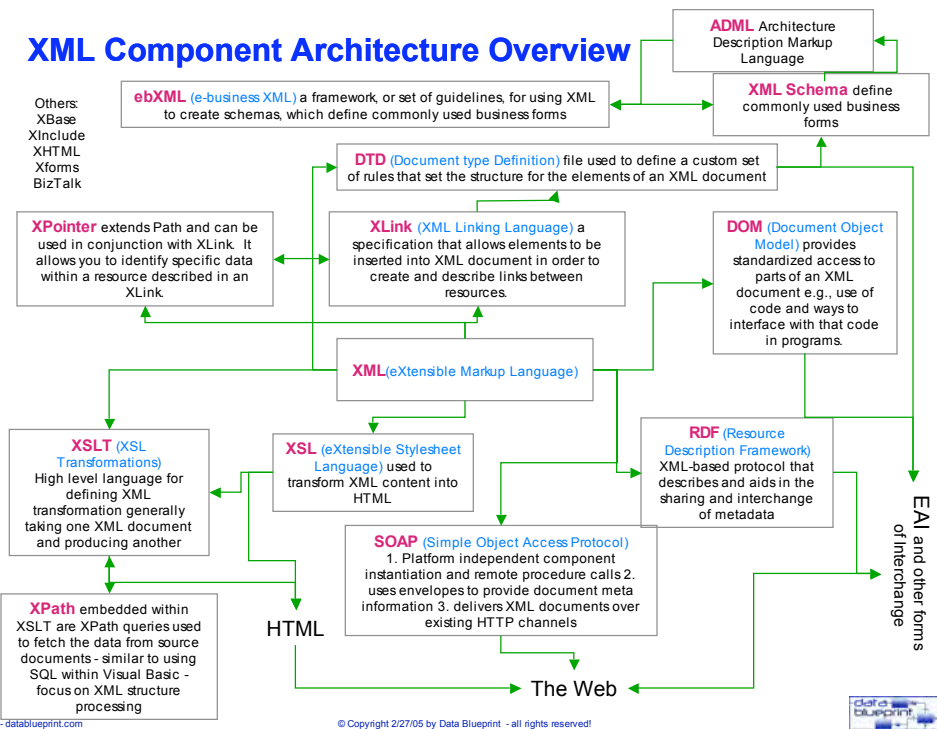
- XML** **Extensible Markup Language** Defines document content using metadata tags and namespaces
- DTD** **Document Type Definition** Defines XML document structure (analogous to DDL schema)
- XSL** **Extensible Style Language XSL or Cascading Style Sheets** (CSS) separate layout from data
- XLL** **Extensible Linking Language** XLL implements bi-directional links (single or multiple)
- DOM** **Document Object Model** Implements a standard API for processing XML in any language
- RDF** **Resource Description Framework** W3 Interoperability Project for data content interchange

3 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



XML Component Architecture Overview



5 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



What is XML?

- eXtensible Markup Language
- A language for describing the content of a document
- World Wide Web Consortium (W3C) Standard Recommendation
- Designed to facilitate Internet-based data transfer
- The Extensible Markup Language (XML) describes a class of data objects called XML documents which are stored on computers, and partially describes the behavior of programs that process these objects.



Definition

- Extensible Markup Language (XML)
- Describes data content
- Describes structure of data and semantics within a document
- Supports Web based presentation
- Supports Web based programming language access
- Hierarchical tree structures



Definition (continued)

- Database Queries
- Simple
- Standardized across browsers
- Content searches
- New Web Page Vocabularies
 - Music Score (MusicML)
 - Mathematical Notation (MathML)
 - Chemical Formula (CML)
 - Business (eBIS-XML)
- Document Search Capabilities

8 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



- **Documents**

- The text file containing the marked-up data

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

XML Parts

- **Elements**

- The components of the XML document, bounded by start and end tags

```
<TITLES>  
  <TITLE CD="Y">  
    <NAME>Data Warehouse Design Solutions</NAME>  
  </TITLE>  
</TITLES>
```

- **Attributes**

- Named values assigned to elements

```
<TITLE CD="Y">
```

- **Entities**

- Named containers of content, both text and non-text

```
<!ENTITY terms SYSTEM "terms.xml" >  
....  
<CONTRACT Standard_Terms="&terms;"/>
```



9 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



All documents have a root element

Hierarchical (nested) format

```
<INVENTORY>
```

```
  <BOOK>
```

```
    <TITLE>East of Eden</TITLE>
```

```
    <AUTHOR>John Steinbeck</AUTHOR>
```

```
    <PRICE>15.00</PRICE>
```

```
  </BOOK>
```

```
</INVENTORY>
```

INVENTORY is the root element



Element Type Declarations

- Elements are the basic building blocks of markup, and consist of the tags, the attributes and the content. The root element is the element which contains all other elements. An XML document **MUST** contain a root element - and only **ONE** root element.
- Defines structure for XML components
- Consists of:
 - Element Name
 - Content Model
 - Element Attributes
 - `<!ELEMENT TITLE (NAME, ISBN)>`



Element Content Model

- Cardinality of sub-elements :
 - **Required**
 - <!ELEMENT TITLE (Name, ISBN)>
 - **Selection**
 - <!ELEMENT TITLE (Name, (#PCDATA | ISBN))>
 - **One or None** – suffix “?”
 - <!ELEMENT TITLE (Name, ISBN, Review?)>
 - **One or More** – suffix “+”
 - <!ELEMENT TITLE (Name, ISBN, Author+)>
 - **Zero or More** – suffix “*”
 - <!ELEMENT TITLE (Name, ISBN, Author*)>

12 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



- Element Content can be:
 - EMPTY** (No data contents)
 - <!ELEMENT br EMPTY>
 -
 -

Element Content Model

- **ANY** (any tags or freeform content)
 - <!ELEMENT address ANY>
 -
 - <address>
 - <street>123 First St</street>
 - <city>Any Town</city> <zip>12345</zip>
 - </address>
- **#PCDATA** (Text only)
 - <!ELEMENT Name (#PCDATA)>
 -
 - <Name>Clive Finkelstein</Name>
- **Other** Elements
 - <!ELEMENT TITLE (NAME, ISBN)>
 - <!ELEMENT NAME (#PCDATA)>
 - <!ELEMENT ISBN (#PCDATA)>

13 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Element Naming Rules

- Start with Letter or characters “_”, “.”
- Remaining portion: letters, digits, and “_”, “.”, “ ”, “_”
- Spaces are not allowed
- Can not start with reserved words



14 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Attribute Declarations

- An attribute is additional data that is associated with an element, rather than being it's own element. Attributes names and their values are both constrained by the document's DTD. If a document is well-formed but not valid (ie. No DTD), then there are no constraints.
- Attribute Types
 - CDATA
 - ID, IDREF
 - Primary and foreign keys
 - ENTITY
 - Enumerated
 - `<!ATTLIST PERSON person_type (Employee | Applicant) "Employee">`
- Attribute Defaults
 - default
 - #REQUIRED
 - #IMPLIED
 - #FIXED fixed_value

15 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Using Attributes

Adds descriptive content to the element tags

```
<? xml version="1.0" ?>  
<! - -This is my first XML Document. - - >
```

```
<INVENTORY>  
  <BOOK inv_number="F00001" >  
    <TITLE>East of Eden</TITLE>  
    <AUTHOR>John Steinbeck</AUTHOR>  
    <PRICE>15.00</PRICE>  
  </BOOK>  
  <BOOK inv_number="SF0123" >  
    <TITLE>Dune</TITLE>  
    <AUTHOR>Frank Herbert</AUTHOR>  
    <PRICE>6.00</PRICE>  
  </BOOK>  
</INVENTORY>
```

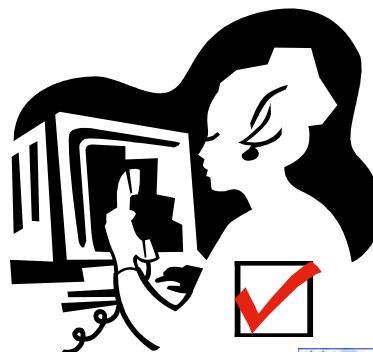
16 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Four Serious Drawbacks

1. But what if the metadata author is a liar?
2. How to generate all that metadata?
3. Microsoft
4. Who decides, within your organization, what information goes on your web page?



17 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Document Type Definitions (DTDs)

- Provide format for documents that can be adopted as (typically a local) standard
- Consists of several components:
 - Element type declarations
 - Attribute declarations
 - Entity declarations
 - Notation declarations
- Describes the documents structure
 - Elements
 - Attributes
 - Text
- Original XML modeling language
- Specifies which elements are allowed



18 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



DTD Functionality

- Designers of a new XML language, must agree on three things:
 1. Which tags will be allowed
 2. How tagged elements may nest within one another
 3. How they should be processed
- The first two--the language's vocabulary and structure--are typically codified in a Document Type Definition, or DTD. The XML standard does not compel language designers to use DTDs, but most new languages will probably have them, because they make it much easier for programmers to write software that understands the markup and does intelligent things with it. Programmers will also need a set of guidelines that describe, in human language, what all the tags mean.

19 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Well-formed Documents

- One root element
- Parent child
- Meta-tags
- Element start tag (< [data name] >)
- Element end tag (< / [data name] >)
- Case sensitive.
(i.e. TITLE does not equal Title)



```
<Institutions>
  <Educational>
    <Universities>
      VCU
    </Universities>
    <Public>
      <High_Schools>
        George Wythe
      </High_Schools>
    </Public>
  </Educational>
</Institutions>
```

Well-Formed Document



Valid XML Documents

- A prolog (preface) that contains a
 - Reference to the associated DTD (Document Type Definition)
 - A version statement
 - A style sheet file link
 - XML document description comments
- Element declaration occurs in DTD
- Well-Formed XML document body
- No browser validation



What XML Developers Want You To Know about XML Document Types

- A **valid** XML document is a “stand-alone” document containing all the information necessary to not only display the document but also do intelligent processing
 - *In other words - it has a DTD!*
- A **well-formed** XML document is a perfectly legal XML document, but does not contain information about structure to allow for intelligent processing
 - *In other words - no DTD!*



Special Keywords



- Content models contain:
 - List of elements
 - Keywords:
 - “#PCDATA” - Parsed Character Data (text entries)
 - “EMPTY” - A leaf element that has no content
 - “ANY” - An element that can contain any other element.
Example: <!ELEMENT name (#PCDATA)>

Occurrence Indicators

- "+" Element must appear once or more times
- "*" Element can appear zero or more times; optional
- "?" Element can appear once or not at all; optional

Example:

```
<!ELEMENT customer_address  
  (name,address,tel*,fax?,email*)>
```

NOTE: An element with no occurrence indicator must appear once and only once

Examples of Errors

- Case mis-match
 - <PUBLIC> George Wythe </Public>
- Incorrect Tag Nesting
 - <Public> <High_School> George Wythe
</Public> </High_School>
- Incorrect data storage (for example)
 - <Institutions> George Wythe
</Institutions> <Public> George Wythe
</Public>
- Syntax violations



DTD (Declaration) Benefits

- Structure enforced
- Useful to populate lists
- Helps the XML processor determine indentation from content
- Reduce the size of documents
 - Default values
 - Fixed values



What are the benefits of using DTDs?

- The structure of a document is enforced by the XML processor as defined in the DTD.
- The application involved accesses this structure which is useful to populate lists.
- The DTD helps the XML processor determine indentation from content.
- Using the DTD can help reduce the size of documents through the declaration of default or fixed values for attributes.
- Ensures that “classes” of documents are uniform in appearance.
- With external DTD subsets, maintenance of only one DTD file is needed for a “class” of documents.

28 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



```
<!DOCTYPE bib [  
<!ELEMENT BIB (BOOK+)>  
<!ELEMENT BOOK (AUTHOR+, TITLE,  
PUBLISHER?, YEAR?)>  
<!ATTLIST BOOK  
isbn CDATA #IMPLIED  
nickname CDATA #IMPLIED>  
<!ELEMENT AUTHOR (#PCDATA):  
<!ATTLIST AUTHOR  
id ID #IMPLIED  
idref IDREF #IMPLIED>  
<!ELEMENT TITLE (#PCDATA)>  
<!ELEMENT PUBLISHER (#PCDATA)>  
<!ELEMENT YEAR (#PCDATA)>  
>
```

**DTD for a
bibliography
example
DTDs define a
grammar for
documents.**

29 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



But let's not linger on DTD's...

- **Because** (according to Len Rosenthal @ www.lazerware.com) **they are not long for this world.**
- A number of companies have made proposals to replace DTD's, and as such the W3C has tasked a subgroup of the XML Coordination Group to decide what to do.
- The subgroup is called the XML Schema Working Group, so the replacement is referred to as XML Schema.



30 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



An XML Challenge: Evolving Standards

- "XML circus of conferences, press coverage and XML developer discussion forums. XML tools vendors and early adopters are being asked to develop and deploy products that are based on proposals that have not yet been ratified as standards.
- Example: XML Schema
 - Near to date of standard ratification
 - Opposition and submission of competing, (some say simpler) proposals
 - Should it be standardized in its current form?
 - Meantime, vendors working to bring XML Schema conformant products to market
 - Unlikely they will encourage significant change

31 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Hysteria is not warranted

The screenshot shows a Safari browser window with the address bar displaying the URL: <http://www.computerworld.com/databasetopics/data/story/0,10801,69907,00.html>. The page features a navigation menu with options like Home, News, Browse Topics, Departments, Services, Subscribe, Events, Store, and Jobs. A prominent banner at the top reads "Cultivate IT to meet business demand" and "Enroll in a free IT Value Management learning series." The main content area displays the article title "GAO says XML not ready for extensive government use" in a red-bordered box. Below the title, it indicates the article is a "News Story by Patrick Thibodeau" dated "APRIL 05, 2002 (COMPUTERWORLD) - WASHINGTON". The article text begins with "XML, the data exchange stender that's being deployed by federal agencies to improve interoperability, isn't a mature standard. And without centralized leadership, XML implementations by various agencies...". A "Related to this topic" section lists several related articles, including "U.S. government, companies warn of critical Oracle flaws" and "SEC: Deadline Delay Signals Sarb-Ox Relief".

32 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



XML Schema

- XML Schema is well-formed XML that allows the user to define datatypes.
- Supports a variety of atomic datatypes, such as string, decimal, and date.
- minOccurs and maxOccurs attributes.
- Distinguishes between sequences and choices - instances of the group element.
- Supports inheritance as part of the more general concept of derivation.
- Defines a grammar for the document.
- However, XML Schema uses XML syntax and is more expressive.

33 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



XML Schema Advantages

- Schemas can incorporate relationship metadata
- Schemas can incorporate business process information
- Incorporates greater control over XML tagging, structure, and data
- Supports definition of custom data domains
- Unlike DTDs schemas are written using XML and are thus manipulated as XML documents
- Tags can inherit definitions and structure from other tag definitions
- Uses XML Syntax – Therefore EXTENSIBLE
- Supports Richer Metadata
- Data Types (WC3-Defined or Custom Built)
- Code Lists
- Regular Expressions
- Minimum / Maximum value constraints
- Logical relationships or relationships to existing data
- Dictionaries and models
- Logical data element pedigree through the use of annotations and types.

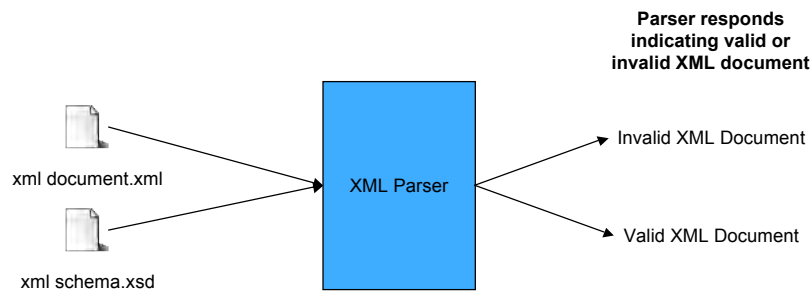
Adapted from Richard Warner - Engelhard Corporation "Managing the XML Data Resource" DAMA International Symposium & Wilshire Meta-data Conference April 30, 2003

34 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



XML Schemas provide structure to XML Documents



35 - datablueprint.com



XML bibliography Schema is more expressive

```
<xsd:schema xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<xsd:element name="BOOK" type="BOOKTYPE"/>
<xsd:complexType name="BOOKTYPE" >
<xsd:element name="AUTHOR" type="xsd:string"
minOccurs="1" maxOccurs="unbounded"/>
<xsd:element name="TITLE" type="xsd:string"/>
<xsd:element name="PUBLISHER" type="xsd:string"
minOccurs="0" maxOccurs="1"/>
<xsd:element name="YEAR" type="xsd:decimal"
minOccurs="0" maxOccurs="1"/>
<xsd:attribute name="isbn" type="xsd:string"/>
<xsd:attribute name="nickname" type="xsd:string"/>
</xsd:complexType>
</xsd:schema>
```



36 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Xpointer, XPath, XLink

- **XPointer**
 - Reference document fragments by pointing to a certain part of a document
 - Addressing specific portions of an XML document - “structure-aware” linking.
 - Uses language of a Family tree called nodes (child, parent, sibling, etc.)
- XPointer uses **XPath** to define fragment identifiers.
 - A fragment identifier can be the
 - value of an attribute of type id
 - a sequence of numbers (for example, /1/4/2/10),
 - a sequence of XPointer expressions (XPointer(...))
- **XLink**
 - Multi-path links to and from different resources
 - “Replacement” for HTML hyperlinking that we do today with. It extends it in new and exotic ways.
- You can use **XLink** without **XPointer**, but not the other way around.

37 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



XPointer

- XPointer
 - Reference document fragments by pointing to a certain part of a document
 - Addressing specific portions of an XML document - “structure-aware” linking.
 - Uses language of a Family tree called nodes (child, parent, sibling, etc.)
- XPointer uses XPath to define fragment identifiers.
- A fragment identifier can be the
 - value of an attribute of type id
 - a sequence of numbers (for example, /1/4/2/10),
 - a sequence of XPointer expressions (XPointer(...))



38 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Review: HTML linking

- `My Home Page`
- Requires a special tag `<a>`
- Uses an attribute (href) to specify the URL
- URL specifies the service type (http), system name (server) and document path - of which the first two are optional
- Can also provide a fragment identifier to go to a named section of a document
`Table of Contents`
`Table of Contents`
- HTML links
 - Go from one single point to one single point
 - Only go in one direction
 - Retrieve the entire document to which it links
 - Require specific knowledge of the target's content
- Only one thing can happen when you click on a linked text
- Linked documents can't be easily reassembled (i.e., next page/previous page)
- Using fragment identifiers requires changes to the linked resource

39 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



What XML Developers Want You To Know about XML

Xlink Capabilities

- Multiple link destinations
- No destinations
- Insert text or images right where you click instead of forcing you to leave the page
- Links point to central database of links rather than pages themselves

40 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



XLink Types

- Simple
 - defines link resource, semantics and behavior
- Extended
 - defines link semantics
- Locator
 - defines remote resource and semantics
- Arc
 - defines traversal rules, semantics and behavior
- Resource
 - defines semantics of resource
- Title
 - provides a title describing the link

41 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



XLink Attributes

- Type - defines link type
- Href - defines resource location
- Role - identifies resource to computer
- title - identifies resource to human
- show - defines how resource will appear
- actuate - defines when resource will appear
- from - defines traversal paths
- to - defines traversal paths



XLink Type / Attribute Grid

Attribute	Link Type					
	simple	extended	locator	arc	resource	title
type	R	R	R	R	R	R
href	O	X	R	X	X	X
role	O	O	O	O	O	X
title	O	O	O	O	O	X
show	O	X	X	O	X	X
actuate	O	X	X	O	X	X
from	X	X	X	O	X	X
to	X	X	X	O	X	X

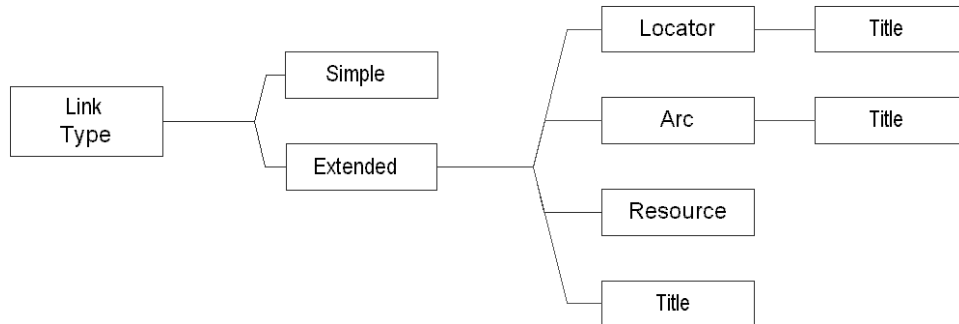
R = Required

O = Optional

X = Not Allowed



Parent-Child Relationships



44 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



XLinking Example

- Using bi-directional links
 - Document being linked to - can know that it is linked to
- Using links with more than one target
 - When wanting to automatically redirect to mirror site during "rush hours"
- Can use metadata about the link to specify what the link points to



45 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



XSL Motivation

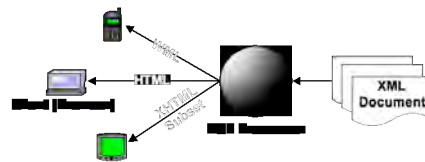
- XML isn't intended to describe how content should be formatted. Instead describes the roles that content plays in a document
- The separation of form from function makes XML documents portable. Write one XML document and then write multiple style sheets
- XML tag set is virtually unlimited - XML is more complex and powerful than HTML - requires a robust and dynamic style tool
- Cascading Style Sheets (CSS) is not robust enough for XML

46 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Introducing XSL



- XSL currently works in two different capabilities:
 - As a transformation language (XSLT, the T stands for transformation) for converting documents written in one XML DTD into another DTD
 - As a formatting language (XSL) for defining how XML documents should be displayed
- XSL style sheets are XML documents with tags, attributes, and all the other standard XML syntax tools
- Current usage of XSL focuses on the transformation function. XML developers use XSL to convert XML documents into HTML or other formatted documents for display

47 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



XSL Specifications

- Extensible Stylesheet Language (XSL) has two distinct vocabularies:
- XSL Transformations (XSLT)
 - Provide a rule-based framework for selecting and processing document content and restructuring it to create new documents
- XSL Formatting Objects (XSL Fos)
 - Assign formatting properties to the result of an XSLT transformation and visually render its result to a wide range of print and web-based devices.

48 - datablueprint.com

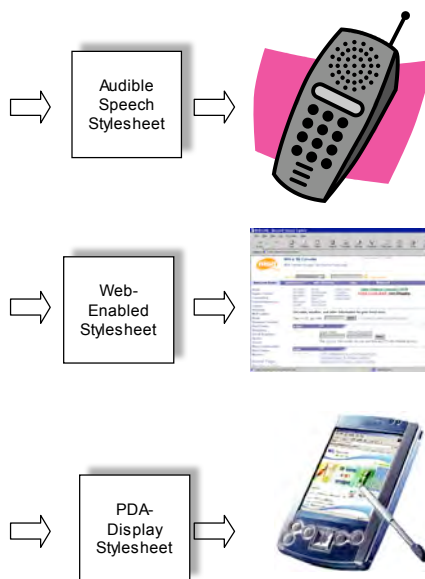
© Copyright 2/27/05 by Data Blueprint - all rights reserved!

[adapted from <http://computer.org>]



XML Stylesheet Capabilities

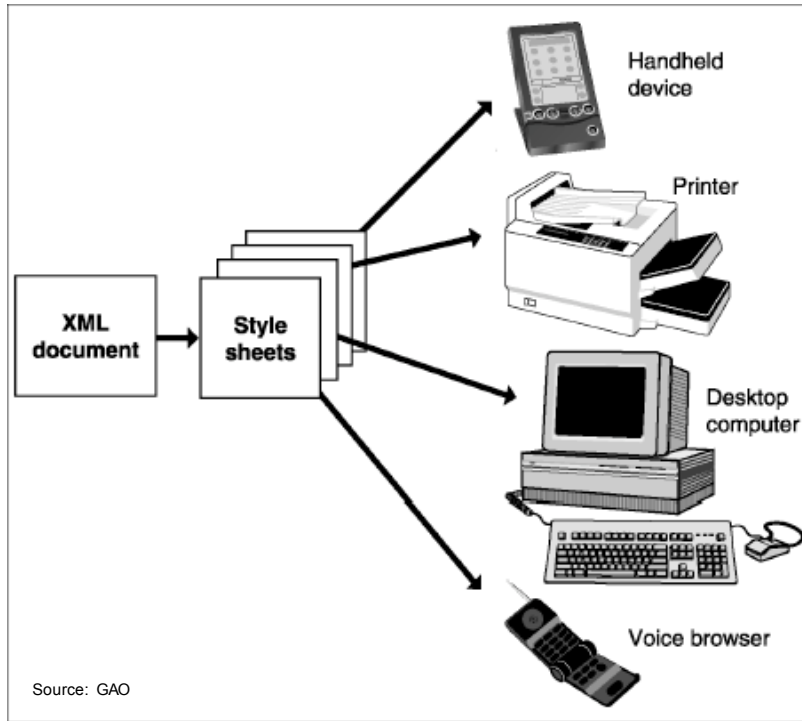
```
<movie>
  <title>Star Trek Insurrection</title>
  <star>Patrick Stewart</star>
  <star>Brent Spiner</star>
  <theater>
    <theater-name>Multiplex 2000</theater-name>
    <showtime>1415</showtime>
    <showtime>1630</showtime>
    <showtime>1845</showtime>
    <showtime>2315</showtime>
    <price>
      <adult-price>8.50</adult-price>
      <child-price>5.00</child-price>
    </price>
  </theater>
  <theater>
    <theater-name>Bigscreen 1</theater-name>
    <showtime>1900</showtime>
    <price>
      <adult-price>6.00</adult-price>
    </price>
  </theater>
</movie>
<movie>
  <title>Shakespeare In Love</title>
  <star>
```



49 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!

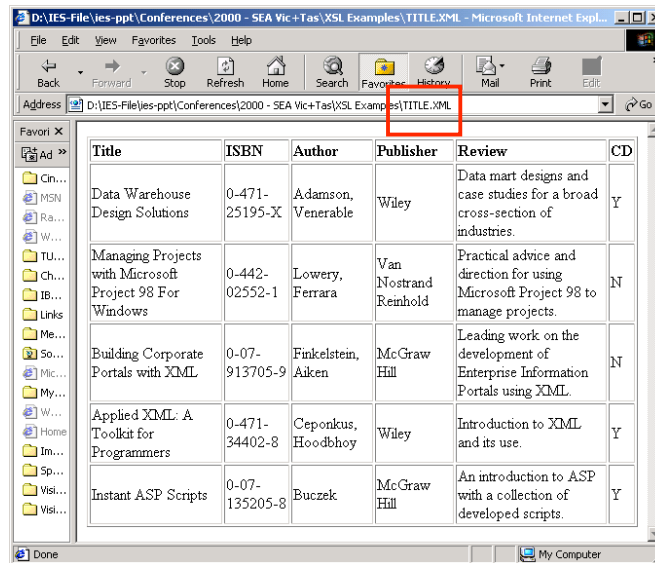




Style Sheet Applications



Internet Explorer 5.0 Example



Internet Explorer 5.0 Example

Title	ISBN	Author	Publisher	Review	CD
Data Warehouse Design Solutions	0-471-25195-X	Adamson, Venerable	Wiley	Data mart designs and case studies for a broad cross-section of industries.	Yes
Managing Projects with Microsoft Project 98 For Windows	0-442-02552-1	Lowery, Ferrara	Van Nostrand Reinhold	Practical advice and direction for using Microsoft Project 98 to manage projects.	No
Building Corporate Portals with XML	0-07-913705-9	Finkelstein, Aiken	McGraw Hill	Leading work on the development of Enterprise Information Portals using XML.	No
Applied XML: A Toolkit for Programmers	0-471-34402-8	Ceponkus, Hoodbhoy	Wiley	Introduction to XML and its use.	Yes
Instant ASP Scripts	0-07-135205-8	Buczek	McGraw Hill	An introduction to ASP with a collection of developed scripts.	Yes

52 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Internet Explorer 5.0 Example

Title	ISBN	Author	Publisher	Review	CD
Data Warehouse Design Solutions	0-471-25195-X	Adamson, Venerable	Wiley	Data mart designs and case studies for a broad cross-section of industries.	
Managing Projects with Microsoft Project 98 For Windows	0-442-02552-1	Lowery, Ferrara	Van Nostrand Reinhold	Practical advice and direction for using Microsoft Project 98 to manage projects.	
Building Corporate Portals with XML	0-07-913705-9	Finkelstein, Aiken	McGraw Hill	Leading work on the development of Enterprise Information Portals using XML.	
Applied XML: A Toolkit for Programmers	0-471-34402-8	Ceponkus, Hoodbhoy	Wiley	Introduction to XML and its use.	
Instant ASP Scripts	0-07-135205-8	Buczek	McGraw Hill	An introduction to ASP with a collection of developed scripts.	

53 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Cool XSL features

- Element matching
 - By context
 - By wildcard
 - By attributes
- Actions
 - Somewhere between CGI and SSI
- Macros (named actions)
- Scripting



54 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Introducing XSLT (XSL Transformation)

- When you build an XSL style sheet, you take each part of a document that you want to display, fit it into a formatting object, and then describe how that formatting object should appear
- XSLT provides a set of rules to convert documents described by one set of tags to documents described by another set of tags. The two sets of tags don't have to look alike
- To write an XSLT style sheet, you simply identify an element in one document and specify how it should be described using a different tag or set of tags. You can grab entire elements or just an element's content. You can even reference attribute values and turn them into tag content or vice versa.
- XSLT is an official recommendation, unlike XSL, which is still a working draft

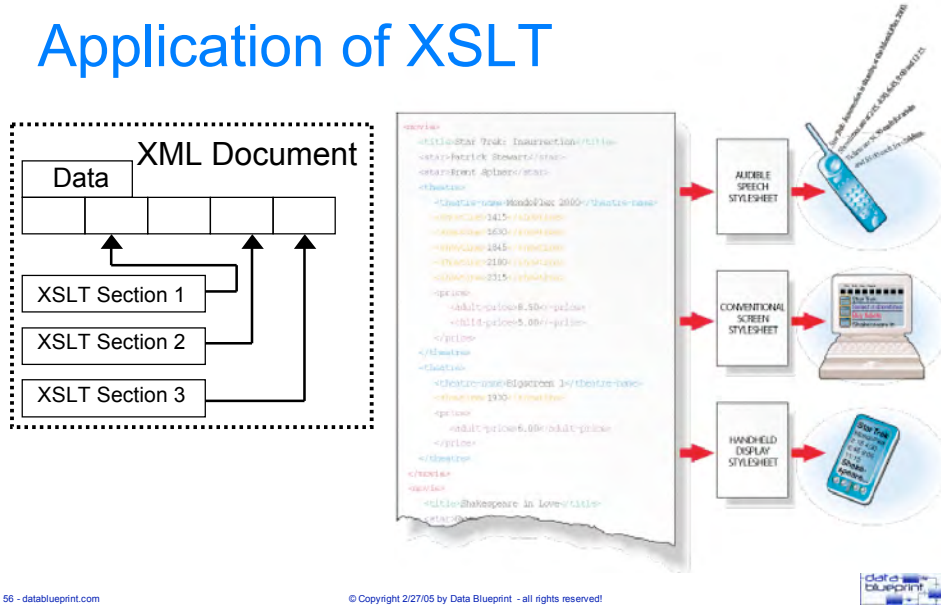
55 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



XML Enabled Reuse via XSLT

Application of XSLT

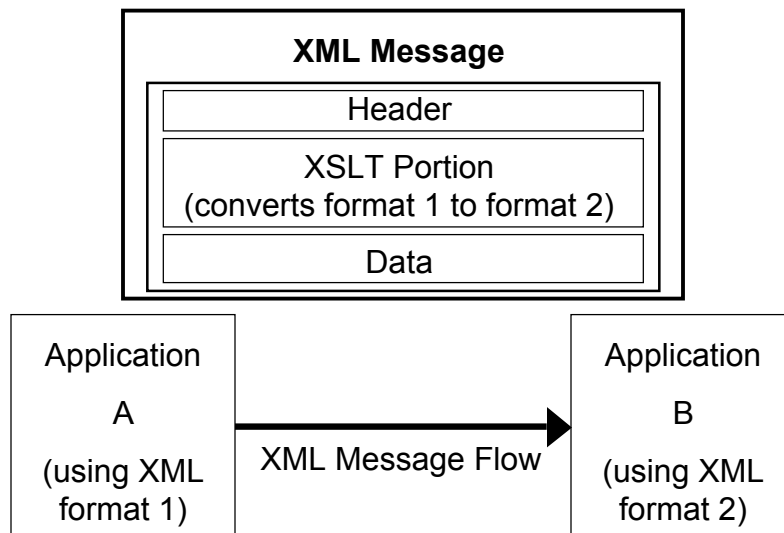


56 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Reconciling XML Formats



57 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



XSLT Capabilities

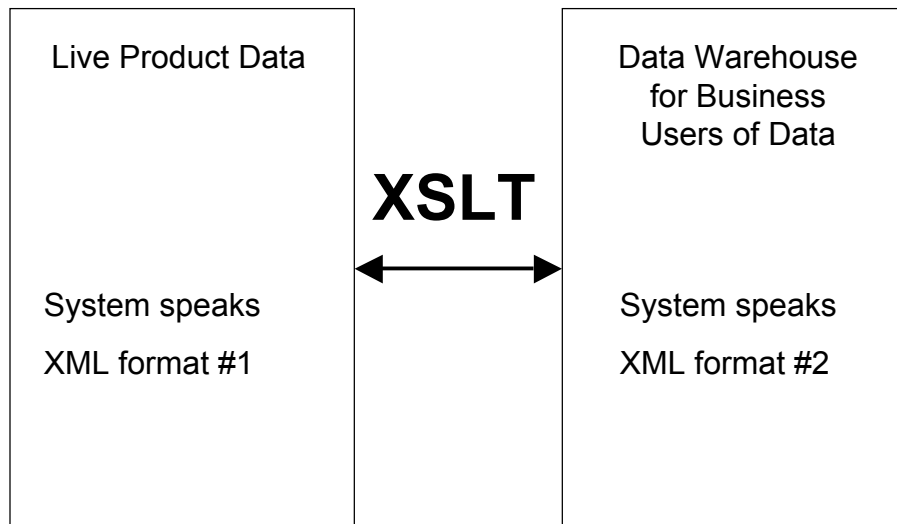
- Provides rules to convert document appearance
- Create a formatting object
- Describe how that formatting object should appear
- Convert entire elements
- Convert just an element's content
- Reference attribute values and convert to tag content or vice versa
- Translations between systems
- Separating data structure from presentation
- Helps deliver the XML promise of a "gateway" data format
- Expands XML's ability to process itself in addition to structuring data

58 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Example XSLT Use: Intrasystem translation

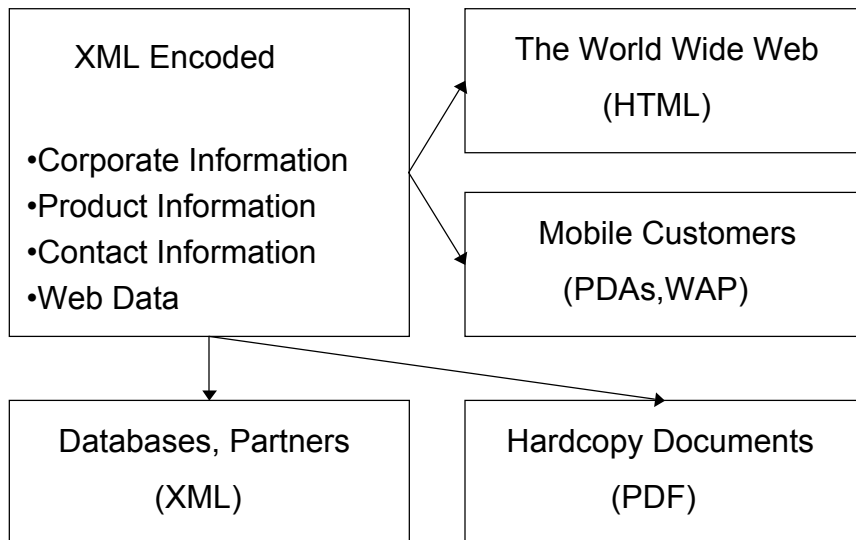


59 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Example XSLT Use: Inter-organizational translation

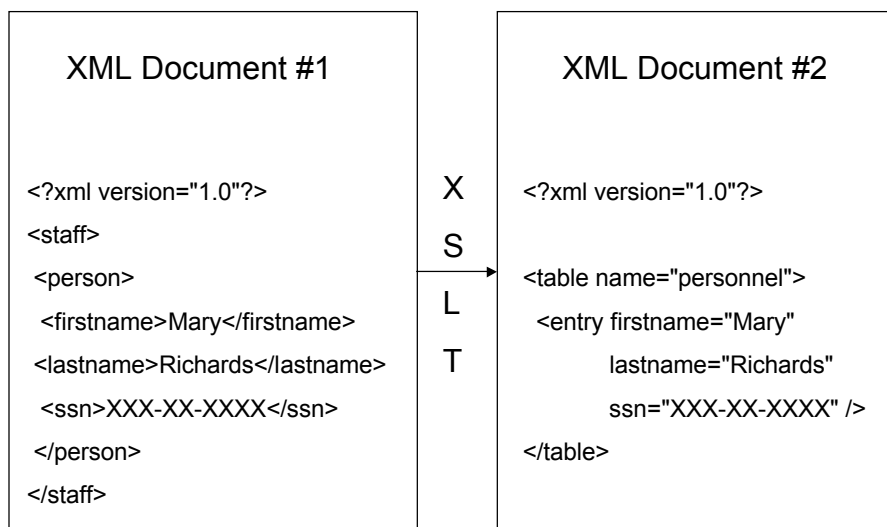


60 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



XSLT Provides XML to XML Conversion

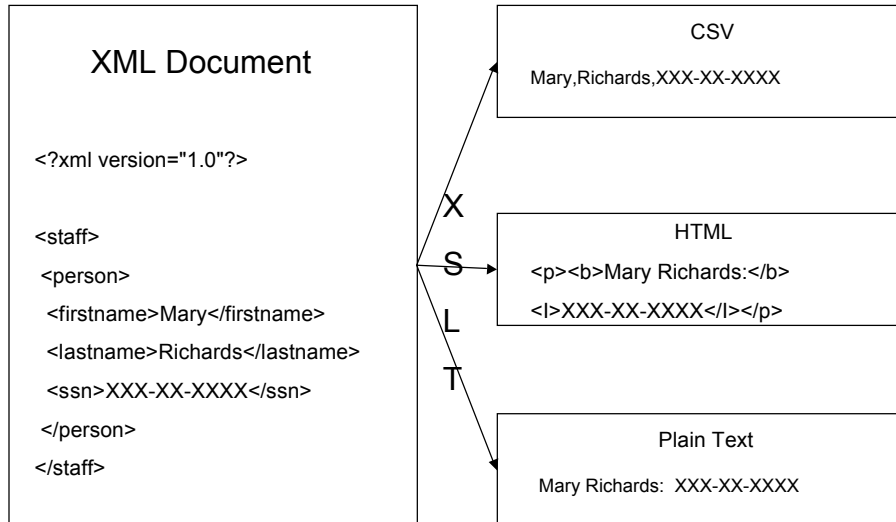


61 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



XSLT Provides XML to (other) Conversion



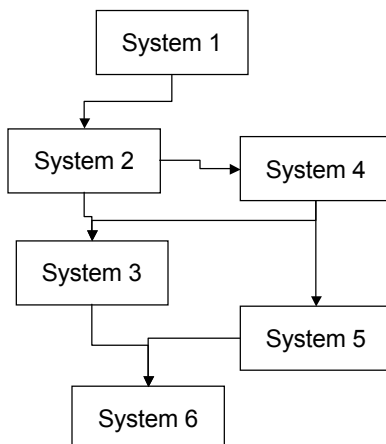
62 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



XML-Based Meta Data Management

Existing

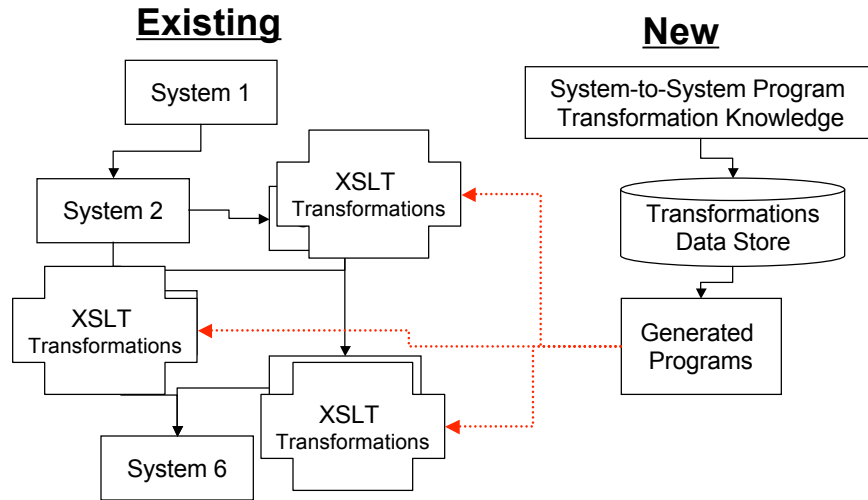


63 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



XML-Based Meta Data Management

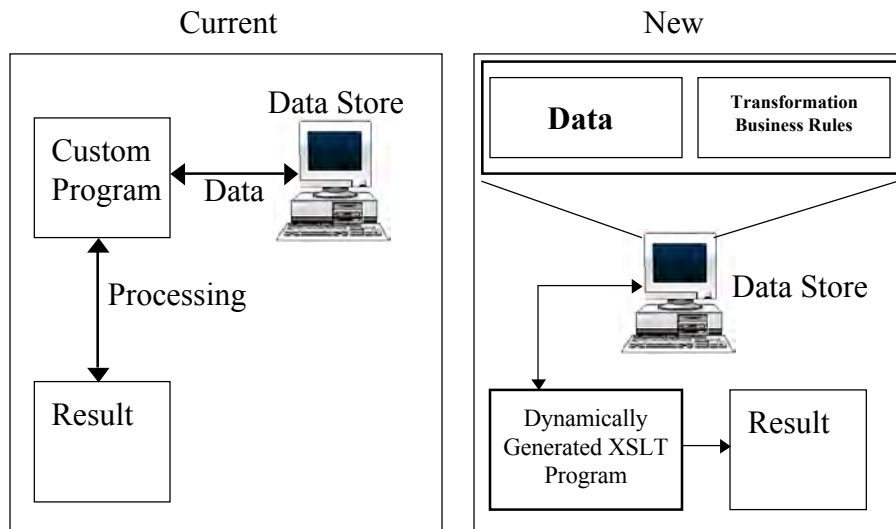


64 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Program Generation

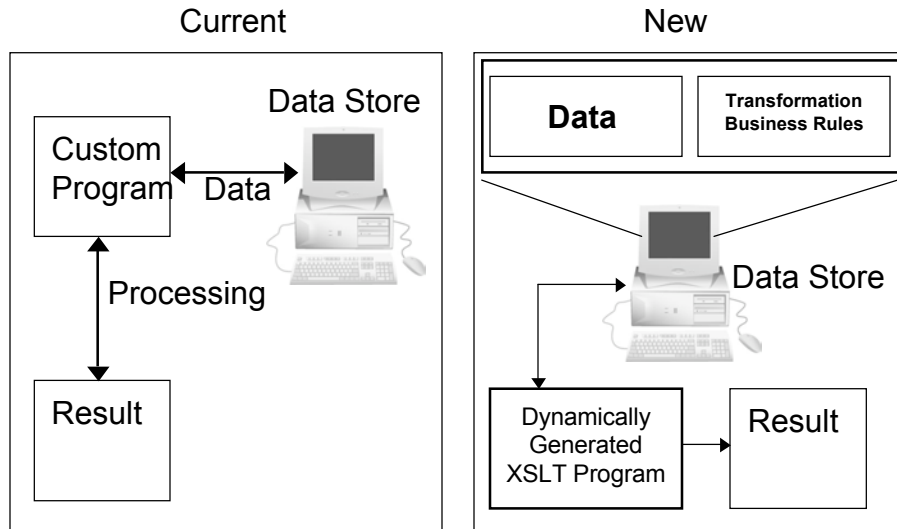


65 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Program Generation



66 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



XSLT is Written in XML

```
<xsl:template match="my-element">
  <xsl:variable name="x">25</xsl:variable> <!-- Set variable x=25 -->
  <xsl:variable name="y">50</xsl:variable> <!-- Set variable y=50 -->
  <xsl:value-of select="$x + $y"/> <!-- Output 25 + 50 -->
</xsl:template>
```

- XSLT is compatible with other XML standards
- XSLT programs can be processed by existing XML tools
- Programs themselves can be treated like data



67 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



XPath

- XPath describes how to move about a document
 - Locate and describe a section of a document targeted for transformation
 - XPath expressions are used to select a node or a whole series of nodes
 - XSLT Applied
 - Transformed from one XML tag set to another or
 - Have styles applied to them
- XPath specification (www.w3.org/TR/xpath) is all about naming the XML paths that run through a document and providing a concise language to describe directions for how to get from one place to another in an XML document.
- So why do you need to know XPath?
 - To find your way to and describe a section of a document that needs to be transformed. This task involves XSLT
 - To be able to point to a certain part of a document. This task involves Xpointers
- XPath uses the language of a family tree called nodes (child, parent, sibling, etc.) as explained in the Document Object Model (DOM)

68 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Xpaths - Nodes

```
<?xml version="1.0" ?>
<!-- clients.xml-->
<Clients>
<!-- This is the root element of clients.xml-->
  <Client id="c1">
    <Name>Jon Smith</Name>
    <Phone type="home">440-123-3333</Phone>
    <Fax>440-123-3334</Fax>
    <Email>jon@acme.com</Email>
  </Client>

  <Client id="c2">
    <Name>Bill Jones</Name>
    <Phone type="cellphone">330-124-5432</Phone>
    <Fax>440-123-5433</Fax>
    <Email>bjones@someinc.com</Email>
  </Client>

  <Client id="c3">
    <Name>Matt Brown</Name>
    <Phone type="work">220-125-1234</Phone>
    <Phone type="cellphone">233-344-4455</Phone>
    <Phone type="home">234-567-8910</Phone>
    <Fax>2200-125-1235</Fax>
    <Email>matthew@hotstuff.com</Email>
  </Client>
</Clients>
```

- Document node contains three children
 - Version declaration
 - Comment node
 - Root element node
- The root element node is Clients
- The Clients node has four children
 - One comment node
 - Three Client nodes
- The third Client node has six children
 - Name
 - Phone type=work
 - Phone type=cellphone
 - Phone type=home
 - Fax
 - Email
- The parent of every phone node is a Client node
- The grandparent of every phone node is the Clients node

69 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Xpaths – Directions & Destinations

Every trip has three parts:

1. The starting point – is usually from the root node of the document. Sometimes, however, it may be another node in a document. Where the journey starts is called the Context node
2. The journey itself – consists of both a direction and a number of steps. XPath describes each of these steps using its syntax. Each step is separated by a forward slash (/)
3. The destination – can either be a single node or a collection of nodes

70 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Xpaths – LocationPaths with Expressions



- Some simple examples of selection paths

```
<?xml version="1.0" ?>
<!-- clients.xml-->
<Clients>
<!-- This is the root element of clients.xml-->
  <Client id="c1">
    <Name>Jon Smith</Name>
    <Phone type="home">440-123-3333</Phone>
    <Fax>440-123-3334</Fax>
    <Email>jon@acme.com</Email>
  </Client>

  <Client id="c2">
    <Name>Bill Jones</Name>
    <Phone type="cellphone">330-124-
5432</Phone>
    <Fax>440-123-5433</Fax>
    <Email>bjones@someinc.com</Email>
  </Client>

  <Client id="c3">
    <Name>Matt Brown</Name>
    <Phone type="work">220-125-1234</Phone>
    <Phone type="cellphone">233-344-
4455</Phone>
    <Phone type="home">234-567-8910</Phone>
    <Fax>2200-125-1235</Fax>
    <Email>matthew@hotstuff.com</Email>
  </Client>
</Clients>
```

71 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



- ♦ **Child :: Clients**
The axis is *child* and the node test is *Clients*. This selects all the child nodes of the context node named *Clients*
- ♦ **Child :: ***
The axis is *child* and the node test is *. This selects all the child nodes of the type element from the context node. Here it selects all the nodes called *Clients*
- ♦ **Child :: Clients[position()=1]**
The axis is *child* and the node test is *Clients*. The expression selects the first child node named *Clients* of the context node.
- ♦ **Child :: Clients[position()=1]/child::Client**
The new context node is the node selected in the preceding step. The next step selects all the Client element children of the root element
- ♦ **Child :: Client[attribute::id="c3"]**
The axis is *child* and the node test is *Client*. The expression is itself another LocationPath with an axis of attribute and a node test of *id="c3"*. This selects all the child elements of *Clients* and the last *Client* element is selected

RDF: Resource Description Framework

XML-based method of representing and communicating arbitrary metadata

- W3C Recommendation
- Represents information as a graph, containing triples:
 - **Subject** (A customer)
 - **Predicate** (includes data about)
 - **Object** (an address)

Like any other tool, you can drop it on your foot if it is used incorrectly.



Resource Description Framework (RDF) is a foundation for processing metadata; it provides interoperability between applications that exchange machine-understandable information (<http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>)

72 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Resource Description Framework (RDF)

- Problems with the Web
 - Hardly any metadata in existence
 - Searches engines use “screen scraping” and “brute force” tactics
 - Only Yahoo! uses metadata
- RDF Proposal
 - Using XML, create more effective and efficient search engines

73 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Industry Initiatives

Standard DTDs

- MathML
- SpeechML
- Synchronized Multimedia Integration Language (SMIL)
- eXtensible Business Reporting Language (XBRL)
- Wireless Markup Language (WML)
- Voice Markup Language (VoiceML)
- Etc etc etc
- XML.ORG
- BizTalk Foundation

74 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Resource Description Framework (RDF)

- What is RDF?
 - XML-based protocol that aids in the sharing and interchange of metadata
 - World Wide Web Consortium's (W3C) effort to create standards for metadata interchange
- What does RDF have to do with XML?
 - Can easily be converted to XML
 - Helps uniquely identify data in a data structure
 - Allows different software to process the same metadata
 - XML essential to RDF's goals

75 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Metadata references in RDF

- Are built on the following:
 - A Resource
 - Anything that can have a URL:
 - Web pages
 - Individual XML elements
 - A Property
 - A resource that has a name and can be used as a property:
 - Examples: Title, Artist, Producer, etc.
 - A Statement
 - Uniquely identifies a particular data element
 - Consists of:
 - » Resource
 - » Property
 - » Value – can be another resource

76 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



What XML Developers Want You To Know about XML

Resource Description Framework (RDF) Capabilities

- This should do for the Web what catalogue cards did for library books
- RDF metadata will make retrieval far faster and more accurate
- *"We expect that RDF will achieve a typically astonishing Internet growth rate once its power becomes apparent."*



Extracted from: XML and the Second-Generation Web: The combination of hypertext and a global Internet started a revolution. A new ingredient, XML, is poised to finish the job by Jon Bosak and Tim Bray Scientific American May 1999

77 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Simple Object Access Protocol (SOAP)

- What Is SOAP?
- Minimal set of procedures for invoking code using XML and HTTP
- Application of XML (and XML Schemas) to HTTP
- Invents no new technology
- Leverages effort already invested in HTTP and XML technologies



What SOAP Does.....

- Defines access to
 - services, objects, and servers
 - platform-independent manner
- Glue between heterogeneous software components
- Microsoft Describes SOAP as a lightweight protocol for information exchange in a decentralized, distributed environment.
- An XML-based protocol that consists of three parts:
 - Envelope that defines framework for what is in a message and how to process it
 - Set of encoding rules for expressing instances of application-defined datatypes
 - Convention for representing remote procedure calls and responses
- Can be used in combination with a variety of other protocols



SOAP Interoperability

- Web sites are developed on a variety of different platforms
 - UNIX, Linux, NT
- In different languages
- Using different object models
 - COM, CORBA, JAVA, RMI
- Public Web site services
- Internal Web site services
- Invoke remote objects on different servers

80 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Impact of SOAP

- The SOAP goal: "Keep SOAP simple and make it easy to implement"
- Using HTTP model, SOAP is highly scalable
- SOAP requests take advantage of
 - HTTP authentication mechanisms
 - SSL for secure channel communications (using HTTPS)

81 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



SOAP Specification

- The client requests server to invoke an object
- Server returns results
- Messages formatted in XML, and encapsulated in HTTP messages
- Works with existing Internet infrastructure
 - routers
 - firewalls
 - proxy servers
- Firewalls impede distributed application
- SOAP relies on HTTP and most firewalls allow HTTP to pass through
- Firewalls can still block SOAP requests using SOAP HTTP headers

82 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



SOAP Flash Presentation



Downloaded from:
http://searchwebservicestechtarget.com/searchWebServices/downloads/what_is_soap.swf

83 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Document Object Model (DOM)

- Programming interface for XML
- Layer of abstraction
- Shields syntax of the data
- Supports Object Oriented Design
- DOM makes an XML document a tree
- Allows "searching" documents like searching a tree
- DOM makes information about any node available
- Many XSLT processors use the DOM to model documents

84 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



DOM Specifications

- Provides a general API for manipulating the contents of XML and HTML documents.
- Make it easy for programmers to access and modify the structure, content, and style of XML and HTML data from within a variety of products using different internal data structures and formats.
- The programming model remains the same regardless of which programming language is being used.
- The DOM API presents a tree-like view of a document as a hierarchy of tags.
- It provides methods for traversing a document's structure.
- It attempts to provide a single programmatic abstract model that can then be applied to numerous types of physical data structures to enable programmers to write applications which work properly on all browsers, servers, and platforms.

85 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!

[adapted from <http://computer.org>]



XSLT & DOM

- XML Document parsed to generate a Document Object Model
- XSL and XSLT style sheets
 - XML documents
 - Parsed and listed as DOM
- DOM Level 1 Core
 - Defines the methods and properties
- DOM Level 1 HTML
 - Extends the Level 1 Core to provide Dynamic HTML

86 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



DOM Applications

- Applications that manipulate XML data need to use DOM interfaces
 - e.g, a script might need to calculate tax on items at an online shopping site
- DOM is used heavily to process XML documents server-side
 - e.g, applying the business logic to complete a transaction
- DOM provides translation mechanism from legacy databases to XML



87 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Architecture Description Markup Language (ADML)

- "A Standard XML-Based Language for Describing Software Architectures to Enable Their Representation, Evaluation, and Analysis"
- "The Open Group, a vendor and technology-neutral consortium dedicated to enterprise interoperability, today announced the availability of ADML Version 1, the first release of the Architecture Description Markup Language, an XML-based mark-up language for describing **system** architectures. A **software** architecture describes the structural properties of the software, typically the components and their interrelationships and guidelines about their use. ADML provides a means of representing an architecture that can be used to support the interchange of architectural descriptions between a variety of architectural design tools. ADML is based on Acme, a software architecture description language developed at Carnegie Mellon University and the Information Sciences Institute at the University of Southern California. ADML is an XML-based version of **Acme** developed by the Micro-electronics and Computer Technology Consortium (**MCC**) as part of its Software and Systems Engineering Productivity (**SSEP**) project. MCC joined The Open Group Architecture Program Group and its members, including NCR and the U.S. Department of Defense, to develop ADML into an Open Group Technical Standard."

88 - datablueprint.com

Extracted from: <http://xml.coverpages.org/adml.html>
© Copyright 2/27/05 by Data Blueprint - all rights reserved!



There is a Big Problem

- Software development no longer the big issue
 - Software design and development tools no longer key issue
 - many good rapid application development tools
- Today's problem: aligning IT solutions with the business / mission
 - Architecture the key issue (business / technology mapping)
 - developers < \$100k / architects > \$100k
 - major product problems come in the architecture
 - Architects deal in the entire life-cycle
 - yet have no tool to support them throughout that life-cycle
- Existing Tools Don't Meet the Need
 - UML intended for high level software design, not architecture

89 - datablueprint.com

Extracted from: <http://xml.coverpages.org/adml.html>
© Copyright 2/27/05 by Data Blueprint - all rights reserved!



There is Value in a Solution

- This problem is well understood by customers
 - Architectural modeling market much larger than normally perceived
 - Customers WILL recognize the value of architectural tools that interoperate with development cycle & analysis tools
- ADML enables you to provide a solution in your tool
 - Don't, and someone else will
 - Do, and you can enable a whole new value proposition

90 - datablueprint.com

Extracted from: <http://xml.coverpages.org/adml.html>

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Remember SQL...

- SQL created a market for relational databases
 - ADML creates a market opportunity for tools
- SQL drove costs down for database vendors
 - Didn't have to invest in creating a language
 - Didn't have to bear the cost of defending a proprietary language
- Open standard SQL attracted attention
 - SQL attracted consumers through
 - promise of single database language
 - new "open" tools that enable better data management and access
 - SQL attracted academics in areas such as
 - optimization
 - distributed relational databases
 - parallelism

91 - datablueprint.com

Extracted from: <http://xml.coverpages.org/adml.html>

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



To service a wider set of needs placed on industrial grade modeling tools

1. Representation - a model must be captured in a format which insures 'shelf life' and a format usable by multiple tools.
2. Views - a model must have the ability to 'morph' other views, including logical, physical, and organization views.
3. Collaboration - a model must have the ability to be partitioned across geographically dispersed work groups.
4. Repository - a model must have fine-grained persistence.



Various Levels of ADML Support

- Level 1
 - Import/Export valid ADML to and from your tool
- Level 2
 - ADML text editing of objects through your tool
- Level 3
 - ADML integration as encoding mechanism for your objects
 - ADML sensitive repository of encoded objects
- Level 4
 - Automatic ADML generation, interchange and management
 - Automated integrity assessments of ADML models
 - Proven interoperation with other tools
 - Integration with ADML-aware “repository”



Level 1 ADML Support

- Import ADML to your tool (e.g., using XSL/XSLT)
 - All ADML elements are mapped on import
 - Assumes some tool/some one has generated ADML
 - Import is done simply by mapping ADML to internal representation
 - ADML to *.ml
 - Allows your tool to re-use information from other tools



Extracted from: <http://xml.coverpages.org/adml.html>

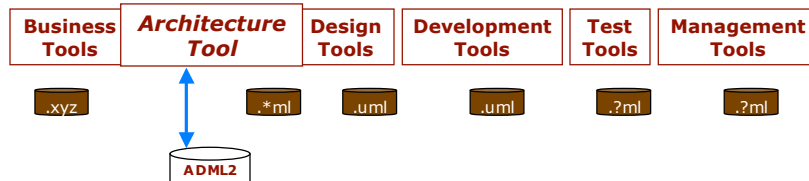
94 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Level 2 ADML Support

- ADML editing of objects through your tool
 - Edit ADML objects directly rather than edit in *.ml then convert
 - right click and edit the ADML (e.g., open XML text editor)
 - Evolve to editing an ADML model



Extracted from: <http://xml.coverpages.org/adml.html>

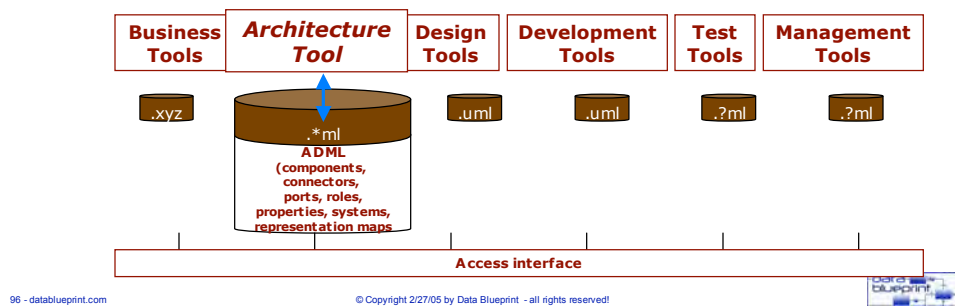
95 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



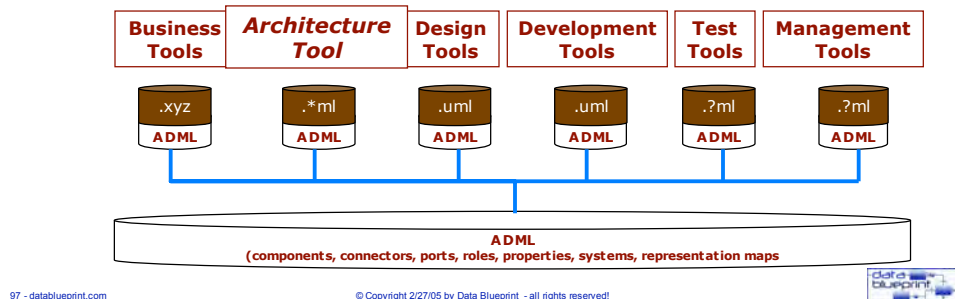
Level 3 ADML Support

- ADML integration as encoding mechanism for your objects
 - Provide ADML editing at same level as *.ml editing
 - Provide ADML access to information per ADML constructs
 - Implies an adoption of the ADML metamodel
 - ADML sensitive repository of encoded objects
 - Provide open access to ADML for other tools



Level 4 ADML Support

- Automatic ADML generation and management
 - Automated integrity assessments of ADML models
 - Proven interoperation with other tools
 - Integration with ADML-aware “repository”



Benefits to Stakeholders

- Customer Architects: Reduced time, cost and risk
 - Procuring effective architecture tools
 - Developing an architecture
 - Procuring products to implement an architecture
- Tools Vendors: Bigger market, bigger market share
 - Supporting interchange of architecture definitions
 - Export / import of building block definitions to / from the Building Blocks Information Base
- IT Solutions Vendors: More effective response to architecture-based procurements
 - Reducing the cost of bidding
 - Securing greater share of procurements



Summary

- You can benefit from ADML
 - Redefined and expanded market
 - significantly greater dollar value
 - More HIGH-VALUE customers
 - Satisfaction of current customers
- You can hedge your bets
 - Aggressive or cautious strategies
 - \$\$profitable either way\$\$
 - Work through your roadmap
- However you place the bet, the odds are good!



Web Services

- Can range from a movie review to a real time weather advisory to a hotel and airline booking package
- Based on assembly of services that may or may not incorporate object technology
- Both legacy and object systems can "plug into" the framework



WSDL

- Web Services Description Language
- Describes how software package can connect and interact with Web services
- Supports building Web-based computing systems that target computer programs rather than humans
- XML-based syntax describes Web service access specifics (such as type and number of parameters passed and type/structure of result returned)



W3C Recommendation version 1.1

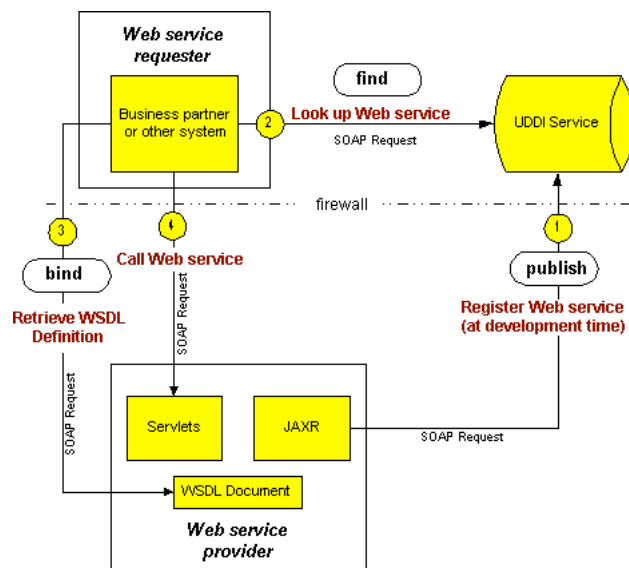
"Once you have developed a Web service, you publish its description and a link to it in a UDDI (Universal Description, Discovery and Integration) repository so that potential users can find it. When someone thinks they want to use your service, they request your WSDL file in order to find out the location of the service, the function calls and how to access them. Then they use this information in your WSDL file to form a SOAP (Simple Object Access Protocol) request to your computer."

102 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



WSDL Execution

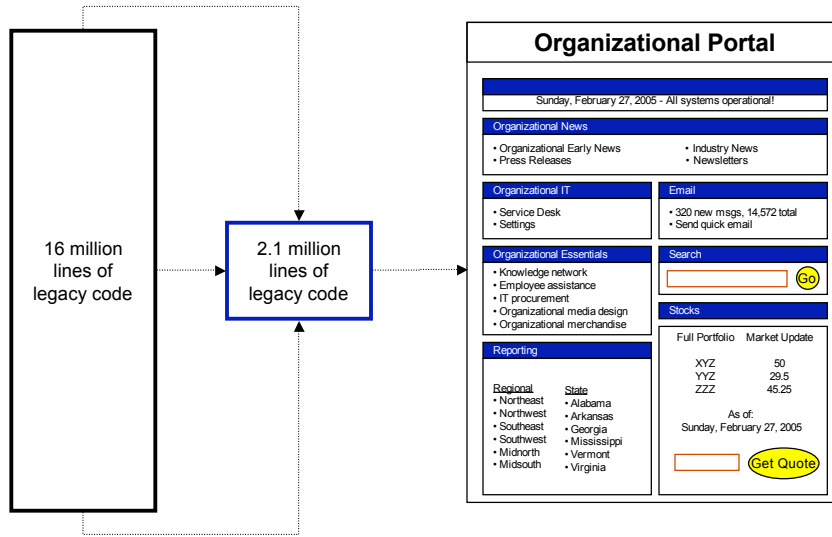


103 - datablueprint.com

- W3C Recommendation version 1.1 - Current March 2001



Evolving applications from stove pipe to web-service-based architectures

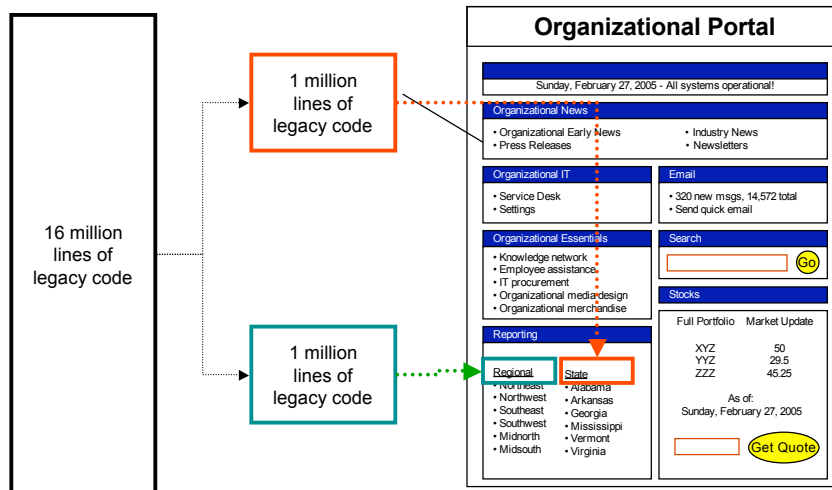


104 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Alternative and more sophisticated means of application reengineering legacy applications into an XML-based portal context



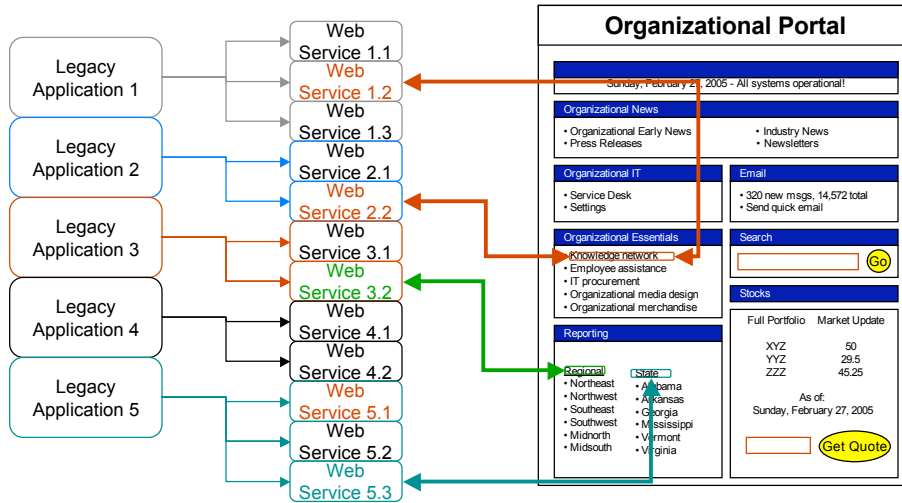
Show one application split into two web-services and another that is combined with functionality of an existing web-service

105 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Legacy Systems Transformed Into Web-services Accessed Through a Portal



106 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Portal/Web Services

Integration Possibilities

- User Interface
- Business Process
- Application
- Data

AV Component

- Well defined components
- Self-contained
- No interdependencies



Analogy derived from D. Barry "Web Services" Intelligent Enterprise 10/10/03 pp. 26-47

107 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



adXML

Sign Up Now!

What is adXML?

adXML News

Download

Links

[-A-@-]

adXML.org is an international, open standard organization, which is defining an advertising XML schema for both on-line and off-line media. adXML was originally conceived and developed by Mediaplex, Inc. as part of the infrastructure to automate the on-line advertising market. adXML now is being expanded to address all advertising modalities and is gathering industry backing as an open international standard.

adXML.org - XML for Advertising

Contact Us at: info@adxml.org

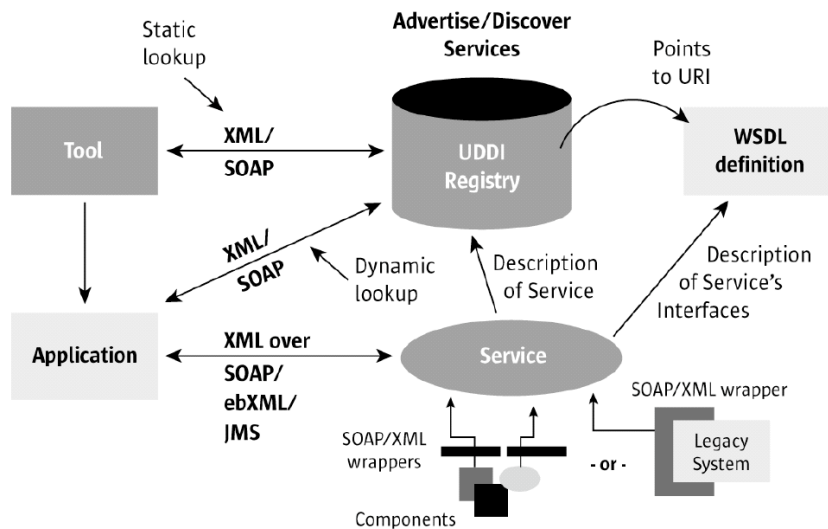


108 - datablueprint.com

© Copyright 2/27/05 by Data Blueprint - all rights reserved!



Use of Services and Registries



110 - datablueprint.com

Sun ONE Architecture Guide <http://www.sun.com/software/sunone/docs/arch/S1-arch-guide.pdf.zip>

