



Data Services World : WADL and URIs as Database Types

Mark Hapner
Distinguished Engineer
Sun Microsystems



Agenda

- Web Application Description Language
 - > A walk through
 - > A WADL for Atom Publishing Protocol
- URI as a database type
 - > Oracle URI types
 - > Further thoughts

WADL

An increasing number of Web-based enterprises (Google, Yahoo, Amazon, etc.) are providing HTTP-based applications that provide programmatic access to their data.

WADL is designed to provide a machine processable description of these applications.

WADL Elements

- Grammars
 - > The schemata describing the information a web app produces and consumes
- Resource Types
 - > An app's URI types and associated methods
- Representations
 - > An app's input/output data representations
- Resources
 - > An app's URI 'dialect'

RelaxNG

During the development of RELAX NG, XML structure validation remained the focus. No compromises were made in deference to other features.

The result is that RELAX NG appears to be the logical successor of XML DTDs and the best tool available to validate the structure of XML documents. RELAX NG's expressive power is such that virtually any XML vocabulary may be described with RELAX NG. That isn't true of W3C XML Schema, nor of DTDs.

Perhaps most important for people who have to write schemas, RELAX NG is also very simple: because it does less, the syntax is intuitive. It has been kept simple. It isn't cluttered with complex limitations that take too much time to learn and remember.

A wander through WADL

Atom Publishing Protocol

- IETF Standard for the web app used to publish and access Atom feeds
- A nice example for demonstrating the ability of WADL to describe a web app
- Atom is another standard that describes its structure with RelaxNG

Atom PP WADL

- Representations
 - > Service Document
 - > Category Document
 - > Feed Document
 - > Entry Document
 - > Media Link Entry Document
 - > Media
- Resource Types
 - > Service
 - > Category
 - > Collection
 - > Feed
 - > Entry
 - > Media Link Entry
 - > Media

A wander through Atom PP WADL

My Atom App

- An Atom service resource I provide
- How I've used the general Atom PP WADL to describe it

A wander through My-Atom-App WADL

Oracle URI Types

New Datatypes Store Uri-references

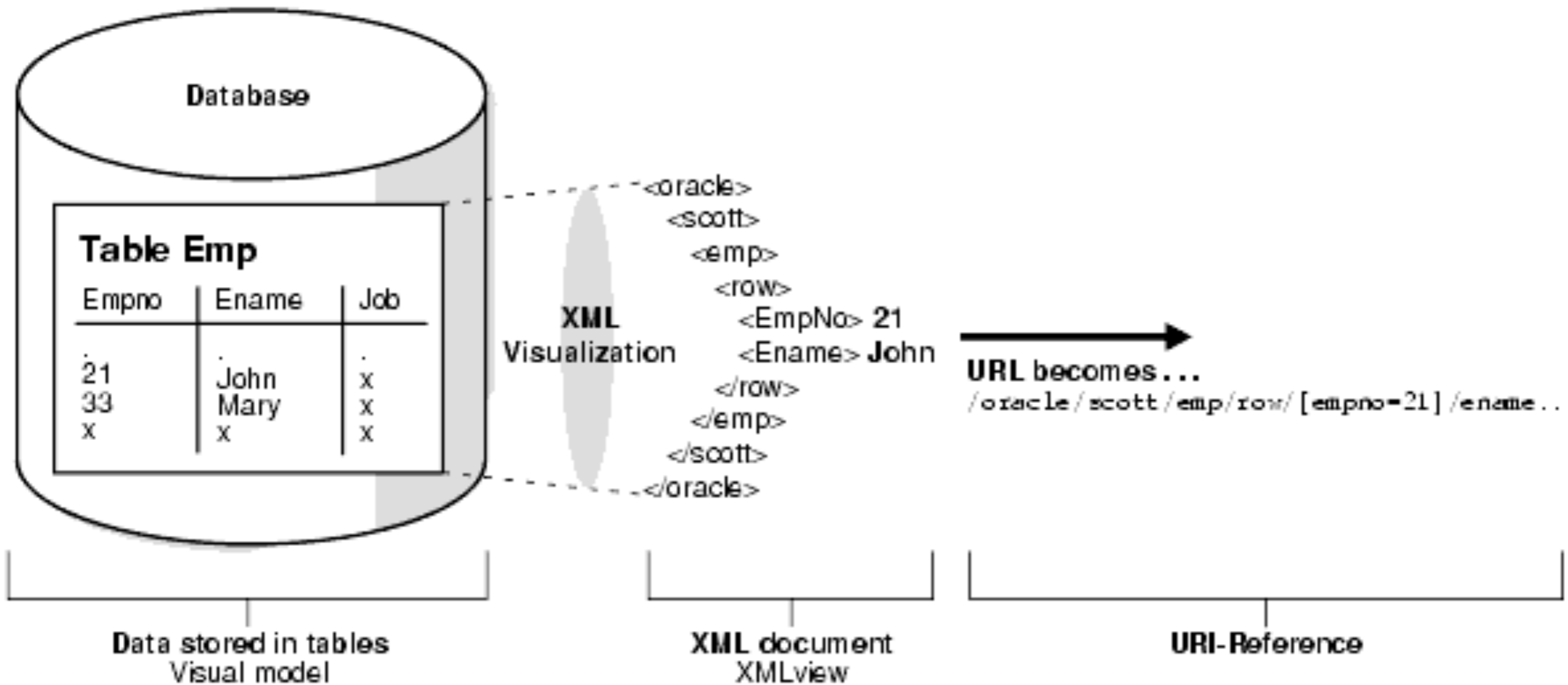
- UriType
 - > An abstract object type that can store instances of HttpUriType or DBUriType
- DBUriType
 - > Intra-Database references
- HTTPUriType
 - > Implements the HTTP protocol for accessing remote pages
- UriFactoryType
 - > A UriType conversion and extension facility

HTTPUri-refs

<i>Create table with URIType</i>	<pre>CREATE TABLE uri_tab (poUrl SYS.UriType, poName VARCHAR2);</pre>
<i>Insert an HTTPUri</i>	<pre>INSERT INTO uri_tab VALUES (sys.UriFactory.getUri('http://www.acme.com/po'), 'P0-256');*</pre>
<i>GET the HTTPUri's representation</i>	<pre>SELECT e.poUrl.getClob() FROM uri_tab e WHERE e.poName='_P0-256';</pre>
<i>Get the HTTPUri value</i>	<pre>SELECT e.poUrl.getURL() FROM uri_tab e WHERE e.poName='_P0-256';</pre>

*The UriFactory strips of the scheme but retains the 'type'.

DBUri-ref



Using DBUri-refs

- DBUri-refs have performance benefits as you interact directly with the database rather than through a web server
 - > DBUri-refs are scoped to database and session
- Who Needs SQL?
 - > You do not need to know SQL to access data stored in the database
- The same DBUri-ref may give different results based on the session context used, particularly if the PUBLIC path is used
 - > For example, /PUBLIC/FOO_TAB can resolve to SCOTT.FOO_TAB when connected as scott, and resolve as JONES.FOO_TAB when connected as JONES

DBUri-ref Examples

<i>A table</i>	XML	<code>/<schemaname>/<tablename></code>	<code>/SCOTT/EMP</code>
<i>A row</i>	XML	<code>/<schemaname>/<tablename>/ ROW[<predicate_expression>]</code>	<code>/SCOTT/EMP/ROW[EMPNO=7369]</code>
<i>A column within a row</i>	XML	<code>/<schemaname>/<tablename>/ ROW[<predicate expression>]/ <columnname></code> <code>/<schemaname>/<tablename>/ ROW[<preciate expression>]/ <columnname>/ <attribute>*</code>	<code>/SCOTT/EMP/ROW[EMPNO=7369 and DEPTNO =20]/ENAME</code> <code>/SCOTT/EMP/ROW[EMPNO=7369]/ ADDRESS/STATE</code>
<i>A column within a row</i>	TEXT	<code>/<schemaname>/<tablename>/ ROW[<predicate expression>]/ <columnname>/text()</code>	<code>/SCOTT/EMP/ROW[EMPNO=7369]/ ENAME/text()</code>

Turning DBUri's into URLs

- Oracle provides OSE default servlet for serving URLs of the form
 - > `http://machine.acme.com:8080/oradb/SCOTT/EMP/ROW[EMPNO=7369]/ENAME`
 - > Where `http://machine.acme.com:8080/oradb` is the default servlet URL
- Or use any other servlet that can open the database connection
- The problem is that DBUri's are not URLs when retrieved and require special knowledge to convert to URLs
 - > DBUri's are no more universal than Oracle object refs

Further thoughts

- WADL Resource Type as a 'first class' database type
 - > It's value would be a URL
 - > It's methods define the CRUD operations it supports
 - > GET, PUT, POST, DELETE, HEAD
 - > Resource parameters
 - > Input representations
 - > Output representations
 - > Faults
- Support navigation through a representation to execute methods on the resources it contains
- Turn databases into a facility for operating on the web's data

Links

- WADL - <https://wadl.dev.java.net/>
- Oracle URI Types
 - > http://download-east.oracle.com/docs/cd/B10464_04/web.904/b12099/adx06uri.htm
- RelaxNG - <http://relaxng.org/>
- Schematron
 - > <http://xml.coverpages.org/schematron.html#mainRefs>
- RelaxNG Book
 - > <http://books.xmlschemata.org/relaxng/page2.html>
- Atom - <http://atompub.org/>
- Google Data APIs - <http://code.google.com/apis/gdata/>

Data Services World : WADL and URIs as Database Types