DB2 Data Management Software

IBM

e business software

Integration von XML Technologie in die Datenbank DB2

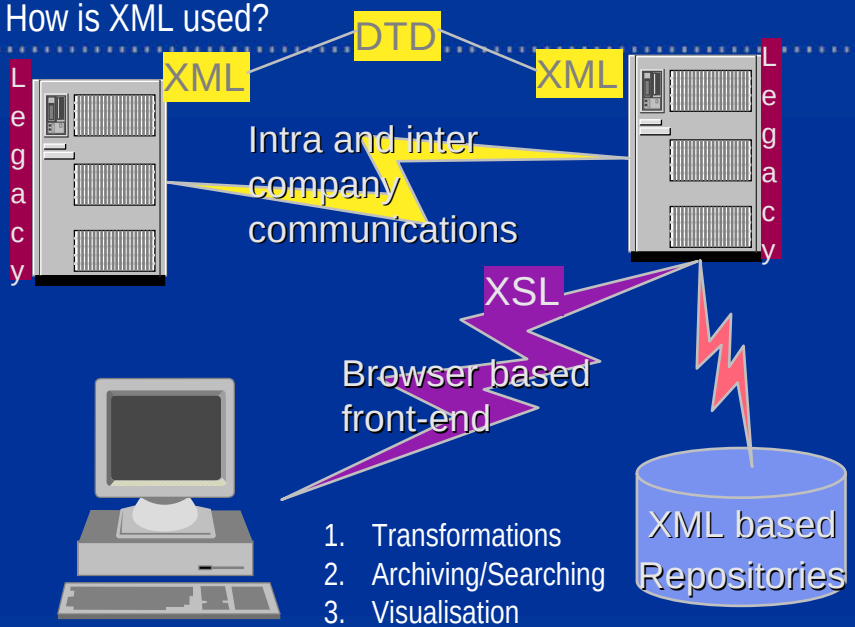
Manfred Päßler, IBM Technical Sales Datamanagement
manfred.paessler@de.ibm.com

IBM Software Group

Agenda

- How XML is used
- Tagging via SQL/XML
- Tagging via table tagging function
- Using XML Extender
- DB2 and Web Services
- Interesting internet adresses

How is XML used?



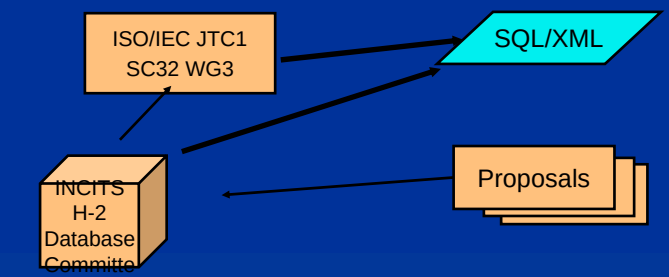
1. Transformations
2. Archiving/Searching
3. Visualisation

Tagging via SQL/XML

- XMLELEMENT function:
  constructs an XML Element out of an Element Expression and/or a list of Attributes
- XMLATTRIBUTE function:
  constructs XML attributes (within XMLELEMENT)
- XMLAGG function:
  returns the concatination of a set of XML data (grouping/ordering with ORDER BY and sort expressions)
- XML2CLOB:
  returns an XML expression (XMLELEMENT and XMLAGG) as a CLOB.

The SQL/XML Background

- SQL/XML: 2nd half of 2000, ANSI H2 and ISO SC32/WG3 approved new project for new part of SQL: "Part 14, XML-Related Specifications (SQL/XML)"
- Official H2.3 task group (formally the informal SQLX workgroup):
  - Mission: explore technologies and develop proposals for SQL extensions for XML
  - Website: http://www.sqlx.org
  - Members: Oracle, IBM, Microsoft, Sybase, FileTech,...



Tagging via SQL/XML – Examples

```
SELECT e.empno,XML2CLOB (XMLELEMENT
                    (NAME "Emp",e.firstnme || " "|| e.lastname))
AS "Result,, FROM employee e WHERE e.edlevel =12
```

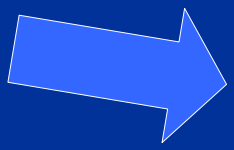
EMPNO Result
000290 <Emp>JOHN PARKER</Emp>
000310 <Emp>MAUDE SETRIGHT</Emp>

```
SELECT d.deptno,XML2CLOB (XMLELEMENT (
                                NAME "Mgr", XMLATTRIBUTES (d.mgrno)))
AS "Result"FROM department d WHERE d.admrdept ='A00 '
```

DEPTNO Result
A00 <Mgr ID="000010"/>
B01 <Mgr ID="000020"/>
C01 <Mgr ID="000030"/>
D01 <Mgr/>

Tagging via SQL/XML – Examples

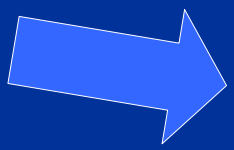
```
SELECT XML2CLOB(XMLELEMENT(NAME "Department",
                           XMLATTRIBUTES(e.workdept AS "name"),
                           XMLAGG(XMLELEMENT(NAME "emp",e.lastname )
ORDER BY e.lastname )))AS "dept_list"
FROM employee e WHERE e.workdept IN ('C01 ','E21 ')
GROUP BY workdept
```



```
dept_list
  <Department name =„C01">
  <emp>KWAN</emp>
  <emp>NICHOLLS</emp>
  <emp>QUINTANA</emp>
</Department>
<Department name ="E21">
  <emp>GOUNOT</emp>
  <emp>LEE</emp>
  <emp>MEHTA</emp>
  <emp>SPENSER</emp>
</Department>
```


Tagging via table tagging function (V7+8) (REC2XML)

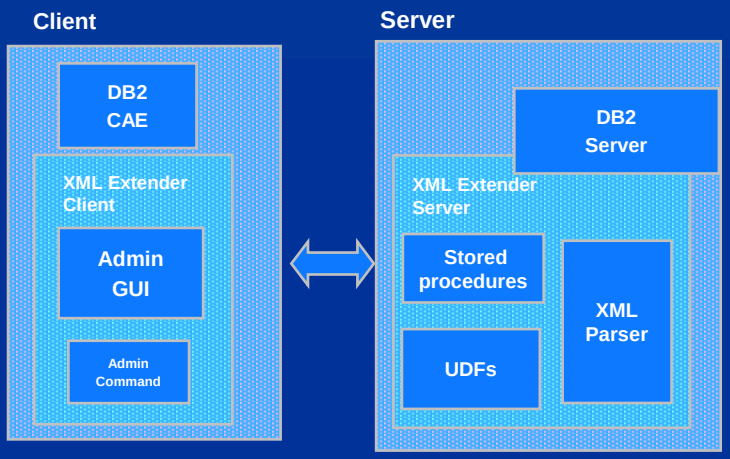
```
SELECT REC2XML (1.0,'COLATTVAL','',DEPTNO,MGRNO,ADMRDEPT)
FROM DEPARTMENT
WHERE DEPTNO ='D01'
```



```
<row>
  <column name="DEPTNO">D01</column>
  <column name="MGRNO"null="true"/>
  <column name="ADMRDEPT">A00</column>
</row>
```

The where clause could again be an select expresssion

Using XML Extender



- UNIX (AIX, Linux, Solaris) and Windows
  - Incorporates ICU, XML4C and XML4J
  - Included in DB2 7+
- zSeries
  - Requires the XML toolkit 1.2 (later 1.3)
  - Included in DB2 V7+
- iSeries
  - XMLC included in the operating system
  - Runs on OS/400 5.1+

IBM

DB2 Extender Concepts

- LOB TYPES
  - Storing objects of any kind up to 2 GB into DB2
  - Data movement and externalisation only if needed (LOB locators)
- USER-DEFINED FUNCTIONS
  - build SQL Functions in any language
  - controlled by DB2
  - overloading functions managed by DB2
  - integrated into DB2 Optimizer
- DISTINCT TYPES
  - create new data types for type safety on DB2 base types
  - type dollar (=integer) not compatible with type DM (=integer)
  - automatic generated casting functions
- TRIGGERS
  - call function by event (event = SQL insert/update/delete)
  - before/after triggers
- Stored Procedures
  - DB function calls to implement complex database logic

SQL Interface!

```
select txt-c1, img-c2, xml-c3
from mm-table
where
  fulltextsearch(txt-c1)
  qbic(img-c2)
  xpath(xml-c3)
```

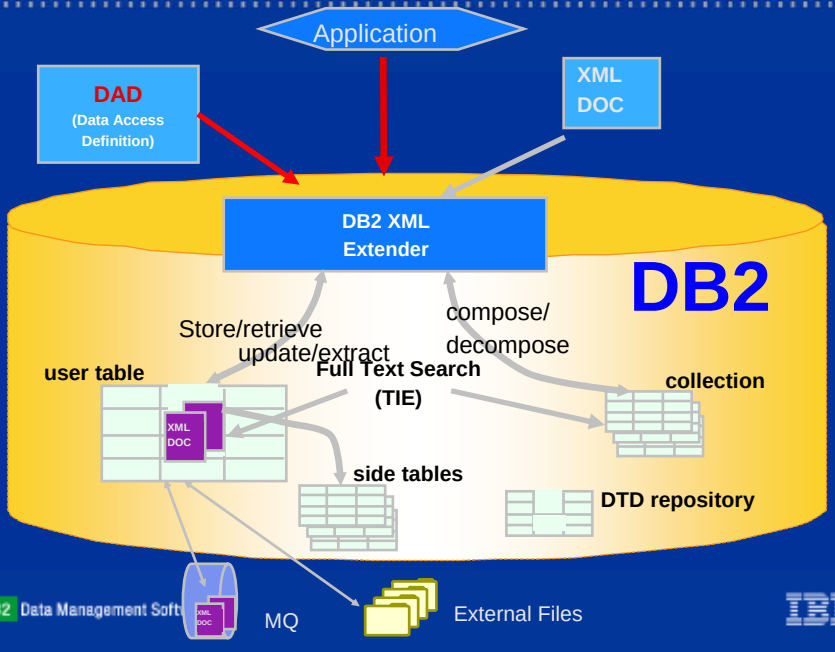

IBM

How DB2 XML Extender supports XML

- Storing XML Docs
  - as is (into a column, external file or message queue) [UDFs]
  - decompostition into relational [Stored Procedures]
- Validate against DTD or X-Schema
- Extract/update attribute/element values from XML docs via XPath Syntax [UDFs]
- DB2 index cruical XML attribute/element values (Side Tables)
  - Fast DB2 search
  - combine XML docs with legacy relational data
- Fast fulltext search in XML structures [Textextender/Text Information Extender]
- Retrieve XML docs [UDFs]
- Composite/Decompostite XML docs from/to relational [Stored Procedures]
- Store/retrieve/extract/update/composite/decomposite into/from message queue [UDFs+MQSeries]
- Transform XML docs via XSLT
- DTD Repository
- Administration Wizzard [Java Program]

IBM

DB2 XML Extender - Two Access and Storage Methods



IBM

Using XML Columns

Document centric approach

- Store entire XML documents as column data in user tables
  - Documents already exist, exist externally
  - Documents primarily read-only, update performance not critical
  - Structural text search with section support
  - Documents with large text blocks
- DAD can create side tables for fast search of selected XML element/attribute values
- Provide UDF and UDT for defining and accessing content

Data centric approach

- Compose & decompose XML documents from and to data in relational tables
  - Data already stored in database
  - Document content update performance is important
  - Rationalize different document sources
- Application specific mapping using DAD
- Can create different documents ("XML views") from same data
- Dynamic XML queries can override DAD
- Stored procedures for composition and decomposition

Composition/Decomposition

```
<order key='99'>
   <customer>Thompson</customer>
   <part key='82' > .... </part>
   <part key='83' > .... </part>
</order>
```

Stored Procedure Call

DAD
(Data Access Definition)

Mapping Methods:
SQL (composition only)
RDB_node (XML-Style for compostion/decomposition)

Collection

order_tab

order_key	customer
99	Thompson

part_tab

order_key	part_key	....
99	82	....
99	83	....

order.dad: SQL Node (Read only)

```
<?xml version="1.0"?>
<!DOCTYPE Order SYSTEM "E:\dtd\dxxdad.dtd">
<DAD> ...
<XCollection>
<SQL_stmt>SELECT o.order_key, c.customer FROM order_tab o, customer_tab c where...
</SQL_stmt>
<prolog>?xml version="1.0"?</prolog>
<doctype>!DOCTYPE Order SYSTEM "E:\dtd\sample.dtd"</doctype>
<root_node>
    <element_node name="Order">          --> identifies the <order> element
        <attribute_node name="Key">                --> identifies the "Key" attribute
            <column name="order_key"/>        --> the "order_key" column to which the element and
                                                  attribute are mapped
        </attribute_node >
        <element_node name="Customer">    --> defines <Customer> as a child element of
                                              <Order>
            <text_node>                            --> CData text for the <Customer> element
                <column name="customer">       --> specifies the "customer" column name mapping
                                                   for the <Customer> element
            </text_node>
        </element_node>
            ...
    </element_node>
</root_node>
</Xcollection>
</DAD>
```

```
...
<Order>  Key="1"  </Order>
<Customer> Rudi Ratlos </Customer>
...
```

DB2 Data Management Software

IBM

order.dad: RDB-Node (R/W)

```
<?xml version="1.0"?>
<!DOCTYPE DAD SYSTEM "c:\dxx\dtd\dad.dtd">
<DAD>
 <dtdid>neworder.dtd</dtdid>
 <validation>YES</validation>
 <Xcollection>
<prolog>?xml version="1.0"?</prolog>
<doctype>!DOCTYPE Order SYSTEM "c:\dxx\samples\dtd\neworder.dtd"</doctype>
 <root_node>
 <element_node name="Order">
  <RDB_node>
   <table name="order_tab" key="order_key"/>  <table name="part_tab" key="part_key"/>
    <condition>order_tab.order_key=part_tab.o_key AND part_tab.part_key=ship_tab.p_key</condition>
  </RDB_node>
  <attribute_node name="Key">
   <RDB_node>
    <table name="order_tab"/>  <column name="order_key" type="integer"/>
   </RDB_node>
  </attribute_node>
  <element_node name="Customer">
   <element_node name="Name">
    <text_node>
     <RDB_node>
      <table name="order_tab"/>  <column name="customer_name" type="varchar(16)"/>
     </RDB_node>
    </text_node>
   </element_node>
 </root_node>
</DAD>
```

```
...
<Order>  Key="1"  </Order>
<Customer> Rudi Ratlos </Customer>
...
```

DB2 Data Management Software

Fast Search with Side Tables

```
<order key='99'>
   <customer>Thompson</customer>
   <part key='82' > .... </part>
   <part key='83' > .... </part>
</order>
```

Store UDF

Triggers

DAD (Data Access Definition)

User Table

Side Tables

order_side_tab

order_key	customer
99	Thompson

part_side_tab

order_key	part_key	....
99	82	....
99	83	....

define Indexes

DB2 Data Management Software

IBM

XColumn Definition in DAD

```
<?xml version="1.0"?>
<!DOCTYPE DAD SYSTEM "c:\dxx\dtd\dad.dtd">
<DAD>
<dtdid>getstart.dtd</dtdid>
<validation>YES</validation>
<Xcolumn>
  <table name="order_side_tab">
    <column name="order_key"
      type="integer"
      path="/Order/@key"
      multi_occurrence="NO"/>
    <column name="customer"
      type="varchar(50)"
      path="/Order/Customer/Name"
      multi_occurrence="NO"/>
  </table>
  <table name="part_side_tab">
    <column name="price"
      type="decimal(10,2)"
      path="/Order/Part/ExtendedPrice"
      multi_occurrence="YES"/>
  </table>
  <table name="ship_side_tab">
    <column name="date"
      type="DATE"
      path="/Order/Part/Shipment/ShipDate"
      multi_occurrence="YES"/>
  </table>
```

XPath

```
<order key='99'>
   <customer>Thompson</customer>
   <part key='82' > .... </part>
   <part key='83' > .... </part>
</order>
```

Side Tables

order_side_tab

order_key	customer
99	Thompson

part_side_tab

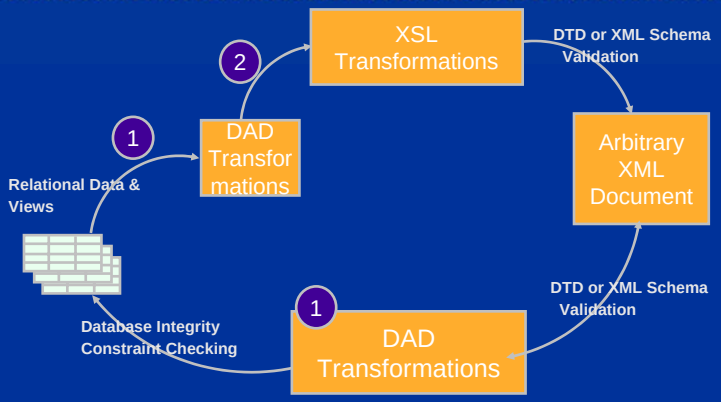
order_key	part_key	....
99	82	....
99	83	....

define Indexes

DB2 Data Management Software

IBM

XML Transformations



- DAD Creation
  - DB2 XML Wizzard, DAD Checker
  - IBM WebSphere Studio Application Developer
  - application ... on the fly ....
  - can include XSL Stylesheet (in advance or on the fly)
    - <stylesheet>?xml-stylesheet type="text/css"href="order.css"?</stylesheet>

IBM

Tooling

- DB2 XML Administration Wizard
  - GUI based front-end for enabling columns and collections
  - Websphere Studio
  - An advanced development environment for J2EE application development
  - Includes comprehensive XML tooling including
    - XML and DTD editors
    - RDB to Database Mapping tool (aka DAD Builder)

IBM

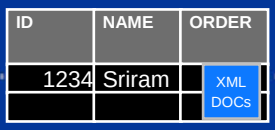
WSAD: XML Tooling Features

- DTD/Schema Editor
  - Visual tooling for working with DTDs/Schemas
  - Create DTDs/Schemas from existing XML documents
  - Convert to/from DTD or Schema
  - Generate JavaBean(s) for creating/manipulating XML documents from DTD/Schema
  - Generate an HTML form from a DTD
- XML Source Editor
  - Design/Source mode
  - DTD/Schema validation
  - Code Assist for building XML documents

WSAD: XML Tooling Features

- XML Mapping Editor
  - Generate XSL to map XML between DTDs/Schemas
- XML to/from Relational Databases
  - Generate XML, XSL, XSD from an SQL Query
- RDB/XML Mapping Editor
  - Map columns in a table to elements/attributes to XML
  - Generate a Database Access Definition (DAD) script to compose/decompose XML documents to/from a database
    - DAD is used with DB2 XML Extender

Using XML UDFs



- Stores a record into an XML-Table with UDF:

```
SQL INSERT INTO sales (ID, NAME, ORDER)
    VALUES( '1234', 'Sriram' XMLCLOBFromFile('c:/dxx/samples/cmd/order.xml'))
```

- Exports XML documents as varchars into files

```
select db2xml.Content(db2xml.xmlvarchar(order), order.xml') from sales where ID=1234
```

- Selects attributes and values from stored XML documents

```
SELECT  extractVarchar(Order, '/Order/Customer/Name') from sales
WHERE id > 10
```

- Updates an XML document with UDF

```
UPDATE sales
    set order = Update(order, '/Order/Customer/Name', 'Heine')
WHERE name = 'Sriram,
```

- **Fulltext search in XML-Doc with Text Information Extender**

```
SELECT Order FROM Sales
WHERE Contains(section(/Order/Customer/Name), "Heine")  = 1
```

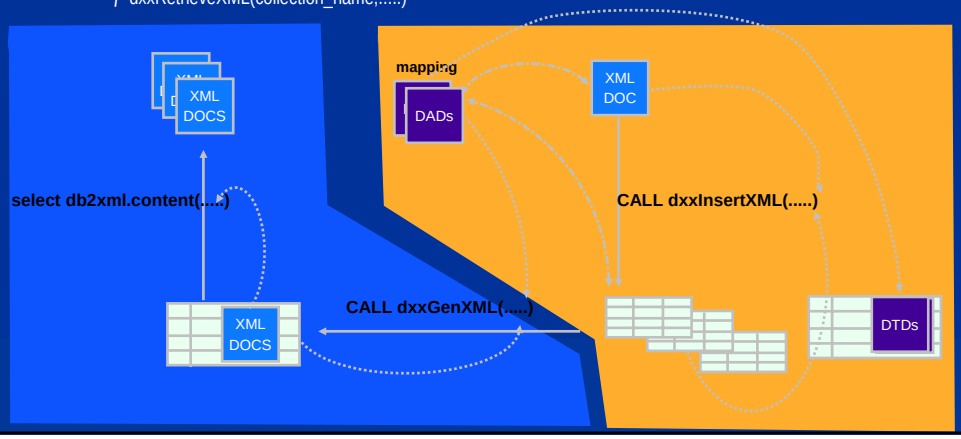

Extracting multible Values UDF Example

- Select * from table(db2xml.extractIntegers(
  db2xml.XMLFile('c:\dxx\samples\xml\book1.xml'), '/book/*/@id')) as x;

```
<book><chapter id="1">  </...>
<footnote id="2">...</...
```

Using XML Collection Stored Procedures

- Decomposition
  - dxxInsertXML(collection_name,XML_Doc,...)
  - dxxShredXML(DAD,XML_Doc,...)
- Composition
  - dxxGenXML(DAD,result_table,max_rows,condition,..)
  - output: result set of composed XML documents
  - dxxRetrieveXML(collection_name,.....)
- Composition into CLOB
  - dxxRetrieveXMLClob(Collection_name, ...,CLOB)
    - dxxGenXMLClob(DAD, ...,CLOB, ...)

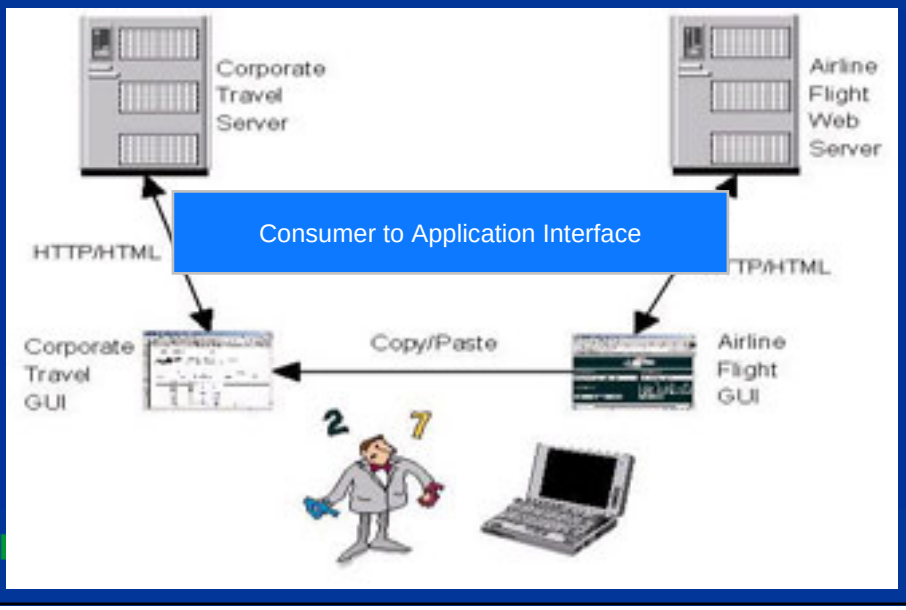


Using XML Extender and MQSeries

- A series of SQL User Defined Functions and Stored Procedures are supplied in DB2 7.2 enabling a DB2 client application to do the following from a single request:
  - To read from MQSeries queues into DB2 XML columns
  - To send and publish messages to MQseries queues from DB2 XML columns
  - To send XML messages composed from relational data to MQSeries queues
  - To decompose (shred) XML messages held in MQSeies queues into relational data
- It is possible to integrate MQSeries operations with DB2 table operations in a single SQL request such as a SELECT

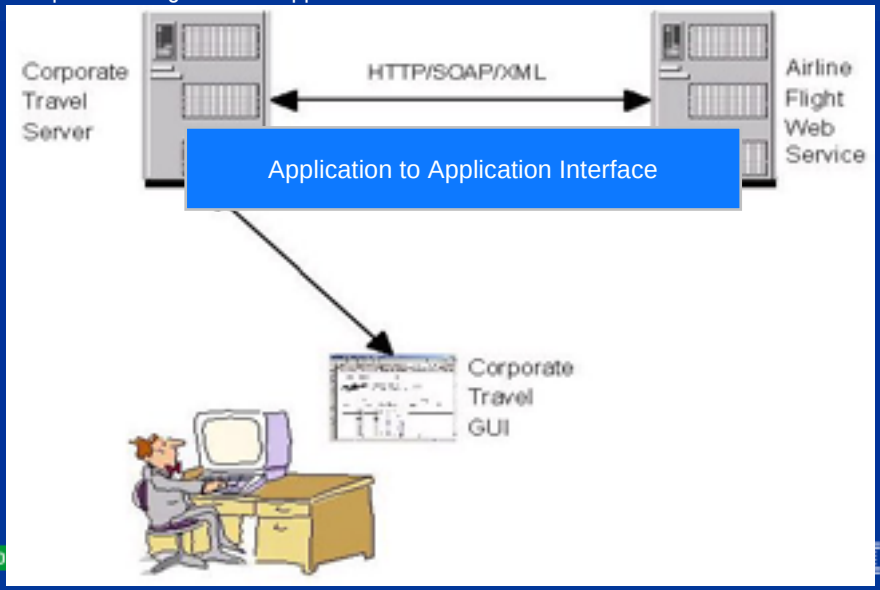
Understandig Web Services

- Week Integration on Application Web



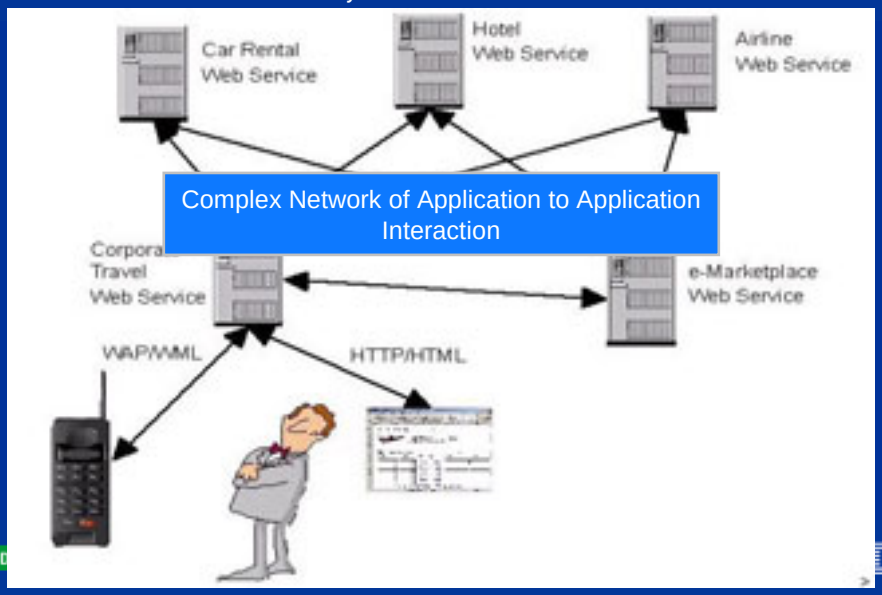
Understandig Web Services continued

- Improved Integration on Application Web



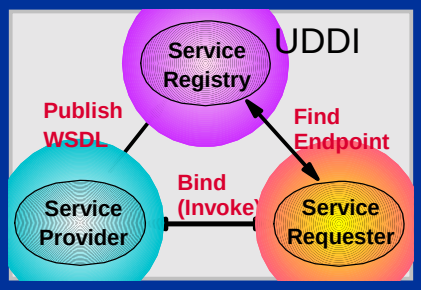
Understandig Web Services continued

- The new Web Service Economy



Web Services Business Model

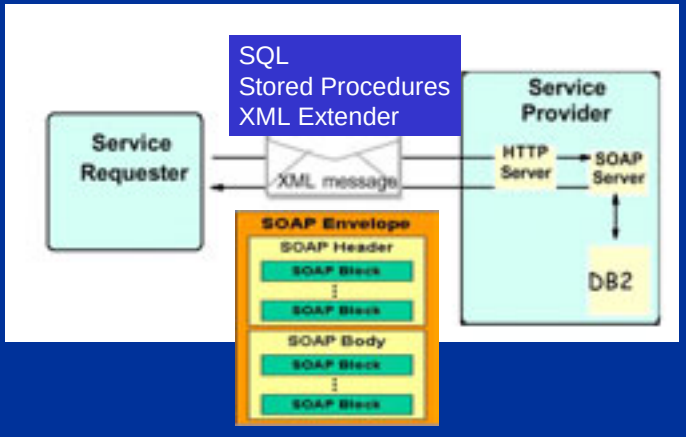
- **XML** defines a universal way of representing any data, making data integration simple
- **SOAP** uses XML as messages to define a universal Web service requests, making process integration simple
- **WSDL** specifies all information needed for integration, making universal application assembly tools possible
- **UDDI** is a special Web service which allows users and applications to locate required Web services
- **WS-I.org** formed to achieve seamless interoperability



Integration Advantages with Web Services

- Support different programming languages
- APIs can change
- different operating systems or hardware platforms
- different software vendors, in-house code
- Loose coupling with SOAP over HTTP (no client software, easy firewall setup)

Invoke a Web Service with DB2 access



Business example

- Company BlueAirways has Stored Procedures that implement a certain business logic
  - Retrieve all flights from city1 to city2 on a certain date
- BlueAirways wants to share this with company TravelPortal to build a travel web portal
  - BlueAirways creates web service definition file for SPs, cities and dates being parameters of the web service
  - Deploys Web Service and sends WSDL to TravelPortal
  - TravelPortal creates web application using BlueAirways's web service
  - Invocation of DB2 WS Provider will execute the SP and return the result as XML
  - Web Service can be shared with other companies too

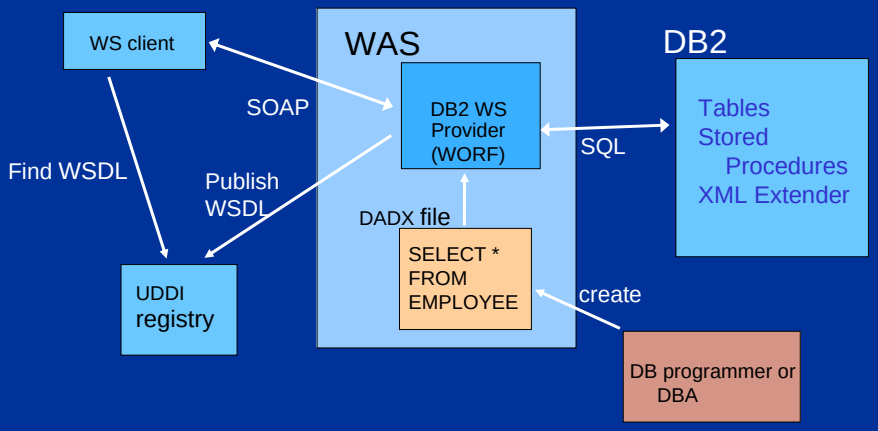
Defining Web Services

- Web Services are defined through **simple** XML files
  - Users list database operations to be exposed as Web Services
- Provider Runtime will help
  - Deploy Web Service
  - Generate WSDL
  - Provide HTML test environment to run Web Service
  - Automatic regeneration if Web Service definition changes
- Deployment
  - WebSphere (including zSeries and iSeries) and Apache Tomcat

```
<DADX>
 <operation name="listDepartments">
  <documentation> Example Employee
  web service</documentation>

  <query>
    <SQL_query>
      SELECT * FROM DEPARTMENT
       WHERE LOCATION = :location
    </SQL_query>
    <parameter name ="location“
       type="xsd:string"/>
  </query>
 </operation>
</DADX>
```

Development Scenario



IBM

DB2 as a Web Service Consumer

- Deploys UDFs for implementing an „SQL SOAP proxy“
- Provides tools for generating DB2 table functions related to WSDL definition (WSAD)
- Usage: invoke Web Services as an SQL function

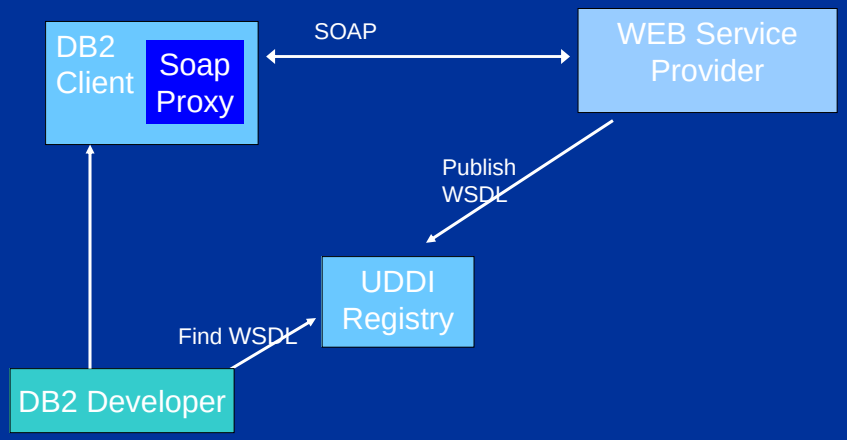
```
SELECT name, symbol,
STOCK_QUOTE (symbol)
AS quote FROM
stock_watch
```

name	symbol
International Business Machines	IBM
Microsoft	MSFT

Dynamic Web Service Invocation

IBM

Development Scenario



Tooling

- WebSphere Studio allows end to end development of DB-oriented Web Services
  - Create SQL queries with SQL editor or XML Extender DADs
  - Create Web Service definition files (DADX)
  - Create DB2 Web Service Table Functions from WSDL
  - Deploys Web Services to WebSphere and Apache Tomcat
  - Publishes WSDL in UDDI
  - Test of Web Service
  - Generation of Java Web Service clients

Interesting Internet Adresses

- DB2 XML Extender Homepage

www-4.ibm.com/software/data/db2/extenders/xmlext/index.html

- Information Integration Technology

http://www7b.boulder.ibm.com/dmdd/library/demos/0203xperanto/0203xperanto.html

preview for information integration via federation and XQuery

- alphaWorks

www.alphaworks.ibm.com - web site for free emerging technologies from IBM.

- developerWorks

www.ibm.com/developer/xml - web site for product- and platform-independent information on e-business application development

- e-business

www.ibm.com/e-business - site for more information on IBM e-business products

- WebSphere

http://www-4.ibm.com/software/webservers/