bmcsoftware

ACTIVATE BUSINESS WITH THE POWER OF I.T.™

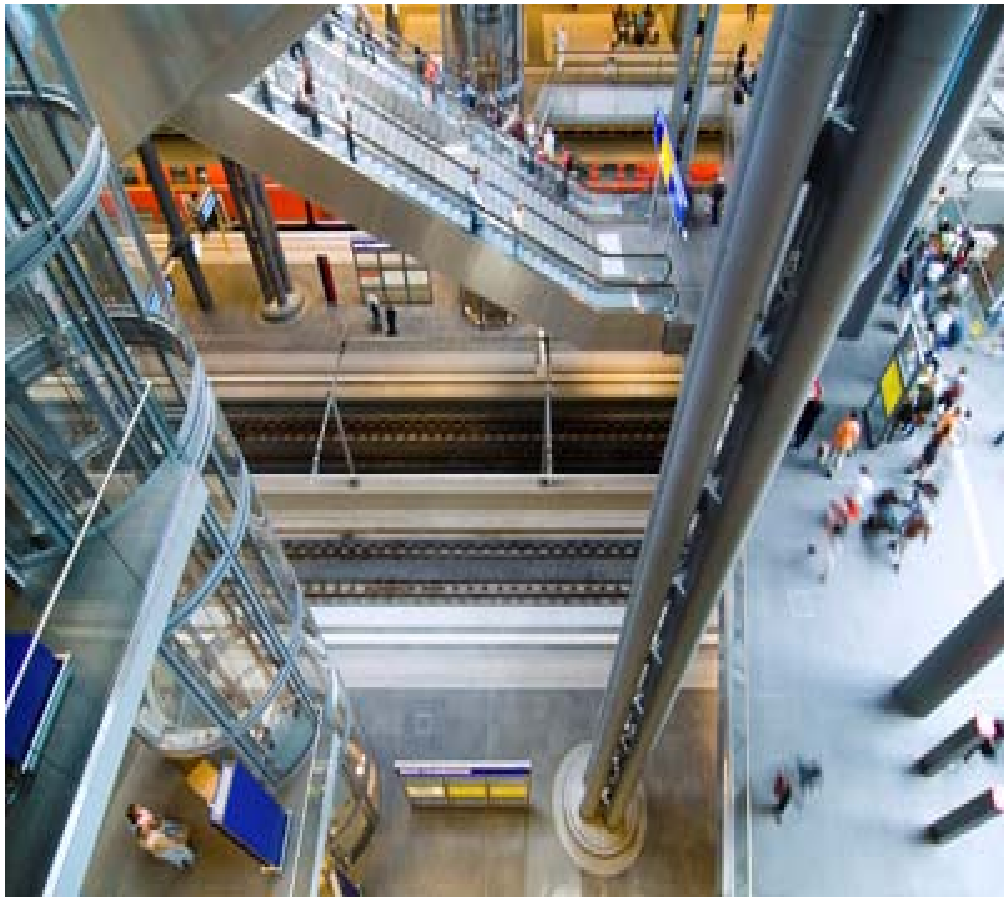
# XML Technical Overview

Bill Arledge, Consulting Product Manager
BMC Software Inc.

11/10/2008

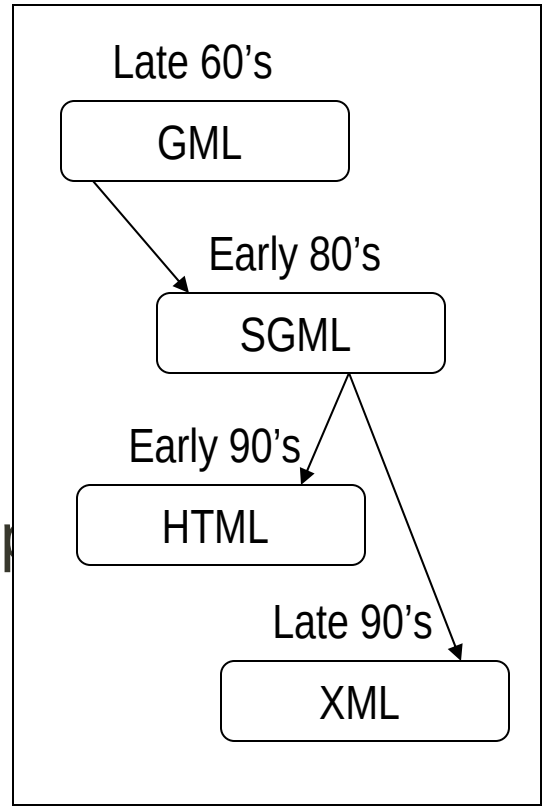
# Agenda

- What is XML?
- Why is XML important to your business?
- PureXML in DB2 9
  - Physical implementation
  - The logical side



# An XML Overview
# Markup Languages in General

- Markup concept
  - Derived from publishing concept of “marking up” a document to provide printing instructions
    - Identify characteristics like typeface, size, and style
- Formalized in computer publishing in the late 60’s
  - IBM GML was the first widely used markup language
    - Pioneered by Charles Goldfarb and partners at IBM
      - GML came from Goldfarb, Mosher, and Lorie
- Evolved to SGML (Standardized Generalized Markup Language) in the early 80’s
  - More focus on structural aspects of a document
- SGML applications/profiles evolved
  - HTML throughout the 90’s
  - XML developed by the W3C to address problem of documents on the Internet
    - W3C delivered first spec in early 1998




XML describes data

HTML describes display

# XML Documents
# An Example

```
<Person DOB='1900/01/01'>
  <Name>
    <Last>Doe</Last>
    <First>John</First>
    <MI>S</MI>
  </Name>
  <Address>
    <Addr-1>900 Any Street</Addr-1>
    <Addr-2></Addr-2>
    <City>Houston</City>
    <Zip>10001</Zip>
  </Address>
</Person>
```

## Well-formed XML Documents

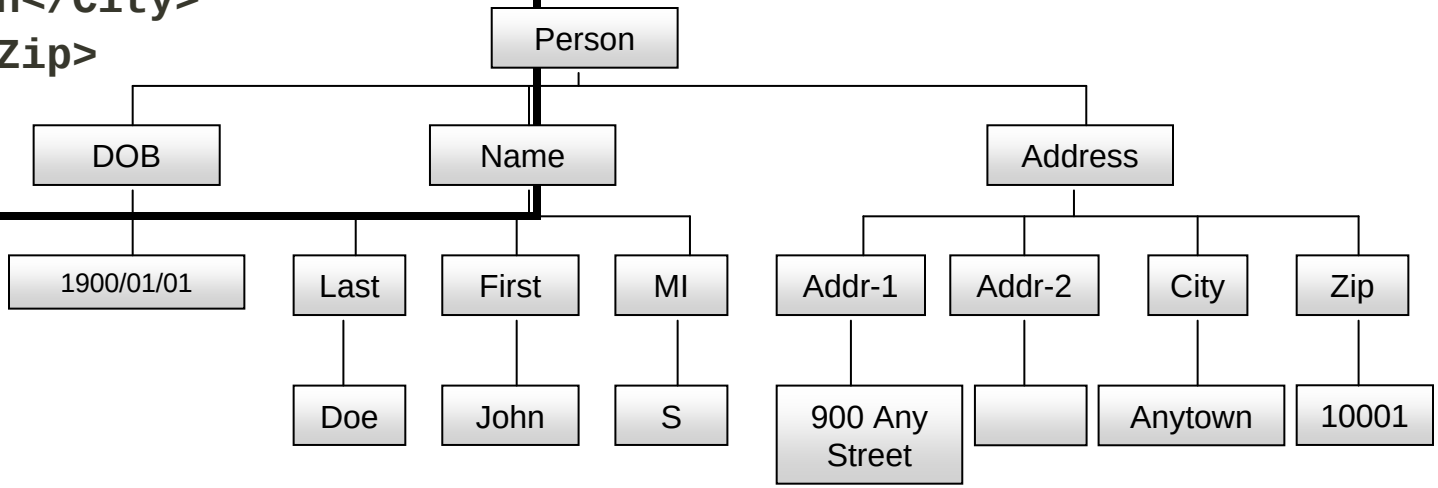
- Must be contained in a single element (root)
- Begin and end tags required
- Elements can't overlap
- Attributes must be enclosed in quotes
- Symbols like > can't be used
- Elements are case sensitive

## Document Characteristics

- XML for DB2 data model follows XPath 2.0 and XQuery 1.0 data models
- Composed of multiple nodes
  - Document (Root)
  - Element
  - Attribute
  - Text
  - Processing instruction
  - Namespace
  - Comments
- Platform independent
- Self describing
- Fits any data that can be visualized as a document
- Key technology for data interchange
- Must be well-formed

# XML Documents
# Hierarchical Construct

```
<Person DOB='1900/01/01'>
  <Name>
    <Last>Doe</Last>
    <First>John</First>
    <MI>S</MI>
  </Name>
  <Address>
    <Addr-1>900 Any Street</Addr-1>
    <Addr-2></Addr-2>
    <City>Houston</City>
    <Zip>10001</Zip>
  </Address>
</Person>
```




bmcsoftware

# XML Documents
# Data Movement

Producing Application

```
<Person DOB='1900/01/01'>
  <Name>
    <Last>Doe</Last>
    <First>John</First>
    <MI>S</MI>
   </Name>
  <Address>
    <Addr-1>900 Any Street</Addr-1>
    <Addr-2></Addr-2>
    <City>Houston</City>
    <Zip>10001</Zip>
   </Address>
</Person>
```

Consuming Application

# Validating XML Documents

- DB2 doesn't require validation against an XML schema
  - Requires documents be well-formed (as discussed)
- XML Schemas primary tool for validation
  - Set of rules for the XML elements and attributes that appear in your XML data
  - XML Schema Repository stored in 7 new catalog tables
    - SYSIBM.XSR*, SYSIBM.SYSXML*
  - Schema maintenance is via stored procedures
    - XSR_REGISTER, XSR_ADDSCHEMADOC, XSR_COMPLETE, XSR_REMOVE
    - Similar support via JDBC and Java calls
  - SYSFUN.DSN_XMLVALIDATE function used to
    - Validate XML values against schema
    - Identify defaults and schema normalized values from schema
- DTD (Documentation Type Definitions) also stored
  - Used for entity reference resolution and defaults

# Extensions of XML Protocol

- Many industries are defining standards for document interchange
  - OASIS: Organization for the Advancement of Structured Information Standards
    - Nonprofit group to develop e-business standards
    - 12 “industry focus areas”
  - ACORD: similar group for the Insurance Industry
  - ISO 20022 – securities and banking
  - WSDL (Web Services Description Language)
  - PIDX (Petroleum Industry Data Exchange)

# Options for managing XML Data

## Primary options prior to DB2 9

- Specialized XML DBMS
  - Separate installation, learning curve, etc.
- "Shredding" XML documents into relational
  - Slow!
  - Schema changes are hard to handle
  - Need space in each row even for columns which are almost never used
  - Multi table joins
- LOB Usage (or something similar)
  - No direct way to index contents
  - No direct way to access a subset of a document

# PureXML™ in DB2 9

- Facilitates storage of XML data in DB2 tables
  - Can be accessed with SQL or XQuery

- Hides complexity from user
- Adds the reliability, security, recoverability of the mainframe, with the flexibility and accessibility of XML
- DB2 V8 supported XML utilizing an XML Extender
  - Still available in DB2 9 but not recommended
    - Native XML storage delivers vast performance improvements and much improved usability
  - Migration to PureXML documented in IBM publications

# PureXML™ in DB2 9
# A Hybrid Database Server

- Relational and XML exist in the same data structure (sort of)
  - Relational columns stored in relational format (traditional)
  - XML columns stored in parsed hierarchical format
  - XML columns have no architectural limitations in size but in DB2 9 on z/OS 2G is the limit (I doubt that's relevant)
  - XML columns stored as Unicode UTF-8
- Multiple XML columns can be stored in a single table
- Defining XML columns – How simple can it get

```
CREATE TABLE INVOICE
 (INV_NO INTEGER NOT NULL,
  CUST_NO INTEGER NOT NULL,
  INV_DATA XML, CUST_DATA XML) ...
```

- XML Spaces
  - Base table in a good old fashioned tablespace
  - XML tablespace required to store the XML data (must be UTS)
  - Specialized columns and indexes required (more later)

# PureXML in DB2 9
# Creating an XML table

```
CREATE TABLESPACE TSXML
  IN DBXML
  USING STOGROUP SYSDEFLT
  CLOSE NO
  CCSID EBCDIC;
COMMIT;


CREATE TABLE RNDWDA.TBXML
  ( CUSTOMERNO        INTEGER
   ,INFO              XML)
  IN DBXML.TSXML
  CCSID EBCDIC;
```

- Your vanilla DB2 tablespace
  - Must be valid in DB2 9 (segmented, partitioned, UTS)
    - Can be created implicitly
  - You can alter a table in a simple tablespace to add an XML column in DB2 9
- Basic CREATE TABLE statement with your XML column
  - What Appears in the catalog

```
NAME                      TBNAME    HIDDEN
-v----1----v----2----v----3----\---2--
CUSTOMERNO                TBXML       N
INFO                      TBXML       N
DB2_GENER..D_FOR_XML      TBXML       P
```

  - DB2 adds a hidden column to the table
    - DB2_GENERATED_DOCID_FOR_XML is the name
    - Hidden attribute to P means won't appear in SELECT * result set
    - Only one of these no matter the number of XML columns

# PureXML in DB2 9
# Creating an XML table

- But wait, under the covers DB2 is creating new objects to service the XML datatype
  - An index on the new BIGINT column added to the table with XML column
  - An XML tablespace and table where the data is actually stored
  - An index into the XML table
- What do you see in the catalog

```
.TS                    TSXML
..TB                                RNDWDA.TBXML
...IX       DUPS                    RNDWDA.IXXML1
...IX                               RNDWDA.I_DOCIDTBXML
.TS         0001       XTBX0000
..TB        XML                     RNDWDA.XTBXML
...IX       UNN                     RNDWDA.I_NODEIDXTBXML
```

User-defined XML index

Index on generated column

Generated XML tablespace

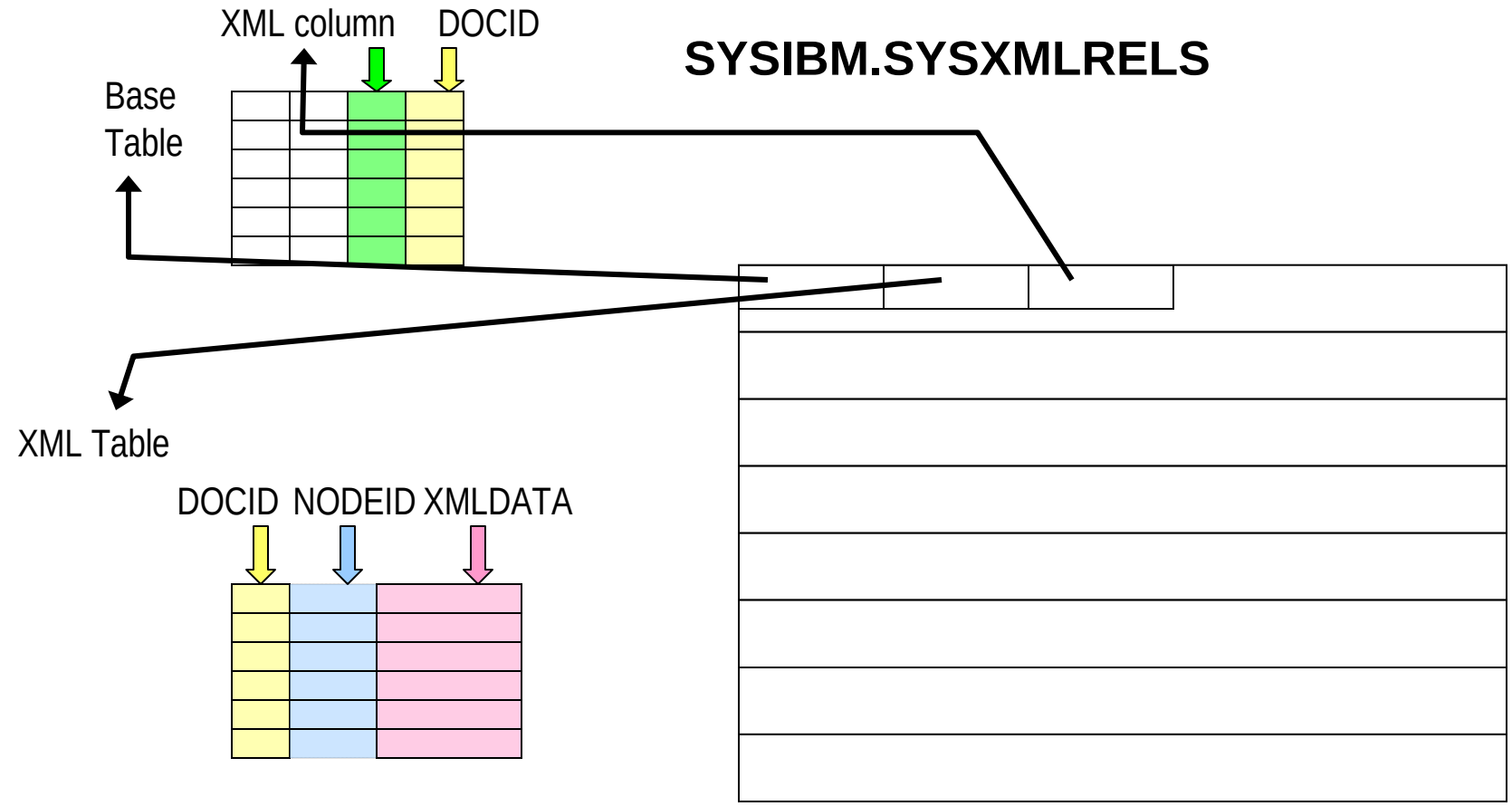
Generated XML table

Generated XML Index

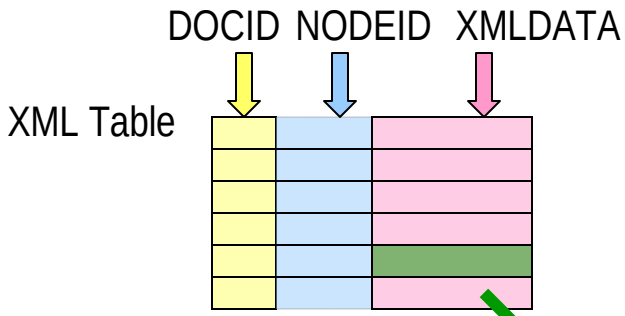
```
Column Name        ColNo Datatyp Length
---1----v----2----v----3----v----4----v-
DOCID                 1 BIGINT        8
MIN_NODEID            2 VARBIN      128
XMLDATA               3 VARBIN    15850
```

```
Index Name                          Column                    Seq
--v----1----v----2----v----3----v----4----v----5----v----
RNDWDA.I_NODEIDXTBXML               DOCID                       1
RNDWDA.I_NODEIDXTBXML               XMLDATA                     2
```

# Catalog Tables Supporting XML Internal Storage Constructs



# Catalog Tables Supporting XML Internal Storage Constructs




**SYSIBM.SYSXMLSTRINGS**

| | | |
|---|---|---|
| | | |
| | | |
| 00010 | Person | |
| 00011 | DOB | |
| | | |
| 00013 | Name | |
| | | |
| | | |

...

```
<Person DOB='1900/01/01'>
 <Name>
  <Last>Doe</Last>
  <First>John</First>
  <MI>S</MI>
 </Name>
 <Address>
  <Addr-1>900 Any Street</Addr-1>
  <Addr-2></Addr-2>
  <City>Any_town</City>
  <Zip>10001</Zip>
 </Address>
</Person>
```

# PureXML in DB2 9
# Indexing Support

- User indexes can be created on XML tables
  - Used to improve performance of queries on XML documents
  - Big advantage over using LOB columns to store XML data
- Uses XPATH specification to identify document nodes to index

```
CREATE INDEX RNDWDA.IXXML1
    ON RNDWDA.TBXML(INFO)
    GENERATE KEY USING XMLPATTERN
    '/Customerinfo/@cid'
    AS SQL VARCHAR (4) NOT PADDED;
```

XPATH specification

Datatype specification

- Index entries in XML documents provide direct access to nodes within the XML document
  - Differs from relational indexes that index entire columns
  - Index values stored as instances of the specified datatype
  - Can define multiple indexes on the same XML column

# Making Changes to XML Data

- INSERT SQL statement  to insert XML documents
  - Must be well-formed document
  - Should Insert data from host variables rather than literals
  - Document can be validated against a registered XML schema using the DSN_XMLVALIDATE function in the INSERT statement
- UPDATE SQL statement used to update XML columns
  - No ability to update individual nodes of an XML document; Entire column is replaced
  - Must be a well formed document
  - If validation against a registered schema is  required use the DSN_XMLVALIDATE function in your UPDATE statement
  - Use the XMLEXISTS function to qualify rows for the update
    - A user index on the XML column is critical if using the XMLEXISTS function
    - More on XMLEXISTS on next slide

# Deleting XML Data

- SQL DELETE or UPDATE statements can be used to remove XML documents
  - Use DELETE to eliminate the row containing the XML document
    - Specify rows to delete using XPath expressions
      - Specify the XPath expressions using the XMLEXISTS predicate
  - Use UPDATE to eliminate the XML document while retaining the row
    - Like the DELETE you can specify XPATH expressions in a XMLEXISTS predicate
- XMLEXISTS predicate
  - Used to qualify XML columns that meet the criteria for the SQL statement being executed
    - Can Qualify SELECT, UPDATE, and DELETE processes
  - Uses SQL XPath Expressions
  - An Example:

```
WHERE XMLEXISTS (//addr[city='HOUSTON']' passing INFO)
```

# XML Query Support

› SQL or XML Functions and predicates used to retrieve data

› SQL is just what it means

– For example:

SELECT POid, POrder FROM PurchaseOrders;

- XML Functions and predicates used to manipulate XML data
  - Funtions include:
    - XMLParse - Convert XML text to XML value
    - XMLSerialize - Convert XML to character type
    - XMLQuery - executes an XPath expression against an XML value

```
SELECT XMLQUERY ('//item[productName = $n]'
       PASSING PO.POrder, P.name as "n")
       FROM PurchaseOrders PO, Product P;
```

    - XMLCast - Cast XML to other types or other types to XML
    - XMLExists - a predicate, which returns TRUE if the XPath expression evaluates to a non-empty sequence

# Some Miscellaneous XML Considerations

- All DB2 utilities support XML data
  - REORG, LOAD/UNLOAD, CHECK DATA/INDEX, COPY, RECOVER, REBUILD
  - Support similar to LOB objects
  - Some utilities have new XML keywords to be specified
- Host Language Support
  - C, C++
  - COBOL
  - Assembler
  - JDBC and SQLJ Type 2 and Type 4 Drivers
  - Even PL/I

# Caveats

- Performance
  - Data design is still necessary!
  - “Extract” static metadata to relational columns
  - Don’t use XQUERY without an XML index
  - Validation is expensive (but you may have to do it)!
  - UPDATE is really delete/insert of XML column
  - LOAD uses INSERT like process

# Caveats

- Maintenance
  - Finding names of implicitly created objects is non-trivial
  - Use ALL option in LISTDEF
  - LOAD must use files for XML columns larger than 32K
- Migration (DW, Test creation)
  - Element and attribute names are not stored with data
  - SYSIBM.SYSXMLSTRINGS matches the id actually stored with the name
  - DSN1COPY is not an option

# BMC Support for XML

- IBM has provided a sound infrastructure for XML
- BMC fully supports this initiative
- Performance and Recovery support XML as of August, 2007
- Admin and Utility releases in 2008 will provide “basic support”
- XML exploitation in the future
  - We want to provide features and tools to help you better use XML
  - We invite your requirements with us

bmcsoftware

# References

- To see the XML 1.0 Standard: http://www.w3.org/TR/xml/
- XPATH and XQUERY: http://www.w3.org/TR/xpath-datamodel/
- Schema: http://www.w3.org/TR/xmlschema-0/
- Listing of XML applications and standards: http://xml.coverpages.org/xmlApplications.html

bmcsoftware

# Questions?