E43

XML Storage in IMS: What's Next

Christopher Holtz

Las Vegas, NV **September 15 – September 18, 2003**

RETURN TO INDEX

Overview

IMS

- **Introduction**
  - **What is XMS?**
  - **What is XML?**
  - **What are XML Schemas?**
- **XMS Methodology**
  - **Decomposed Storage**
  - **Intact Storage**
- **XMS Tooling**
  - **Metadata Generation**
- **XMS Java Implementation**
  - **SQL UDF Interface**
  - **Future**

- **A methodology for storing and retrieving XML documents into and out of standard IMS databases**
  - **Language Independent Design**
  - **XML Schema Metadata (Structural Metadata)**
  - **DL/I Metadata (Physical Metadata)**
  - **Two storage types**

- **XMS Java is the Java enablement of XMS using an extended IMS Java JDBC interface**

Why use XMS

IMS

- **A World-wide movement towards XML as the standard data interchange language.**

- **Retrieve existing IMS data in standard, easily exchangeable XML format**
- **Store, Index, Search and Retrieve valid new XML documents into new or existing IMS databases**

- **35 years of storage and management of Hierarchical data**
- **35 years of performance, stability and reliability**

What is XML

- **A Standardized, Simple, and Self-Describing Markup Language for documents containing structured or semi-structured information.**

```
<A>
   <f1>~~~~</f1>
   <f2>~~~~</f2>
   <f3>~~~~</f3>
   <B>
      <f4>~~~~</f4>
      <f5>~~~~</f5>
   </B>
   <B>
      <f4>~~~~</f4>
      <f5>~~~~</f5>
   </B>
</A>
```

- **Standard Internet Data Exchange Format**
  - **Handles encoding**

    <xml? version="1.1" encoding="ebcdic-cp-us"?>

  - **Handles byte ordering**

    <OrderNumber>**110203**</OrderNumber>

  - **Human Legible?**
  - **Easily Parsed**
  - **Standard**

What is XML

- **Data-centric**
  - **Highly structured**
  - **Limited size and strongly typed data elements**
  - **Order of elements generally insignificant**
  - **Invoices, purchase orders, etc.**
- **Document-centric**
  - **Loosely structured**
  - **Unpredictable sizes with mostly character data**
  - **Order of elements significant**
  - **Newspaper articles, manuals, etc.**

Well formed vs. Valid XML Document

IMS

- **Well formed – Obeys the XML Syntax Rules**
  - **must begin with the XML declaration**
  - **must have one unique root element**
  - **all start tags must match end-tags**
  - **XML tags are case sensitive**
  - **all elements must be closed**
  - **all elements must be properly nested**
  - **all attribute values must be quoted**
  - **XML entities must be used for special characters**
- **Valid – Conforms to a specific XML Schema**

An XML language for defining the legal building blocks of a valid XML document

An XML Schema:

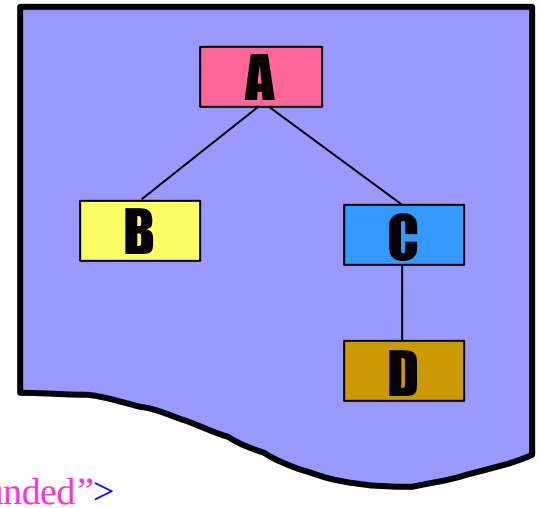
- **defines elements and attributes that can appear in a document**
- **defines which elements are child elements**
- **defines the order and number of child elements**
- **defines whether an element is empty or can include text**
- **defines data types for elements and attributes**
- **defines default and fixed values for elements and attributes**

Defines an agreed upon communication contract for exchanging XML documents

XML Schema Example

IMS

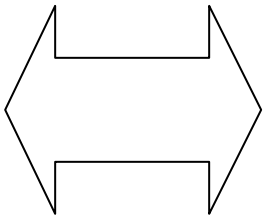
```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns="http://www.myNamespace.net"
        targetNamespace="http://www.myNamespace.net"
        elementFormDefault="qualified">


  <xsd:element name="A">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Ainteger" type="xsd:int"/>
        <xsd:element name="Astring" type="xsd:string"/>
        <xsd:element name="B" minOccurs="0" maxOccurs="unbounded">
          <xsd:element name="Bfield" type="xsd:string"/>
          …
        </xsd:element>
        <xsd:element name="C" minOccurs="0" maxOccurs="unbounded">
          <xsd:element name="D" minOccurs="0" maxOccurs="unbounded">
          </xsd:element>
      …
</xsd:schema>
```

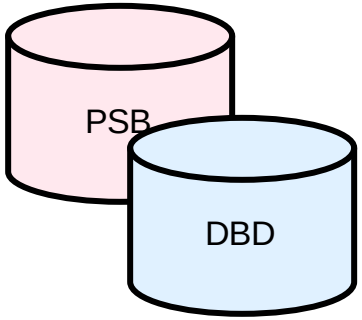


- **Natural mapping between hierarchic XML data and hierarchic IMS database definitions.**

XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd ="http://www.w3.org/2001/XMLSchema"
        xmlns:ims ="http://www.ibm.com/ims "
        xmlns ="http://www.ibm.com/ims/PSBName/PCBName"
        targetNamespace="http:// www.ibm.com/ims/PSBName/PCBName'
        elementFormDefault ="qualified">

  <xsd:annotation>
    <xsd:appinfo>
      <ims:DLI mode="store" PSB="AUTPSB11" PCB="AUTOLPCB"
            dsg ="DATASETG" meanLength ="1000" numDocs ="100"/>
    </xsd:appinfo >
  </xsd:annotation>

  <xsd:element name="A">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="field1" type="xsd:int"/>
        <xsd:element name="field2">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string ">
              <xsd:maxLength value="30"/>
            </xsd:restriction>
  …
```





IMS to XML mapping metadata

IMS

- **Physical Metadata**
  - **Segment Hierarchy (***field relationships – 1-to-1, 1-to-n***)**
  - **DBD Defined Fields**

  Defined in DBD

  - **Application Defined Fields**
  - **Field Type, Type Length, Byte Ordering, Encoding, etc.**
  - **Offer Field/Segment Renaming** *(lift 8 char restriction)*

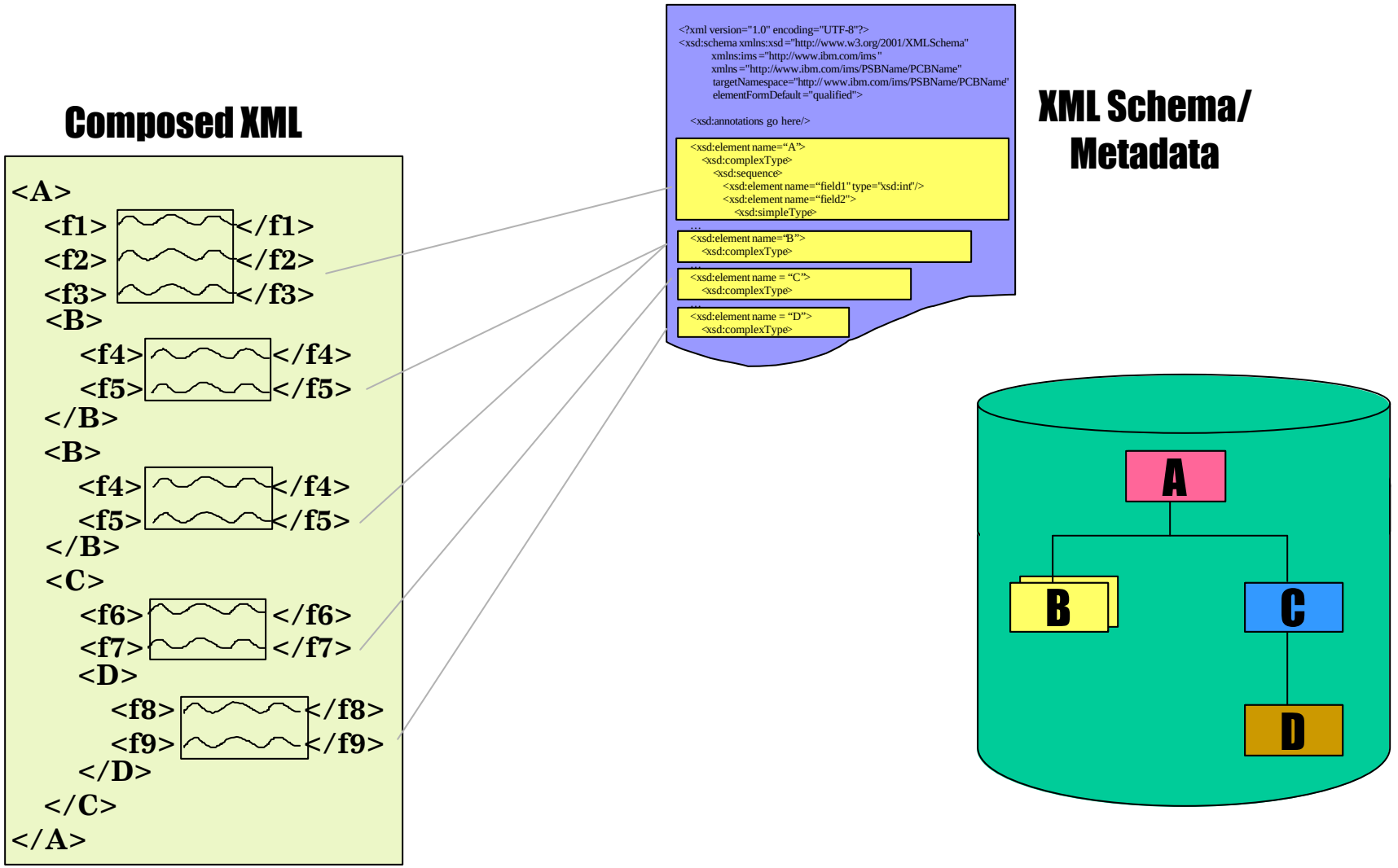
  Defined in Copylibs (IMS Java)

- **Logical Metadata**
  - **XML layout for fields (***field relationships must still match***)**
  - **Element vs. Attribute (***names must match***)**
  - **Type Restrictions, Enumerations, etc.**

  Defined in XML Schema

Decomposed XML Retrieval in IMS

IMS



Decomposed vs. Intact Storage

IMS

- **Decomposed (*data-centric storage*)**
  - **XML tags are stripped from XML data**
  - **Identical as current IMS storage**
  - **Strict data-centric XML Schema validated data**
  - **EBCDIC encoding**
  - **Searching on IMS Search Fields**

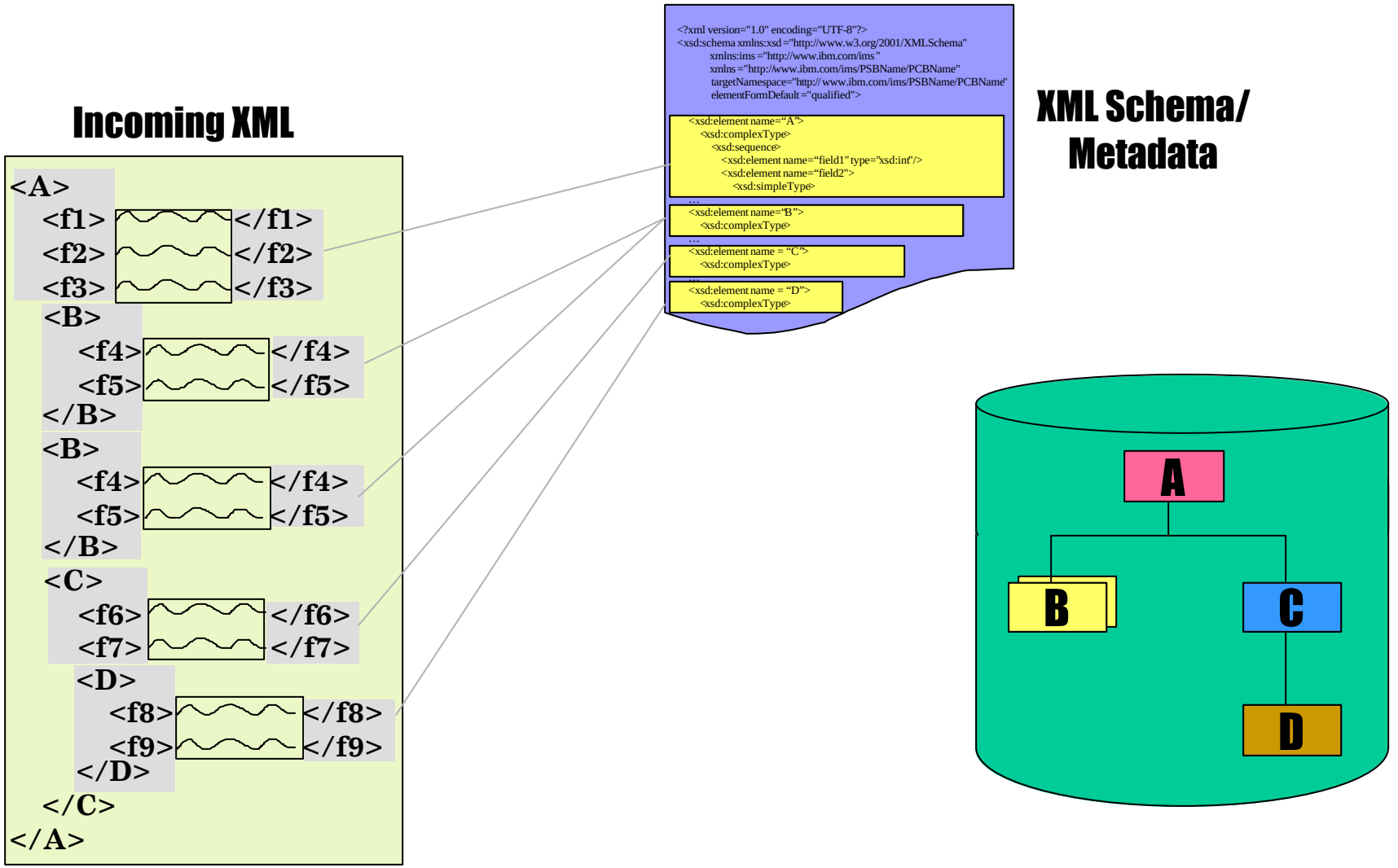
- **Intact (*document-centric storage*)**
  - **Entire XML document is stored (including tags)**
  - **Relaxed un-validated data**
  - **Any desired encoding is possible**
  - **Searching is through XPath specified and generated Secondary Indexed Side Segments**

Decomposed Storage

- **XML document must be parsed and validated.**
- **Data must be converted to *traditional* IMS types**
  - **COMP-1, COMP-2, etc.**
  - **EBCDIC CHAR, Picture Strings**
- **Stored data is searchable by IMS and transparently accessible by non-XML enabled applications.**

Decomposed XML Storage in IMS

IMS



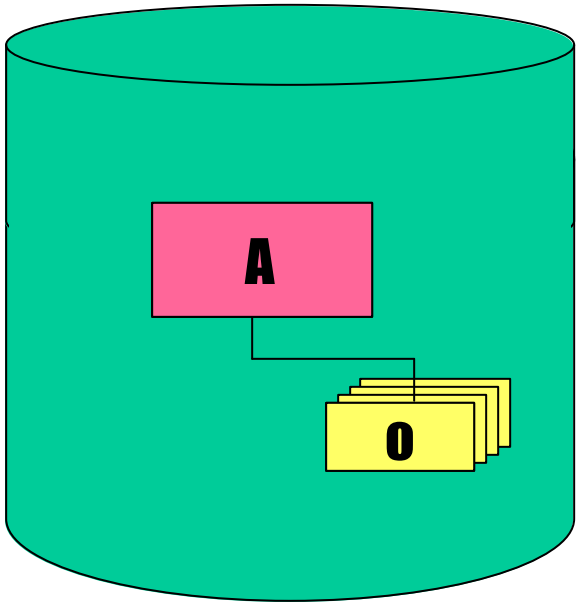
- **No (or little) XML Parsing or Schema validation**
  - **Storage and Retrieval Performance**
- **No (or little) data type conversions**
  - **Unicode storage**
- **Stored documents are no longer searchable by IMS and only accessible to XML-enabled applications**
  - **XPath side segments**

Intact XML Storage in IMS

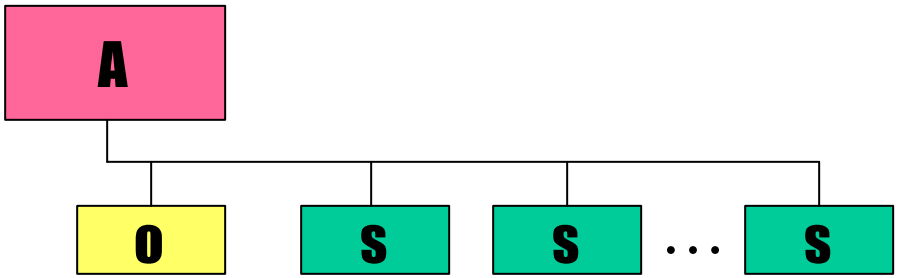
IMS

Incoming XML

```
<A>
   <f1>~~~~</f1>
   <f2>~~~~</f2>
   <f3>~~~~</f3>
   <B>
      <f4>~~~~</f4>
      <f5>~~~~</f5>
   </B>
   <B>
      <f4>~~~~</f4>
      <f5>~~~~</f5>
   </B>
   <C>
      <f6>~~~~</f6>
      <f7>~~~~</f7>
      <D>
         <f8>~~~~</f8>
         <f9>~~~~</f9>
      </D>
   </C>
</A>
```



- **XPath expression identifying Side Segments**
  - **Side segment is converted to *traditional* data type and copied into segment.**
- **Side Segments are secondary indexed with documents root as target.**



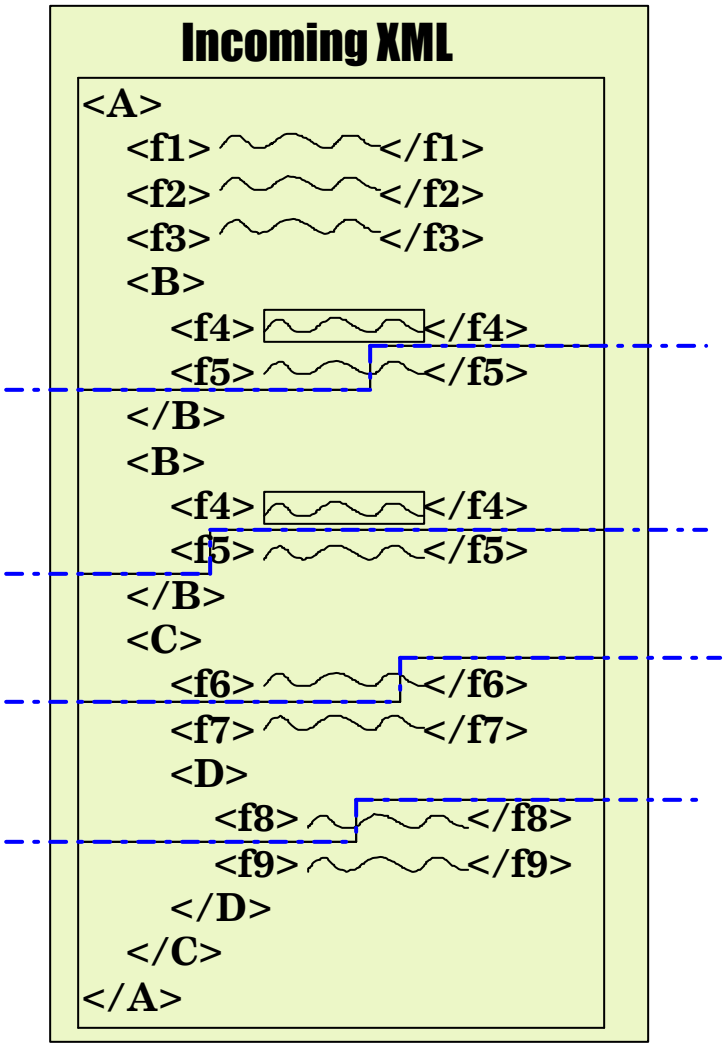
Example:

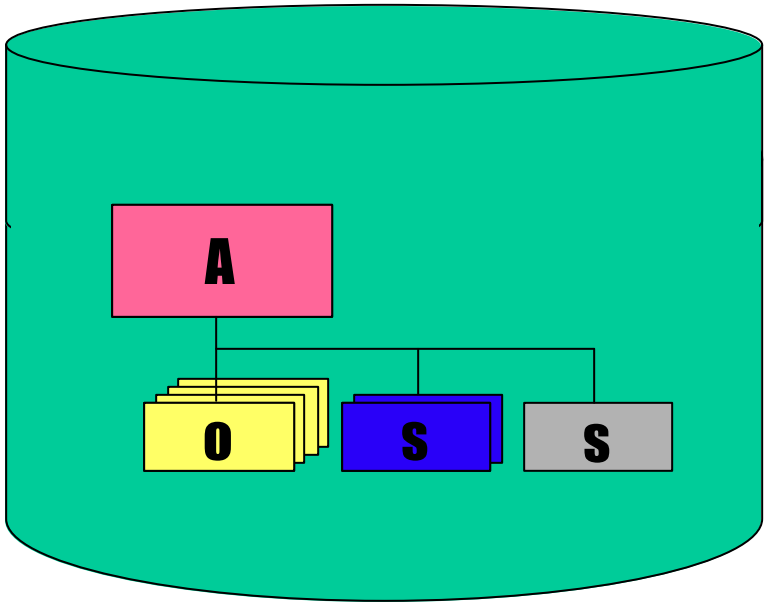
XPath="/Dealer/DealerName"

XPath="/Dealer/Model[Year>1995]/Order/LastName"

Intact XML Storage in IMS

IMS





Example:

XPath="/A/B/f4"
XPath="/A/E/f1"

Overview

IMS

- Introduction
  - What is XMS?
  - What is XML?
  - What are XML Schemas?
- XMS Methodology
  - Decomposed Storage
  - Intact Storage
- XMS Tooling
  - Metadata Generation
- XMS Java Implementation
  - SQL UDF Interface
  - Future

DL/I Model Utility

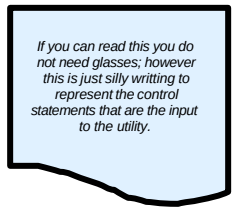
IMS

Control statements:

1) Choose PSBs/DBDs

2) Choose copybook members

3) Aliases, data types, new fields.



COBOL
copybook members

PSB

DBD

DL/I Model Utility

XMI 1.2

XML Schema(s)

IMS Java classes

IMS Java report

DL/I Model Schema Generation

IMS

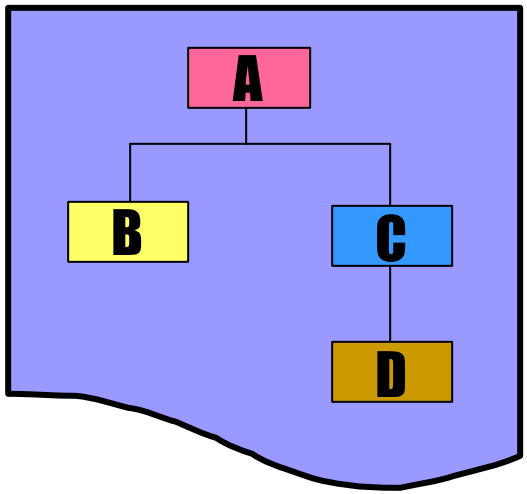
- **Additional Control Statements Keywords**

```
OPTIONS PSBds=PSB.SOURCE.PDS     DBDds=DBD.SOURCE.PDS
        GenJavaSource=YES        JavaSourcePath=output/dir
        Package=test.db.psb4     ReportPath=output/dir
        GenXMLSchema=YES         XMLSchemaPath=output/dir
        Outpath=output/dir

PSB psbName=AUTPSB4 Javaname=AutoDealershipDatabase
    PCB PCBName=PCB1 JavaName=MyXMLView GenXMLSchema=YES

// Physical Segments for DEALERDB
SEGM DBDName=DEALERDB SegmentName=DEALER
    FIELD Name=DLRNO    JavaType=INTEGER JavaName=DealerNo
    FIELD Name=DLRNAME  JavaType=CHAR    JavaName=DealerName
          …
          …
```

Logical Metadata (XML Schema)

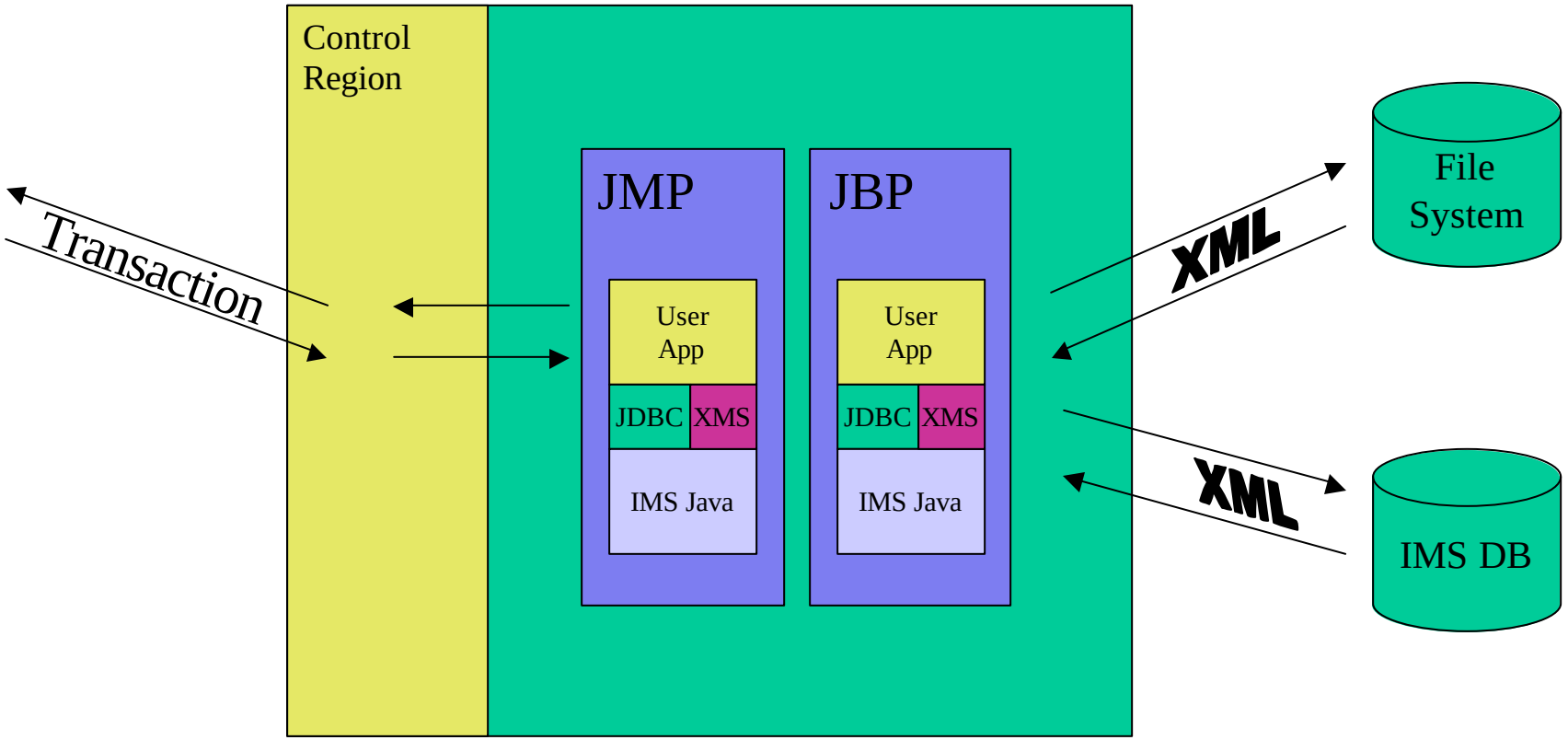
IMS

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns="http://www.ibm.com/ims/PSBName/PCBName"
        targetNamespace="http://www.ibm.com/ims/PSBName/PCBName"
        elementFormDefault="qualified">

  <xsd:element name="A">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="field1" type="xsd:int"/>
        <xsd:element name="field2">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:maxLength value="30"/>
            </xsd:restriction>
          </xsd:simpleType>
        <xsd:element name="B" minOccurs="0" maxOccurs="unbounded">
        </xsd:element>
        <xsd:element name="C" minOccurs="0" maxOccurs="unbounded">
          <xsd:element name="D" minOccurs="0" maxOccurs="unbounded">
          </xsd:element>
      …
</xsd:schema>
```



XMS Java Interface

IMS



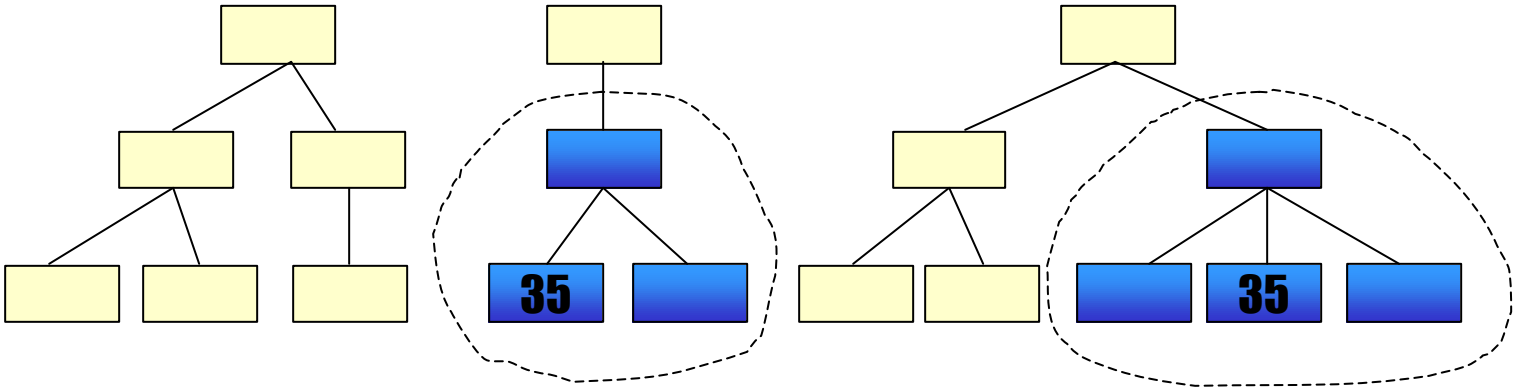
XMS Java Interface

IMS

- **Adds 2 User Defined Funtions (UDF) to the IMS Java JDBC SQL interface**
  - **retrieveXML()**
  - **storeXML()**
- **Runs as an IMS Java Application**
  - **JDR (JMP, JBP)**
  - **DB2 Stored Procedure**
  - **CICS**
  - **WebSphere**

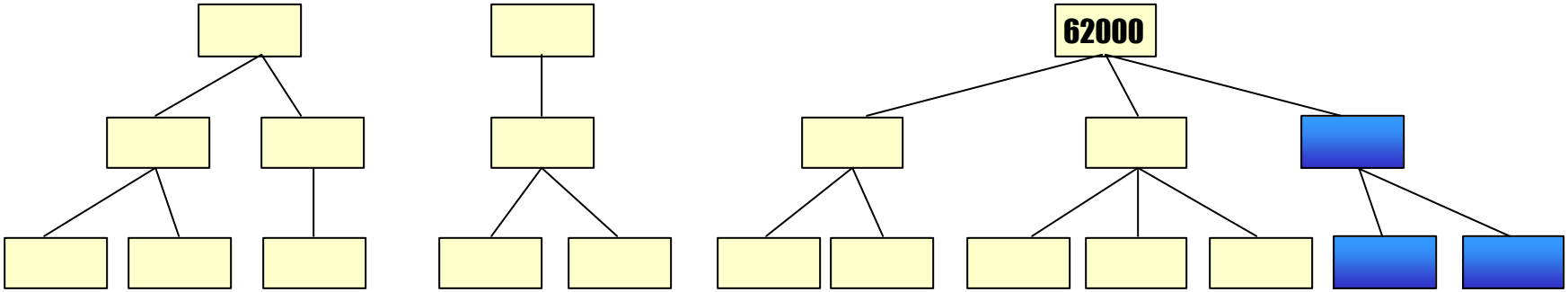
RetrieveXML() UDF

SELECT retrieveXML(B)
FROM C
WHERE C.fieldA = ’35’



**Two Rows of XML CLOBs in the ResultSet*

INSERT INTO B (storeXML())
VALUES (?)
WHERE A.fieldA = ’62000’



**Insert Statement must be a Prepared Statement*

- **SQL is a really poor XML/IMS interface**
  - **Hierarchical DB**
  - **Hierarchical Data**
  - **Relational Query Language??**
- **SQL/XML**
  - **Still relational**
- **XQuery**
  - **Only query right now**
  - **Still under development**