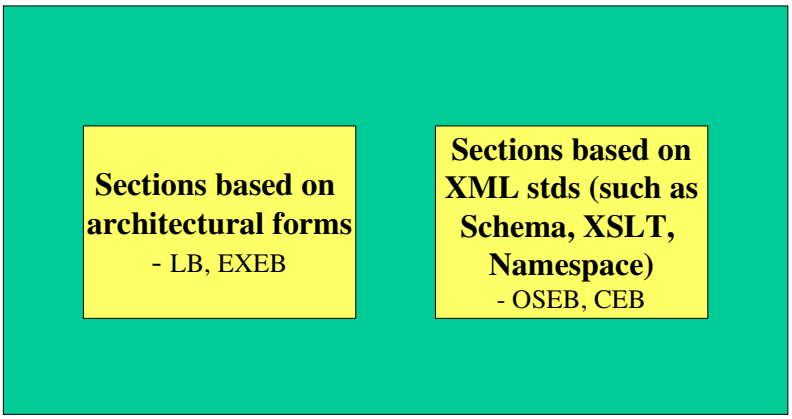
XML Exchange Based on ISO 10303-28

Object Serialization Early Binding

****NOTE:** These slides reflect the state of the OSEB prior to the London workshop (4/25/00-4/28/00). Except items indicated by *

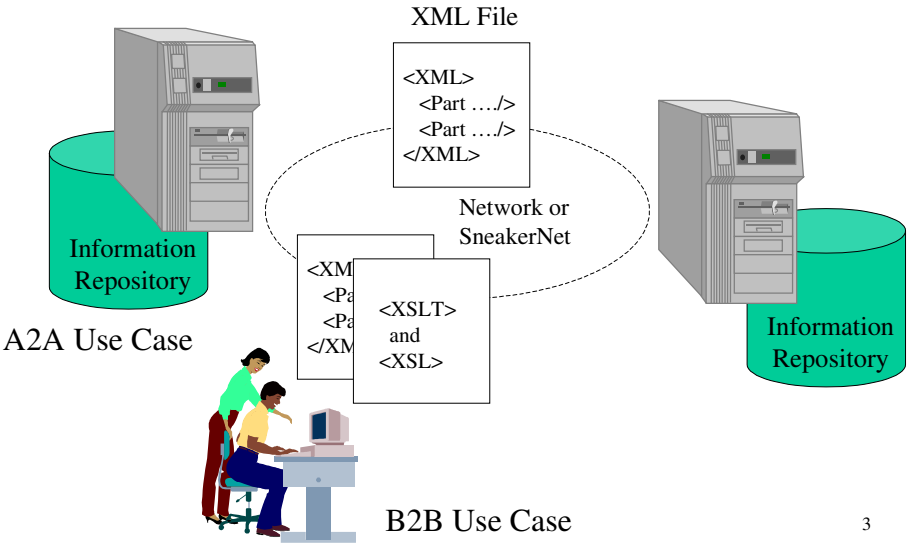
1

Part 28



2

Interoperability Use Cases



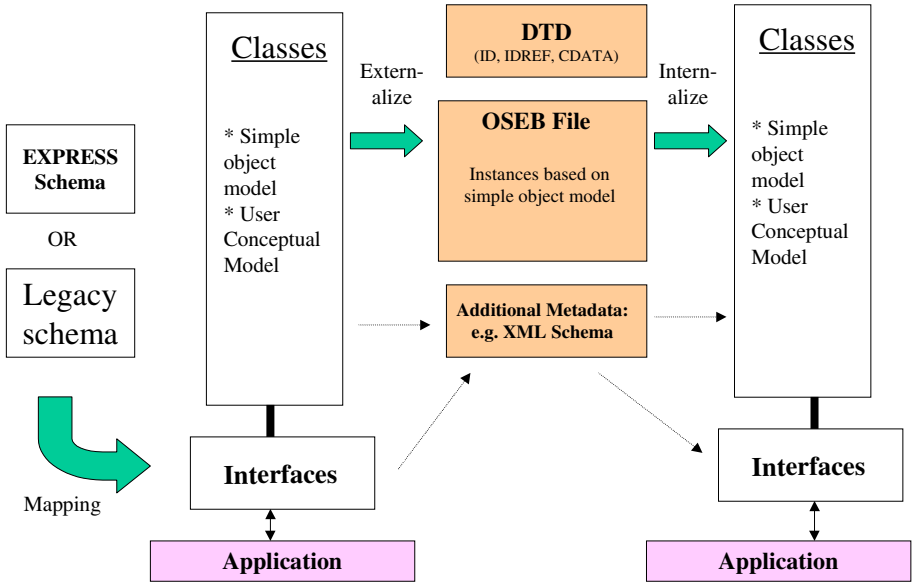
3

A2A Implementation Constraints

1) Should "re-use" widely available, standards-based tools
   - •Should be implementable using the widely implemented Serializer design pattern
   - •Should be implementable without modifying application code
   - •Should be able to re-use XML metadata/validation tools (DTD and XML Schema)
   - •Should completely separate instance data from metadata (Need to use early binding style)
   - •Should be able to re-use event-driven parsing tools (need to minimize nesting)
2) Should be implementable with tools that allow for re-configuration by non-programmers
3) Should support transformation to and from a canonical or neutral format
4) Should utilize a representation that is as simple and concise as possible
   - •Should be able to represent data as objects
   - •Should use a representation that maps directly to a simple object model (in order to minimize complexity/special cases of the implementations)
   - •Should contain minimum amount of information needed to effect serialization
5) Should be able to marshal/unmarshal every semantic concept representable in the EXPRESS information modeling language (ISO 10303-11)

4

OSEB interchange



Use cases

- Serialize instances from a Java application in one process to the same Java application on another process
- Serialize instances of Java data classes from one process to another process
  - Data classes are independent of implementation classes
  - Support the ability to develop new behavior without re-compiling the data classes
  - Data classes provide the data structures for the implementation classes that provide application behavior; data need not be copied from data classes to execute the application
- Serialize the essential state from Enterprise Data Objects(s) as XML.
- Import the essential state from Enterprise Data Object(s) to create new (replica) EDO(s) that maintains connection to the original EDOs
- Import the essential state from Enterprise Data Object(s) to create new (duplicate) EDO(s) that is an independent copy and has no links back to the original
- Import the essential state from Enterprise Data Object(s) into an application component
- Import the essential state from Enterprise Data Object(s) for viewing in a browser

6

Xchange File - Objects

- Each object is represented by an XML element with the name of the type of which it is a direct instance.
- Each object contains an attribute list that is the union of the attributes of all its ancestor types.
- Each object has an attribute “x-id” of type ID, which is its unique identifier within the file.

7

Xchange File - Objects (cont’d)

- All primitive attributes shall be of type CDATA.
- All object reference attributes shall be of type IDREF.
- All collection attributes shall be of type IDREFS for collections of entities and white-space separated CDATA for collections of primitives. *

8

Xchange File - Primitives

- In most cases, primitive types (float, double, int, long) shall be represented as attribute values of type CDATA.
- Enumeration shall be represented as an XML enumerated attribute.
- Primitive types shall be represented by wrapper elements, where necessary.

9

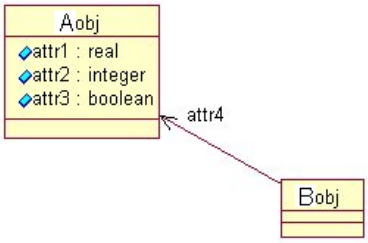
EXPRESS ENTITY

```
ENTITY aobj;
attr1    :       REAL;
attr2    :       INTEGER;
attr3    :       BOOLEAN;
END_ENTITY;

ENTITY bobj;
attr4    :       aobj;
END_ENTITY;
```

10

UML



11

Java DataClasses

```
public class Aobj {
        public double attr1;
        public long attr2;
        public boolean attr3;
};

public class Bobj {
        public Aobj attr4;
};
```

12

XML DTD

```
<!ELEMENT Aobj EMPTY>
<!ATTLIST Aobj
        x-id    ID              #REQUIRED
        attr1   CDATA           #REQUIRED
        attr2   CDATA           #REQUIRED
        attr3   (true | false)  #REQUIRED>

<!ELEMENT Bobj EMPTY>
<!ATTLIST Bobj
        x-id    ID              #REQUIRED
        attr4   IDREF           #REQUIRED>
```

13

XML Instance Data

```
<Aobj x-id = "id-1" attr1 = "47.534" attr2 = "234"
        attr3 = "true" />

<Bobj x-id = "id-2" attr4 = "id-1"/>
```

14