EXup:

An Engine for the Evolution of XML Schemas and Associated Documents

Federico Cavalieri

DISI - University of Genoa, Italy

Why EXup?

- Context
  - XML Schema
  - Requirement changes over time
- Schema evolution
  - Schema modification
  - Document adaptation

Agenda

- Specifying a schema evolution with EXup
  - Identification of schema components
  - Specification of schema modification
  - Definition of document adaptation

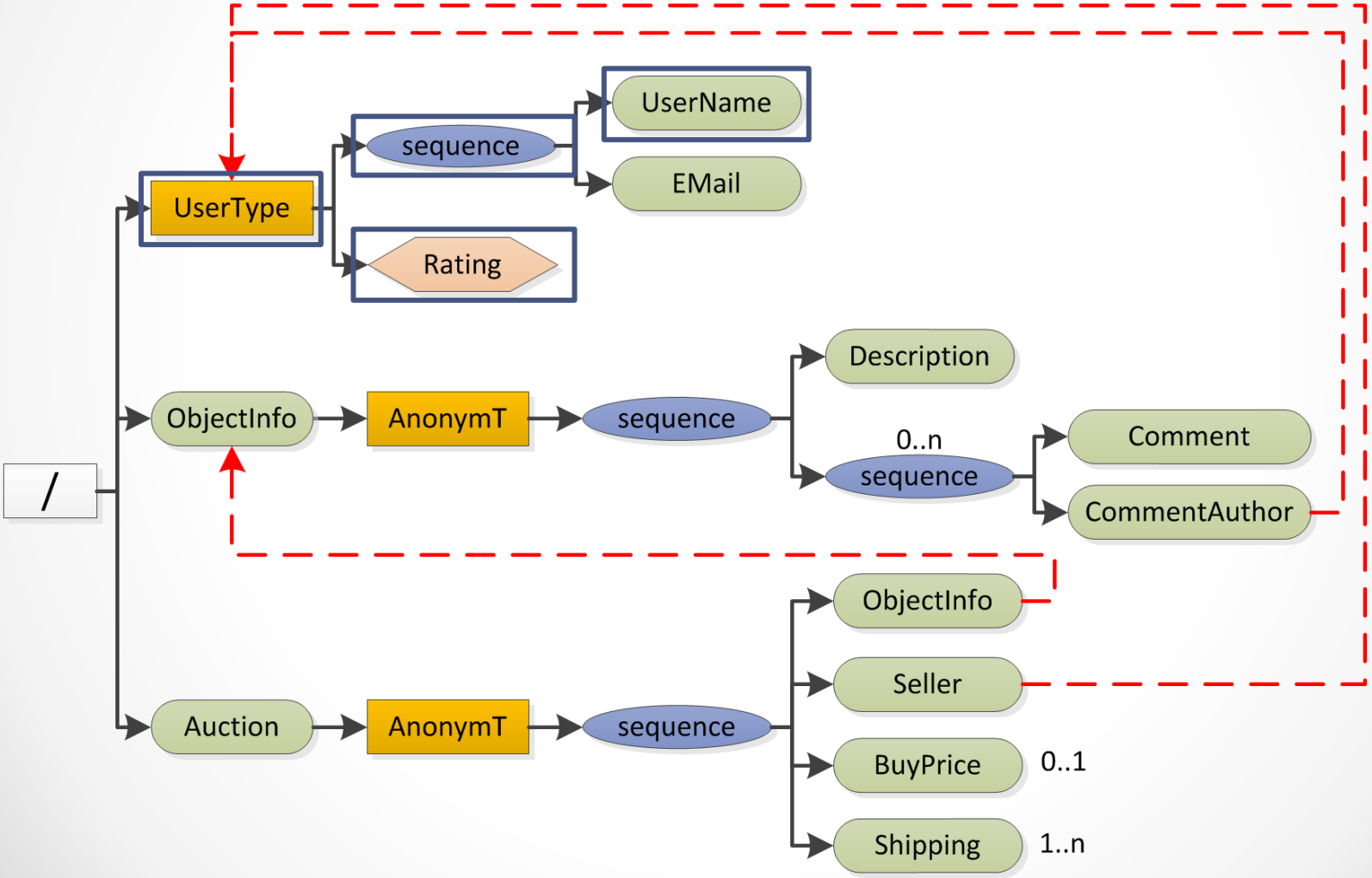
  } XSUpdate
- EXup interface
- Evaluating schema evolutions
- Conclusions

Identification of schema components

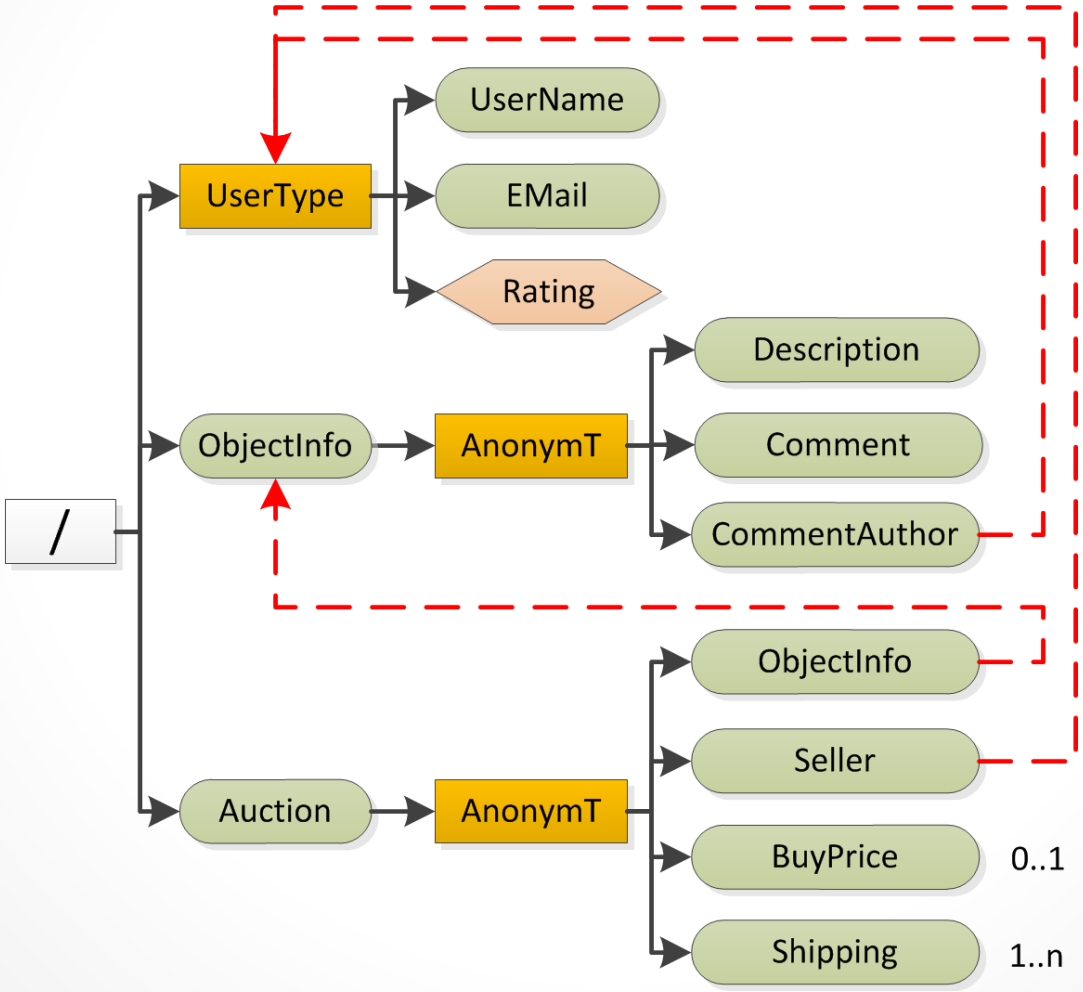
Schema Components

```
1  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3    <xs:complexType name="UserType">
4      <xs:sequence>
5        <xs:element name="UserName" type="xs:string"/>
6        <xs:element name="EMail" type="xs:string"/>
7      </xs:sequence>
8      <xs:attribute name="Rating" type="xs:nonNegativeInteger" use="required"/>
9    </xs:complexType>
10   <xs:element name="ObjectInfo">
11     <xs:complexType>
12       <xs:sequence>
13         <xs:element name="Description" type="xs:string"/>
14         <xs:sequence minOccurs="0" maxOccurs="unbounded">
15           <xs:element name="Comment" type="xs:string"/>
16           <xs:element name="CommentAuthor" type="UserType"/>
17         </xs:sequence>
18       </xs:sequence>
19     </xs:complexType>
20   </xs:element>
21   <xs:element name="Auction">
22     <xs:complexType>
23       <xs:sequence>
24         <xs:element ref="ObjectInfo"/>
25         <xs:element name="Seller" type="UserType"/>
26         <xs:element name="BuyPrice" type="xs:float" minOccurs="0"/>
27         <xs:element name="Shipping" type="xs:string" maxOccurs="unbounded"/>
28       </xs:sequence>
29     </xs:complexType>
30   </xs:element>
31 </xs:schema>
```

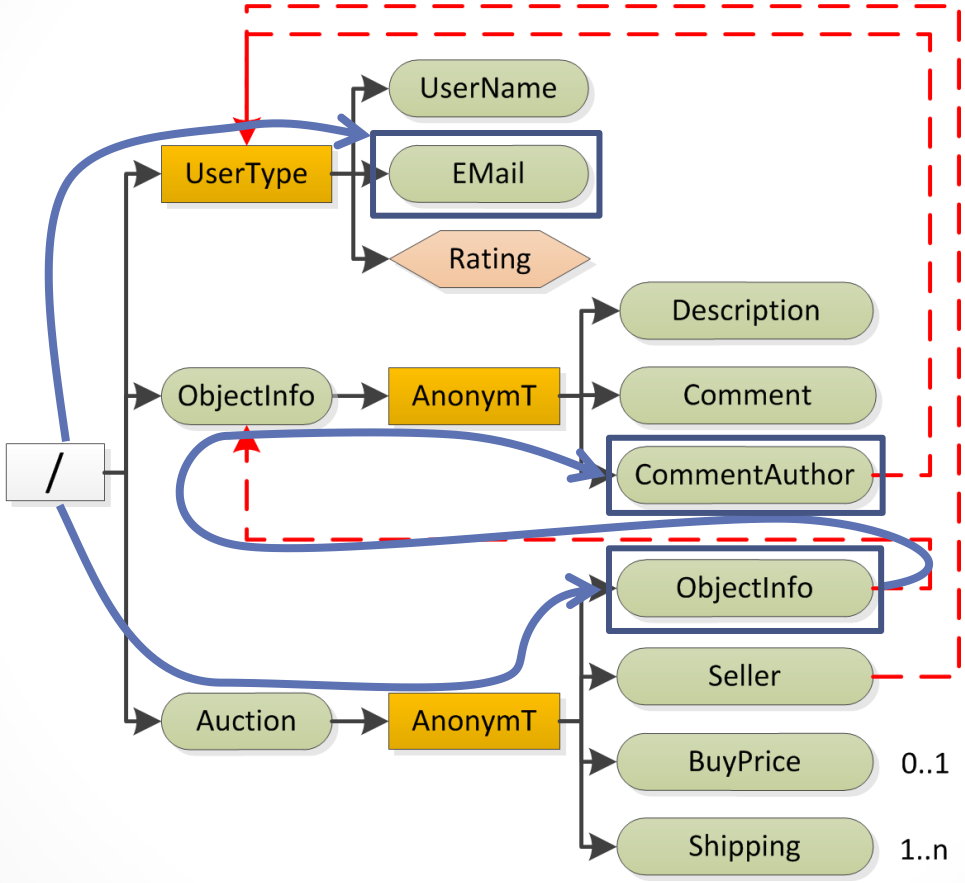
Data model – Low level



Data model – High level

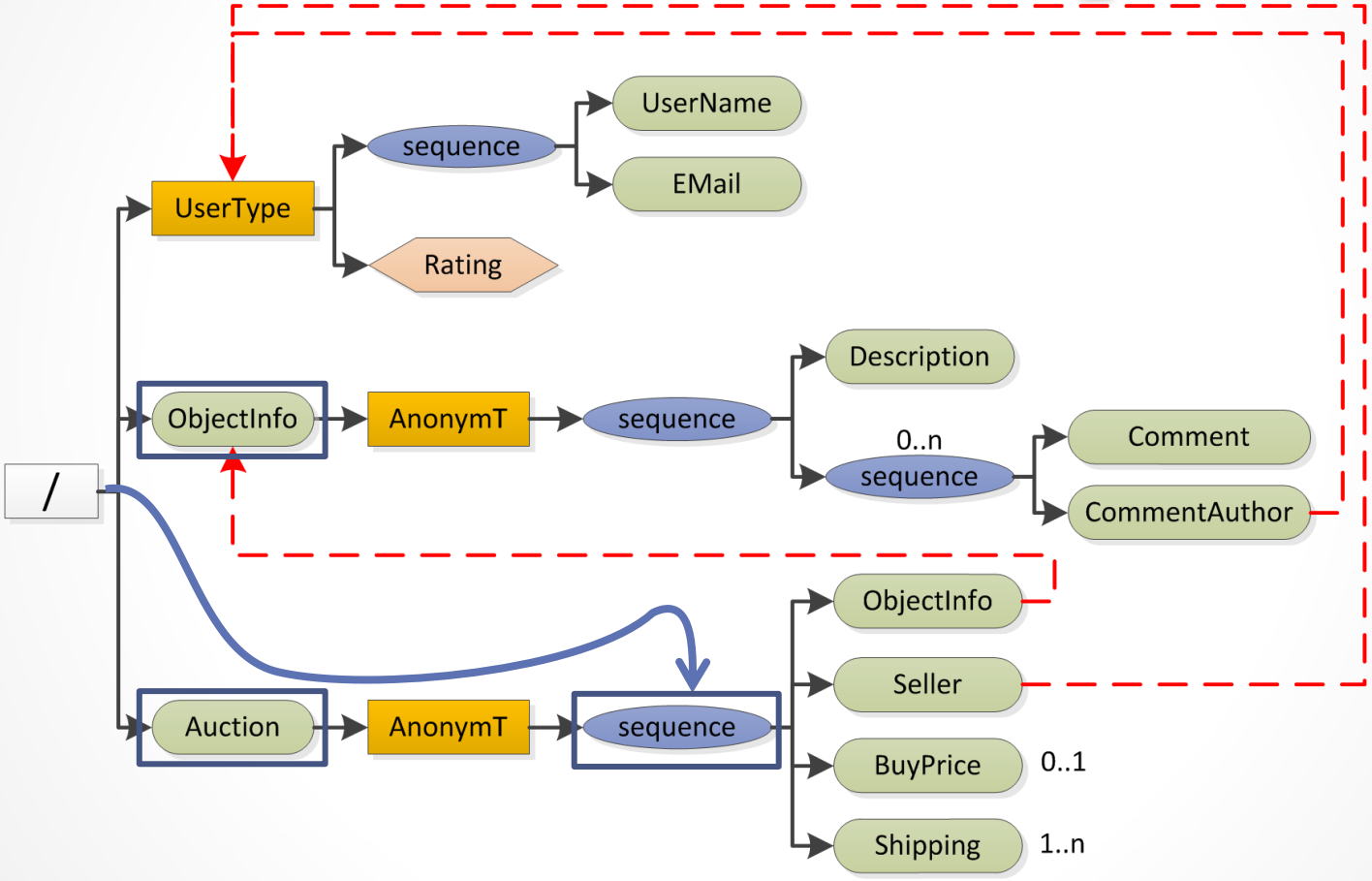


XSPath – Examples



/#UserType/EMail

XSPath – Examples



/Auction!sequence

Specification of schema modification and document adaptation

...

Schema modification

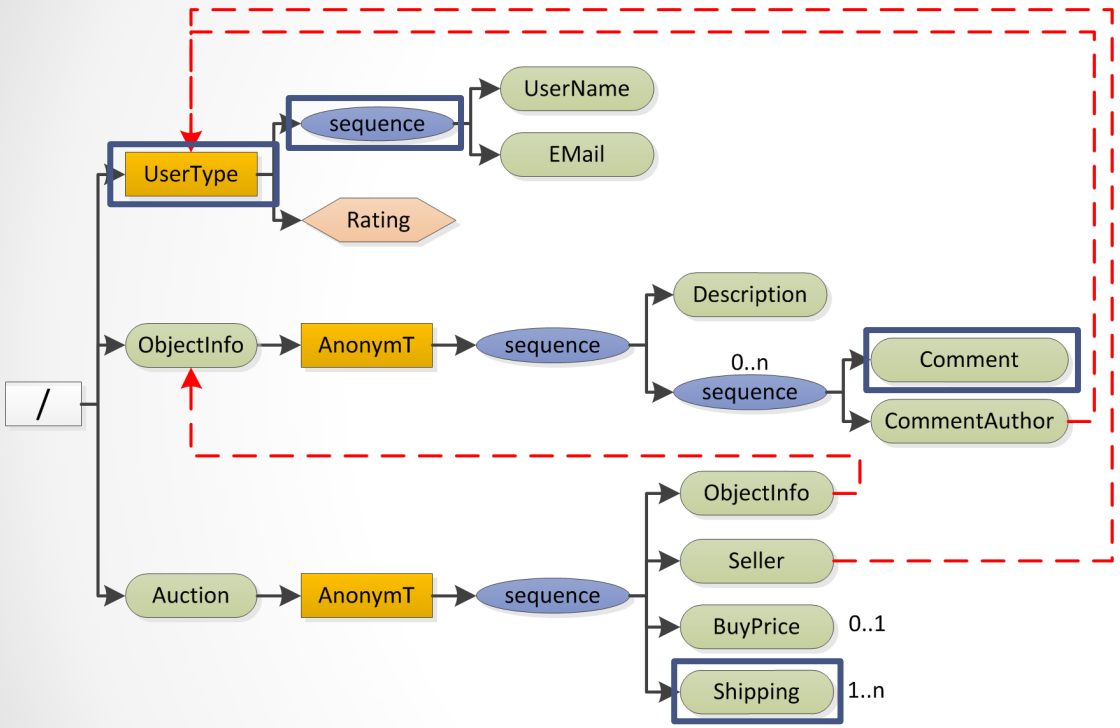
Insertion/Deletion	Modification	Simple types
Insert Attribute	Change Cardinality	Add Facets
Insert Element	Change Operator	Add Members
Insert Operator	Change Type	Change Base Type
Insert Substructure	Move	Remove Facets
Insert Type	Migrate	Remove Members
Delete	Rename	Replace Facets
		Replace Members

Document adaptation

- Three adaptation approaches
  - No adaptation
  - Automatic adaptation
  - User-defined adaptation
- Document revalidation

```
UPDATE SCHEMA ("Auction.xsd")/#UserType/Device
INSERT ELEMENT Insurance OF TYPE xs:string
AUTO ADAPT
```

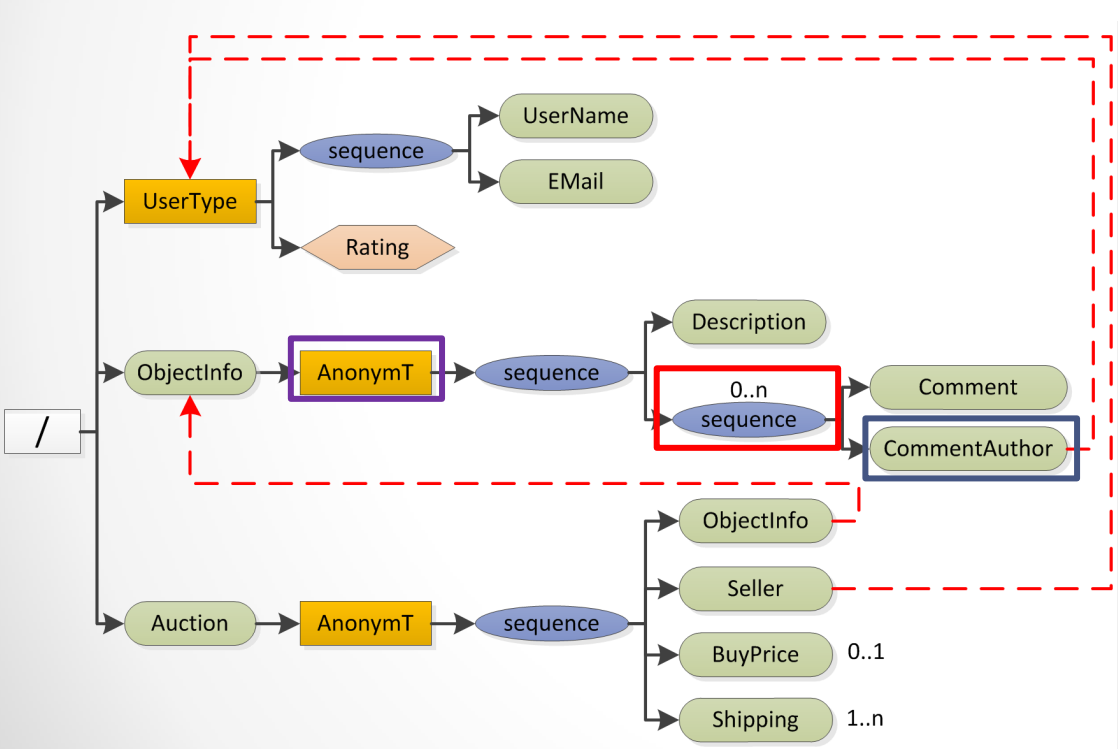
(Sets of) instances

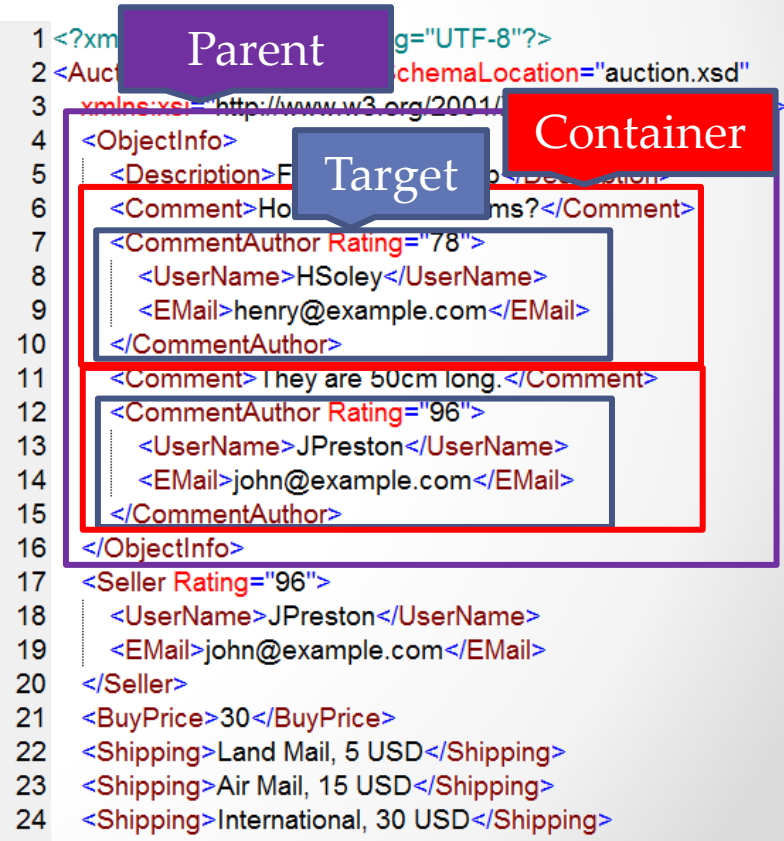



```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <Auction xsi:noNamespaceSchemaLocation="auction.xsd"
3    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
4    <ObjectInfo>
5      <Description>Flexible desk lamp</Description>
6      <Comment>How long are the arms?</Comment>
7      <CommentAuthor Rating="78">
8        <UserName>HSoley</UserName>
9        <EMail>henry@example.com</EMail>
10     </CommentAuthor>
11     <Comment>They are 50cm long.</Comment>
12     <CommentAuthor Rating="96">
13       <UserName>JPreston</UserName>
14       <EMail>john@example.com</EMail>
15     </CommentAuthor>
16   </ObjectInfo>
17   <Seller Rating="96">
18     <UserName>JPreston</UserName>
19     <EMail>john@example.com</EMail>
20   </Seller>
21   <BuyPrice>30</BuyPrice>
22   <Shipping>Land Mail, 5 USD</Shipping>
23   <Shipping>Air Mail, 15 USD</Shipping>
24   <Shipping>International, 30 USD</Shipping>
25 </Auction>
```

U.D. adapt - Modifications

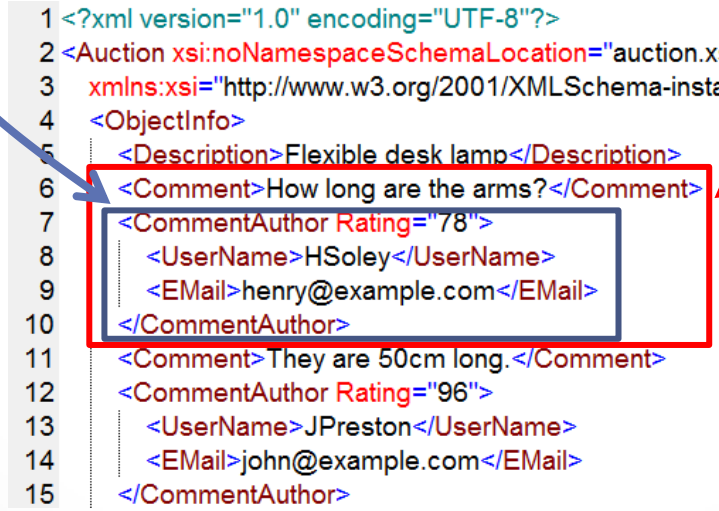
UPDATE SCHEMA ("Auction.xsd")/Auction/ObjectInfo/CommentAuthor
DELETE
XQueryUpdate(environment)





U.D. adapt - Modifications

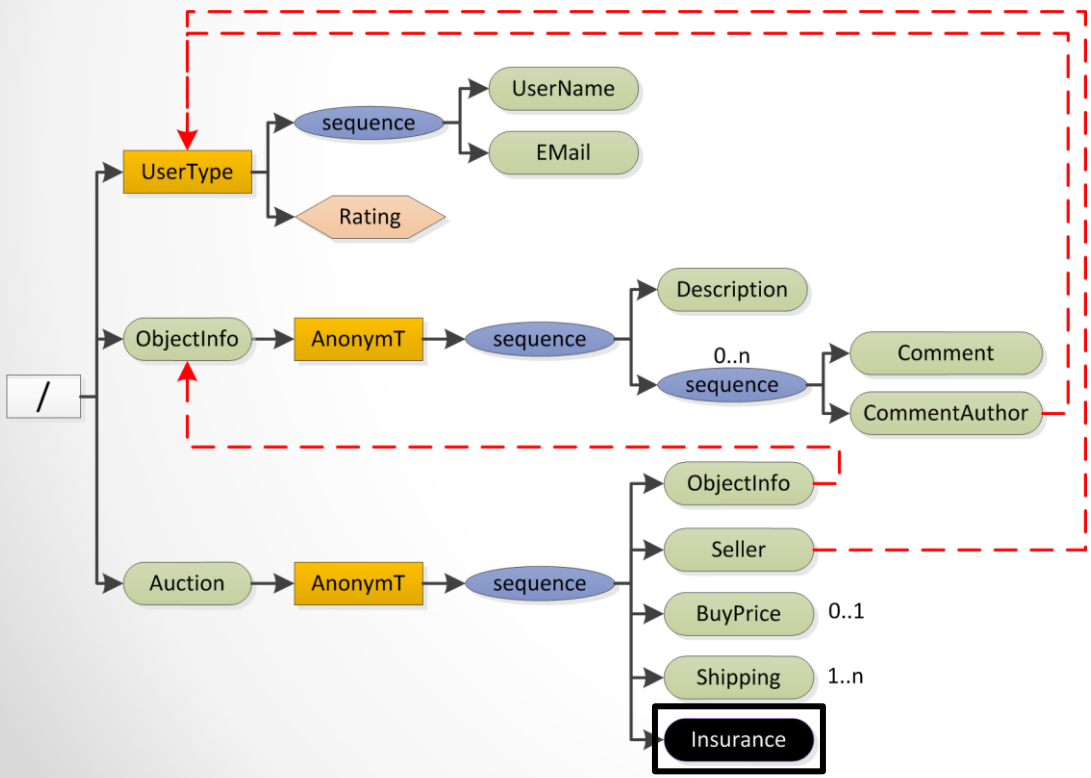
```
UPDATE CURRENT SCHEMA /Auction/ObjectInfo/CommentAuthor
DELETE
FOR EACH ENVIRONMENT
REFERENCING TARGET AS $CommAuthor, CONTAINER AS $CommInfo DO
{[
delete nodes $CommAuthor, replace value of node $CommInfo[self::Comment]
with concat($CommAuthor,$CommInfo[self::Comment])
]}
```



```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <Auction xsi:noNamespaceSchemaLocation="auction.xsd"
3    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
4    <ObjectInfo>
5      <Description>Flexible desk lamp</Description>
6      <Comment>How long are the arms?</Comment>
7      <CommentAuthor Rating="78">
8        <UserName>HSoley</UserName>
9        <EMail>henry@example.com</EMail>
10     </CommentAuthor>
11     <Comment>They are 50cm long.</Comment>
12     <CommentAuthor Rating="96">
13       <UserName>JPreston</UserName>
14       <EMail>john@example.com</EMail>
15     </CommentAuthor>
```

U.D. adapt - Insertions

UPDATE SCHEMA ("Auction.xsd")/Auction!sequence
INSERT ELEMENT Insurance OF TYPE xs:string
XQueryUpdate(environment)




```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <Auction xsi:noNamespaceSchemaLocation="auction.xsd"
3    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
4    <ObjectInfo>
5      <Description>Flexible desk lamp</Description>
6      <Comment>How long are the arms?</Comment>
7      <CommentAuthor Rating="78">
8        <UserName>HSoley</UserName>
9        <EMail>henry@example.com</EMail>
10     </CommentAuthor>
11     <Comment>They are 50cm long.</Comment>
12     <CommentAuthor Rating="96">
13       <UserName>JPreston</UserName>
14       <EMail>john@example.com</EMail>
15     </CommentAuthor>
16   </ObjectInfo>
17   <Seller Rating="96">
18     <UserName>JPreston</UserName>
19     <EMail>john@example.com</EMail>
20   </Seller>
21   <BuyPrice>30</BuyPrice>
22   <Shipping>Land Mail, 5 US
23   <Shipping>Air Mail, 15 USI
24   <Shipping>International, 30 USD</Shipping>
25 </Auction>
```

Ins. Pos.

U.D. adapt - Insertions

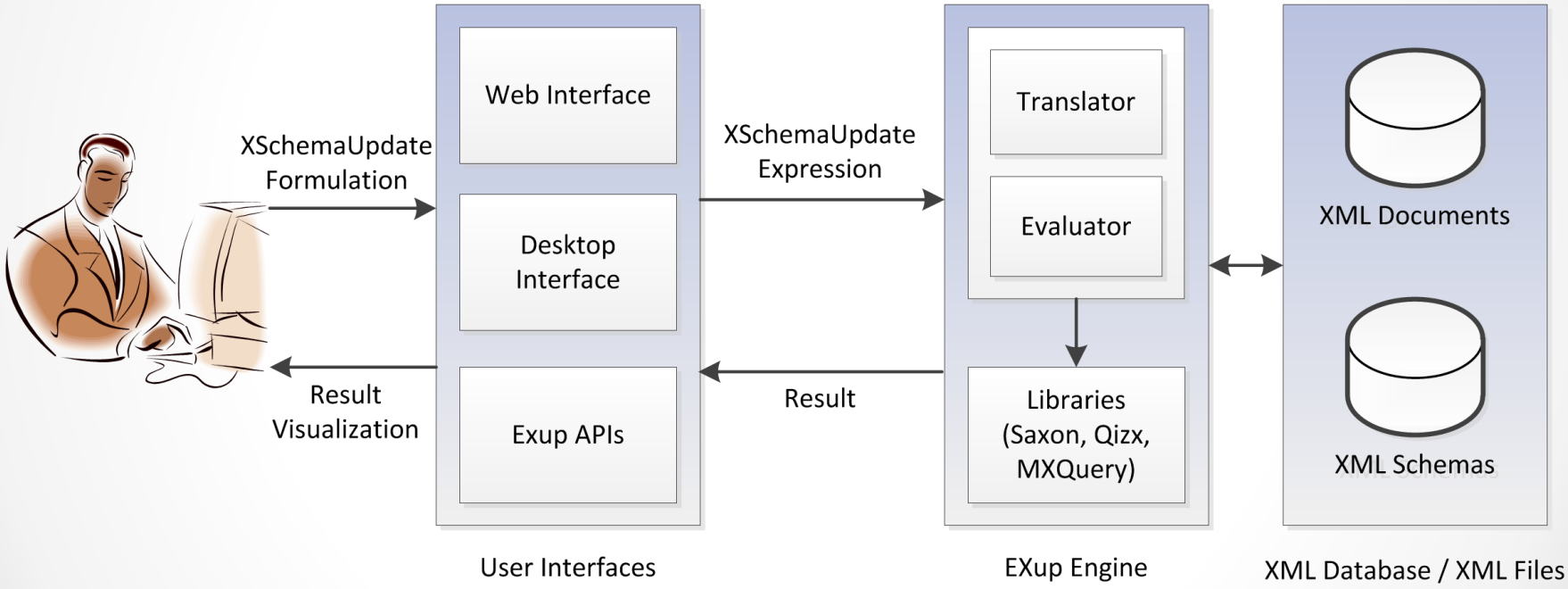
```
UPDATE SCHEMA (“Auction.xsd”)/Auction!sequence
INSERT ELEMENT Insurance OF TYPE xs:string
FOR EACH ENVIRONMENT
REFERENCING PARENT AS $Auction DO
{[
local:insertAtCurrentPosition(element Insurance {
if ($Auction/Seller/@Rating>95) then (“Insured”) else (“Not Insured”)})
]}
```

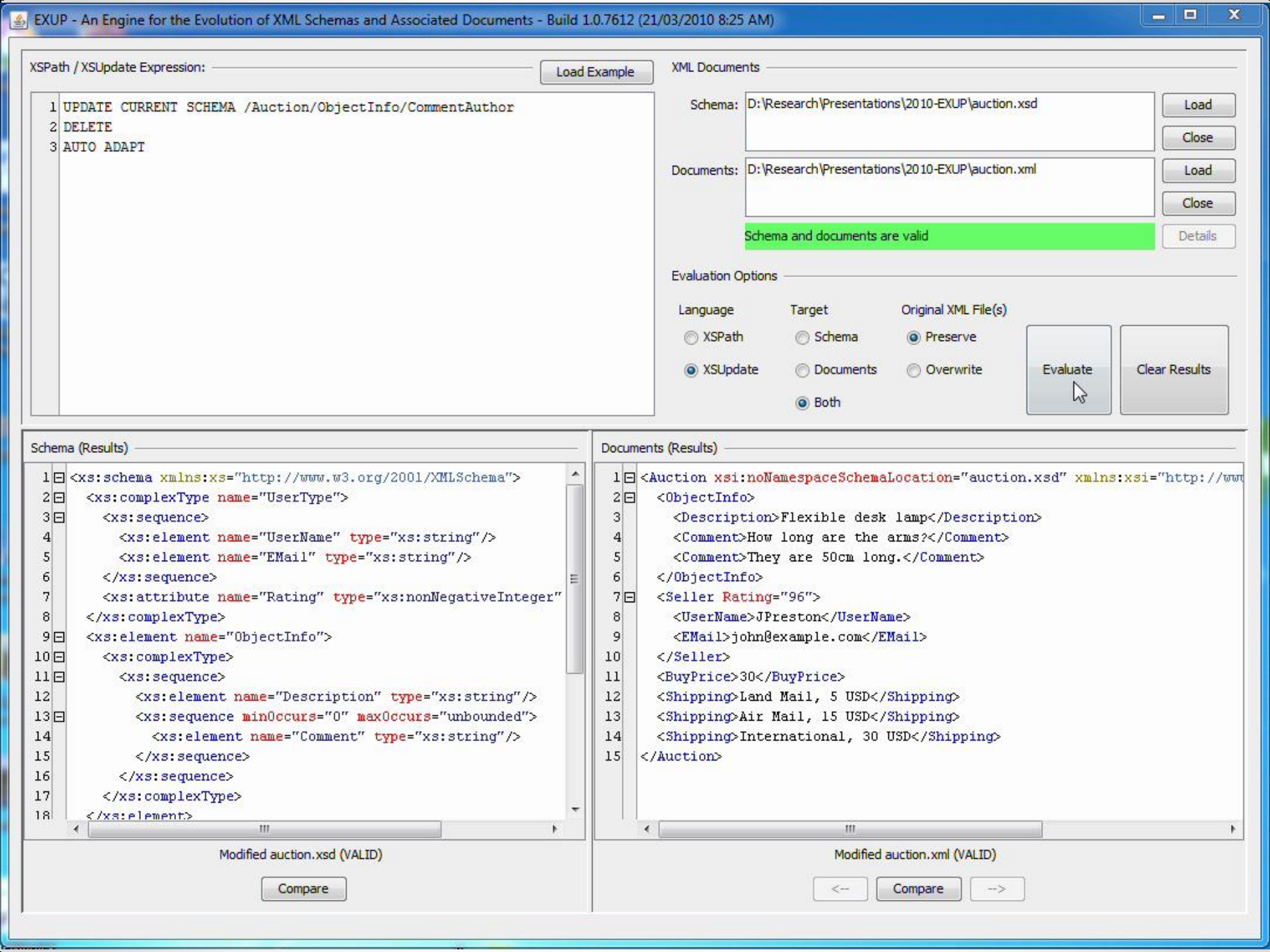
EXup

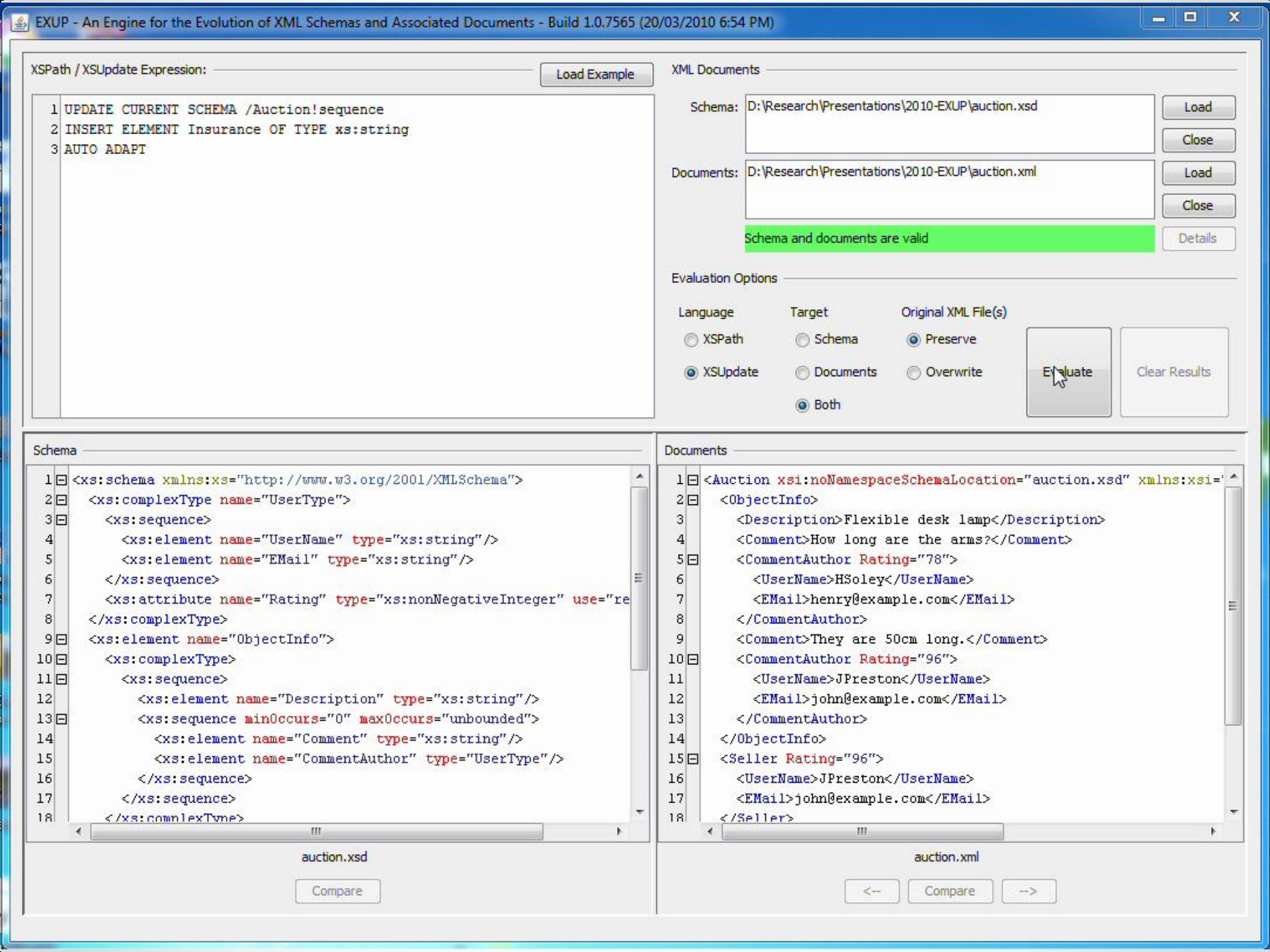
• • •

Evaluation of evolution specifications

EXup Architecture




EXUP - An Engine for the Evolution of XML Schemas and Associated Documents - Build 1.0.7612 (21/03/2010 8:25 AM)
XSPath / XSUpdate Expression:
Load Example
1 UPDATE CURRENT SCHEMA /Auction/ObjectInfo/CommentAuthor
2 DELETE
3 AUTO ADAPT
XML Documents
Schema:
D:\Research\Presentations\2010-EXUP\auction.xsd
Load
Close
Documents:
D:\Research\Presentations\2010-EXUP\auction.xml
Load
Close
Schema and documents are valid
Details
Evaluation Options
Language
XSPath
XSUpdate
Target
Schema
Documents
Both
Original XML File(s)
Preserve
Overwrite
Evaluate
Clear Results
Schema (Results)
1 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
2 <xs:complexType name="UserType">
3 <xs:sequence>
4 <xs:element name="UserName" type="xs:string"/>
5 <xs:element name="EMail" type="xs:string"/>
6 </xs:sequence>
7 <xs:attribute name="Rating" type="xs:nonNegativeInteger"
8 </xs:complexType>
9 <xs:element name="ObjectInfo">
10 <xs:complexType>
11 <xs:sequence>
12 <xs:element name="Description" type="xs:string"/>
13 <xs:sequence minOccurs="0" maxOccurs="unbounded">
14 <xs:element name="Comment" type="xs:string"/>
15 </xs:sequence>
16 </xs:sequence>
17 </xs:complexType>
18 </xs:element>
Documents (Results)
1 <Auction xsi:noNamespaceSchemaLocation="auction.xsd" xmlns:xsi="http://ww
2 <ObjectInfo>
3 <Description>Flexible desk lamp</Description>
4 <Comment>How long are the arms?</Comment>
5 <Comment>They are 50cm long.</Comment>
6 </ObjectInfo>
7 <Seller Rating="96">
8 <UserName>JPreston</UserName>
9 <EMail>john@example.com</EMail>
10 </Seller>
11 <BuyPrice>30</BuyPrice>
12 <Shipping>Land Mail, 5 USD</Shipping>
13 <Shipping>Air Mail, 15 USD</Shipping>
14 <Shipping>International, 30 USD</Shipping>
15 </Auction>
Modified auction.xsd (VALID)
Compare
Modified auction.xml (VALID)
<--
Compare
-->


EXUP - An Engine for the Evolution of XML Schemas and Associated Documents - Build 1.0.7565 (20/03/2010 6:54 PM)
XSPath / XSUpdate Expression:
Load Example
1 UPDATE CURRENT SCHEMA /Auction!sequence
2 INSERT ELEMENT Insurance OF TYPE xs:string
3 AUTO ADAPT
XML Documents
Schema:
D:\Research\Presentations\2010-EXUP\auction.xsd
Load
Close
Documents:
D:\Research\Presentations\2010-EXUP\auction.xml
Load
Close
Schema and documents are valid
Details
Evaluation Options
Language
XSPath
XSUpdate
Target
Schema
Documents
Both
Original XML File(s)
Preserve
Overwrite
Evaluate
Clear Results
Schema
1 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
2 <xs:complexType name="UserType">
3 <xs:sequence>
4 <xs:element name="UserName" type="xs:string"/>
5 <xs:element name="EMail" type="xs:string"/>
6 </xs:sequence>
7 <xs:attribute name="Rating" type="xs:nonNegativeInteger" use="re
8 </xs:complexType>
9 <xs:element name="ObjectInfo">
10 <xs:complexType>
11 <xs:sequence>
12 <xs:element name="Description" type="xs:string"/>
13 <xs:sequence minOccurs="0" maxOccurs="unbounded">
14 <xs:element name="Comment" type="xs:string"/>
15 <xs:element name="CommentAuthor" type="UserType"/>
16 </xs:sequence>
17 </xs:sequence>
18 </xs:complexType>
auction.xsd
Compare
Documents
1 <Auction xsi:noNamespaceSchemaLocation="auction.xsd" xmlns:xsi="
2 <ObjectInfo>
3 <Description>Flexible desk lamp</Description>
4 <Comment>How long are the arms?</Comment>
5 <CommentAuthor Rating="78">
6 <UserName>HSoley</UserName>
7 <EMail>henry@example.com</EMail>
8 </CommentAuthor>
9 <Comment>They are 50cm long.</Comment>
10 <CommentAuthor Rating="96">
11 <UserName>JPreston</UserName>
12 <EMail>john@example.com</EMail>
13 </CommentAuthor>
14 </ObjectInfo>
15 <Seller Rating="96">
16 <UserName>JPreston</UserName>
17 <EMail>john@example.com</EMail>
18 </Seller>
auction.xml
<--
Compare
-->

XSUpdate Translation

- Evolution object identification (XSPath)
  - XPath or XQuery
    - Generic
    - Specific
- Schema modification
  - XQueryUpdate
    - Generic
    - Specific
- Document adaptation
  - XQueryUpdate

XSPath generic translation

1. Split path into atomic components
2. Translate each component independently
3. Combine translated fragments

/Auction/type()[isAnonym()]/Description

HL::child::type()[isAnonym()]

- Child axis at high level
  - Path expression
- "type()" node selector
  - Predicate
- "isAnonym()" predicate
  - Predicate
- Step translation
  - Concatenation of translated parts
- Path translation
  - Translated fragments concatenation

On-the-fly optimization

- Translation parts can be omitted
  - Knowing the kind of nodes on which the step is evaluated
  - Analyzing the step components

- Child axis identifies
  - ~~Child declarations~~
  - ~~Declarations in the current type structure~~
  - ~~(Declarations in) referenced types~~
  - ~~(Declarations in) referenced elements~~

Schema modification translation

1. Identify preconditions
   - Preconditions as conditional statements
   - Static preconditions evaluation
2. Instantiate primitive application expression

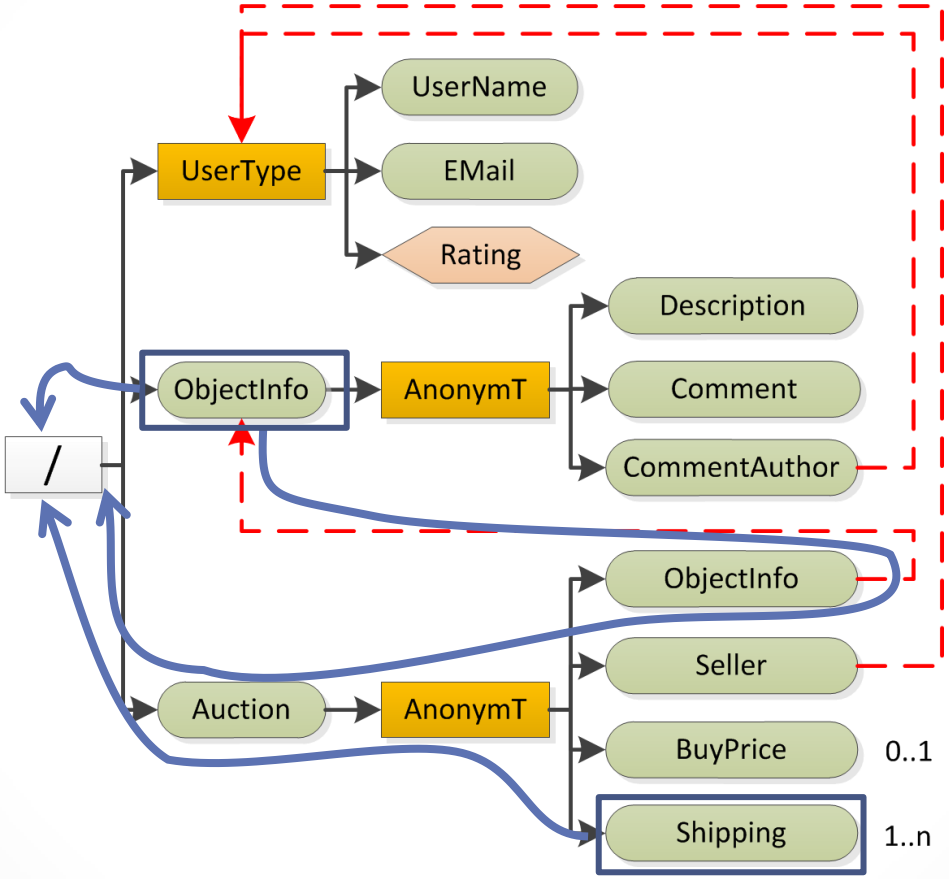
```
declare function local:evoObject()
{… Translation of evolution object expression…}

for $evo in local:evoObjects()
where applicability preconditions hold
return primitive application
```

ADD FACET maxInclusive="5"

Instances identification

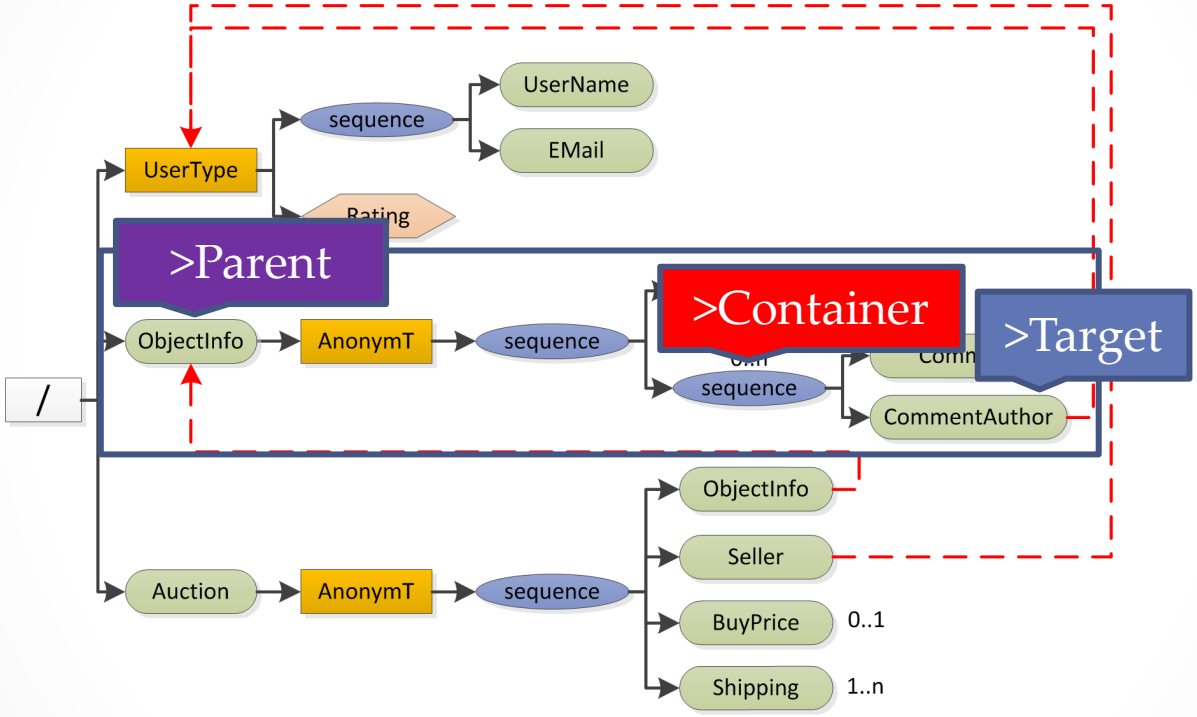
- Backward navigate incoming links
- Consider only named element nodes along navigated paths



/Auction/ObjectInfo
/ObjectInfo

Env. identification

UPDATE SCHEMA ("Auction.xsd")/Auction/ObjectInfo/CommentAuthor DELETE



1) Identify the part(s) of the grammar involved
2) Identify the part(s) of the document involved
3) Match the grammar part(s) against the document part(s)

Document adaptation

UPDATE SCHEMA (“Auction.xsd”)/Auction/ObjectInfo/CommentAuthor
DELETE
FOR EACH ENVIRONMENT
REFERENCING TARGET AS $CommAuthor, CONTAINER AS $CommInfo
DO {[XQueryUpdate(environment)]}

declare updating function local:adaptFunction($CommAuthor, $CommInfo)
{XQueryUpdate(environment)}

let $evoObject :="/Auction/ObjectInfo/CommentAuthor"
let $modPrimitive= " DELETE"
let $schema := doc("Auction.xsd")

for $document in $documentCollection
for $env in local:identifyEnvironments($schema, $doc, $evoObj, $modPrimitive)
local:adaptFunc(local:getTarget($env),local:getContainer($env))

(Incremental) validation

- Validate only the parts of the documents affected by the evolution
- Effort required depends on modification primitive and adaptation approach

Primitive	No Adapt.	Auto Adapt.	U.D. Adapt (Environment)
REMOVE	Path / Static	Statical	Env. Revalidation
INSERT ELEMENT	Path / Static	Statical	Env. Revalidation

Conclusions

- EXup features
  - Evolution specification / evaluation
  - Multiple user-interfaces
  - Multiple libraries

- Future developments
  - Complete automatic adaptation
  - Enhance XML Schema support
  - Evolution composition
  - Enhance incremental validation in case of user defined adaptation

Thank you for your attention

Questions are welcome