SchemaCom – An XML Schema Comparator

Ihe Onwuka

Systems Engineer
LS Technologies

Ihe.Onwuka@gmail.com

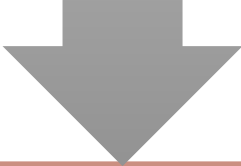# About Me

Been working with XML since 2003.

Was tricked into learning about XML by our then Consulting Director – who told me "It's just like COBOL"

Currently System Engineer with LS Technologies architecting complex data models for the US Federal government

SchemaCom What's Your Problem Then

Schemas change but how they've changed isn't always communicated

⬇

Need to port existing data to be conformant with the new version of the schema.

⬇

Need to know how and what to map between instances of schema versions

Schema Comparison is a hard problem

That's quite a challenge, because so many differences are possible: you may have to make compromises depending on what you actually encounter. (For example, you may or may not need to bother with named model groups, depending on whether either of the schemas you are comparing actually uses them. Most schemas don't.)

https://softwarerecs.stackexchange.com/questions/48067/diff-tool-for-xml-schema-file

You really don't want to do this by hand and for sure nobody is going to write such a complex method from scratch, it doesn't fit the Q&A nature of StackOverflow. However, several tools exist that can do this for you:

https://stackoverflow.com/questions/32522975/comparing-two-xsd-files-for-differences-in-elements

https://docs.deltaxml.com/xml-compare/latest/comparing-xml-schema-15008896.html

Schema Comparison – Survey of whats out there

- schemaDiff - https://www.membrane-soa.org/soa-model-doc/1.4/cmd-tool/schemadiff-tool.htm

The changes break the interface.
Element createResponse has changed:
  ComplexType has changed:
    Sequence has changed:
      Element NewElementForTest added.

ComplexType GetAllType has changed:
  Annotation removed.

Imported schema http://predic8.com/material/1/ has changed:
  Element manufacturer added.
  ComplexType ArticleType has changed:
    Annotation added.
    Sequence has changed:
      Position of element name changed from 1 to 2.
      Position of element description changed from 2 to 3.
      Position of element price changed from 3 to 5.
      Position of element id changed from 4 to 1.
      Element availability added.

Imported schema http://predic8.com/common/1/ has changed:
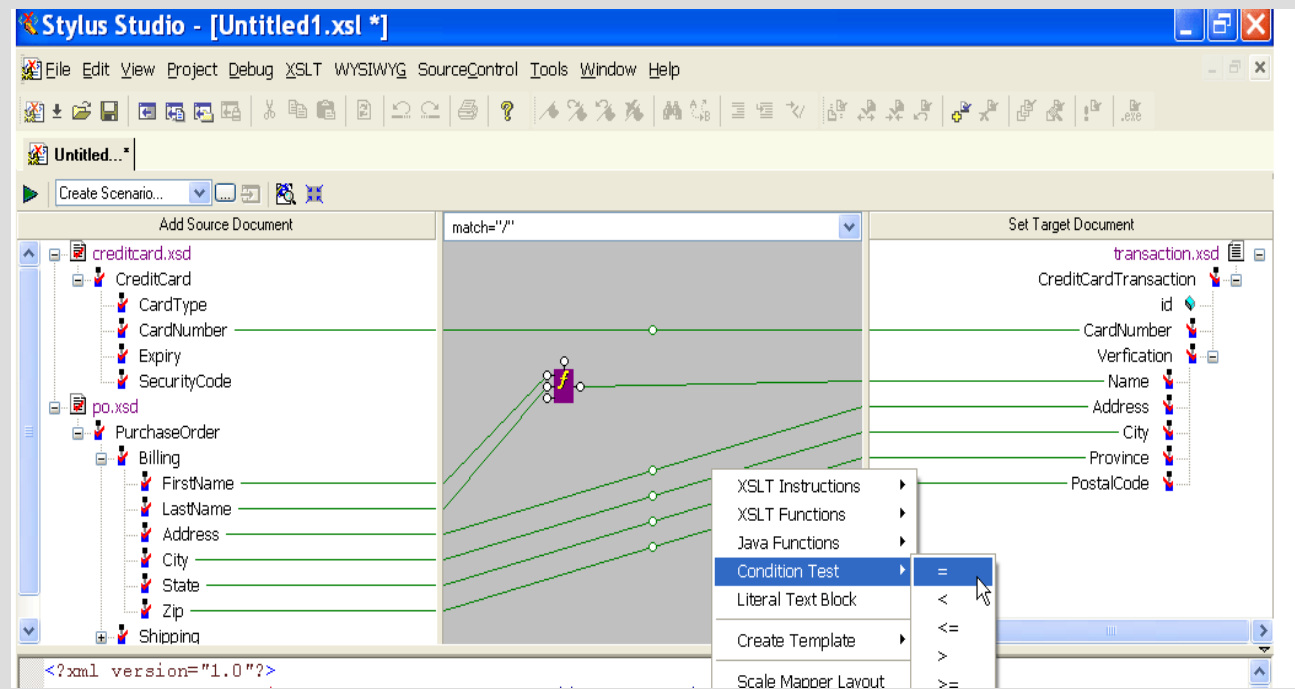 ComplexType PersonType has changed:
    Sequence has changed:
      Element id has changed:
        The type of element 'id' has changed from xsd:string to tns:IdentifierType.

Schema Comparison – Survey of whats out there

- Stylus Studio – http://www.stylusstudio.com/xsd-mapping.html

- https://www.altova.com/diffdog#XML_Schema_compare

What's missing

No guidance on how to populate what the tools tell us has been added.

Several ways in which schemas can change without impacting mapping.

- If an annotation is added or altered does it affect the mapping?
- If the ordering in a content model changes does it affect the mapping?
- If the type of primitive changes does it affect the mapping?

The information that these tools give us on changes is limited

Tools that allow us to do the mappings but offer no guidance on what the mapping should be.

Mapping Aids
What does FeatureType map to in the new version

Mapping Aids
What does FeatureType map to in the new version

A Mapping Aid

FeatureType Became ProvenanceType
Who knew?

Mappings that are hard to spot

Select Schemas to Compare `fixm/4.0 vs fixm/4.1` ⌄ | Search Types `fac` | **To expand a types details click on it's button below**

Source: Fixm/4.0-Target: Fixm/4.1

Dropped Complex Types

New Complex Types

DepartureDelayType

TfdmDepartureType

Matched Complex Types

SurfaceEventsType-fixm/4.0
stddsSpotIn(fb: Time Type)
SurfaceEventsType(complexType)
stddsSpotOut(fb: Time Type)
SurfaceEventsType(complexType)
stddsWheelsOff(fb: Time Type)
SurfaceEventsType(complexType)
stddsWheelsOn(fb: Time Type)
SurfaceEventsType(complexType)

SurfaceEventsType-fixm/4.1
@stddsSpotIn(fb: Time Type)
SurfaceEventsType(complexType)
@stddsSpotOut(fb: Time Type)
SurfaceEventsType(complexType)
@stddsWheelsOff(fb: Time Type)
SurfaceEventsType(complexType)
@stddsWheelsOn(fb: Time Type)
SurfaceEventsType(complexType)

Identical Complex Types

Mappings that are hard to spot

Select Schemas to Compare [fixm/4.0 vs fixm/4.1 ▼] | Search Types [datal] To expand a types details click on it's button below

Source: Fixm/4.0-Target: Fixm/4.1

Dropped Complex Types   **New Complex Types**   **Matched Complex Types**   **Identical Complex Types**

CommunicationCapabilitiesType-fixm/4.0
@otherDataLinkCapabilities(fb:CharacterStringType)
CommunicationCapabilitiesType-fixm/4.1
@otherDatalinkCapabilities(fb:CharacterStringType)

Select Schemas to Compare [fixm/3.0 vs fixm/4.0 ▼] | Search Types [fac] To expand a types details click on it's button below

Source: Fixm/3.0-Target: Fixm/4.0

Dropped Complex Types   **New Complex Types**   **Matched Complex Types**   **Identical Complex Types**

SurfaceEventsType

TelephoneContactType-fixm/3.0
@facimile(ff:TextPhoneType)
TelephoneContactType-fixm/4.0
@facsimile(fb:TextPhoneType)
NasFlightType-fixm/4.0

Superset content models

A Validation Aid – Beware false positives caused by namespaces

Select Schemas to Compare: saml/1.1 vs saml/2.0 — Search Types [   ] To expand a types details click on it's button below

Source: Saml/1.1-Target: Saml/2.0

Dropped Complex Types
- AudienceRestrictionConditionType
- DoNotCacheConditionType
- SubjectStatementAbstractType
- NameIdentifierType
- AuthenticationStatementType
- AuthorityBindingType
- AuthorizationDecisionStatementType
- AttributeDesignatorType
- AuthenticationQueryType
- AuthorizationDecisionQueryType
- ResponseAbstractType

New Complex Types
- BaseIDAbstractType
- NameIDType
- SubjectConfirmationDataType
- KeyInfoConfirmationDataType
- AudienceRestrictionType
- OneTimeUseType
- ProxyRestrictionType
- AuthnStatementType
- AuthnContextType
- AuthzDecisionStatementType
- SharedSecretChallengeResponseType
- PublicKeyType
- AuthnContextDeclarationBaseType
- IdentificationType
- TechnicalProtectionBaseType
- OperationalProtectionType
- AuthnMethodBaseType
- GoverningAgreementsType
- GoverningAgreementRefType
- PrincipalAuthenticationMechanismType
- AuthenticatorBaseType
- ComplexAuthenticatorType
- AuthenticatorTransportProtocolType
- KeyActivationType
- KeySharingType
- PrivateKeyProtectionType
- PasswordType
- RestrictedPasswordType

Matched Complex Types
- AssertionType-saml/1.1
- AssertionType-saml/2.0
- ConditionsType-saml/1.1
- ConditionsType-saml/2.0
- AdviceType-saml/1.1
- AdviceType-saml/2.0
- SubjectType-saml/1.1
- SubjectType-saml/2.0
- SubjectConfirmationType-saml/1.1
- SubjectConfirmationType-saml/2.0
- SubjectLocalityType-saml/1.1
- SubjectLocalityType-saml/2.0
- EvidenceType-saml/1.1
- EvidenceType-saml/2.0
- AttributeStatementType-saml/1.1
- AttributeStatementType-saml/2.0
- AttributeType-saml/1.1
- AttributeType-saml/2.0
- RequestAbstractType-saml/1.1
- RequestAbstractType-saml/2.0
- RequestType-saml/1.1
- RequestType-saml/2.0
- SubjectQueryAbstractType-saml/1.1
- SubjectQueryAbstractType-saml/2.0
- AttributeQueryType-saml/1.1
- ResponseType-saml/1.1
- ResponseType-saml/2.0
- ResponseType-saml/2.0

Identical Complex Types
- ConditionAbstractType
- StatementAbstractType
- ActionType
- StatusType
- StatusDetailType

Is that a duplicate definition of ResponseType or a false positive because namespaces were ignored.

Project Management Uses – Identifying Churn Rates

Above shows less volatility between v4.0 and 4.1

Project Management Uses – Identifying Churn Rates

Only 2 complexTypes remained completely unchanged between v4.0 and v4.3

Other Approaches

Walmsley, Priscilla. "Comparing and diffing XML schemas." Presented at Balisage: The Markup Conference 2015, Washington, DC, August 11 - 14, 2015. In *Proceedings of Balisage: The Markup Conference 2015*. Balisage Series on Markup Technologies, vol. 15
(2015). https://doi.org/10.4242/BalisageVol15.Walmsley01.

Why was this was not my development experience?

The approach

- Use XSLT. (What else?)
- Use a pipeline (in Ant) to break it down into several simpler steps.
- Build it "iteratively". Which means:
  1. Write some code.
  2. Run it.
  3. Bang head against wall.
  4. Reread note to self.
  5. Goto 1.
- Current status: basic working code, much untested code, long list of future enhancements.
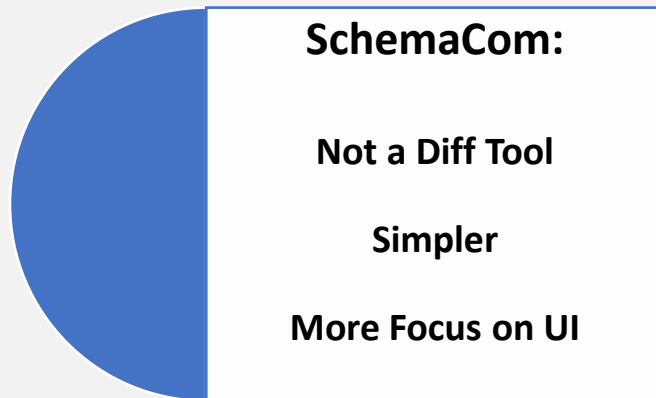
Steps

1. Canonicalize
2. Flatten
3. Simplify
4. Sort
5. Find Diffs
6. Determine Backward Compatibility
7. Report

SchemaCom only entails steps 2, 5 and 7

190 lines of XSLT split over 4 modules

120 lines of XQuery split over 2 modules

122 lines of CSS

SchemaCom:

Not a Diff Tool

Simpler

More Focus on UI

Designed to assist in the derivation of mappings for vocabulary translation – only incidentally a diff tool.

Not concerned with subsetting or backward compatibility

Only tested on XSD 1.0

Compares versions which could consist of multiple schemas rather than physical schemas

Ignores factors that don't visibly impact the instance e.g changes to compositors, annotations and documentation.

Compares after inlining references to elements, groups and attributeGroups

Not concerned with changes to element sequencing as translation algorithm can be based on the sequence in the target vocabulary

Comparison is based on entity names. No attempt is made to compare anonymous types.

Design – Generate Schemas XML

AllSchemas.xsl (Standalone Step)
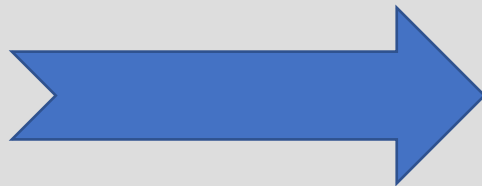Place all the schema files into a parent folder for the whole vocabulary

```
<allSchemas>
    <schemas version=…
        for each version in vocabulary
            for each schema in version
                inline groups and attributeGroups
                remove annotations documentation
```



```
▼<allSchemas>
  ▶<schemas version="fixm/3.0">
    ...
  </schemas>
  ▶<schemas version="fixm/4.0">
    ...
  </schemas>
  ▶<schemas version="fixm/4.1">
    ...
  </schemas>
  ▶<schemas version="fixm/4.2">
    ...
  </schemas>
  ▶<schemas version="fixm/4.3">
    ...
  </schemas>
</allSchemas>
```

Design – Generate Comparsions

SchemaDeltas.xq

```
<deltas>
    for each $schemaPair in pairUpSchemas
        fn:transform(complexDeltas.xsl, $schemaPair)
```

deltaForms.xq

Generates Xform UI

Rendered as xhtml

Delta files from previous step

Bind to UI controls

Xform model

Reiterating what made this an approachable problem

Can choose to ignore changes that won't be visible in the XML instances.

We are aiming for a best guess and not unerring accuracy.

Whatever has been achieved here is the result of naively trying to solve a similar but different problem.

Future Work?

Performance degradation of UI when the size of the XForm model is large (say 1MB)

Adapting interface to allow the specification of mappings, however challenge to find space for additional UI interactions

Encoding a schema as a function of it's similarities and differences to another schema

Acknowledgements

Hans-Jurgen Rennau reviewed the paper, application and provided valuable and extensive feedback.