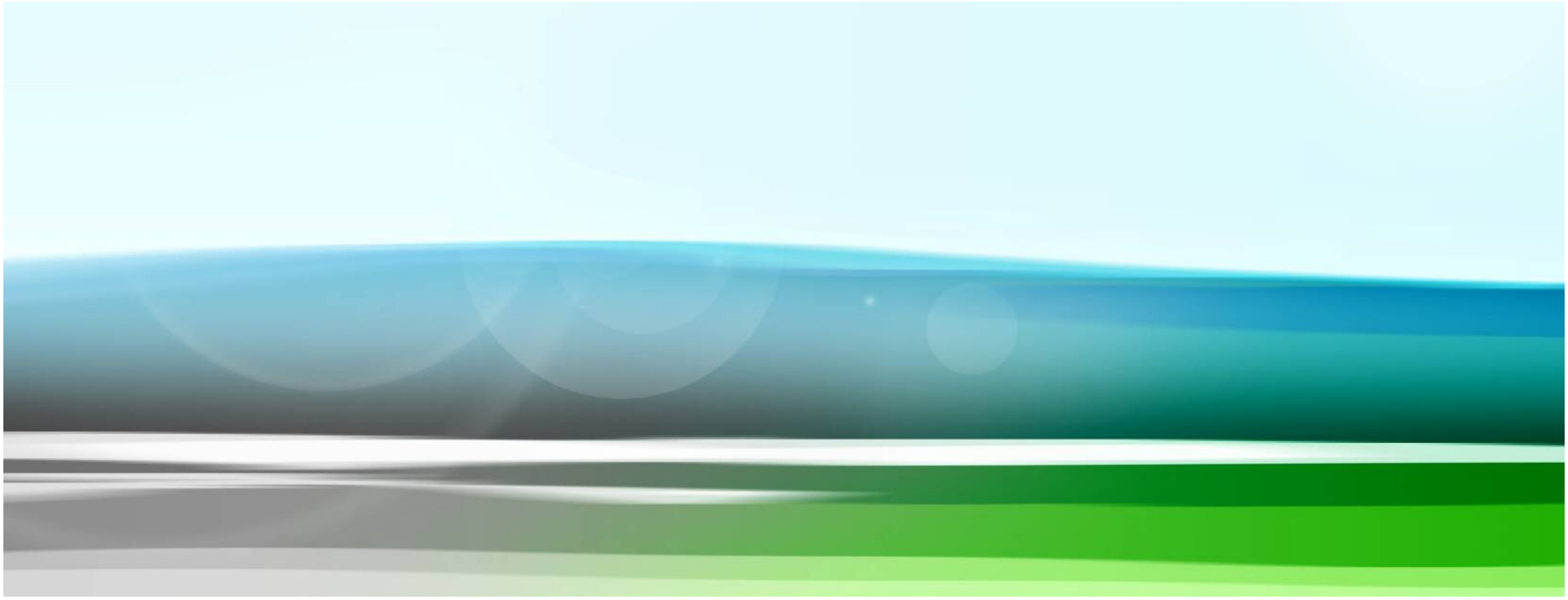


XML schema versioning

Martin G. Skjæveland
15 June 2009

Overview

- Best Practices
- Current status
- Versioning approaches
- Proposals
- Results

- XML Schema Versioning,
  - xml-dev list group, a leading technical forum for XML discussion
  - http://www.xfront.com/Versioning.pdf

Best Practices

- Capture the schema version in the XML schema.
- Identify in instance documents what version of the schema with which the instance is compatible.
- Make previous versions of the XML schema available.
- Come up with convention for schema versioning indicating significant changes or just extensions.
- If a new schema is only extended, one should not invalidate existing documents.
- If a new schema invalidates existing documents, one should change the target namespace.

Current status

- DDR 1.0:

```
<xsd:schema targetNamespace="http://www.witsml.org/schemas/131" version="1.3.1(NPD)">
  <xsd:documentation>WITSML - Daily Drillling Report - Norwegian Profile</xsd:documentation>
```

- DDR 1.1:

```
<xsd:schema targetNamespace="http://www.witsml.org/schemas/1series">
  <xsd:documentation>WITSML - Daily Drillling Report - Norwegian Profile</xsd:documentation>
```

- MPR 1.0:

```
<xsd:schema targetNamespace="http://www.witsml.org/schemas/131" version="1.3.1">
```

- DPR 1.0:

```
<xsd:schema targetNamespace="http://www.witsml.org/schemas/131">
```

- No own versioning.
  - Uses WITSML namespace.
- Difficult to tell schemas apart. Difficult to tell instances apart?

Versioning approaches

1. Change the internal schema version attribute.
2. Create a `schemaVersion` attribute.
3. Change the schema's `targetNamespace`.
4. Change the location of the schema.

Versioning approach no. 1

1. Change the internal schema version attribute.

```
<xsd:schema targetNamespace="http://www.witsml.org/schemas/131" version="1.3.1(NPD)“>
```

Pros:

- Easy. Part of schema language.
- Does not invalidate old instances.
- Versioning information is available to applications.

Cons:

- A validator ignores the `version` attribute. It is not an enforceable constraint.

Versioning approach no. 2

2. Create a `schemaVersion` attribute (to root element).

```
<xsd:schema ...>
   ...
   <xsd:attribute name="schemaVersion" type="xs:decimal" use="required" fixed="1.0"/>
```

Pros:

- An enforceable constraint. Instances must have the correct version number.

Cons:

- The `schemaVersion` number must match exactly. An instance document cannot indicate validity against multiple versions.
- Extends the schema with a new attribute, not standardized.

Versioning approach no. 3

3. Change the schema's `targetNamespace`.

```
<xsd:schema targetNamespace="http://www.witsml.org/schemas/131" version="1.3.1(NPD)“>
```

Pros:

- An enforceable constraint. Instances must use the correct namespace.
- Applications are notified of the change of namespace, e.g., a validator would not recognize the new namespace.

Cons:

- Invalidates old instance documents.
- All other schemas importing the new schema need to update to new namespace.

Versioning approach no. 4

4. Change the location of the schema.

```
...MonthlyProductionReport/1.1/schema.xsd (schema)
xsi:schemaLocation=“ .../MonthlyProductionReport/1.1/schema.xsd“ (instance)
```

Pros:

- Easy.
- All versions are available.

Cons:

- Forces all instance documents and related schema to import the new schema at different location.
- For users of old versions the change to new location is not ”discovered”.
- The `schemaLocation` attribute in instance document is optional. It is only a hint to locate the schema. Relying on this attribute is not good practice.

Proposal version numbering

- Use version numbering with one decimal number.
  - E.g., 1.0, 1.20, 2.0
- Increment to next integer if the change is “significant”.
  - Big changes to either content or structure.
  - E.g., 1.x to 2.0
- Increment by one decimal if the change is “small”.
  - E.g., 1.0 to 1.1

- Easy.
- Using backward compatible measurement is unnecessary strong requirement.

Proposal versioning system

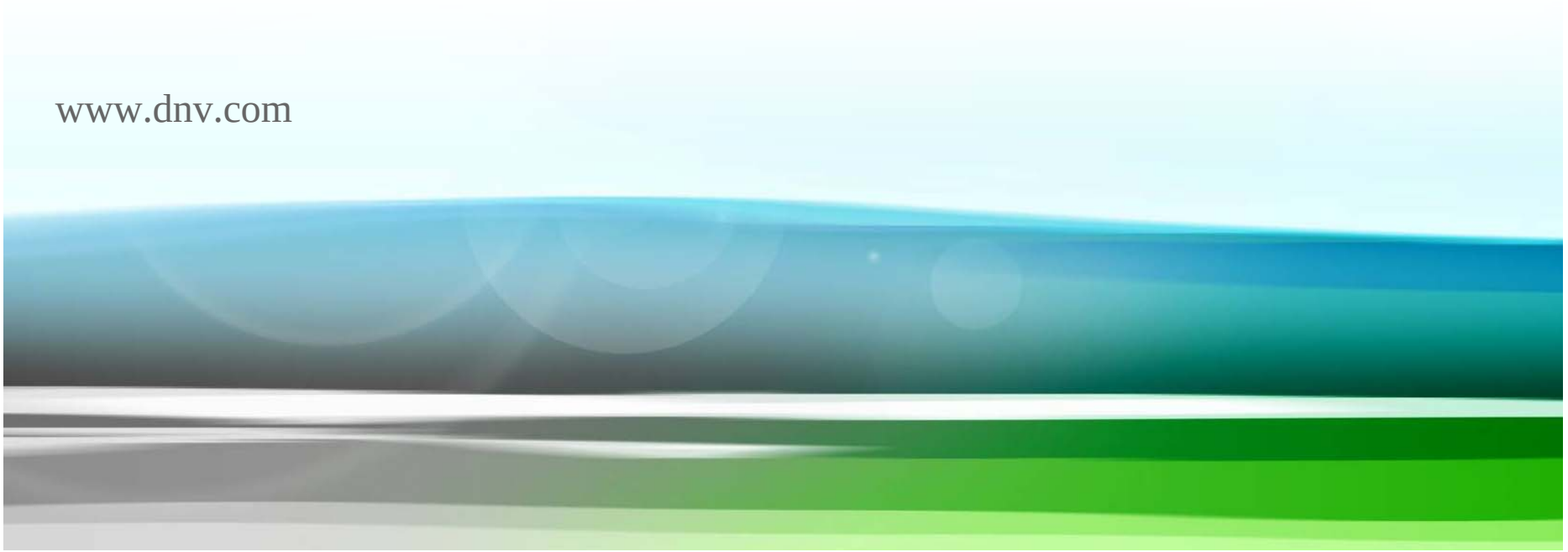
- Change `targetNamespace`
  - `http://www.epim.no/standards/DailyDrillingReport/1.0/[filename].xsd` ?
  - `http://www.epim.no/standards/WITSML/1.3.1/DailyDrillingReport/1.0/[filename].xsd` ?
- Publicise schemas at `targetNamespace` location.
  - Users expect to find the schema at the namespace URL.
  - Publicise latest version always in same location?
    - `.../DailyDrillingReport/"current"/[filename].xsd` (Note: no version information.)
- Rationale:
  - Identifies standard, version, organization by namespace.
  - Most changes are significant? Instances will be invalid against a new schema anyway.
  - Similar to WITSML versioning system.

Results

- Capture the schema version in the XML schema.
- Identify in instance documents what version of the schema with which the instance is compatible.
- Make previous versions of the XML schema available.
- Come up with convention for schema versioning indicating significant changes (e.g. 1.0 to 2.0) or just extensions (e.g. 1.0 to 1.1).
  - If a new schema is only extended, one should not invalidate existing documents.
- If a new schema invalidates existing documents, one should change the target namespace.

Safeguarding life, property and the environment

www.dnv.com



MANAGING RISK