INSPIRE

Infrastructure for Spatial Information in Europe

Alternative approaches for updating the Annex I xml schemas

Title	Alternative approaches for updating the Annex I xml schemas
Creator	Michael Lutz
Date created	2014-12-22
Date modified	2014-12-23
Subject	Two alternative approaches for updating the Annex I xml schemas
Publisher	EC and EEA INSPIRE Team
Type	Text
Description	This document presents two alternative approaches for updating the INSPIRE Annex I xml schemas to reflect the changes have been introduced to the Annex I data models in the amendment of the Implementing Rules on interoperability of spatial data sets and services (Commission Regulation (EU) No 1253/2013) and the corresponding updated versions of the data specification Technical Guidelines.
Contributor	
Format	PDF
Identifier	http://inspire.ec.europa.eu/mig-t/annex-I-schema-update
Language	En
Status	Final

Alternative approaches for updating the Annex I xml schemas

This document presents two alternative approaches for updating the INSPIRE Annex I xml schemas to reflect the changes have been introduced to the Annex I data models in the amendment of the Implementing Rules on interoperability of spatial data sets and services (Commission Regulation (EU) No 1253/2013) and the corresponding updated versions of the data specification Technical Guidelines[1].

Since both alternatives have benefits and drawbacks and may affect existing implementations of Annex I data sets that have already been transformed in accordance with the IRs and of client software ingesting such data sets, it has been agreed in the 13th MIG-T meeting on 27/11/2014 to present have the MIG-T vote on the preferred approach to be taken.

Context

In the amendment of the Implementing Rules and the corresponding updated versions of the data specification Technical Guidelines, a number of changes have been introduced to the Annex I data models (for details, see Annex II of Commission Regulation (EU) No 1253/2013 and the updated data specifications), namely:

1. Some candidate types and placeholders and the references to them have been removed (since they have been replaced by other types in the Annex II+III themes).
2. Some candidate types and placeholders have been re-included in the Physical Waters package in the Hydrography theme
3. Some references have been updated to types in Annex II+III themes
4. Some additional sub-types of HydroObject have been created in the Sea Regions theme
5. An additional data model has been developed for Maritime Units.
6. One minor change of a geometry type (from GM_Surface to GM_MultiSurface) has been made in the candidate type Shore.

These changes should be reflected in the Annex I xml schemas.

---

[1] http://inspire.ec.europa.eu/index.cfm/pageid/2

Furthermore, the encoding rules for the encoding of code lists were updated during the Annex II+III development process. According to the updated encoding rule, code list-valued properties are encoded using `gml:ReferenceType` instead of `gml:CodeType`. Again, the Annex I schemas should be updated to allow data providers to use the same approach for encoding code lists for Annex I, II and III data sets.

In the following sections, the two possible approaches are described in detail.

Approach 1: Backwards-compatible update

The main goal for this approach is that, where possible, **any data sets and, where possible, software that have already been created according to the current schemas (v3.0) should also be valid according to the updated schemas.**

Backwards-compatibility

This means that all changes made to the schemas have to be backwards-compatible. To achieve this, only changes in the update that relax the schemas, e.g.

- removed elements are deprecated and made optional
- new code list encoding is implemented using "union" type between gml:CodeType and gml:ReferenceType

For a complete overview of how the required changes can be implemented in a backwards-compatible way and the open discussion issues for such an implementation, see the proposal on the MIG-T wiki.

Client software needs to be adapted to accept the relaxed schemas, e.g. they need to accept data that do not contain the elements that have been made optional in the updated schemas.

Versioning

The current proposal for versioning of xml schemas (inspired by the rules set up for schema versioning by the OGC) is included in the Annex.

According to these rules, since all changes to the schemas are backwards-compatible in this approach, the new version of the schemas should be considered as a minor revision (in most cases v3.1).

EXAMPLE v3.1 of the Addresses xml schema will be published at http://inspire.ec.europa.eu/schemas/ad/3.1/Addresses.xsd

NOTE All schema documents that include imports to the updated Annex I schemas have to be updated as well. In order not to cause a cascade of new schema versions the updates of import statements in the importing schemas should be treated as bugfix releases (i.e. the importing schema should be replaced with a new schema document with updated import statements).

Namespaces

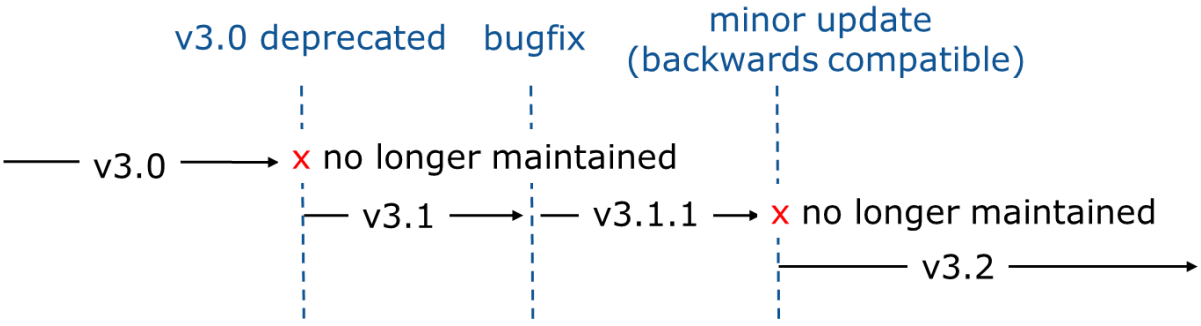
In this approach, the current xml schema namespaces would stay the same in the updated schemas, so that the namespace used in the xml instance documents for existing data would not have to be changed for the data to remain valid also according to the updated schemas.

EXAMPLE The namespace for v3.1 of the Addresses xml schema will be `urn:x-inspire:specification:gmlas:Addresses:3.0`

NOTE This approach for namespaces would means that in some cases, the version number of the schema (and in the schema location path) is not reflected in the namespace. An alternative would be to create a new namespace (e.g. `urn:x-inspire:specification:gmlas:Addresses:3.1`) and to state that all v3.0 data will be valid according to the new schema, if the namespace in the XML instance document is changed to the 3.1 namespace and the rest of the file is left unchanged.

Schema deprecation and maintenance

The following schema summarises the deprecation and maintenance of schema versions for approach 1.



Approach 2: Non-backwards-compatible update

The main goal for this approach is to **use a methodologically clean approach for the update and to clearly communicate where there have been non-backwards-compatible changes** to the data models and schemas.

Backwards-compatibility

In this approach, the changes made to the schemas would not be backwards-compatible. This means that all Annex I xml schemas would be updated to reflect the changes made in Commission Regulation (EU) No. 1253/2013 (e.g. removed elements would be deleted in the schemas) and the new code list encoding.

Existing implementations using the current Annex I schemas would need to be updated over an agreed period of time, during which both the current and the updated schemas would be maintained.

For the agreed maintenance period, also client software would need to be adapted to accept both the current and the updated schemas.

Versioning

According to the versioning rules in the Annex, since all changes to the schemas are non-backwards-compatible in this approach, the new version of the schemas should be considered as a major revision (v4.0).

EXAMPLE v4.0 of the Addresses xml schema will be published at http://inspire.ec.europa.eu/schemas/ad/4.0/Addresses.xsd

NOTE As for approach 1, there are dependencies due to import statements that could cause a cascade of new schema versions. In this approach, dependencies between Annex I schemas can be treated in the updated versions. Dependencies from Annex II+III schemas should be treated as bugfix releases (i.e. the importing schema should be replaced with a new schema document with updated import statements).

Namespaces

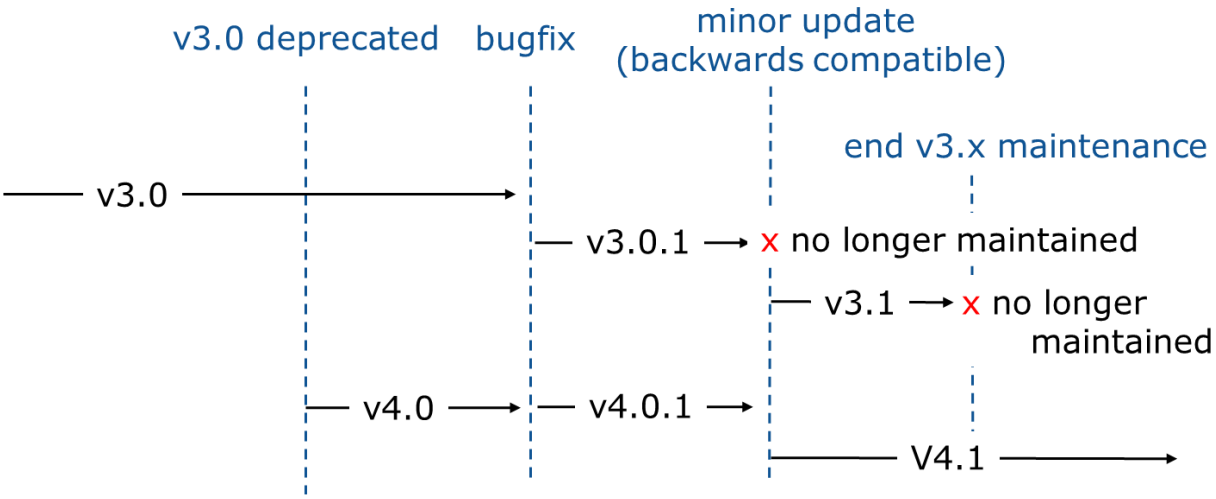
In this approach, the xml schema namespaces would be updated in correspondence with the version number. The current URN namespaces would

be replaced by the http URI namespaces also used for the Annex II+III schemas.

EXAMPLE The namespace for v4.0 of the Addresses xml schema will be `http://inspire.ec.europa.eu/schemas/ad/4.0/`

Schema deprecation and maintenance

The following schema summarises the deprecation and maintenance of schema versions for approach 2.



Comparison

The table below includes a comparison of the two approaches for different aspects.

	Approach 1	Approach 2
Complexity to implement new schemas	complex, e.g. new union code list type, dependencies of deprecated elements	relatively simple, because changes need not be backwards-compatible
Maintenance effort	Only one schema version to be maintained	Two schema versions to be maintained
Clarity of approach	hard to communicate, e.g. discrepancy between	easier to communicate: two different development lines

	version number and namespace	
Interoperability	Only one schema version (but with many options) → higher interoperability? → no incentive to move towards new schemas	Two schema versions (but with no options) → lower interoperability? → incentive to move towards new schemas
Effort for data implementers	low if staying with the old options; medium if moving to new options	low if staying with the old schemas; medium if moving to new schemas
Effort for client implementers	medium to support new options	medium to support two schemas; difficulties for schema-unaware implementations?

Annex: Versioning approach proposed by the DT DS in 2010 in "INSPIRE Maintenance and Implementation - Discussion Paper"

The following system for revisions is proposed (mainly based on the new approach being recently developed within OGC) for INSPIRE Guidance documents.

There are three different types of changes, expressed in different levels of version numbers. Versions are described by a three-level version number X.Y.Z, e.g. 1.2.1.

The three levels of changes are:

- **Corrigendum (Z)**: A pure bugfix release that replaces the previous X.Y.(Z-1) version. The previous version should not be used any longer as it is broken. All data providers and software vendors need to update to the new corrigendum. No semantic changes are allowed.

  As a corrigendum does not change any semantics it is not appropriate to apply the methodology or involve all stakeholders in the process, i.e. there should be a simpler mechanism to develop and adopt the corrigendum.

  Examples for changes:
  - A technical error in an XML Schema was detected that leads to validation errors.
  - An attribute with an error in the multiplicity
  - The definition of a spatial object type is inconsistent with the FCD at the time of adoption of the data specification.
    A major error should immediately lead to a corrigendum. Smaller errors may be collected first and introduced in the next corrigendum.

- **Minor revision (Y)**: A backwards compatible revision, i.e. all data sets conforming to version X.(Y-1).Z still conformant with revision X.Y.0. This should be the typical approach for revisions unless fundamental changes are required.

  This has some implications on the types of changes that may be applied in a minor revision. In particular, outdated items are marked as

deprecated but kept in the specification.

Examples for changes are the introduction of a new spatial object type, a new attribute in a spa-tial object type or a new listed value in an enumeration.

- **Major revision (X)**: A revision introducing significant changes. Where feasible and appropriate, a major revision may also still be backwards compatible. This type of revision should only be targeted if absolutely necessary for the domain, e.g. to introduce a significant number of additional spatial object types to a theme, or if there is consensus to upgrade the Generic Conceptual Model or a data specification in a fundamental, incompatible way.

  Besides explicit change requests from submitting organisation, it might be considered to periodi-cally, e.g. every six years, ask Member States for requirements for changes to the adopted data specifications. A similar process is used within ISO.

The types of changes that are possible / applicable in each of the versions should be described in more detail (once there is a general agreement on the classification system).

It should be discussed if the encoding(s) of a data specification are versioned separately from the data specification itself, to decouple conceptual issues from technical issues on different encoding platforms.