NOAA S-100 Metadata REST API

Metadata XML Schemas Feedback

Aug 7th 2020

IIC Technologies Inc.
Suite 303, 1124 Lonsdale Ave.
North Vancouver, BC
Canada V7M 2H1
P: +1 (604) 904-4402
F: +1 (604) 985-7512
edward.kuwalek@iictechnologies.com
www.iictechnologies.com

Table of Contents

1 Introduction 3
2 S-100 XML Schemas 3
3 Metadata Schema Flexibility vs Encoding Guidance 4
4 Individual Metadata Observations 5
4.1 S – 100 Exchange Set Catalogue Identifier 5
4.2 Metadata File Identifier 5
4.3 Dataset and Data Coverage Scales 6
4.4 Data Coverage and Bounding Box 6
4.5 Support File Grouping 7
4.6 Digital Signatures and Exchange Set Packaging 7
5 Conclusions 8

1 Introduction

The NOAA S-100 Metadata REST API system provides the functionality needed to store all dataset discovery metadata records in the metadata database and use them to construct the exchange set catalogues ready to be consumed by other systems. Additionally, the API subsystem provides the support functionality needed to manage the full life cycle of all records and any axillary information in the database, including the ability to create, read, update and delete individual records as needed. Users are also able to export the resulting XML exchange set catalogues and perform their validation based on the underlying S-100 metadata XML schemas. Currently the system supports four S-100 products, namely S-101, S-102, S-104 and S-111. The system also provides an online metadata editor that can be used to make metadata record modifications using an interactive, web browser based, interface.

The main aim of the project was to provide an optimized metadata database implementation along with a full-featured API to allow other developers and data production systems to interact with the metadata system programmatically. At the same time, having to implement metadata profiles for four S-100 products provided a unique opportunity to have a very close look into the related metadata record content and explore the challenges related to their practical implementation. This report outlines metadata implementation findings and provides recommendations how they could be potentially be resolved.

2 S-100 XML Schemas

Currently there is a relatively well-established set of S-100 XML schemas that can be used to validate the structure of an exchange set for various individual S-100 products. Although these schemas are actively managed and stored in the GitHub repository, there are a few challenges using them in an optimal way, as follows:

- The latest schema versions are currently not operational online and accessible dynamically as most of the typical live XML schemas are. For this reason it is not possible to point to an official online location for any S-100 XML exchange set catalogue schema and validate XML instances against it. Instead the users must download them from the GitHub master repository and configure local schema versions to work with. While this approach is often used for development purposes to speed up record validation, it is a must to have a live, actively managed schemas also accessible directly online for the S-100 ecosystem to function properly.
- The existing S-100 schemas have internal dependencies on other schemas, for example GML or ISO. These additional schemes are live documents themselves and implement changes in line with their own needs and timelines. This creates multiple issues for S-100 community: a) it adds a requirement to monitor and periodically synchronize changes implemented by other communities b) it adds a need for an active change propagation mechanism to be in place, one that does not require technology dependent workarounds such as the currently used custom link redirection so

any system can properly validate against the live schemas without any additional custom settings as is currently the case c) it establishes a strong requirement to have a stable, self-contained set of S-100 XML schemas always active ensuring the services needed by the S-100 community are available online 24/7 regardless of any impending changes being implemented by other communities.

- The current S-100 schemas mostly provide the ability to validate XML document structure and while the content validation using Schematron technology is illustrated for S-101 this functionality it is not fully developed as generally needed in a mature live production environment.
- The current S-100 schemas do not support all products yet. For example, the schemas required for S-104 were not available as the S-104 PS is still under development.

To address the above generic S-100 XML schema challenges IIC recommends establishing a fully operational live schema dynamically accessible online along with a suitable management processes, infrastructure and personnel to keep it up to date at all times. Ideally there should be a formally appointed body responsible for keeping all S-100 schemas operational, perhaps similar to how the IHO Registry is currently managed.

3 Metadata Schema Flexibility vs Encoding Guidance

As currently designed and implemented the S-100 discovery metadata offers a significant degree of flexibility. This can be very beneficial as it provides options for the users to capture discovery metadata in the most appropriate way for their individual needs. At the same time, the S-100 standard and the individual product specifications provide limited guidance or best practices to help users take advantage of the inherent flexibility while keeping things consistent and easily inoperable in practice. There are multiple examples where best encoding practices could be established to not only help the implementors with the metadata capture but also to keep the resulting implementations consistent agency to agency. This includes establishing naming conventions for the key elements, such as the identifiers for exchange set catalogues, datasets, support files as well as prescribing the desirable file folder structure for the exchange set content. It also includes providing a clear guidance for encoding of all formatted strings and external code list values. Although the existing S-100 schema package includes XML examples for some of the products they are not 100% consistent and do not include some of the more advanced concepts such as handling of multiple coverage areas with various interior/exterior polygon boundaries.

IIC recommends reviewing the above-mentioned items and adding the required technical guidance to achieve encoding consistency in practice. Similarly, IIC recommends reviewing and enhancing the existing XML samples to cover all products in a consistent and comprehensive manner while factoring in any new best practices being established.

4 Individual Metadata Observations

Overall, the existing S-100 XML schema package proved to be largely sufficient as the input for the NOAA Metadata API development project and it was certainly possible to develop a full-featured solution covering numerous advanced requirements using it. At the same time, doing so presented numerous challenges that our development team had to overcome during the project. The most notable one was perhaps the difficulty of onboarding new software developers. The learning curve was generally very steep and the information provided in the underlying specifications took long time to absorb. We believe this challenge will be mostly addressed by establishing best practices and more comprehensive XML examples as described in section 3.

Outside of the overall onboarding challenge, the team has run into a few XML authoring intricacies with the current schemas that are worth looking into. These items are described individually in the sections below.

4.1 S – 100 Exchange Set Catalogue Identifier

Currently each exchange set catalogue (and other similar elements such as datasets, support files, catalogues etc.) need to be uniquely identifiable. This requires a proper naming convention, or at least a basic guideline, to be established which is currently not available. It could be something as simple as US_111_20200515_042100_01 indicating country, product, date, time, unique id or something more elaborate and strictly formatted. Either option would go a long way to help data producers and software developers to develop consistent handling of all such elements.

Recommendation: develop and provide naming convention guidelines for exchange set catalogue identifiers and similar items that need to be uniquely identifiable.

4.2 Metadata File Identifier

Each dataset discovery metadata record is required to have a unique metadataFileIdentifier element. This element is an equivalent to the unique metadata file identifier required by the ISO 19115 intended to provide the means to uniquely identify any 19115 metadata files as well as to support parent-child relationship references between them. In contrast to other unique elements, there does not appear to be any effective use for metadataFileIdentifier element. It appears to be simply a carryover concept from ISO 19115 that should likely be removed altogether to simplify things unless there is a valid use case for it.

Conversely, if a valid use case for it can be established there should be a clear guidance for generating such identifiers and ensuring their uniqueness. Additionally, notes for metadataFileIdentifier element in the S100_DatasetDiscoveryMetadata table specify: *For example, for ISO 19115-3 metadata file.* This note should be corrected as it gives an impression that metadataFileIdentifier is related to ISO 19115 dataset metadata and this is not the case.

Recommendation: remove metadataFileIdentifier element unless a proper use case can be established for it.

4.3 Dataset and Data Coverage Scales

Each dataset discovery metadata record can have three scale related elements: optimumDisplayScale maximumDisplayScale minimumDisplayScale at the dataset level. The same elements can also be captured for each data coverage present in a dataset. This opens the door for possible data duplication and encoding inconsistencies. As these elements are closely related to the data content within each data coverage polygon and each data coverage polygon is effectively a sub-dataset it would make sense to always encode these scale related elements at the data coverage polygon level and remove them from the dataset level.

While it is possible to have a more elaborate system with two levels of scale values potentially working together, the potential benefit of doing so is limited and does not seem to out weight the benefits of consistent metadata capturing.

Recommendation: remove optimumDisplayScale maximumDisplayScale minimumDisplayScale at the dataset level and encode them consistently at the data coverage polygon level.

4.4 Data Coverage and Bounding Box

Currently each dataset can have zero or more data coverage elements. In turn, each data coverage element must have one bounding box and one or more bounding polygons indicating the actual data limits within each bounding box.

There appears to be disconnect between the above model and the definition of boundingBox element (*The extent of the dataset limits*). The way things are currently handled each bounding box is related to a specific data coverage and, since multiple data coverages are allowed, effectively each bounding box covers only part of dataset limit when there are more than one data coverages present in a dataset. In this case determining the actual dataset limits would require recomputing them from all bounding boxes.

Historically, the bounding box elements aimed to provide quick access to dataset limits i.e. one all-inclusive bounding box per dataset was computed and readily provided for direct use. The definition used in S-100 aligns with this concept, but the implantation does not.

Recommendation: revise the model and schemas to have one bounding box per dataset indicating the overall dataset limits; all data coverage polygons withing one data coverage must have the same scale values as there is only one set of scale related elements allowed per data coverage.

4.5 Support File Grouping

Currently, there are two main ways to package the support files inside an exchange set. The first one is to collocate them inside individual folders with one dataset and all the related support files grouped inside it. This makes things very straightforward as each folder includes all files required for using that dataset. This is also how the S-57 exchange sets have been typically packaged. The second option is to consolidate all support files into one large grouping, which removes possible data redundancy as any support files used by multiple datasets need to be only provided once. This can make things a bit more complex to manage when an exchange set is created or used in some cases. At the same time, pragmatically removing file duplication can significantly reduce data transfer sizes. The second option is the most optimal approach for machine to machine communications even if it is potentially harder to implement. Since both options are currently supported, questions about which one of them is best to use are common therefore it would be sensible to provide a guidance or maybe even refine the design to single option only.

Recommendation: revise the model and the schemas to support one, preferred option for support file packaging.

4.6 Digital Signatures and Exchange Set Packaging

The use of digital signatures is currently supported and the essential metadata elements are in place to provide the relevant information. This S-100 feature is indented to be a dynamic live system functioning fully online. Similar to the S-100 XML schemas, this infrastructure needs to be developed and made operational. In relation to that a proper process for exchange set packaging should be also established as there are often questions about whether all datasets should be digitally signed first or compressed and then signed individually or as one big archive. At the metadata level all datasets and support files need to be currently signed individually and yet the compression flag is set for the entire exchange set however, the exchange set is not signed. This suggests that the individual exchange set components need to be signed first and, only after that, the entire structure should be compressed. The lack of signature at the exchange set level make signature verification rather complex since the structure needs to be uncompressed first and then individual components need to be verified one by one. In contrast, in most machine to machine transmission data package gets verified as a whole once it is received.

Recommendation: establish digital signature infrastructure online and make it operational; establish the workflow for exchange set packaging and verification and revise metadata to match it if needed.

5 Conclusions

Overall, the current set of S-100 discovery metadata XML schemas is relatively well-established and it is possible to develop a sensible implementation around it. The NOAA Metadata API and Editor developments are good examples of how this can be achieved in practice and how such systems can be used to provide compensative discovery metadata information. There are a few areas that require some additional attention and should be optimized as outlined above. At the metadata content level these changes are mostly smaller refinements that will streamline things rather than dramatically change them. The bigger challenges appear to be the need to have more sensible technical guidance, more comprehensive representative metadata samples for all products, and the need for fully functioning live infrastructure making both the schemas and digital signature operational online.