S100WG02-08.8

Paper for Consideration by the S-100 Working Group

Recommended Changes to Support a Generic S-100 Viewer

Submitted by: ***Executive Summary:*** ***Related Documents:***	SPAWAR Atlantic Summarizes issues encountered while developing a generic S-100 viewer. S-100 Specification
Related Projects:	IHO S-100/S-101 Test Bed Project

Introduction / Background

This paper discusses issues encountered while attempting to develop a generic, product-agnostic S-100 viewer, capable of loading and displaying any S-100 based product without hard-coding of product specific knowledge. A generic viewer should not contain code based on information contained within S-101 or any other product type.

When an application receives an S-100 based dataset, it must perform the following operations prior to installing/using the dataset:

- Verify the signature
- Decryption / decompression
- Read the encoding
- Determine the product type
- Determine the appropriate feature catalogue
- Determine the appropriate portrayal catalogue

In developing a generic S-100 viewer challenges have been encountered with each of these steps. This paper describes the challenges and provides recommendations for overcoming them.

Analysis / Discussion

1. The first challenge for a generic viewer is that the S-57 converter does not generate an exchange set. The exchange set is required for signature verification, decryption, decompression, and determination of the encoding. We recommend the converter be modified to generate a complete *S100_ExchangeSet* for all converted data[1].

2. Assuming an exchange set is present, the method used to store the signature is specified via *S100_DatasetDiscoveryMetaData:digitalSignatureReference*. This is currently described as "*Digital Signature of the file*". We recommend changing the description to "*Algorithm used to compute the digital signature*" for clarity[2].

   digitalSignatureReference is a *CharacterString*, which is not specific enough for the application to determine the algorithm to apply. For instance, an application may treat "SHA-1", "sha-1" and "SHA1" as distinct algorithms. We recommend adding a class specifying the allowed algorithms, similar to how *S100_DataFormat* specifies the encoding for each dataset[3].

3. Once the signature is verified, the application attempts to decrypt/decompress the dataset(s). Again assuming a complete exchange set is available, the application can determine the presence of compression and/or encryption via *S100_ExchangeCatalogue:compressionFlag* and *S100_DatasetDiscoveryMetaData:dataProtection*. If either of these flags are set, the application encounters another challenge: determining the type of compression and/or encryption. The relevant values are stored in *S100_ExchangeCatalogue:algorithmMethod* and *S100_DatasetDiscoveryMetaData:protectionScheme*.

Similar to the issue with *digitalSignatureReference*, these values are stored in an unconstrained *CharacterString*. We recommend adding classes specifying the allowed algorithms, similar to how *S100_DataFormat* specifies the encoding for each dataset[4,5].

4. After decompression/decryption, the application needs to be able to determine the encoding prior to loading the dataset(s). Note that the *S100_DataFormat* value of "Other" cannot be loaded without product specific knowledge.

   We recommend discouraging the use of "Other", or removing it from *S100_DataFormat*[6]. We also recommend that "ISO/IEC 8211 ASCII" be removed, and have provided a companion paper ("Removal of ISO/IEC 8211 ASCII") supporting this recommendation[7].

5. After determining the encoding, the application can attempt to load the encoding. In order to do this, it must map the encoded data to the product-specific application schema. The application schema is based on the S-100 General Feature Model as described in S-100 Part 3. The mapping of S-100 part 10 encoding to the product-specific application schema poses many challenges, and is addressed in a companion paper ("General Data Model in S-100").

6. Assuming the application can successfully load the encoding, it then must determine the product type of the dataset, and associate the correct version of that product types feature catalogue with the dataset. The product type can be determined from either the metadata contained within the encoding, or from the dataset discovery metadata.

   The HDF-5 encoding does not specify any required metadata. We recommend that the HDF-5 encoding specify metadata matching the ISO-8211 and GML encodings[8].

   The product type metadata is stored in the ISO-8211 and GML encoding as unconstrained text. The use of unconstrained text means that "INT.IHO.S-101.1.0.0", "INT.IHO.S-101.1", "S-101", and "101" are all valid values which could be used to specify the S-101 product type.

   The product type is also available from the exchange set metadata in *S100_DatasetDiscoveryMetaData:S100_ProductSpecification*. *S100_ProductSpecification:name* is unconstrained text, which means that ambiguous values can be used for the product type.

   Since applications need to unambiguously determine the product type, we recommend adding an integer attribute *S100_ProductSpecification:number*[9]. This change would also support determination of the product type when installing feature and portrayal catalogues, since *S100_CatalogueMetadata:productSpecification* is of type *S100_ProductSpecification*.

7. Assuming the product type is determined, the application needs to associate the dataset(s) with the appropriate feature catalogue. The product type can be used to look up the appropriate installed feature catalogues, however there is no metadata to indicate which of the available feature catalogues should be used.

   We recommend metadata be added to the encodings and/or the *S100_DatasetDiscoveryMetaData* to specify the appropriate feature catalogue[10]. The signature in *S100_CatalogueMetadata:digitalSignature* could be used if its multiplicity were changed to "1", and it was guaranteed to be unique. We further recommend that the S-57 converter be modified to store the signature / identifier of the feature catalogue used to convert each dataset[11].

8. Assuming the feature catalogue is determined, the final step prior to portrayal is to determine the appropriate portrayal catalogue. This should be the latest portrayal catalogue which was generated from the feature catalogue determined in the previous step. However, there is no metadata available to make this association.

   We recommend metadata be added to the portrayal catalogue to unambiguously specify the associated feature catalogue[12]. The schema of this metadata should match the *S100_CatalogueMetadata* used to distribute the feature catalogue.

9. A final recommendation is that feature and portrayal catalogues be required to be delivered as part of an exchange set so that the metadata is available for development and testing[13]. Currently the feature catalogue and portrayal catalogue are provided stand-alone, without the required metadata.

10. In order to work around the challenges noted in this paper, the current version of the testbed viewer has the following limitations:

    - The digital signature is not currently validated.
    - Encryption and compression are not currently supported.
    - The GML and HDF-5 encodings are not currently supported.
    - When loading a dataset, the user must manually select the appropriate feature catalogue and portrayal catalogue.

Recommendations

1. Modify the S-57 converter to generate a complete *S100_ExchangeSet* for all converted data.
2. Modify S-100 3.0 Part 4a *S100_DatasetDiscoveryMetaData:digitalSignatureReference* description to read "Algorithm used to compute the digital signature".
3. Modify S-100 3.0 Part 4a *S100_DatasetDiscoveryMetaData:digitalSignatureReference* to have a value from a digitalSignature class, similar to S100_DataFormat.
4. Modify S-100 3.0 Part 4a *S100_ExchangeCatalogue:algorithmMethod* to have a value from a compressionAlgorithm class, similar to *S100_DataFormat*.
5. Modify S-100 3.0 Part 4a S100_DatasetDiscoveryMetaData:protectionScheme to have a value from an encryptionAlgorithm class, or specify the allowed algorithm values in the "Remarks" column.
6. Discourage use of, or remove "Other" from S100_DataFormat.
7. Remove "ISO/IEC 8211 ASCII" from S100_DataFormat.
8. Add metadata to the HDF-5 encoding (S-100 Part 10c) matching the Part 10a and Part 10b.
9. Add integer attribute S100_ProductSpecification:number to unambiguously specify the product type.
10. Add metadata to the encodings and/or the S100_DatasetDiscoveryMetaData to specify the appropriate feature catalogue.
11. Modify the S-57 converter so that it stores the metadata necessary to identify the feature catalogue associated with each converted dataset into the encoding / S100_DatasetDiscoveryMetaData.
12. Modify the portrayal catalogue schema to unambiguously specify the feature catalogue.
13. Provide future feature and portrayal catalogues as part of an exchange set.

Action Required of S-100 WG

The S-100 working group is invited to:

a. note the paper
b. discuss the recommendations
c. take action on each recommendation described herein