Generic Open Packaging Convention (OPC) Binding Profile

1. Introduction

The term labelling is the process of determining the appropriate metadata for a given data object, creating the metadata label and binding the metadata label to the data object. A binding is a relationship between a data object and a metadata label. A binding is realized by applying a binding mechanism. If a metadata label must be bound to a data object, both the metadata label and the data object are input to the binding mechanism. The output of the binding mechanism is the binding of a data object and metadata label (see Figure 1) which says that the data object and the metadata label belong together. The binding can be recorded as a structured data object, known as a Binding Data Object (BDO).

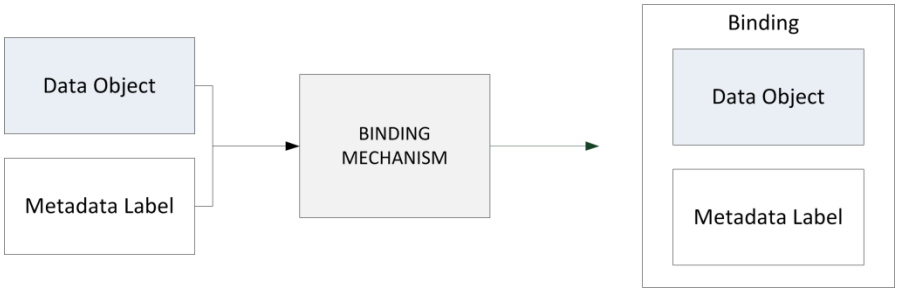

Figure 1 Creation of a binding

STANAG 4778 (Reference [3]) standardizes the binding of a data object and metadata label by specifying a common binding mechanism and a syntax for representing the BDO. However, to support information management and information sharing requirements it is necessary to further profile the application of STANAG 4778 to facilitate locating a BDO and embedding a BDO in data objects.

2. OPC Introduction

This profile defines a generic packaging mechanism, based upon the Open Packaging Container (OPC) defined in ISO/IEC 29500-2:2008 (Reference [1]), to associate any arbitrary file that do not use the Office Open XML (OOXML) format (Reference [1]) or have no specific profile for supporting the *BindingInformation* with their own file format.

In OPC terminology, the term *package* corresponds to a ZIP archive and the term *part* corresponds to a file stored within the ZIP. Every part in a package has a unique URI-compliant part name along with a specified content-type expressed in the form of a MIME media type. A part's content-type explicitly defines the type of data stored in the part, and reduces duplication and ambiguity issues inherent with file extensions.

3. Identification

The profile for generic OPC is uniquely identified by the Canonical Identifier shown in Table 1.

Table 1: Profile Identifiers

Type	Identifier
Canonical Identifier	urn:nato:stanag:4778:profile:gopc
Version Identifier	urn:nato:stanag:4778:profile:gopc:1:0

It is recognized that this profile may evolve during its review cycle. For example, a review might identify:

- changes to the base OPC standard
- improvements to the existing profiles based upon operational feedback

Therefore this version of the profile is uniquely identified by the Version Identifier shown in Table 1.

Subsequent versions of this profile will maintain the same Canonical Identifier, but define a new Version Identifier.

4. Standards

[1] ISO/IEC 29500-2 “Office Open XML File Formats - Part 2: Open Packaging Conventions”, at http://standards.iso.org/ittf/PubliclyAvailableStandards/c061796_ISO_IEC_29500-2_2012.zip, August 2012

[2] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium

[3] STANAG 4778, Metadata Binding Mechanism, Brussels, Belgium

5. Notational Conventions

- The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [IETF RFC 2119, 1997].
- Words in *italics* indicate terms derived from Reference [3].

6. File Package

One of the common ways to package a number of files together is to use the archive file format. An archive file may contain a number of different files and an associated folder structure.

This profile adopts the Open Packaging Conventions (OPC) as defined as Part 2 of the Office Open XML specification (Reference [1]).

By adopting OPC this profile provides a structured and consistent mechanism for associating *BindingInformation* with a data object within an archive file.

This profile uses the same customXml files and relationships within the archive file as those defined in the OOXML Binding Profile, as shown in **Figure 2**.

Specifically:

- A top-level relationship within the package SHALL be defined which identifies the file with which the *BindingInformation* will be associated.
- The file SHALL be held in a folder called “files”
- The *BindingInformation* SHALL be held within a file called “customXml”.
- The *DataReference* URI attribute SHALL be specified as the full path to the file.
- As such, a relationship is defined between the file and the *BindingInformation*.

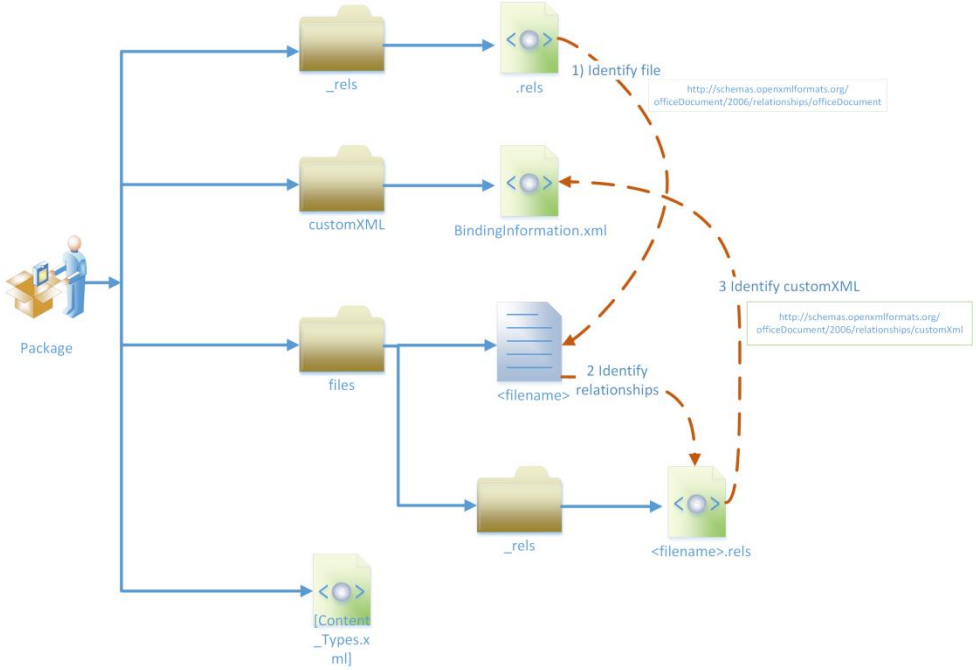

Figure 2: OPC Structure for packaging BindingInformation with an arbitrary file

This approach allows multiple files, of different types, to be held within the same package and be bound to distinct metadata. **Figure 3** shows an example customXML file for a package containing the file “image1.jpeg”. This example uses Confidentiality Metadata Labels (Reference [2]) as example metadata.

```
<mb:BindingInformation
  xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
  xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
  <mb:MetadataBindingContainer>
   <mb:MetadataBinding>
    <mb:Metadata>
     <slab:originatorConfidentialityLabel
      xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
      <slab:ConfidentialityInformation>
       <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
       <slab:Classification>UNCLASSIFIED</slab:Classification>
      </slab:ConfidentialityInformation>
      <slab:CreationDateTime>
       2016-11-10T12:30:00Z
      </slab:CreationDateTime>
     </slab:originatorConfidentialityLabel>
    </mb:Metadata>
    <mb:DataReference URI="files/image1.jpeg" xmime:contentType="image/jpeg" />
   </mb:MetadataBinding>
  </mb:MetadataBindingContainer>
 </mb:BindingInformation>
```

Figure 3: Example Packaged CustomXML file