## Embedding foreign OPC parts

Markup consumers are able (but not required) to preserve OPC parts with unrecognized relationship types during save operations. Such foreign parts are best suited to data (either binary or XML) that the creator desires to be preserved during round-trip operations.

One good use of a foreign part would be for an embedded video file attached to a WordprocessingML document. Three steps are needed. Firstly, the file would be added to the OPC package:



{Note from CRA: the new diagram needs to look like the following hierarchy.}

\_rels

customXml

docpProps

word

[Content\_Types].xml

movies

 extra.mov

Secondly, a relationship would be added to the \\_rels\.rels part:

<Relationships
 xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
 <Relationship Id="rId5" Type="<http://example.org/myexample>"
 Target="movies/extra.mov"/>
</Relationships>

Thirdly, an element would be added to the [Content\_Types].xml stream, unless an appropriate element was already present.

<Default Extension="mov" ContentType="video/quicktime"/>

The content will likely be preserved on round-trip through non-understanding applications, and because there is no requirement to serialise it into XML this extension mechanism is well-suited to binary data such as video or images.

A second good use of a foreign part would be to embed richer metadata than supported in the Core Properties part of an OPC package (IS 29500-2, §10) or the Custom Properties part of an OOXML document (IS 29500-1, §15.2.12.2). For example, an ONIX for Books record could be embedded in the package for a WordProcessingML document using the same approach as outlined above. Although the ONIX record would be in XML, applications would not be required to parse or understand the record.

For an ONIX record in a part with name example\_ONIX.xml, an appropriate location would be in a folder called "meta". The relationship could be:

 <Relationship Id="rId56" Type="http://ns.editeur.org/onix/3.0/reference"
 Target="meta/example\_ONIX.xml"/>

If not already present in the [Content\_Types].xml stream, the following addition would be appropriate:

<Default Extension="xml" ContentType="application/xml"/>