ISO/IEC JTC 1/SC 34/WG 6

OpenDocument Format

Convenorship: BSI (United Kingdom)

Document type: National Body Contribution

Title: Comparison of extensibility features in ODF and OOXML

Status: This paper has been prepared by Francis Cave as an expert contribution for discussion by the Working Group.

Date of document: 2017-05-30

Expected action: INFO

Email of convenor: francis@franciscave.com

Committee URL: http://isotc.iso.org/livelink/livelink/open/jtc1sc34wg6

ISO/IEC JTC 1/SC 34/WG 6 N 132

Comparison of extensibility features in ISO/IEC 26300 and ISO/IEC 29500

Francis Cave, WG 6 Convenor

Introduction

Both ISO/IEC 26300 (ODF) and ISO/IEC 29500 (OOXML) include mechanisms to support the extension of the standard document format to meet new and niche market requirements. The aim of this discussion paper is to compare the extensibility features provided by these formats and suggest ways in which each format might be enhanced by learning from design choices made in the other.

What is meant by “extensibility feature”?

In the context of document formats the term “extensibility feature” means any feature provided as part of the standard format that enables document instances to contain data whose syntax and semantics are not defined as part of the standard format. Such data is often referred to as being “foreign” to the standard format. Given that the standard formats of both ODF and OOXML are package formats, and the main parts of a package are XML documents conforming to standard XML schemas in specified XML namespaces, foreign data implies XML data in foreign namespaces, i.e. in namespaces not defined by the standards.

Generally excluded from this definition are features that allow non-XML data, such as images or executable scripts in non-XML-based syntaxes, to be included in a document instance. The term “extensibility” implies the inclusion of features that extend the existing XML-based format, not the inclusion of self-contained, foreign objects that don’t extend existing features.

Extensibility features in ISO/IEC 26300

OpenDocument Extended Document

The main extensibility feature in ISO/IEC 26300-1:2015 (ODF v1.2, Part 1) is provided by the concept of an OpenDocument Extended Document, as defined in Section 2.2.2. An OpenDocument Extended Document can contain foreign elements and attributes in accordance with Section 3.17, which specifies how a conforming ODF consumer application should process foreign elements and attributes.

A legacy feature of earlier versions of ODF, the attribute `office:process-content` is deprecated in ODF v1.2, but its original use was to specify whether or not an ODF consumer is expected to process the foreign element on which this attribute is specified.

Section 3.17 draws a distinction between foreign elements that occur as descendants of elements `<text:h>` and `<text:p>` (both elements that usually contain text) and those that do not. In the case of foreign elements that are descendants of `<text:h>` and `<text:p>`, a consumer is expected to **unwrap** the foreign element, or **delete** it if empty. In the case of elements that are not descendants of `<text:h>` and `<text:p>`, a consumer is expected to ignore the foreign element.

Attribute values that are names in foreign namespaces

Although not an extensibility feature as such – it is not a mechanism for extending the format – ODF does define a number of attributes whose values are interpreted as names in foreign namespaces. The following attributes, specified in Part 1, have this characteristic: 19.29 `config:name;` 19.429 `script:event-name;` 19.430 `script:language;` 19.468 `style:condition;` 19.581 `table:algorithm;` 19.596 `table:condition;` 19.635 `table:expression;` 19.642 `table:formula;` 19.777 `text:condition;` 19.805 `text:formula`.

OpenDocument Extended Package

The concept of an OpenDocument Extended Document in ODF v1.2 Part 1 is complemented to some extent by the concept of an OpenDocument Extended Package in Part 3. An OpenDocument Extended Package, as specified in Part 3 Section 2.2.2, conforms to all the requirements of an OpenDocument Package, as specified in Section 2.2.1, except that the package may contain files with relative path beginning “META-INF/” in addition to a manifest file and signature files. It is clearly the intention that such files contain what can broadly be termed metadata, not document content, although such metadata could presumably have the purpose of supporting the interpretation of foreign elements in an OpenDocument Extended Document (Part 1 Section 2.2.2).

Alternative content representations in frames

The ODF concept of a frame is specified in Part 1 Section 10.4. A frame, represented by the element `<draw:frame>`, typically contains an object such as a text box or image, and the format allows for multiple representations of the same object to be included, so that the consumer implementation can choose whichever representation it is best able to handle.

Extensibility features in ISO/IEC 29500

ISO/IEC 29500-3:2015 – Markup Compatibility and Extensibility

OOOXML Part 3 is devoted to the specification of various extensibility features under the general heading Markup Compatibility and Extensibility (MCE). Part 3 defines a set of XML elements and attributes in the MCE namespace that allow documents to contain elements and attributes in foreign namespaces and specify how document consumers are to handle documents containing elements and attributes in foreign namespaces. A general overview of MCE is provided in Clause 6.

The MCE attribute `Ignorable` specifies which foreign namespaces can be ignored by a consumer that does not understand them. A namespace is ignorable if an element or attribute in that

namespace is within the scope of an `Ignorable` attribute that specifies this namespace to be ignorable.

The MCE attribute `ProcessContent` specifies which elements in ignorable namespaces must be processed by being "unwrapped".

The MCE attribute `MustUnderstand` specifies which foreign namespaces must be understood by a consumer in order to consume the document, unless it falls within the scope of an `Ignorable` attribute or is in an alternate content block that has been ignored (see next paragraph).

The MCE element `<AlternateContent>` specifies alternative blocks of content from which a consumer may choose one that it understands and ignore the remainder. This MCE element contains one or more child `<Choice>` elements and, optionally, a `<Fallback>` element. Typically, a `<Fallback>` element would not contain any extensibility features.

MCE also defines the concept of an application-defined extension element, an element that is specified by the standard file format but whose content is not specified by the standard format. The key difference between an application-defined extension element and an element in a foreign namespace is that the application-defined extension element *and all its contents* are always passed to the consumer application, whereas an MCE pre-processor will remove ignorable elements and attributes in foreign namespaces (unless they are contained in an application-defined extension element). Specific application-defined extension elements are defined in Part 1 of OOXML, and include the element `<extLst>` in the SpreadsheetML namespace.

Guidance on the use of MCE in OOXML is to be found in ISO/IEC TR 30114-1:2016. It should be noted that MCE is also used in file formats other than OOXML, such as XPS.

Foreign OPC Parts in OOXML documents

OOXML documents are generally packaged using the Zip-based packaging format specified by ISO/IEC 29500-2 Open Packaging Conventions (OPC). In OPC a "part" is the equivalent of a "file contained in the Zip file", as defined by ISO/IEC 26300 Part 3 Section 2.2.1.

OPC does not restrict the names of foreign parts, but does require that the Content Type of each foreign part must always be specified, and that a relationship between its source (another part or the entire OPC package) and the target foreign part must always be specified.

ISO/IEC TR 30114-1 provides guidance on how "foreign parts" can be included in an OOXML document, as specified by OPC.

Comparison of extensibility features in ODF and OOXML – some observations

In many ways the extensibility features in ODF and OOXML are quite similar. Comparable concepts in both include:

- The concept of allowing documents to contain foreign content that cannot be interpreted by a conforming consumer implementation.

- The concept of either “ignoring” or “unwrapping” foreign elements that cannot be understood by the consumer implementation.
- The concept of allowing document to contain files/parts whose format is not specified as part of the standard.
- The concept that a consumer implementation may choose to preserve foreign elements that are understood, but this is an implementation choice and preservation is not a requirement.
- The concept that the same content object may be held within a document in different alternative representations, allowing a consumer implementation to choose any one of them to process.
- The concept of an application-defined extension element, whose content a consumer may choose to round trip. In ODF `<text:p>` and `<text:h>` both play this role as a secondary role (their primary role being containers for text), while in OOXML `<extLst>` plays this role exclusively.

The principle differences between the extensibility mechanisms provided by ODF and OOXML can be summarised as follows:

- ODF does not specify any circumstances in which a conforming consumer implementation must understand foreign elements in a document, whereas MCE uses the `MustUnderstand` attribute to specify which foreign namespaces must be understood and the context within which they must be understood.
- ODF is quite specific about the context in which foreign elements must be unwrapped, whereas MCE uses the `ProcessContent` attribute to specify the context in which elements in a foreign namespace must be unwrapped when not understood. This is, presumably, similar to the original intention of the now-deprecated `office:process-content` attribute in ODF.
- In ODF there is only one context in which alternative representations of the same content may be included in a document, which must be in the element `<draw:frame>`. In MCE the element `<AlternateContent>` allows alternative forms of the same content to be included in a wide range of contexts. It should also be pointed out that ODF specifies (in Section 10.4.2) the alternative representations that are allowed as child elements of `<draw:frame>`, whereas the MCE element `<AlternateContent>` allows alternative forms of the same content in any foreign namespace.
- In ODF the elements `<text:p>` and `<text:h>` act in a similar way to application-defined extension elements in OOXML, but the main difference is that the contents of an application-defined extension element (`<extLst>`) will always be passed to a consumer implementation of OOXML, whereas an ODF consumer implementation can unwrap foreign elements within `<text:p>` and `<text:h>`.