

CAP: Editorial Principles for OpenDocument Format 1.2

Patrick Durusau, Editor

1 Introduction:

OpenDocument 1.2 differs in content from OpenDocument 1.0/1.1 only from correction of errors and the addition of new material. However, in appearance it is substantially different from those versions and it may be helpful if I enunciate the principles that have led to that difference. The principles that I list below are often found in ISO standards but are by no means limited to being used in connection with ISO standards. Good standards, without regard to their issuing organization, will follow these rules or ones very similar to them.

CAP: stands for **C**onsistency, **A**ccuracy, **P**recision. While that sounds trivial and is easy to say, producing a standard that follows those principles is a non-trivial exercise.

In my role as editor of OpenDocument Format 1.2, I have carefully reviewed the entire text multiple times. My comments are not meant as a criticism of anyone who has contributed to the drafts up to this point. I contributed to those drafts as well and I am sure some of my contributions will be corrected as well.

In addition to enunciating the principles by which I am editing the next ODF draft, I have included examples of the application of those principles to the current text. I am hopeful that such examples will elicit discovery of issues I have yet to uncover.

2 Consistency:

Since OpenDocument Format 1.2 is destined to return to ISO, it is important that it conform to the requirements for ISO standards. Only an initial submission is permitted to vary from the formatting requirements for ISO standards. But the consistency I address here is not the issue of formatting under the ISO rules. Rather, the concern is with consistency of presentation of information in OpenDocument Format 1.2.

2.1 Attributes

If you look at the draft dated 14 April 2008, you will see that I have removed all the attribute listings from the sections that define elements. The reason for that removal was that before that draft, we would list attributes as a bulleted list, sometimes with multiple items per bullet, sometimes not, or in some sort order that was not readily apparent or consistent, or simply fail to list them at all, with variations in between.

Even a cursory review in ISO would result in such a text being returned to the editor for the imposition of consistent presentation of attribute information for elements. Not to mention that inconsistent presentation of information simply increases the difficulty of using the standard.

With regard to attributes, I have moved all the attributes to the end of the text, in alphabetical order. It should be noted in the 14 April 2008 draft that you can still see multiple definitions of the same attribute. That is also a consistency issue.

The objection will likely be made that such extensive changes will require a vast increase in proofing of the final text. No doubt the text will need to be proofed but if done properly, **the resulting text will require less time and be more accurate than prior versions.**

Note that I have changed the section headings to element and attribute names. If we following the principle that every element and attribute should be defined once and only once, then the job of proofing the new draft becomes a matter of rote inspection.

For example, if the attributes for the <text:p> element are automatically generated from the schema and then a lookup is done on the section headers, the resulting attribute listing as a cross-reference should have **one and only one target** for each attribute. If an attribute is not presented as a cross-reference, then its proper target is missing and if there is more than one target, then there is more than one definition for that attribute. The same principle applies to child element listings.

Such auto-generated cross-references should be sorted by namespace order and appear in the same order every time they are listed.

Granted, in a mixed prose text like the draft prior to this one, such a revision would be very difficult to proof as there is no systematic way to proof the necessary linking. As a matter of fact, current cross-references are not consistently entered in the text, such that following the link is necessary to ascertain its target.

2.2 Attribute Definitions

With the replication of attribute definitions, we have fallen into inconsistent definition of attributes. For example, we define xlink:href using varying prose to mean the same thing. And since xlink:href is defined in another standard that we cite normatively, we should simply cite that definition once and only once. For every element with that attribute, if we mean to normatively constrain the result of following the value of an xlink:href, then we should say so. It is completely unnecessary for us to re-define xlink:href to achieve that goal.

Another problematic area with attribute values is the use of draw:name to identify an element. For the most part we define that as being unique, but not always. Moreover, we also use text:id as a means to identify elements. And we are about to add xml:id with the new metadata section to identify elements as well. I don't think we did that intentionally but having the attributes scattered throughout the text not only added to the complexity of the display of the text but lead to problems such as these.

3 Accuracy

In many cases, we use attributes from other standards. Unfortunately, we then go on to say what the other standards defines for that attribute, plus give a reference to the standard. If we are going to follow the other standard, then a normative reference is sufficient. It is not necessary and is in fact dangerous to restate what has been defined in another standard. The SMIL attributes are good examples of our practice in this regard.

I address the issue of schema fragments later in this document but this is a good place to note that the

issue of including schema fragments is a question of accuracy.

In the latest version before the one of 14 April, you can find the following fragment in Chapter 4:

```
<define name="text-p">
  <element name="text:p">
    <ref name="paragraph-attrs"/>
    <zeroOrMore>
      <ref name="paragraph-content"/>
    </zeroOrMore>
  </element>
</define>
```

The question you have to ask is how "accurate" is that fragment? There is no doubt that it is "accurate" as far as it goes, but is it accurate in the sense of being useful to anyone who is simply reading the text? What does "paragraph-attrs" mean?

Or, should we have:

```
<define name="text-p">
  <element name="text:p">
    <optional>
      <attribute name="text:class-names">
        <ref name="styleNameRefs"/>
      </attribute>
    </optional>
    <optional>
      <attribute name="text:cond-style-name">
        <ref name="styleNameRef"/>
      </attribute>
    </optional>
    <optional>
      <attribute name="text:style-name">
        <ref name="styleNameRef"/>
      </attribute>
    </optional>
    <optional>
      <attribute name="text:id">
        <ref name="string"/>
      </attribute>
    </optional>
    <zeroOrMore>
      <ref name="paragraph-content"/>
    </zeroOrMore>
  </element>
</define>
```

I did not expand "paragraph-content" as it would be too large to be meaningful.

As you will read later, I favor having a "view" of the standard that includes an expanded notion of schema fragment, for proofing and well as other purposes but not as part of the normative text.

4 Precision

I won't go on at length about this issue but suffice it to say that when we are defining elements or attributes for OpenDocument we should make our definitions clear and precise. That means that every

time we define an element or attribute, it should use the same verbal format. That is to say it should say: "The *attributeName* attribute defines...." every time an attribute is defined. The reader should not have to hunt through prose to discover what element or attribute is being defined.

While that has overtones of consistency, note that so defining an element or attribute does not involve comparison to another standard. We either define our elements or attributes or make a normative reference to a definition elsewhere. There are no OpenDocument elements that are "like" elements in other standards, at least not in the normative text. Quite possible for that to occur in supplemental or tutorial material but not in the normative text.

The rule is either we define an element, attribute, or behavior or we don't. If we don't, then as a standard we should leave it strictly alone. To observe that some element is "like XSL-FO" implies semantics that we do not define.

5 Why No Schema Fragments?

I assume if you have read this far into an editorial principles document you deserve to know why the schema fragments are missing from the 14 April draft. Well, the truth is that the schema fragments were preventing me from re-editing the text. It wasn't possible to ignore them and to recast them into another presentation of the same content models would have taken a very long time. Actually it took several months of struggling with the text (I have a version of chapter 8 that is almost all red) to realize that with the schema fragments in the text, they were driving the structure of the prose.

That is a bad thing because it means the text becomes overly complicated and the result is duplicate attribute definitions, and other problems.

That is not to say that we can't "re-insert" the schema fragments (assuming the schema is properly modularized and generated for each display) but that will be only in a "view" of the standard.

Here is how I see that shaping up:

1. OpenDocument Format 1.2 (the standard, no schema fragments, no examples, this is the version we approve in OASIS and submit to ISO)
2. OpenDocument Format 1.2 + schema fragments, quite possibly the edition that we use to proof version 1.2.
3. OpenDocument Format 1.2 + schema fragments + examples, which can have examples for every section, etc., as many examples as anyone wishes to insert.

All of that is possible with the new metadata mechanisms which are inherent in OpenDocument Format 1.2.

I am aware that there is a concern that there may not be any implementations that actually can dynamically display the additional information for views 2 and 3, but I was not asked to write a standard against a particular implementation but the best version of OpenDocument Format possible. All current implementations can display the results of the processing of the metadata and since we don't require any particular processing of metadata capability, that should not affect the issue of "implementations" of OpenDocument Format 1.2.

We have been promising people the advantages of markup documents for over 20 years now. I think it is time for us to deliver on that promise, at least when we are producing markup standards.

6 Timing, Past and Future

The obvious questions at this point are why did it take so long for me to report these issues and how long will they take to resolve?

The answer to why it took so long to report these issues is that it took months to realize the need to re-conceptualize the text and then to be able to produce a draft that evidences that re-conceptualization took a bit of effort. Realize that for months I worked with the tradition schema framework, which resulted in chapters with so many edits that the interface would almost stop responding. I finally had to simply abandon change tracking altogether.

Once I realized the impact that the schema fragments was having on my view of the text, it was necessary to produce a draft that represented what I had realized about the text. The problem was that I could only see that because I had read the text probably five or six times in as many months. The problem I faced was how to communicate that to others. The 14 April draft is a first attempt to convey a different view of the text.

How long will it take seems to be the next most popular question. Assuming that the procedures I have briefly outlined for generation of reference content are put into place, I actually don't think completion of OpenDocument Format 1.2 will take us beyond the end of May or possibly June of 2008. One of the reasons for the suggestions I have made is to make it easier and not harder to proof the text. It is hard to proof the prior versions because the section names bear no consistent relationship to their content and attributes are defined at various places and in various ways. Once we put a stop to that, proofing becomes a good deal easier.

This document itself is a rough draft which is subject to revision and refinement.

Comments and suggestions are always welcome.

Patrick Durusau