



LibreOffice
The Document Foundation



ROME
CONFERENCE

Taming the ODF Dragon: Complexity Reduction by improved Tooling

Svante.Schubert@gmail.com

ROME | 11 October 2017

Agenda

- **Why?**
- **What?**
- **How?**

Why?

Why? Leverage ODF standard!

- ODF is..
 - Blueprint for all ODF applications
 - Telling what are valid ODF documents
 - An OASIS and ISO standard (government argument)
 - created against vendor lock-in (interoperability by format)

What?

Improve ODF for developer!

- Ease ODF application development
 - Save time by avoid repeated work!
 - Reducing complexity!
- Share as much as possible among ODF app developers..
 - Feature tests and test documents at standard level (similar JPEG2000 ISO standard)
 - Make specification machine readable..
 - Generate (as much as possible)..

Illustration							
model	Krups Controlline KH442	Tefal TT 5500	Severin AT 2509	Severin AT 2514	WMF Stelio Toaster	Severin AT 2287	Grundig TA 6330
comparison result	 Vergleich.org Review 1.2 very good 09/2017	Vergleich.org Rating 1.3 very good 09/2017	Vergleich.org Review 1.5 good 09/2017	Vergleich.org Review 1.6 good 09/2017	Vergleich.org Review 1.7 good 09/2017	Vergleich.org Rating 1.8 Good 09/2017	Vergleich.org Excellent 1.9 good 09/2017
Customer rating at Amazon	★★★★☆ 4 Reviews	★★★★☆ 364 reviews	★★★★☆ 2 business days	★★★★☆ 963 ratings	★★★★☆ 158 reviews	★★★★☆ 511 reviews	★★★★☆ 44 Reviews
Sheets per pass	2	2	4	2	2	2	2
browning levels	6	8th	6	6	7	7	6
Power (watts)	720 watts	1,200 watts	1,400 watts	850 watts	900 watts	700 watts	850 watts
Dimensions (LxWxH)	33.2 x 24.4 x 20.2 cm	40 x 23.4 x 22.8 cm	12.6 x 37.1 x 18.2 cm	27.1 x 15.5 x 18.3 cm	32.5 x 20 x 27.5 cm	32 x 18 x 18.5 cm	34 x 21.5 x 24.5 cm
mass	0.8 kg	3.1 kg	1.0 kg	1.5 kg	1.9 kg	1.2 kg	2.0 kg
Heat insulation	✓	✓	✓	✓	✓	✓	✓
rolls rust	✓	✓	✓	✓	✓	✓	✓
Bread disc centering Bread disc	✓	✓	✓	✓	✓	✓	✗
Toast lifting function	✓	✓	✓	✓	✓	✓	✓
defrost	✓	✓	✓	✓	✓	✗	✓
manual stuffing function	✓	✓	✓	✓	✓	✓	✓
	⊕ Testieger Foundation Warentest 04/2016	⊕ including egg cooker and egg pans	⊕ Housing is not hot ⊕ also suitable for bread	⊕ very quiet ⊕ integrated roll holder	⊕ tans evenly ⊕ illuminated key labels	⊕ integrated roll holder ⊕ good workmanship	⊕ stable crumb tray ⊕ good workmanship

Can I use @page

[?](#) [Settings](#)

1 result found

CSS Paged Media (@page) - WD

Global

61.43% + 11.61% = 73.03%

CSS at-rule (@page) to define page-specific rules when printing web pages, such as margin per page and page dimensions.

[Current aligned](#) [Usage relative](#) [Date relative](#) [Show all](#)

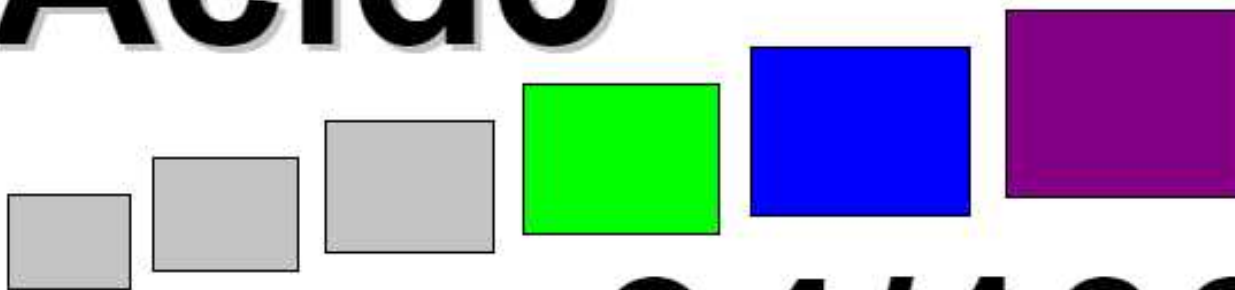
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49			10.2			
		55	60	10.1		10.3		4.4	
11	15	56	61	11	47	11	all	56	61
	16	57	62	TP	48				
		58	63		49				
		59	64						

[Notes](#) [Known issues \(1\)](#) [Resources \(6\)](#) [Feedback](#)

Currently no browsers appear to support the marks & bleed properties from the latest version of the specification.

Does not support the size property

Acid3



94/100

To pass the test, a browser must use its default settings, the animation has to be smooth, the score has to end on 100/100, and the final page has to look exactly, pixel for pixel, like [this reference rendering](#).

We shall aim for..

- Conformance comparison of ODF Applications (like Toasters)
- Automated ODF conformance tests (like ACID3 for HTML/CSS)
 - only work as *HTML has a browser Run-Time-Model (DOM)*
- No **ODF XML Run-Time-Model**, but common **Semantic Model**
- Need for **ODF Feature** description
 - There is more than a monolithic ODF block
- Need for **ODF Feature change** description
 - User already expect to change ODF documents
 - Do no longer 'send the repository', send an ODF DIFF
- Some feature example: Page layout... (see PageLayout.odt)

An ODF feature example..

- Page Layout test document (PageLayout.odt)
 - Two pages (one portrait, one landscape layout)
 - Three paragraphs (two on first page, third on second)
 - Different header/footer on each page

```
<?xml version="1.0" encoding="UTF-8"?>
<office:document-content xmlns:office="urn:oasis:names:tc:opendocument:xmlns:office:1.0" xmlns:style="urn:oasis:names:tc:opendocument:xmlns:style:1.0" xmlns:text="urn:oasis:names:tc:opendocument:xmlns:text:1.0" xmlns:font-face="urn:oasis:names:tc:opendocument:xmlns:font-face:1.0" xmlns:automatic="urn:oasis:names:tc:opendocument:xmlns:automatic:1.0" xmlns:sequence="urn:oasis:names:tc:opendocument:xmlns:sequence:1.0">
  <office:scripts/>
  <office:font-face-decls>
    <style:font-face style:name="Lucida Sans1" svg:font-family="&apos;Lucida Sans&apos;" style:font-family-generic="swiss"/>
    <style:font-face style:name="Liberation Serif" svg:font-family="&apos;Liberation Serif&apos;" style:font-family-generic="roman" style:font-pitch="variable"/>
    <style:font-face style:name="Liberation Sans" svg:font-family="&apos;Liberation Sans&apos;" style:font-family-generic="swiss" style:font-pitch="variable"/>
    <style:font-face style:name="Lucida Sans" svg:font-family="&apos;Lucida Sans&apos;" style:font-family-generic="system" style:font-pitch="variable"/>
    <style:font-face style:name="Microsoft YaHei" svg:font-family="&apos;Microsoft YaHei&apos;" style:font-family-generic="system" style:font-pitch="variable"/>
    <style:font-face style:name="SimSun" svg:font-family="SimSun" style:font-family-generic="system" style:font-pitch="variable"/>
  </office:font-face-decls>
  <office:automatic-styles>
    <style:style style:name="P1" style:family="paragraph" style:parent-style-name="Standard" style:master-page-name="Standard">
      <style:paragraph-properties style:page-number="auto"/>
    </style:style>
    <style:style style:name="P2" style:family="paragraph" style:parent-style-name="Standard" style:master-page-name="First_20_Page">
      <style:paragraph-properties style:page-number="auto"/>
    </style:style>
    <style:style style:name="P3" style:family="paragraph" style:parent-style-name="Header">
      <style:paragraph-properties fo:text-align="center" style:justify-single-word="false"/>
    </style:style>
  </office:automatic-styles>
  <office:body>
    <office:text text:use-soft-page-breaks="true">
      <text:sequence-decls>
        <text:sequence-decl text:display-outline-level="0" text:name="Illustration"/>
        <text:sequence-decl text:display-outline-level="0" text:name="Table"/>
        <text:sequence-decl text:display-outline-level="0" text:name="Text"/>
        <text:sequence-decl text:display-outline-level="0" text:name="Drawing"/>
      </text:sequence-decls>
      <text:p text:style-name="P2">Hello</text:p>
      <text:p text:style-name="Standard">(some standard style text</text:p>
      <text:p text:style-name="P1">World!</text:p>
    </office:text>
  </office:body>
</office:document-content>
```

```

128     <style:header-footer-properties fo:min-height="0cm" fo:margin-top="0.499cm"/>
129     </style:footer-style>
130 </style:page-layout>
131 <style:page-layout style:name="Mpm1">
132     <style:page-layout-properties fo:page-width="29.7cm" fo:page-height="21.001cm" style:num-format="1" style:print-orientation="landscape" fo:margi
133     <style:footnote-sep style:width="0.018cm" style:distance-before-sep="0.101cm" style:distance-after-sep="0.101cm" style:line-style="solid" st
134 </style:page-layout-properties>
135 <style:header-style>
136     <style:header-footer-properties fo:min-height="0cm" fo:margin-bottom="0.499cm"/>
137 </style:header-style>
138 <style:footer-style>
139     <style:header-footer-properties fo:min-height="0cm" fo:margin-top="0.499cm"/>
140 </style:footer-style>
141 </style:page-layout>
142 </office:automatic-styles>
143 <office:master-styles>
144     <style:master-page style:name="First_20_Page" style:display-name="First Page" style:page-layout-name="Mpm3" style:next-style-name="Standard">
145     <style:header>
146         <text:p text:style-name="Header">First Page Header</text:p>
147     </style:header>
148     <style:footer>
149         <text:p text:style-name="Footer">First Page Footer</text:p>
150     </style:footer>
151 </style:master-page>
152     <style:master-page style:name="Standard" style:page-layout-name="Mpm1">
153     <style:header>
154         <text:p text:style-name="MP1">Default Style Header (landscape)</text:p>
155     </style:header>
156     <style:footer>
157         <text:p text:style-name="Footer">Default Style Footer (landscape)</text:p>
158     </style:footer>
159 </style:master-page>
160     <style:master-page style:name="Landscape" style:page-layout-name="Mpm2">
161     <style:header>
162         <text:p text:style-name="Header"/>
163     </style:header>
164 </style:master-page>
165     <style:master-page style:name="Index" style:page-layout-name="Mpm4"/>

```

The Semantic Model vs. XML

- ODF XML Logic (XML Model)
 - Paragraph → Style → Master Page → Page Layout → Text/Size
- User Logic (Semantic Model)
 - Align **paragraph** with **page layout** (header & footer and size)
 - Decouple XML from user logic
 - Define feature machine readable in spec & grammar

Specify Semantic Model!

- Define semantic model in ODF spec & grammar
 - How does the ODF XML change, when we **add / modify / delete** the feature
 - Allow to send DIFF instead of documents
 - Make Libreoffice Online, Office365, Google Docs and GIT interoperable!
- Enable to generate the model from the ODF spec & grammar
 - How the run-time-model is changed when adding / modifying / deleting the feature
 - Import & export of ODF XML into application are dependent like inverse functions and should be able to be generated as well

Specify Semantic Model!

- Problem:
 - ODF grammar is more than 18.000 lines text file
 - Hard to read / analyze

How?

Tricks to analyse ODF

- **Original Idea:** Chaos Computer Congress on [Source Code Analysis](#)
 - Instead of using 'grep' on source code text files
 - Mapping source code semantic to graph database ([TinkerPop 3](#))
 - Analyze the graph with stored procedure (Gremlin script)

Tricks to analyse ODF

- **Original Idea:** Chaos Computer Congress on [Source Code Analysis](#)
 - Instead of using 'grep' on source code text files
 - Mapping source code semantic to graph database ([TinkerPop 3](#))
 - Analyze the graph with stored procedure (Gremlin script)
- **Copied Idea:** Transformed to our ODF domain:
 - Instead of looking into [ODF grammar text file](#) (>18k lines)
 - Map the ODF grammar into a graph database ([Tinkerpop 3](#))
 - Analyze the ODF grammar graph with reproducible queries (using Gremlin script within GraphDB)

How? - Mapping into GraphDB

- Map the ODF grammar into a graph database (Tinkerpop 3)
 - ODF grammar (RelaxNG XML) file is the source
 - Instead of parsing and resolving it myself, reuse [OpenSource MultiSchemaValidator](#) and parse its memory dump :)
 - Written a tool by using ANTLR 4.0 (parser generator) mapping the MSV input to a Graph XML file (to load into any GraphDB)
see [JIRA #458 of Apache ODF Toolkit \(incubating\)](#)

Reproducible Analysis

- Analyze the ODF grammar graph
- Having reproducible queries
 - using Gremlin script within GraphDB
 - What is the difference between a heading (text:h) and paragraph (text:p)?
 - Is it possible that paragraphs are being nested?
 - What is the minimal text document?
 - ...

@Developer: To Do List

- Identify all ODF patterns (semantic model, feature change API)
- Make all ODF information machine readable in the specification
 - e.g. annotate ODF elements in grammar, which start a logical user object
- Generate as much as possible from ODF spec
 - Move feature tests and test documents to ODF standard

Thank you!



All text and image content in this document is licensed under the Creative Commons Attribution-Share Alike 4.0 License (unless otherwise specified). "LibreOffice" and "The Document Foundation" are registered trademarks. Their respective logos and icons are subject to international copyright laws. The use of these therefore is subject to the trademark policy.