### **Testing Interoperability of Office Applications**

This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License.

### Miloš Šrámek

### Society for<sub>1</sub>Open Information Technologies Slovakia

SOIT Society for Open Information Technologies (Slovakia)



# **Importance of Interoperability**

- Interoperability: application and format independent rendering (of office documents)
- Important when
  - Living in a heterogeneous environment
  - Moving from one application to another
- Expectations:
  - Full application and format independent rendering of office documents



# **Users' Opinions**

- •A frequent one: MS is always right
- A not that frequent one: MS (mis)uses formats to protect its market dominance
- •Both are (usually) based on random experience with random documents
- To know the real situation we need a solid framework for interoperability testing



### **Earlier Work**

### • Presented at Plugfest 2011 in Berlin:

- ODF document overlays for a detailed visual inspection
- An *overlap index* for numeric comparison

Conversion tools at <a href="http://www.officeshots.org/used">http://www.officeshots.org/used</a>

										100	hansar		
	File	#	0	1	2	3	8	4	5	6	7	9	10
	AbiWord 2 8 6 Linux	0	100	81,92	81,92	24,54	24,55	24,54	24,6	24,6	24,6	24,3	24,3
	AbiWord 2 9 0 Linux	1	81,92	100	100	24,51	22,89	22,89	22,95	22,95	22,95	22,71	22,71
	AbiWord 2 9 1 Linux	2	81,92	100	100	24,51	22,89	22,89	22,95	22,95	22,95	22,71	22,71
100 March 100 Ma	KOffice 2 3 3 Linux B	3	24,54	24,51	24,51	100	34,44	34,53	34,3	34,3	34,3	33,02	33,02
	StarOffice_9_2_Linux_	8	24,55	22,89	22,89	34,44	100	99,51	79,01	79,01	79,01	56,29	56,29
	LibreOffice 3 3 2 Linu	4	24,54	22,89	22,89	34,53	99,51	100	79,32	79,32	79,32	56,33	56,33
	LibreOffice 3_3_3_Linu	5	24,6	22,95	22,95	34,3	79,01	79,32	100	100	100	60,8	60,8
	LibreOffice_3_4_1_Linu	6	24,6	22,95	22,95	34,3	79,01	79,32	100	100	100	60,8	60,8
	OpenOffice_org_3_4Be	7	24,6	22,95	22,95	34,3	79,01	79,32	100	100	100	60,8	60,8
	Microsoft_Office_2007_	9	24,3	22,71	22,71	33,02	56,29	56,33	60,8	60,8	60,8	100	100
	Microsoft_Office_2010_	10	24,3	22,71	22,71	33,02	56,29	56,33	60,8	60,8	60,8	100	100
			5	8	8	8		3	3	8	3	3	3 3



# **The 'DoCmp' Framework**

### Compare and grade document pairs

- Four independent error measures:
  - Per page (we assume single-page documents):
    - Page Height Error (PHE): characterizes errors in line spacing:
      - PHE = abs(h(d1)-h(d2))
    - Line Number Difference (LND): missing lines
  - Per line:
    - Feature Distance Error (FDE): maximum distance between features of aligned lines
    - Line Position Error (LPE): horizontal shift of dominating line segments

• Segmentation of documents (as images) in individual lines is required



### **The Feature Distance Error**

The Document Foundation

CONFERENCE2013

Observation: Maximum value of abs(df(L1) – df(L2)) is equal to a distance between different local features
Characterizes difference on a character level



# The Comparison Algorithm

### • For both compared documents (pdf-s)

- Convert the pdf-s in images d1 and d2
- Crop both images and compute Page Height Error
- Segment both images in lines and interline spaces
- Get LND by counting lines in both images
- For each pair of lines of d1 and d2
  - Align the pairs horizontally and vertically
  - Compute the distance field and find the difference maximum
- Find FDE as the maximum of the distance field difference maxima for all document lines
- Find LPE as the maximum of the horizontal misalignments for all document lines



### The DoCmp framework: Part 1, preparation of test files

- Prepare **source** test documents in the native format of each tested application (manual)
  - Tested applications: LO3.6, LO40, LO41, AO34, AO40, MSO10, MSO13, CW27, AW29...)
  - Native formats:
    - LO/AOO: ODF
    - MS20XX: OOXML
  - Example file: bullets.LO41.odt (the name matters)
- Convert the source document to all tested formats by all applications (automatic)
  - Example file: bullets.LO41.docx
  - Tested formats: odt, docx, rtf, doc



# The DoCmp framework: Part 2, printing to pdf

- Print the source and converted documents to pdf (automatic)
  - Example 1: bullets.LO41.odt.LO41.pdf
    - Read: source file created in LO41 in odt and printed by LO41 to pdf
  - Example 2: bullets.LO41.odt.MS13.pdf
    - Read: source file created in LO41 in odt and printed by MS13 to pdf
  - Example 3: bullets.LO41.docx.MS13.pdf
    - Read: source file created in LO41 in odt, converted by LO41 to docx and printed by MS13 to pdf



# The framework: Part 3, evaluation of the similarity measures

- Evaluate similarity measures in pairs with the **same source** pdf file:
  - bullets.LO41.odt.LO41.pdf vs. bullets.LO41.docx.MS13.pdf
- •Used measures:
  - Page height error, normalized to full page (PHE)
  - Line number difference (LND)
  - Feature distance error (FDE, per line, maximum of all lines taken)
  - Line position error (LPE, per line, maximum of all lines taken)
- Measures values stored in a csv table
- Repeated for all cases, source and target formats and applications



# The framework: Part 4, creation of reports

- Automatically generated report (ODT file, odfpy used):
  - Summaries per format, case and application
  - Detailed data on source target application pairs
    - Tables with grading
    - Images of worst case lines according to the feature distance error
- The used grading scale:

	Grade 0 Ideal	Grade 1 Very good	Grade 2	Grade 3	Grade 4	Grade 5 Failure
PHE [mm]	0	5	10	15	20	> 20
LND	0	-	-	-	-	> 0
FDE [mm]	0	0.5	1	2	4	> 4
LPE [mm]	0	2	4	6	8	> 8



# Visual comparison

- Alternative / supplement to the report
- A combined view of the printed converted and source files. For example:
  - bullets.LO41.odt.LO41.pdf vs. bullets.LO41.docx.MS13.pdf
- Three types of display:
  - Page overlay view, shows predominantly errors in line spacing
  - Aligned row overlay view, line details well visible
  - Side-by-side view, perhaps useful for user studies



# The Page Overlay View

#### Page overlay cyan: numbering/LO41/numbering-LO41.odt.LO41.pdf red: numbering/LO41/numbering-LO41.docx.MS13.pdf PgPxOvI[: 11.3 LnPxOvI[: 77.6 LnMaxDist avg: 0.29 std: 0.05 max: 0.4 med: 0.28 PgHeightErr: 68.91 HorLnShiftMax:0.13 nLinesDif: 0 Basic numbering, Font Times new roman 1. one 1. one 1. one one 1. one one one 1. one offic one one - longer text for second line indent testing ... abcd efgh ijkl mnop qrst uvwxyz abcd efgh ijkl mnop qrst uvwxyz 1. one one one - longer text for second line indent testing ... abcd efgh ijkl mnop qrst 3. uvwxyz abcd efgh ijkl mnop qrst uvwxyz 2. one two 2: one two 2: one two

- 3. 3. one one one three
  - 1. one three one
- 2. 2voone two



# The Aligned Row Overlay View

Aligned row overlay

cyan: numbering/MS13/numbering-MS13.docx.MS13.pdf red: numbering/MS13/numbering-MS13.docx.AO40.pdf PgPxOvI[%]: 36.6 LnPxOvI[%]: 76.2 LnMaxDist avg: 6.50 std: 6.34 max: 18.1 med: 6.03 PgHeightErr: 0.00 HorLnShiftMax:18.03

Basic numbering, font times new roman

- 1. one
  - 1. one one

1.1. 1. one one one - longer text for second line indent testing ... abcd efgh ijkl mnop mnop qrst uvwxyz abcd efgh ijkl mnop qrst uvwxyz

- 1.2. 2. one one two
- 1.3. 3. one one three
  - 2. one two
- 2.1. 1. one two one
  - 3. one three
- 3.1. 1. one three one
- 2. two
  - 1. two one
- **1.1. 1**. two one one



### The Side-by-Side View

#### Original

Basic numbering, Font Times new roman

1. one

#### 1. one one

- one one one longer text for second line indent testing ... abed efgh ijkl mnop qrst uvwxyz abed efgh ijkl mnop qrst uvwxyz
- 2. one one two
- 3. one one three
- 2. one two
  - 1. one two one
- one three
  - one three one

#### 2. two

- 1. two one
  - 1. two one one
- two two
   two two one
  - two two one
     two two two
  - two two two
     two two three
  - 5. two two uni

two three

Basic numbering, Font Times new roman

#### Test 8

1.	one
1.	one one
1.1.	one one one - longer text for second line indent testing abed efgh ijkl mnop qr
uvw	xyz abed efgh ijkl mnop qrst uvwxyz
1.2.	one one two
1.3.	one one three
2.	one two
2.1.	one two one
3.	one three
3.1.	one three one
2.	two
1.	two one
1.1.	two one one
2.	two two
2.1.	two two one
2.2.	two two
2.3.	two two three
3.	two three
Basi	c numbering. Font Times new roman



### Implementation

- A bash script for format conversion and printing to pdf (on Windows using Cygwin)
  - Only applications with command line interface:
    - AOO, LO, MSO, GoogleDocs, Abiword, Calligra Words, ...
- Support programs enabling the script-based conversion:
  - MSO (the OfficeConvert tool, C#)
  - AOO (communication with AOO running as a server)
  - GoogleDocs (using Google API)



# **The Testing Workflow**

- 1.Prepare source documents, store them in a predefined directory structure (all platforms)
- 2. Run the conversion script (all platforms)
  - Bash, to be repeated on all platforms
- 3. Run the printing script (all platforms)
  - Bash, to be repeated on all platforms
- 4.Run the script for evaluation of the measures and preparation of views (bash, python, only Linux)
- 5. Run the script for report generation (python, only Linux)
- 6.Enjoy the report and see how good/bad the interoperability is



# **The Experiment: Test Cases**

### •One page documents featuring

- Formatted text
- Bullets
- Bullets advanced
- Numbering
- Numbering advanced
- Chapters
- Chapters numbered
- Tables
- Tables advanced
- Image insertion



### **Numeric Results for the 'Numbering' Test Case**

ODT	AO40	LO40	LO41	MS13
AO40	[0, 0, 0, 0]	[0, 0, 0, 0]	<u>[0, 0, 0, 0]</u>	<u>[0, 1, 1, 0]</u>
LO40	[0, 0, 0, 0]	[0, 0, 0, 0]	[0, 1, 0, 0]	<u>[0, 1, 1, 0]</u>
LO41	[0, 1, 0, 0]	[0, 1, 0, 0]	[0, 0, 0, 0]	[0, 1, 1, 0]
MS13	[0, 1, 1, 0]	[0, 1, 1, 0]	[0, 1, 1, 0]	[0, 0, 0, 0]

DOC	AO40	LO40	LO41	MS13
AO40	[0, 0, 0, 0]	[0, 0, 0, 0]	[0, 0, 0, 0]	[3, 1, 1, 5]
LO40	[0, 0, 0, 0]	[0, 0, 0, 0]	[0, 1, 0, 0]	[ <u>2, 1, 1, 5</u> ]
LO41	[0, 1, 0, 0]	[0, 1, 0, 0]	[0, 0, 0, 0]	[2, 1, 1, 5]
MS13	[0, 1, 1, 0]	[0, 1, 1, 0]	[0, 1, 1, 0]	[0, 0, 0, 0]

 Poor grades should be analyzed visually

DOCX	AO40	LO40	LO41	MS13
AO40	-	-	-	-
LO40	[0, 5, 5, 0]	[0, 0, 0, 0]	[0, 1, 0, 0]	[ <u>5, 1, 1, 0]</u>
LO41	[0, 5, 5, 0]	[0, 1, 0, 0]	[0, 0, 0, 0]	[ <u>5, 1, 1, 0</u> ]
MS13	[0, 5, 5, 0]	[0, 1, 1, 0]	[0, 1, 1, 0]	[0, 0, 0, 0]

The Document Foundation



# 'Numbering': Visual evaluation

●ODT, LO40→LO41: grade 1



• DOCX, LO41 $\rightarrow$ AO40: grade 5

one one one - longer text for second line indent testing

• DOCX, MS13 $\rightarrow$ AO40: grade 5

**1.1. 1.** one one one - longer text for second line indent testing

### • DOCX, LO40, LO41 $\rightarrow$ MS13: grade 5

Basic numbering, Font Times new roman

1. one 1.  $\Theta ne_{1}$  one one

- 1. One one one longer text for second line indent testing ... abcd efgh ijkl mnop qrst uvwxyz abcd efgh ijkl mnop qrst uvwxyz
  - 1. one one one longer text for second line indent testing ... abcd efgh ijkl mnop qrst
  - 3. uvwxyz abcd efgh ijkl mnop qrst uvwxyz

### ●DOC, AO40, LO40, LO41→MS13: grade 5

• The last line missing in MS13



### Summaries

### • ALL cases, LO41, AO40 and MS13:

format	text height err.	feature dist. err.	line position err.	line number diff.
doc	0.9	1.5	1.3	1.1
docx	1.1	2.0	1.9	0.0
odt	1.0	1.6	1.0	0.0
Worst grades	:			
format	text height err.	feature dist. err.	line position err.	line number diff.
doc	5.0	5.0	5.0	5.0
docx	5.0	5.0	5.0	0.0
odt	5.0	5.0	5.0	0.0

### •All cases, ODT

Average grades:			
source/target	LO41	AO40	<u>MS</u> 13
LO41	0.0	0.4	1.8
AO40	0.5	0.0	1.9
MS13	1.5	1.4	0.7
Worst grades:			
source/target	LO41	AO40	MS13
LO41	0.0	1.0	5.0
AO40	5.0	0.0	5.0
MS13	5.0	5.0	5.0



# Summary: LO41 & AO40 vs. MS13

### • LO41 vs. MS13:

Average grades:

format	text height err.	feature dist. err.	line position err.	line number diff.
doc	0.8	1.5	1.2	1.2
docx	1.3	1.2	1.1	0.0
odt	1.0	1.7	1.2	0.0

Worst grades:

format	text height err.	feature dist. err.	line position err.	line number diff.
doc	3.0	5.0	5.0	5.0
docx	5.0	3.0	5.0	0.0
odt	5.0	5.0	5.0	0.0

### • AO40 vs. MS13:

Average grades:

format	text height err.	feature dist. err.	line position err.	line number diff.
doc	0.9	1.4	1.2	1.2
docx	0.6	1.7	1.4	0.0
odt	1.2	1.6	1.2	0.0

Worst grades:

format	text height err.	feature dist. err.	line position err.	line number diff.
doc	5.0	5.0	5.0	5.0
docx	5.0	5.0	5.0	0.0
odt	5.0	5.0	5.0	0.0

(tested both reading and writing)



# **Shortcomings and Future Work**

- The error measures are perhaps not optimal
- The grading scale is perhaps not optimal
- Correlation between the proposed approach and user perception is unknown
  - Run a user study?
- Can the results be used for LO marketing?
  - Results published at independent site, e.g. Phoronix?
- Where to publish the code?
  - Github?



### Download

• The code and test cases: https://gitorious.org/docmp

- The test cases presented in Milano, converted and printed files(52MB): http://ubuntuone.com/0sBUtiZdFyTYeI6VM1Wcal
- The test cases presented in Milano, reports: http://ubuntuone.com/2nlBtaXkY8flQtvFOMrbW5

# **Thank You!**





## Is an implementation correct?

- Tests show differences, but do not tell us what is correct
- •Assumption:
  - An application may fail in many ways, but correct is only in one way
- Conclusion:
  - If two **independent** applications display one file in the same way, then
    - The file is correct
    - Both applications can read the file correctly
    - The application which created the file wrote it correctly

