Opening Schrödingers Library: Semi-automatic QA Reduces Uncertainty in Object Transformation

Lars R. Clausen

The State and University Library
Århus
Denmark

Abstract. Object transformation for preservation purposes is currently a hit-or-miss affair, where errors in transformation may go unnoticed for years since manual quality assurance is too resource-intensive for large collections of digital objects. We propose an approach of semi-automatic quality assurance (QA), where numerous separate automatic checks of "aspects" of the objects, combined with manual inspection, provides greater assurance that objects are transformed with little or no loss of quality. We present an example of using this approach to appraise the quality of OpenOffice's import of Word documents.

1 Introduction

Libraries are central players in the long-term preservation business. While commercial businesses may think of 5 years as long-term, and public institutions might be happy to keep objects around for a few decades, libraries must think in terms of centuries when we plan for preservation. One central pillar of preserving digital objects is the ability to understand the way their information is encoded in a series of bits. This encoding, loosely known as a "file format", is typically a highly complex system, but frequently taken for granted because they "just work". Current file formats, which at present seem ubiquitious and easy to access, will one day be things of the past, or at the very least will be heavily modified. The PDF format exists in eight official versions[4] as well as several derivations, and the wide-spread JPEG format has two worthy successors lined up already[6,1].

To preserve access to the wealth of information currently stored in digital objects, two complementary methods are commonly suggested: Emulation and object transformation. In this article, we shall not try to determine which of the two is better, but will assume that object transformation will be the strategy of choice for a significant amount of objects now and in the future.

Once the decision to perform object transformation has been made, the question of "how do we preserve our information" becomes "how do we ensure that the information from the old objects also exists in the new objects". The prevalent way of answering this question at the moment is to take a (hopefully representative) sample of objects and examine before-and-after versions manually for differences. This method has several fundamental flaws:

1. We have no idea if the objects picked are representative, or particularly good or bad examples.

L. Kovács, N. Fuhr, and C. Meghini (Eds.): ECDL 2007, LNCS 4675, pp. 186–197, 2007.

2. We have little idea if the differences seen are caused by the transformation, or are merely an artifact of the tools used to examine the objects.
3. We may overlook errors in some significant properties in the examination if the errors are not obvious with the tools used.
4. For those objects left unexamined, we have only a statistical assurance that the information is intact.

These flaws together put us in the situation of ending up with a "Schrödingers Library": a large number of unopened boxes of information, where only the act of opening the boxes some day in the future will tell us if the information is alive, or has been dead for decades.

In order to collapse this digital superposition of states to certainly alive or certainly dead, quantum mechanics teaches us that we need to observe the objects, for instance by measuring them. Each measurement we make will collapse one or more dimensions of uncertainty, hopefully allowing us to notice transformation failures early enough to make another attempt. By combining the quality measures of a number of smaller tools, rather than attempting one big, complex check, we average out the shortcomings of each tool and gain a more detailed, yet more reliable, view of the quality of the transformation.

In the next section, we discuss some prior work and the state of the art. We then give a more detailed description of the concept of "aspects" in section 3 and use them as a mental framework for semi-automatic quality assurance in section 4. Section 5 shows an example where semi-automatic QA is applied to importing of Word documents in OpenOffice 2.0. Finally, in section 6 we conclude and look at future work.

2 Related Work

Jeff Rothenberg describes a number of the problems with object transformation[8] and argues that emulation is a more viable way of preserving access. While we do not agree that emulation is always the better solution for digital preservation, he eloquently describes many of the problems that we seek to solve.

Rauber and Rauch describes in [7] how to use Utility Analysis to decide on the best preservation strategy. They present a method for analyzing file formats and goals for preservation which could possibly be adopted for our purposes.

Most current transformations happen on an ad-hoc basis, using whatever tools happen to be available for the purpose. There is some work being done on a more systematic approach, creating generic frameworks for describing file formats and file contents. One such framework is the XCEL/XCDL system[2], which uses XML descriptions of the structure of a file plus an automatically generated extractor to turn current files into a more durable format. A similar approach is that of persistent objects[3], where a description of the encoding format, including structures and relationships, is preserved. Both these approaches require a deep understanding of the formats in question, a formidable task for complex objects like PDF. Additionally, they do not provide any guarantee that the understanding is correct. The approach described in this article could complement these approaches by providing some assurance that their understanding of the old format is actually in line with reality.

Our method is related to factor analysis[10], in that we use multiple measureable factors (aspects) to investigate what can be seen as a single unmeasurable cause (quality of transformation). In our case, the causes are errors in the transformation, either caused by problems with the transformation system or due to errors in the original documents. We may consider using factor analysis techniques to determine which underlying errors cause the most problems, indicating where to apply improvements to the transformation system. However, the main use of our analysis is to point out which documents are poorly transformed, such that these can be used to investigate the underlying errors.

Surowiecki describes in [11] examples of how averaging multiple independent guesses gives significantly better answers than trying to agree on one single answer. His examples come from areas as different as trivia shows, stock markets and the finding of lost submarines, but all show that a multitude of independent, diverse guess are better than even the most expert single guess. Even if each of the guessers only have access to a very limited amount of information, the combined estimate averages out errors to such a degree that the combined guess is surprisingly precise. Our work can be seen as applying this principle to the realm of digital preservation, where for practical reasons we cannot find the most poorly transformed objects ourselves, but must rely on an educated guess to find the problematic objects.

3 Aspects

First proposed by Anders Johansen in connection with the PLANETS project[5], aspects are an abstraction of the information stored in digital objects. Originally envisioned as a replacement for the concept of file formats, we now view it as a complementary way to discuss digital aspects and how to access the information in them.

An *aspect* is an abstract view of (a subset of the) information in one or more digital objects. Example aspects could be ICC profiles in JPEG images, metadata in MP3 files, or page breaks in Word documents. Aspects commonly correspond to certain parts of files, but they don't have to. *Implicit aspects* are aspects that require some processing of the data in the objects to be found, e.g. word counts of text files, histograms of image colors or bounding boxes of CAD objects. While explicit aspects normally can be found at a specific place in a digital object, and thus should be present in both the source and target files of a transformation, implicit aspects are mostly useful for quality control. An implicit aspect may be easier or more meaningful to compare than the raw information.

Aspects are not constrained to a given file format. An aspect like "creator metadata" may be found in all manner of formats, sometimes under different names. It is important to be specific about the meaning of aspects when using them across different formats or even when considering the same format written by different systems that may have different interpretations of the specifications.

We can fruitfully consider hierarchies of aspects or overlapping aspects. For instance, the five significant properties – content, context, appearance, functionality and structure – defined in [9] can be considered major aspects that in turn encompass numerous smaller aspects like those described earlier. As such, people have been discussing aspects in various ways before, but by unifiying these views and thinking of them all as aspects, we promote novel approaches to existing problems.

4 Semi-automatic Quality Assurance

Approaching QA from an aspect point of view suggests doing QA on each aspect separately. As part of the preservation planning, repository administrators should enumerate which aspects are to be preserved and which, if any, can be considered irrelevant. It is useful to start this by thinking about the five significant properties (content, context, appearance, behaviour and structure), but any information present in the file should be considered to belong to some aspect. Rauber and Rauch describes in [7] how to use Utility Analysis to identify important parts of a format, each of which can be seen as an aspect.

Most explicit aspects that one can come up with are likely to be worthy of preservation, especially in a library context, but some might be considered artifacts of the old format or are simply internal housekeeping. Examples of such could be a separate list of page breaks when page break markers exist in the text (a sort of non-normalized data), or a table of colors in an indexed pixmap when the target format uses full RGB representation. Such irrelevant aspects should be noted in transformation metadata.

For all other aspects, we need one or more checks to ensure that they are preserved in the transformed object. While some transformation tools may have built-in checks of the transformation quality, it is the exception rather than the rule, and would share implementation deficiencies with the transformation tool. It is preferable to use external tools to extract the aspects and compare them, and the more the better. Particularly useful are tools that don't share the same code base, since they will be less susceptible to systematic errors.

4.1 Finding Tools

The easiest way to get such tools is if somebody has already made them. Most file formats in active use will have their own ecosystem of tools available for various uses: Checking conformity, extracting information for display or debugging, automatic cataloguing etc. While they may be written for other purposes, they can frequently be used with little or no modification to extract or compare aspects. They can range from something as simple as the Unix `grep(1)` command to advanced signal processing systems. The simplest possible tool may just show the file size, for cases where the sizes of the source and target files can be easily correlated (e.g. audio and video).

Any tool that can read either the source or the target format is potentially useful, too. Even if no output is given, error messages can be used to indicate encoding problems or semantic errors. Tools that can read a format and write a radically different one can allow different kinds of comparisons than those that can just give the same kinds of format. For instance, reducing a CAD diagram to a low-resolution bitmap may allow one to check the overall positioning of objects without interference from finer details (see section 5.6 for an example).

4.2 Making Tools

Once the existing tools have been examined, if there are still some aspects that are not sufficiently checked, it is time to consider making new tools. One approach to this

is using a generic file format abstraction framework like XCEL/XCDL[2] or persistent objects[3]. These allow a system-independent description of the layout of a file and a way to automatically extract the information thus encoded. While creating a full machine-readable description of complex formats like PDF may be an insurmountable task, it should be quite feasible to make one to, say, extract metadata information or list embedded hyperlinks.

If source code is available for a program that can read either format, it may be a minor task to make from it a tool that emits a certain aspect in a form amenable to comparison. While it may be tempting to check all aspects this way if a complete reader is available, it cannot be recommended as it leaves one open to systematic error. Diversity in error checking leads to higher quality.

4.3 Comparing Measurements

Once tools for extracting aspects are assembled, one should look to how to compare them. This may be as easy as comparing two numbers or running `diff`, or could involve complex comparison algorithms and further conversions. Each comparison should yield but a single quality number on some given scale. One should not try to check too many things at a time; it is better to have many local but precise measurements than a few larger but muddled ones.

Let us consider for example a diagram format like the one used by Microsoft Visio. The appearance could be checked by converting to a high-resolution bitmap and comparing those. However, that would be less informative than if we separately compared object placements, object size, fonts used and other more detailed features. Doing a single comparison would not only leave us with only one measurement, it would also be conflating a number of different possible errors. Comparing low-resolution bitmaps would still be useful as one of several checks, as it can be a check of the overall layout that does not get affected by font rendering details or the shapes of arrowheads, but it cannot stand alone.

The output of each comparison should be a quality index of some sort. It doesn't matter if the output has no obvious units or even an identifiable meaning — as long as there is a correspondence between the output and some possible conversion error, the measurement is useful (though it is more useful if you can reason about it). The measure will be combined with a number of other measurements to pin-point the bad transformations. If the individual measures on occasion gives high marks to a bad transformation or vice versa, it need not be a disaster, as combining it with the other measurements will average out the occasional error. Having measures that overlap somewhat, or that measure the same thing in different ways, gives extra insurance against measurement errors. If an aspect is considered particularly critical, one would want to be extra careful that it gets measured accurately, and multiple measures help with that.

Once all the measurements are taken on a particular object, they should be normalized to a uniform scale. The obvious choice for normalizing is to use the average error as the midpoint of the scale, and use the standard deviation to determine the endpoints. Thus if the quality (inverted error) for a measure m of an object o is $Q_m(o)$, and the average and standard deviations of the measure for all objects is E_m and σ_m, the *normalized quality* of the object for that measure is $\frac{Q_m(o)-\max(0,E_m-\sigma_m)}{2\sigma_m}$ capped to the

range $[0, 1]$. The normalized quality of the transformation of an object is the average of the normalized qualities of that object for all measures.

The normalized quality will tell how overall well the transformation of that object went compared to the other ones. Manually examining objects with the highest and lowest normalized quality will give an indication of the quality range of the transformation, and plotting the averages can tell something about the overall quality distribution. When the transformation is performed on a large number of objects, the outliers should be manually checked to see if they are still within the bounds of acceptable quality. Also, objects with non-normalized measurements far outside the range of the standard deviation should be examined to see what caused such abberations.

Rauber and Rauch[7] uses a separate value "Not Acceptable" in their quality assessments. Such a value could also be used in this context, for characteristics that are critical and accurately checked by a single measurement. If this value is assigned for an object, all other measurements are overruled and the conversion is considered of unacceptable quality. This approach should probably be reserved for cases where resources become virtually unusable without a specific aspect correctly transformed, but could provide a shortcut to finding critical failures.

Objects measured as being of very low quality will have to be checked manually. A manual check can be aided both by the individual measurements that went into the normalized quality rating (possibly before capping to the normalized range) and by specialized or modified tools to compare originals and converted objects. Details of how to implement such aids are beyond the scope of this paper.

5 An Example: Reading Word Files in OpenOffice

As a proof of concept, we took a semi-random selection of 46 Word files from the Danish archive site vaerkarkivet.dk and investigated how well OpenOffice 2.0 could understand them. To facilitate comparison, we exported the documents to PDF using Adobe Acrobat in Word and OpenOffice's native PDF exporter. Various features of the resulting PDF files were then compared.

5.1 Setup

50 Microsoft Office files were downloaded at random from Værkarkivet, a Danish public archive of digital objects (`http://pligt.kb.dk`. Of these, 2 turned out to be Excel files rather than Word files, 1 was removed since OpenOffice could not convert it, and 2 were removed later when one of the tools failed to process it, leaving us with 45 files. These were first converted with Adobe Acrobat 7.0 Professional (Danish version) into one group of PDFs (the Acrobat conversions), and then loaded one at a time into OpenOffice 2.0.3 (Danish version), where they were exported into PDF (the OpenOffice conversions). All this was done on the same Windows XP machine.

5.2 Measure 1: Number of Pages

The first measure was the number of pages in the PDFs. The `pdftodsc` tool was used to extract this from both sets of PDFs. Only 25 of the files had exactly the same number

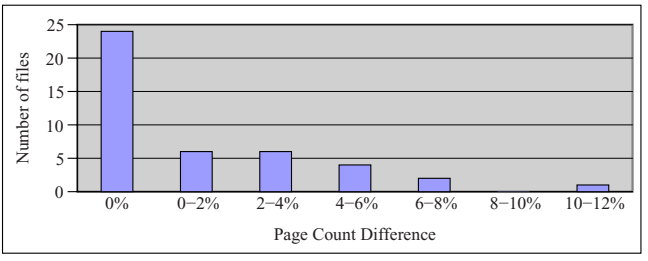

Fig. 1. Count of differences in page count

of pages. The remaining files had differences of mostly less than 5%, with a few going as high as 10%. Figure 1 shows the distribution of differences in page counts.

5.3 Measure 2: Metadata Similarity

The second measure was of the metadata found in PDF files. Using the `pdfinfo` tool, various metadata fields could be extracted, of which three (Title, Author and Page Size) could reasonably be expected to be taken from the original document. The measurement was simplistic: We measured how many of the metadata fields were the same. Only twelve differences were found, and only for one file did two of the three fields differ. Most differences were either encoding errors or fields that had been truncated. Figure 2 shows the distribution of number of metadata fields that differed.

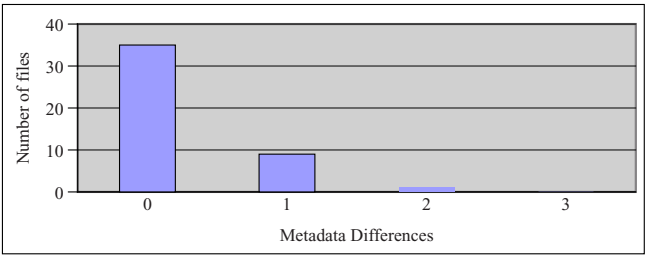

Fig. 2. Errors in metadata fields

5.4 Measure 3: Font Substitutions

The `pdffonts` utility extracts a table of which fonts are used. For this example, we merely compared the names of the fonts to see how many fonts were missing or added. It is interesting here to notice that pdffonts consistently complained that the Acrobat conversions did not embed TrueType fonts as required by Adobe's specifications. On average, the 58% of the fonts were the same in the original and the converted PDFs, with only 6 files having exactly the same set of fonts. It is unclear how much influence this has on the actual rendering, but it can still be used as a measure of difference. Figure 3 shows the distribution of number of fonts added or removed.

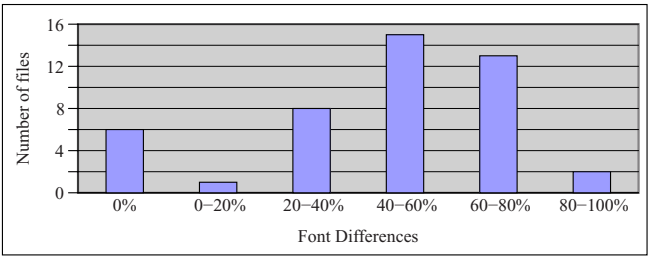

Fig. 3. Differences in included fonts

5.5 Measure 4: Text Similarity

One would hope that the text is preserved reasonably intact when transforming a text document. To check this, we used the `pdftotext` utility, which extracts into plain UTF-8. We then sorted the words and ran `diff` on them to see the number of words added or removed.

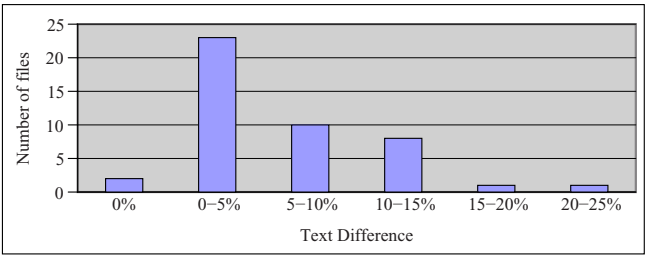

Fig. 4. Differences in words in text

On the average, 5.5% of the words had changed. Most of the changes were either due to differences in hyphenation or different layout of the titles, table of contents or index. Figure 4 shows the distribution of the percentage of words added or removed.

5.6 Measure 5: Layout Similarity

As a final measurement, we converted each page into a 40x40 pixmap with the `convert` program from ImageMagick, and then compared these using `compare` from the same package using the Mean Average Error metric. Two files from the OpenOffice set could not be converted, but caused the `convert` program to die with "unrecoverable error". Of the rest, none were exactly the same, but some exhibited significantly larger differences than others. Figure 5 shows the distribution of differences in layout

A main reason for layout changes is that lines and paragraphs get broken in different ways. If this happens near the end of a section, extra pages may be added or removed, causing the layout of pages to go out of sync. There is a correlation (0.51) between the layout similary and page count similarity.

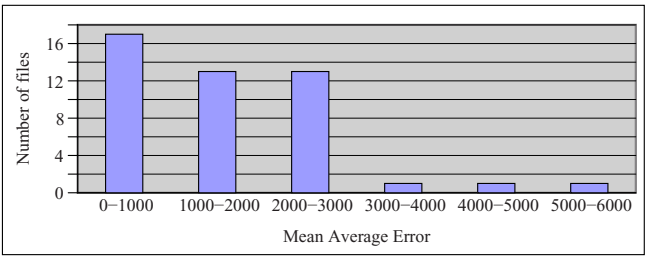

Fig. 5. Differences in low-resolution versions of pages

5.7 Combining the Measures

As described in section 4.3 above, the quality measurements are normalized based on the standard deviation. The normalized quality is then the average of the five normalized measurements. Figure 6 shows the combined error rates (inverse quality) for all documents, sorted by overall quality.

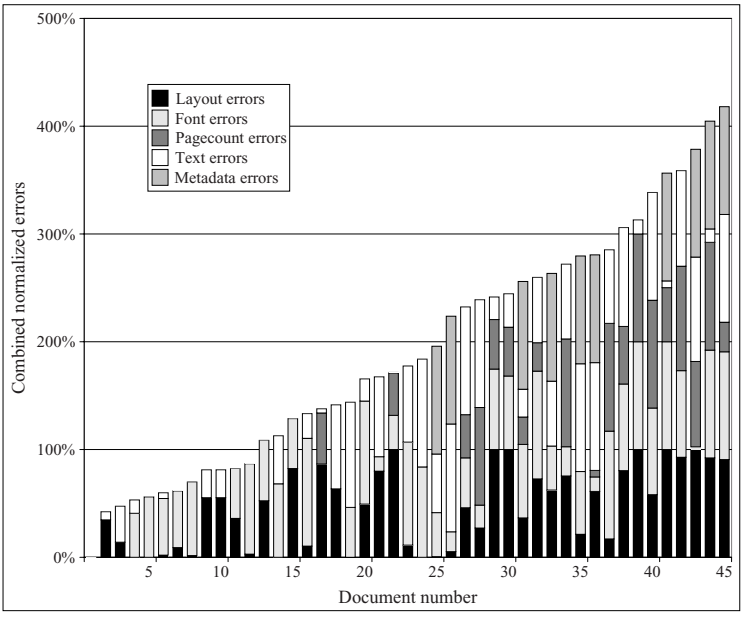

Fig. 6. Combined errors rates for all documents, sorted by total error

Correlation testing shows that the largest correlation between measurements is between the layout measurement and the pagecount measurement, with a correlation coefficient of 0.51. This is not surprising, as changes in page count must lead to some amount of change in the layout when different pages are compared. Other than that,

there are correlations of between 0.2 and 0.3 between the layout and the metadata measures, between the fonts and the pagecount measurements, between the fonts and the metadata measurements, and between the pagecount and text measurements. Outside of the correlations with metadata, these may be the result of different hyphenation systems and font substitutions causing layout changes.

5.8 Result

Based on the combined results from above, we manually examined the original Word files in Microsoft Office 2003 and OpenOffice 2.0.3 by displaying them next to each other and looking for visible differences. We examined a random sample of 10 documents as well as the 5 highest-quality and 5 lowest-quality documents.

The most common problem was changes in line breaks and page breaks, partly due to differences in font rendering, hyphenation and spacing. This would have a marked effect on the bitmap and page count measures and some effect on the text measure. Almost all the documents examined had some amount of difference in page breaks.

The manual inspection did, as expected, not give a perfect match to the calculated quality. However, the 5 highest quality documents did turn out to have very high quality conversions, with only two having any significant shifts in page breaks, and all having markup, foot notes, table of contents etc. essentially the same.

Among the five conversions with the lowest quality, all had significant to major shifts in layout, and each displayed one or more other errors, including graphics superimposed on text, table contents fused together, images missing, spontaneously appearing elements or major additions in table of contents.

It was obvious from the inspection that in this example, differences in word- and line-breaking had too much weight in the overall quality measure. The presence of a correlation of over 0.5 indicates that two of the measures measure the same error to some degree. Extra tools to extract diagrams, table of contents and other features would have pinpointed errors more accurately, as would extraction of text in a way that disregarded hyphentation. 5 measures is not enough to give a reliable quality indication, but does give strong hints.

It is particularly noteworthy that the quality measures worked given the chain of processing the data went through. Rather than comparing Word files and OpenOffice files, both were converted to PDF and in one case further converted to PNG. If used for preservation, such a chain of conversions is normally expected to accumulate errors from each transformation. However, since these transformations are allowed to drop information not relevant to the measurement being taken, such error accumulation is limited enough to vanish in the measurement errors. Thus, we can use a plethora of tools to perform our measurements without worrying overmuch about whether each tool makes a perfect transformation.

6 Conclusion and Future Work

Quality assurance is a critical part of transforming digital objects from one file format to another or to a newer version of the same format. There are numerous things that

can go wrong in such a process, and on the scale of a digital library, manual inspection of all transformed objects is a Herculean task. Thus in many cases, we inspect only a small number of objects and hope that they give enough indication of errors for the transformation process to be sufficiently debugged. This leaves us with a majority of objects in an uncertain state of accessibility until somebody in the future attempts to access them, by when it may be too late to correct the errors.

To avoid this uncertainty, we have proposed a method called *semi-automatic QA*, in which a multitude of separate quality measurements are taken and their results combined to give an overall quality rating. We base this approach on the concept of *aspects*, in which we view digital objects not as specific file formats but through a prism of smaller parts, ranging from the five significant properties of [9] over very specific aspects such as "creator metadata" or "color profile name" to implicit aspects calculated from the data in the objects.

We have given an example of using semi-automated QA to assess the quality of reading Word files in OpenOffice 2.0. In order to facilitate the comparison, we converted the documents to PDF from both Word and OpenOffice, and subsequently compared the PDF files. Despite having a small number of measures and a long chain of conversions, the highest-rated objects were indeed well transformed, while the lowest-rated objects turned out to have transformation errors of types that we had not checked explicitly for. We conclude that semi-automated QA can give a higher degree of confidence in the quality of digital object transformations by allowing practical, early pin-pointing of errors. We also make note that for measurement purposes, a longer chain of conversion programs does not necessarily accumulate errors impeding the measurements.

The concept of semi-automatic QA holds promise as a part of a digital preservation strategy, however more research needs to be done on it. Methods for defining aspects and the measures based on them needs to be developed and collected, and a firmer statistical founding needs to be incorporated. In particular, there is a need for ways to help identify the desired aspects and to analyze whether the measures in place cover all desired aspects with sufficient overlap and in a sufficiently independent manner. Methods from factor analysis can surely be applied to some of these problems, while other parts of them are a problem for preservation planning systems more than for the actual transformation systems.

It may be possible to use the approach described herein for object characterization as well. For instance, one could use fine-grained aspects to determine properties that all or nearly all objects in a collection has, either for validation of expected properties or for improving the description of the collection.

Another intriguing notion suggested by one of the anonymous reviewers is to apply entropy minimization principles to determine that all quantifiable information has been characterized. If this is feasible, it would provide guarantees that the aspects cover everything. However, it should be kept in mind that the semi-automatic QA method does not infer anything about the format of the digital objects being investigated, only about the information content.

The connection between the aspect approach and the "definitive description" approaches [2,3] has yet to be investigated. Besides aspect providing a complementary view of the objects, partial definitive descriptions could be used to extract aspects as

well: even if no complete description of a format is available, making an automatable description of the aspects that no other tools can easily extract could be much more feasible. As mentioned earlier, aspect-based investigations can also be used to verify the validity of definitive descriptions.

There is also an open area of methods to assist manual checking of quality. Several approaches can be envisioned, such as automatic side-by-side viewing, flipping back and forth between images, or highlighting automatically detected differences. The opportunities and problems in this area needs investigation, and in particular methods for data other than images need to be developed.

The methods discussed herein can also be used to extend the work of Rauber and Rauch[7], whose tests methods for transformation tools do not include anything equivalent to our implicit aspects.

References

1. Hd photo specification v1.0. Technical report, Microsoft (2006)
2. Heydegger, V., Neumann, J., Schnasse, J., Thaller, M.: Basic design for the extensible characterisation languages. Technical report, Universität zu Köln (October 2006)
3. Moore, R.: The san diego project: Persistent objects. In: Proceedings of the Workshop on XML for Digital Preservation, Urbino, Italy (October 2002)
4. Pdf reference: Technical report, Adobe Systems Incorporated (November 2006)
5. Planets: Preservation and long-term access through networked services (2006), URL `http://www.planets-project.eu`
6. Rabbani, M., Joshi, R.: An overview of the jpeg 2000 still image compression standard. Signal Processing: Image Communication 17(1), 3–48 (2002)
7. Rauch, C., Rauber, A.: Preserving Digital Media: Towards a Preservation Solution Evaluation Metric. In: Chen, Z., Chen, H., Miao, Q., Fu, Y., Fox, E., Lim, E.-p. (eds.) ICADL 2004. LNCS, vol. 3334, pp. 203–212. Springer, Heidelberg (2004)
8. Rothenberg, J.: Avoiding technological quicksand: Finding a viable technical foundation for digital preservation. Council on LIbrary and Information Resources (CLIR) (January 1999)
9. Rothenberg, J.: Carrying Authentic, Understandable and Usable Digital Records Through Time. RAND Europe, Leiden, The Netherlands (1999)
10. Rummel, R.J.: Applied Factor Analysis. Northwestern University Press, Evanston (1979)
11. Surowiecki, J.: The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations. DoubleDay (May 2004)