ARCHIVING CONFERENCE
https://www.ingentaconnect.com/contentone/ist/ac/2008/00002008/00000001/art00011

Analysing the Impact of File Formats on Data Integrity

Volker Heydegger; Universität zu Köln; Cologne, Germany

Abstract

The concept of file format is fundamental for storage of digital information. Without any formatting rules, bit sequences would be meaningless to any machine. Due to various reasons there exists an overwhelming mass of file formats in the digital world, even though only a minority has a broader relevance. Particularly in regard of specific domains like long-term preservation of digital objects, the choice of the appropriate format can be a problematic case. Thus, one of the basic questions an archivist needs to get an answer for is: Which file format is most suitable for ensuring the longevity of its information?

In this study a particular criteria for long-term preservation suitability is picked up: the robustness of files according to their bit error resilience. The question we address is: Up to what extent does a file format, as a set of formatting rules, contribute to the long-term maintainability of the information content of digital objects? Or in other words: Are there any file format basing factors promoting the consistency of digital information?

Introduction

Among several other criteria [9], one considered to be crucial for the decision which file format to choose for digital preservation refers to the capability of file formats to keep its information, as it is, over a long period against the evil of bit rot. The single reasons for corrupted files are manifold. Nevertheless there are two main categories: First, bit errors in files occur in consequence of degradation of the storage medium, e.g. caused by poor physical storage conditions, just as a natural decay of the medium or as a consequence of massive usage. This is especially true for storage of data on optical disks [5]. Hard disks are also exposed to such errors although less severe [7][12]. Second, bit errors result from transmission procedures. However, e.g., in case of data migration, these errors can be prevented if methods for checking the integrity of the data are implemented.

The nature of the corruption of files can also be manifold. Bit errors can be located to special areas of the file, they can also be distributed [5]. The actual location of bit errors within a file strongly depends on the underlying reason for corruption: E.g. consider a DVD which was damaged by the influence of strong heat. In this case the distribution of bit errors may vary according to the strength of direction of the heat source. On the other hand, files can be corrupted in a way that not only single bits are flipped from zero to one or vice versa but also that they totally get lost. In such cases, the effect on data integrity increases dramatically. In this study we focus on bit errors in the sense of flipped bits and on equally distributed errors. In fact there is actually no general tendency of error location in files as a consequence of the manifold reasons for corruption we mentioned before.

The current strategy to get the problem of file corruption under control targets at hardware-sided solutions. Determined by the storage medium, data is usually stored according to particular methods which again follow international standards. Specific codes for error correction are adapted to the processes of reading and writing data from and to the storage medium. The devices which deal with the medium are constantly refined in their ability to handle it with higher precision, thus improving the quality of the data as well. New technologies using different methods and materials for storage media, e.g. holography, promise to push on the durability of the medium while increasing the storage capacity at the same time. However, all of these efforts are not primarily the result of a basic sense for the necessity of keeping data as safe as possible; most notably they arise from the necessity to cope with the advancing technological complexity of such devices and storage media.

Even if it would succeed to get a grip on the problems of storage technology in terms of durability and capacity of storage media more accurately: If it comes to make long-term preservation of data also feasible in an economical sense, there is no doubt to follow up additional strategies for improving data integrity. The proposal to keep data by redundant copies, additionally locally distributed, is a simple and useful approach but may suffer from additional cost effects [1][10].

The study on hand takes up this necessity to find backing solutions and moves away from the problem of physical and technical restrictions of storage media. The focus is now on logical representation and organisation of data as files, which is determined by a set of given rules, commonly called file format. The concept of file format is the fundament for data to become meaningful. Data interpreted by a machine according to the underlying file format is not only raw data but information.

So the question we address is: Up to what extent does a file format, as a set of formatting rules, contribute to the long-term maintainability of the information content of digital objects? Or in other words: Are there any file format basing factors promoting the consistency of digital information?

The consequence of clarifying this question is obvious: If there is indeed a significant relation between file format and information constancy, it will be possible, in due consideration of the revealed determinants, to improve the long-term preservation of digital objects: E.g., existing file formats could be optimized, newly created file formats could be, with the help of the updated knowledge, conceived including aspects of longevity.

Studies in this area focused one specific aspect of representing data in files: Data compression [2][3][8]. This is not surprising, for data compression is a major feature of file formats, especially in terms of data integrity of files. It arises from certain technological facts which originate in the information technologies past, for capacity of storage media and efficiency of data processing systems were formerly quite more limited than they are now. Nevertheless these are still factors to be considered. Though technological progress may lessen these limitations, the mass of digital data still increases. After all this will be more and more a domain specific question. The question if to store data as compressed or not does not arise for digital objects like movies;

however, for an archivist who wants to keep his images for long-term preservation, this may be a question worth asking.

Indeed, especially in terms of long-term preservation, one was sceptical about the usage of data compression for a long time: Compressed data is extremely prone to consecutive faults caused by bit errors. Therefore, besides other reasons, JPEG 2000 was also developed with the goal in mind to make compressed data more robust against bit errors. Since then the discussion on usage of compressed data for preservation purpose is sparked again [2][4][8].

Although this study takes up this special point, we also focus on other aspects of file format, namely which kind of data is captured by the file format and how data is structured and related among each other.

Additionally we concentrate on image files as our practical subject of research here. Therefore the following remarks have a strong relation to image file formats.

General Implications on File Format Data

Usage and Processing

In the context of this study, a file format is a set of rules constituting the logical organisation of data and indicating how to interpret them. The quantity of set of rules may vary to a great extent and depends strongly on the information intended to be represented. In the context of this study we call all information that can be described through one or more files, their formatted data respectively, a digital object.

The complexity of digital objects may variegate also in a certain span. But even within similar categories of information, digital objects can be described by file formats in an extremely different level of complexity. A digital object of domain 'image' may be modelled in a raw data format, using quite few formatting rules. If it is intended to be transferred and represented through a specific software like a web browser, the functionality of a raw data format usually does not last anymore. Or as another example: An image intended to be represented not only statically as a whole but from which certain parts of it are matter of interest may be expressed best way using JPEG 2000 file format. The question on which format to choose for a digital objects data is in terms of temporary usage a question concerning the scope of application of that object.

An essential conclusion that can be drawn from these considerations is: Every digital object is provided with a basic content of information. This is directly reflected through the data which represent that information. Additionally the basic content of information can also be modified and enriched by added functionality.

Information is exactly that in what humans as the users of data are interested in. Exaggerated: A user does not care about data. From the users point of view a perfectly preserved digital object presents the same information to him or her as originally intended. With respect to a categorization of file format data, this should be seen from a different perspective : A perfectly preserved digital object presents the same information as originally intended after its data has been processed by a file format data processing software following the rules given by a file format specification.

The relation is now contrariwise: The software does not care about information but data.

Software which has to cope with the task of transforming data to useful information needs to rely on the readability of data. Data must be processible according to the underlying file format.

Which conclusions can be drawn out of this regarding bit error corrupted files? For simplification of the following example let us presume that a given file format defines as smallest processible unit one byte (as it is indeed usual in most formats). If so, a single bit error causes a one byte error, this is an error rate of 1:1. We call this plain information loss. In this case, the actual change in the bit state corresponds to the actual information loss (given one byte as smallest processible unit) since it affects the information which is represented by exactly one (the corrupted) byte. Consider a comparison of two files A (this is the original, uncorrupted file) and B (the original version as corrupted file), where B differs from A in exactly one byte. A program that is able to perform a pairwise comparison of the byte values of the files then recognizes exactly one different byte. In a sequence of unformatted bits every change in the bit state is definite and irreversible. For data described by a file format this is not necessarily so. E.g., file formats which allow for error correction codes within the data potentially enable the processing software to recapture the original byte (bit) value.

Sometimes a file format specification defines a byte value as fixed value. In such cases it is also possible to recapture the original byte value from the affected byte. However, such format specific definitions must be implemented by the processing software. Conventional software applications which implement a file format compliant to its rules should not accept such an error (by the way: this is exactly what a file format recovery tool does not).

Simple bit errors do not always cause plain information loss with 1:1 error rates as shown in this example. The error rate is expected to be multiplied if a file format defines logical information units by more than one single byte. We call this kind of information loss, logical information loss. E.g., for the case of a file format assigning four byte for representation of big numbers: the information loss for an one bit error then increases to an error rate of 1:4 (again in terms of byte as reference unit).

A third kind of information loss is much more effective regarding information loss. We term it conditional information loss. Such information loss produces error rates of much higher extent than those discussed so far. In the extreme case it causes the content of the entire file to not being processible, with the result of error rates increasing up to 100%. TIFF file format for example allows for placing the pixel data of an image at any position within a file except the first eight positions which are always fixed for special usage. This file format rule necessitates to set an offset to that position within the file where the pixel data can be found. This is done in the so called image file directory, which also can be placed arbitrarily within the file (again except starting at one of the first eight positions in the file). It is once more necessary to set an offset that tells the processing software where to find the beginning of the image file directory. A bit error occurring in the offset data to the start of the pixel data, not only causes an error, in the sense of a logical information loss, within the offset data per se. As an aftereffect, at least any 'conventional' processing software does not find out anymore where exactly the pixel data is located within the file. In this case we have a conditional information loss to the amount of the number of those data indicated via the offset. More worse, such bit errors raise the conditional information loss to the maximum if, like in this example, the error already occurs in the offset to the image file directory. Repairing such an error is even for a file format recovery tool a hard job to do. To adjust such errors in corrupted files is a real challenge for file format recovery tools.

Functionality and Categorization

Data, organized according to a file format, is in its basic function an information carrier. The primary task of data-processing software is to read data with respect to the file format and to capture its information content. Such processed data can then finished according to the aspired purpose. An image viewer for example reads data as defined by the image file format from a file to transfer it to one of the image viewers concurrent purposes.

Even though all of the data described through a file format always represent some kind of information, the nature of this information is different, at least in terms of functionality. That is why a file format assigns functional meaning to data, according to its information content.

Which kind of information is represented by file format data? We generally differentiate between two main categories, which are also the basis for the robustness indicators described in the following section. The first category relates to aspects relevant for usage, the other to data related to processing tasks.

The basic content of information of a digital object that was already discussed in the previous section is reflected in the first of these two main categories. Such information and its carrying data respectively is essential for representation of the object.

Data relevant for usage can be distinguished in three sub-categories. Those of the data relevant for usage which carry the basic information of a digital object are called basic data. In case of a raster image rendered to a display, the carrier of these information are the processed pixel data. Or, in case of a simple text encoded in HTML, this is those data which map the text as, for example, accessible via a web browser. In case of an audio file, this is all of the data interpretable as sound, basically all sample data.

A second sub-category of data relevant for usage can be characterised as not directly carrying a digital objects information; nevertheless this data represents information which is indirectly necessary for adequate representation of the information content of the base data. We call that kind of data derivative data. Data on picture coordinates, bit depth or compression method are representatives from the image domain. In case of text domain, this can be data relating to text formatting information, for example font style, font size or space settings.

Another sub-category of data relevant for usage is commonly known as descriptive metadata. It adds such information to digital objects that is irrelevant for basic representation of the object. Data about creator, author, date of creation or producing software are examples for that sub-category. We call it supplemental data.

The second main category of file format data introduced here is data concerned with the structural organisation and the technical processibility of any other file format data, i.e., in its core this is data relevant for processing tasks.

At a first glance, such kind of data seems to play a minor role opposite to the object-related information carrying data. However, this is not the case. Often such data is essential for the processibility of the entire file. The example for TIFF file format we discussed in the previous section deals with data of that category.

Processing-relevant data is distinguished in two sub-categories. Such data supporting the structural configuration of the entire data is called structural data. Structural file format data describe the logical units of the file organisation. Examples for this category are the tag numbering in TIFF files, offsets to the position of certain related data, or data that functions as filler data. Structural data is directly related to the structure of the data described by the file format.

Another sub-category includes data giving information on the validity of subsequent data units. We call it definitional data. By its application on target data, data of that kind gives an answer to the question if a certain sequence of data units (the target data) are valid or not according to the parameter defined through that data. Error correcting codes or indications on the data-type to be used are two examples for that category. In contrary to structural data, definitional data asks for an interpretation on any target data.

The advantage of such a categorization should be evident. Bit errors can now related to a categorization scheme. A close analysis of the distribution of these categories on different file formats can indicate which kind of data loss is to be expected. The results of quantities analysis of errors in corrupted files can be discussed by means of a distinct vocabulary. It is also possible to derive measures for information loss using these categories. Recently, even though in a slightly different context, the assumption of general file format data categories has led to the development of new comprehensive practical approaches to the characterisation of file format data [14].

Measuring Information Loss

Robustness Indicators

Building on the theoretical foundations we examined in the previous chapters, metrics for measuring information loss in corrupted image files were derived. These metrics are called robustness indicators (according to reflections in [13]). They give us a hint on the robustness of a file format in terms of the categorized file format data. Thus, in difference to similar existing metrics (e.g., RMSE, simple match coefficient), these metrics explicitly refer to our categorization of file format data.

The robustness indicators can not be interpreted as image quality measures. They are prepared for giving information on information loss that is caused by data which has changed or which original information content can not be captured anymore; this can be the case if a byte, as the information carrying unit, is directly corrupted (plain information loss) or because a certain number of bytes can not be processed adequately (logical or conditional information loss). Again: Information loss is always reflected by data as carrier of information. In the following, we present those Robustness Indicators which we applied to the test corpus in the next section.

RB is defined as a robustness indicator for file formats which relates to the basic data of usage:

$$R_B = \boldsymbol{\Delta} (b_0 ,b_1) / m \qquad (1)$$

where
b_0 is the basic data of usage before being corrupted,
b_1 is the basic data of usage after being corrupted,
m is the absolute number of corrupted data units.

R_{Bt} additionally includes the relation to the total number of basic data of usage:

$$R_{Bt}= R_B / n \tag{2}$$

where
n is the total number of basic data of usage.

A Test Implementation for Measuring Information Loss

We have implemented a software tool that is able to simulate data corruption, which can recognize data according to the file format data categories we defined, that is able to process and translate the relations between the data categories and that finally computes the robustness indicators.

In its core procedure it analyses files (which represent the underlying file format) in several subsequent processing parts, using both the original (error-free) file and a manipulated (bit-corrupted) version of it. The latter is prepared by the manipulation module of the software, also taking compressed data into account by trying to decompress the corrupted files. After that, the tool analysis the original file as to the data categories defined in the model. Another module transforms the data of both files into an internal normalised representation, processing file format specific data allocations as described in the file format specification. In a last move, the data of the normalised corrupted representation is used to compute RIs.

We have also built a corpus of test files for a number of image file formats. In this study we report on the results for four of them: TIFF, PNG, BMP (windows) and JP2. The corpus comprises files which consider various basic characteristics and features of each file format. The results reported in here relate to a 'real world image', i.e. a colored image, standard 24-Bit RGB. For some of the file formats we created different test files reflecting potentially important characteristics in terms of the expected data effects on data integrity. In this case we added compression characteristics (for details see table 1). As already discussed, they so far played a leading role in the discussion of file formats robustness and their potentiality for long-term preservation respectively.

Table 1 shows the results for Robustness indicator R_{Bt}. For better readability the results for R_{Bt} are transformed to base 100 (i.e. expressed in percentage). The single file formats and compression characteristics are put in the first column. The given ratios relate to compression ratio understood as ratio between uncompressed size and compressed size of the files. The indication in brackets is the compression ratio in terms of space savings. The other columns contain the single results for R_{Bt}. We have performed test series on the base of byte errors with corruption rates of exactly one byte (which results in individual percentage corruption rates based on the original file size (second column, indication in brackets) and three more for percentage corruption rates of 0.01, 0.1 and 1.0, since they seem to be sufficient enough to clearly show the effects on file corruption in general and with respect to R_{Bt} in specific. For each file type and corruption rate we performed the corruption procedures 3000 times always using a different set of random numbers per single corruption, generated by Mersenne Twister algorithm [6] that guarantees equal distribution of errors, as we intended to have for this part of the study. We also made sure that none of the single random numbers per set occurred twice or more to avoid imprecision of R_{Bt} values. We also cross-checked the results with confidence intervals indicating a deviation of the R_{Bt}s of less than three percentage in all cases.

Table 1: Results for R_{Bt} (in percentage) for various file formats

	1 Byte	0.01	0.1%	1.0%
TIFF				
uncompressed	0.00 (0.00063)	0.56	6.64	48.83
JPEG compressed, ratio 1:2.60 (62%)	2.14 (0.00166)	13.03	-	-
JPEG compressed, ratio 1:10.72 (90%)	2.44 (0.00505)	13.32	-	-
LZW compressed, ratio 1:1.01 (2%)	1.37 (0.00064)	18.79	77.95	99.34
ZIP compressed, ratio 1:1.28 (22%)	27.12 (0.00081)	84.92	98.47	-
PNG				
ZLIB compressed, unfiltered	18.21 (0.00074)	79.15	97.63	-
ZLIB compressed, filtered	25.05 (0.00085)	81.83	98.08	-
BMP (windows)				
uncompressed	0.00 (0.00063)	0.14	1.92	15.29
JP2				
lossless, ratio 1:1.36 (27%)	17.53 (0.00086)	76.22	94.29	-
lossy, ratio 1:7.42 (87%)	33.31 (0.00166)	51.86	95.03	-
lossy, ratio 1:2.64 (62%)	22.61 (0.00468)	72.93	95.62	-

Discussion of the Results

The results reveal a strong correlation between usage of compression and data integrity. As compression is a widely used feature in many file formats, for some explicitly dedicated to (e.g., JP2), compression can be considered as one of the most important features of file formats and therefore is one of the crucial factors for a file formats impact on data integrity. In almost all cases of compression usage, 0.1 percentage of byte corruption is enough to produce R_{Bt} values of more than 90 percentage (in case of TIFF with JPEG compressed data we were not able to compute R_{Bt} with sufficient exactness since the errors provoked serious software crashes). For example TIFF with ZIP compressed data : More than 98 percentage of the basic data of the corrupted file is changed compared to the original data. Or in other words: More than 98 percentage of single information units have changed according to the change in the data which carries this information.

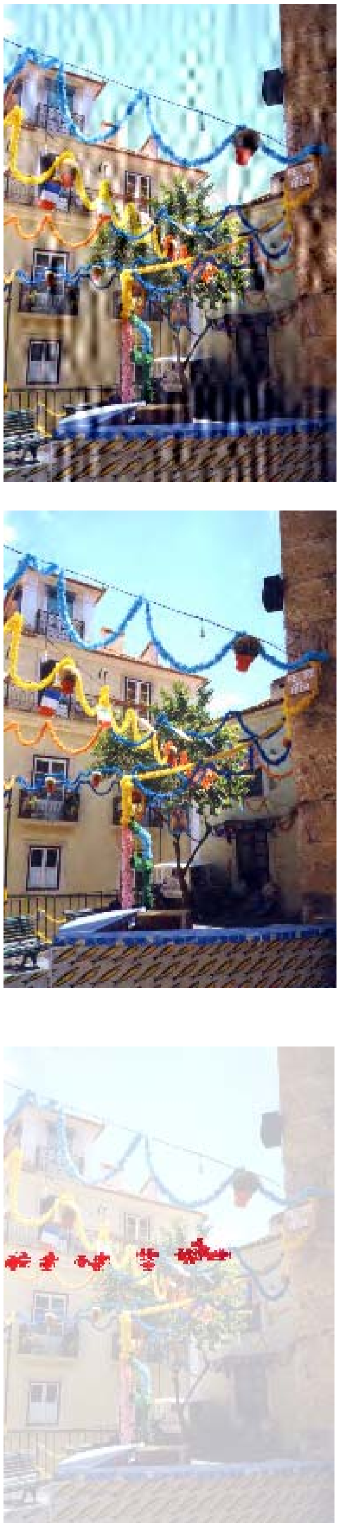
Almost more amazing are the results for one byte corruptions. In case of JP2, a one byte error causes, as a consequence of conditional information loss, a change in basic data of about 17 percentage for lossless compressed data (corruption rate: 0.00086), up to 33 percentage for lossy compressed data (corruption rate: 0.00166) in moderate compression ratio (JP2 is able to produce much higher compression ratio). Conditional information loss is symptomatic to compressed data and seems to not depending on whether data is compressed lossless or in a lossy mode.

Table 2: Totally failed test files (in percentage)

	1 Byte	0.01%	0.1%	1.0%
TIFF				
uncompressed	0.00	0.36	3.60	32.00
JPEG compressed, ratio 1:2.60 (62%)	0.13	0.67	-	-
JPEG compressed, ratio 1:10.72 (90%)	0.11	5.63	-	-
LZW compressed, ratio 1:1.01 (2%)	0.03	1.20	13.43	72.40
ZIP compressed, ratio 1:1.28 (22%)	0.07	0.50	3.77	-
PNG				
ZLIB compressed, unfiltered	0.00	0.70	4.30	-
ZLIB compressed, filtered	0.00	0.10	4.30	-
BMP				
uncompressed	0.00	0.10	1.67	11.07
JP2				
lossless, ratio 1:1.36 (27%)	0.40	0.40	11.10	-
lossy, ratio 1:7.42 (87%)	0.20	2.00	12.10	-
lossy, ratio 1:2.64 (62%)	0.10	1.30	10.40	-

Particularly for the JP2 results, R_{Bt} may be a convenient measure for reflecting the characteristics of JP2 files after being corrupted. Already for low corruption rates, the rendered versions of corrupted JP2 files can be extremely different (Figure 1). This is not a JP2 specific issue. Nevertheless JP2 compression is, compared to other compressions, quite successful in producing images which keep their visual quality, especially in case of low corruption rates, although there are moderate differences in pixel data (see Figure 1 also). However, the effects of bit corruption on the rendered files can vary to a great extent. Right due to that, R_{Bt} values reflect the actual information loss, not influenced by the deficiency of humans visible system. If it is our task to make a clear statement on whether the data of a file is in danger to be

Figure 1: Two JP2 images, both with the same degree of corruption (one single byte); the second image shows no visual difference to the rendered version of the original uncorrupted file (not illustrated) although there are actual changes in pixel data (as shown in the third pseudo- image, where different pixel data is marked in red).



changed after a bit corruption, the visual appearance of the object after being rendered is not a matter of interest.

Again, this is the task of quality measures. So while considering JP2 as a candidate for long-term storage, this still remains a point for discussion, at least if one decides that error resilience should be an important issue for long-term preservation.

Those files not using compression (TIFF uncompressed, BMP uncompressed), proved to be much more stable. For one-byte errors, none of the two file formats showed serious problems (R_{Bt} values of 0.00). Table 2 shows the number of files that totally failed during processing (also in percentage). The reasons for such a phenomenon can be found in corruption of extremely significant data. This always the case for derivative data, structural data or definitional data. As a result, this causes destructive conditional information loss. We already discussed another example for conditional information loss in TIFF files in the section before. Expectedly, the values for R_{Bt} increase according to the corruption rate.

Nevertheless there are differences, especially with increasing corruption rates. Just for TIFF and BMP uncompressed, there is a clear tendency. BMP uncompressed appears to be quite stable in its file format structure. A closer look at its file format structure shows that BMP is in deed quite simple in it. Most of the lengths and positions of the data fields are predefined. In contrary, TIFF allows for advanced features like to stripe pixel data or to freely choose positions of data fields within the file. File formats which support advanced features tend to be more complex in their structure. This is not surprising since this requires concessions to the processing software. However, complex file formats tend to get into trouble with keeping their data against bit errors.

Conclusions and Outlook

With the results of this study we give some direction for all those people who are concerned with question of file format and its usability, especially for long-term storage. The choice they make surely depends on factors which are widespread and not only depending on error resilience. To great extent they are often a matter of organizational needs. Despite all that, we regard robustness of file formats against bit corruption as a main factor: As long as it is possible to constantly check files for data integrity, error resilience may be less a hard problem. But consider a scenario where the keepers of the data are not able to do so anymore, may it be because of financial, technical, societal or whatever else shortcomings; then, robustness of file formats against bit corruption is in deed the more crucial.

Robustness Indicator is a simple measure for quantitative analysis of file format data. It does not claim to be a measure of quality analysis. The results we reported in here are part of a larger study. In the future we will focus on enlarging the set of measures for file format, also including measures which are already proven as useful for such issues. This will enable us to additionally refine the model of file format data categorization as well as the findings so far.

We will also refine the analysis of the exact data categories responsible for the specific kind of information loss we diagnosed. This is done by in depth analysis of file formats supported by additional test implementation features. This will help us to find a close understanding of the relation between file format and its error resilience.

We also will extend our research on file formats from other domains especially for formats of text or hybrid content. This will validate and/or improve our given data categorization towards a common model of file format data. Additionally, it is expected to reveal so far unidentified impact of file formats on data integrity for those domains.

References

[1] Bradley, K.., Risks Associated with the Use of Recordable CDs and DVDs as Reliable Storage Media in Archival Collections: Strategies and Alternatives. UNESCO (2006). http://unesdoc.unesco.org/images/0014/001477/147782E.pdf

[2] Buckley A., JPEG2000, A Practical Digital Preservation Standard? DPC Technology Watch Series Report 08-01 (2008). http://www.dpconline.org/docs/reports/dpctw08-01.pdf

[3] Buenora, P., Long lasting digital charters. Storage, formats, interoperability, Presentation held on Digital Diplomatics, Munich (2007). http://www.cflr.beniculturali.it/Progetti/FixIt/Munich.ppt

[4] DPC/BL Joint JPEG 2000 workshop, June 2007. http://www.dpconline.org/graphics/events/0706jpeg2000wkshop.html

[5] Iraci J., The relative stabilities of Optical Disk Formats, Restaurator, Vol.26, Number2 (2005).

[6] Matsumoto M. , Nishimura T., Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator, ACM Trans. on Modeling and Computer Simulation Vol. 8, No. 1, January, pp.3-30 (1998).

[7] Panzer-Steindel, B., Data Integrity, CERN/IT (2007). http://indico.cern.ch/getFile.py/access?contribId=3&sessionId=0&resId=1&materialId=paper&confId=13797

[8] Rog, J., Compression and Digital Preservation: Do They Go Together?, Proceedings of Archiving 2007 (2007).

[9] Rog, J., van Wijk, C., Evaluating File Formats for Long-term Preservation, National Library of the Netherlands, (2008). http://www.kb.nl/hrd/dd/dd_links_en_publicaties/publicaties/KB_file_format_evaluation_method_27022008.pdf

[10] Rosenthal, D. S., Reich, V., Permanent Web Publishing (2000). http://lockss.stanford.edu/freenix2000/freenix2000.html

[11] Santa-Cruz D., Ebrahimi T., Askelof J., Larsson M., Christopoulos C.A.,JPEG 2000 still image coding versus other standards, Proceedings of SPIE, Vol. 4115, pp. 446-454 (2000).

[12] Schroeder B., Gibson G.A., Disk Failures in the Real World, 5th USENIX Conference on File and Storage Technologies, San Jose, CA (2007). http://www.cs.cmu.edu/~bianca/fast07.pdf

[13] Thaller, M., Preserving for 2016, 2106, 3006. Or: Is there a life for an object outside a digital library? Presentation held at ‘DELOS Conference on Digital Libraries and Digital Preservation’, Tallin, Estonia (2006). http://www.hki.uni-koeln.de/events/tallinn040906/tallinn040906.ppt

[14] XCL. Extensible Characterisation Language (2007). http://planetarium.hki.uni-koeln.de/XCL/

Author Biography

Volker Heydegger is a PhD researcher at the University of Cologne, department of Computer Science for the Humanities (Historisch-kulturwissenschaftliche Informationsverarbeitung). Since 2004 he is working in European research projects in the field of digital preservation, currently for PLANETS. His research focus is on characterisation of file format content for digital preservation and on preservation aspects of file formats.