Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-29 (2016) 253–258

File Formats - Characterization and Validation

Lavdërim Shala *; Ahmet Shala**

**University of Freiburg / Technical Faculty-Computer Science, Freiburg, DE 79110 Germany*
e-mail: shalal@informatik.uni-freiburg.de .
***University of Prishtina / Faculty of Mechanical Engineering, Prishtina, XXK 10000 Kosovo*
e-mail: ahmet.shala@uni-pr.edu

Abstract: Nowadays, most of the information is stored digitally. Digital information is from a high level of view it is just an array of bits. In order to figure out its real meaning special software which interprets it is required. Therefore, if by evolution of technology this software cannot be executed anymore there is potential risk that also the data interpreted by it becomes not useful. The goal of Digital Preservation is to stop occurrences of such phenomenon. Data is commonly stored in files each file has a specific format or structure, by knowing it user can figure out the real meaning of raw data stored in the file as an array of bits. Digital Preservation considers valid file format as a perquisite for file to be in usable form, with valid is meant that a specific file is structured conform its declared file format. In this paper we throw a spotlight on the accuracy and capability of these file validation tests. Therefore, we present some open source software which are able to automatically identify and verify the file format. We focus more on file types they can identify, and how they work in large scale data sets.

Keywords: File Format, Digital Preservation, JHove, Droid, Exiftool.

1. INTRODUCTION

This Evolution of computer systems in the last two decades, especially the wide usage of them all over the world by people of every profile has impacted the way how people store information. In the time when this paper is written (2016) people prefer to store their information digitally in their electronic devices [1]. The information itself in the electronic storage is saved in objects called files where each file consists an array of bits 0 and/or 1. This is done so independently from the information real meaning no matter whether it is text, photo, audio or something else it is stored digitally as an array of bits. On the other hand, there are built computer software which make these arrays of bits meaningful by converting these bits to the real interpretation of them and vice-versa. Without these software the information is meaningless and it is just an array of bits.

One important characteristic of the world of computers is the continuous evolving and the wide range of computer software and hardware. Currently, there is a high number of operating systems, and software to manipulate different kind of information e.g. text being offered to be used by everybody. In addition, this number is rapidly increasing, and, moreover, current software is regularly updated. On the other hand also the hardware implementation and architecture is occurring changes towards better performance. In summary, this variety and evolution of computer systems is leading to multiple ways of storing and interpreting the digital data. This leads to cases like the one mentioned above when the user has still the information, but is unable to figure out its real meaning. This can be due to the fact that the way this information is structured is not supported by the current software and hardware that he uses which can be an updated successor of the one used to create the data or a completely new one. Here Digital Preservation comes into consideration. The main goal of it is keep the digital information usable over the time, therefore it tries to make it neutral against continuous technological evolution. Digital Preservation tries to find methods, strategies, and activities help it achieve its goals [1].

In order to view the content of a digital information (file), the user should know which software to use to open the file. On the other hand, in order to show the user the real meaning of the file the software should know how the array of bits which is inside the file is structured, where in the array specific information needed by software is noted. In order to solve these two issues there exists a concept called file format.

File format defines the structure how information is ordered in the array of bits stored in the digital storage (disk). Normally, in operating systems the file format is declared as an extension at the end of the file name it is preceded by a dot. Therefore, the identification of file format seems to be an easy job. But, in contrast, it is not as easy as it looks since the format extension can be modified at any time by user. Therefore, from the software point of view, it is not only important to figure out what format does the file have by looking at the extension, but it also important that the file itself represents a valid structure of the format it is thought to be otherwise if the file does not represent it, the file is considered to be not useful [2].

Due to the fact that the goal of the Digital Preservation is to keep the file in useful form it also treats the problem mentioned above by providing mechanisms which identify and verify the format of a file. Once a file is verified by such mechanism it is guaranteed that it is structured in the proper way that a software dedicated to open its format can open it. In this paper we put a light on the way how file format is treated by Digital Preservation.

This paper is organized as follows: In the second chapter we briefly explain the analysis that that Digital Preservation

2405-8963 © 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.11.062

conducts towards file format in order to proof that it is usable. Next, in the third chapter we introduce file format registries which are public databases where file format specification is stored. In the fourth chapter we present a brief description of three tools named JHOVE, DROID, and Exiftool which use the information in databases explained in the previous chapter to identify, retrieve characteristics, and validate file format of files for different file formats whose specification is available in these databases. Next in chapters five and six we present my experiment done against a data set of 927 files using tools mentioned above. In chapter five for each tool individual results towards analysis on a large scale data set are presented. Next in the sixth chapter we try to merge the results from each individual tool from previous chapter. Here we throw a spotlight on the conflicting results from different tools. Moreover, in this chapter we present a framework proposed by K., B.,[3] for merging results from different tools including also tools presented in chapter four, this framework tries to solve conflicts at its best. Finally, in the last chapter we make a brief discussion about current situation into file format identification, characterization, and verification, gives suggestions what is important in the future.

2. DIGITAL PRESERVATION APPROACH TOWARDS FILE FORMAT

All the work done by digital preservation in order to ensure that the file is useful in terms of its format can be grouped into three processes [1]:

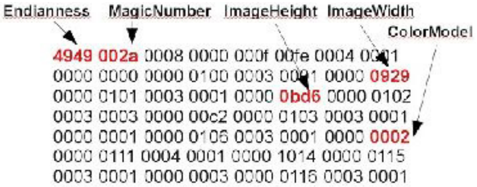
- * Identification - The process of identifying what format the given file is likely to be.
- * Characterization - The process of extracting specific information from the file.
- * Verification - The process of verifying if the file matches the structure given by specification.

2.1 Identification

Identification as described above has to do with determining what format a file is likely to have. This can be in interest of human or software, to achieve it there are two approaches. The first one relies on fact that when a file is stored in file system in its name it is also noted the format as an abbreviation called extension. More precisely extension is the text after the last dot ('.') [4]. Therefore, a file named in the file system as foo.txt is declared to have the file format text. The problem with this approach is that these extensions are not by default verified by system. As a result, a user can store a file to disk and nobody stops him from declaring any file format to this file. Therefore, by just having an extension declared in file systems it not guaranteed that the content of the file matches the structure of the declared file format so a file named foo.txt is not guaranteed to have some text stored inside but it is likely to be so. To solve this issue a second approach towards identification is used. This more advanced method ignores the fact that file format is declared and tries to determine the file format by analysing the structure of the file, and comparing it if it matches any of the known file format specifications [1]. These known file format specifications are stored in so called File Registries which are databases which consist structural specification for different file formats, they are discussed later in this paper. Also in order to analyse the structure it is needed to extract some information from the file this is known as Characterization and is described below in this chapter [2].

2.2 Characterization

Characterization is the process of extracting specific characteristics from the file. This extracted information is used for two main purposes. First it is used by software to check if the file consists the needed information to match the file format it is thought to be, this is called file format validation and is explained later in this chapter, Second use of the extracted data after a validation is proved is to use them to construct the meaningful interpretation from the file, and present it to user [1]. Extracting specific information from files is computationally expensive since each file should be loaded in memory and be analysed. Therefore, to solve this problem digital libraries where huge amounts of data is stored usually do this extraction only once and then save the extracted data apart from the original file but in the same repository so when another extraction is needed this saved extracted data is read instead of another extraction from the original file which is more computationally expensive \cite {preservation Thesis}.Ford [2] explains characterization by an example which we will also show below. He takes a well-known image format named TIFF which is widely used to store image files. The extension for TIFF files is .tif, below in Fig. 1 the raw presentation of some part of a random TIFF file where the array of bits is noted in hexadecimal form.



```
4949 002a 0008 0000 000f 00fe 0004 0001
0000 0000 0000 0100 0003 0001 0000 0929
0000 0101 0003 0001 0000 0bd6 0000 0102
0003 0003 0000 00c2 0000 0103 0003 0001
0000 0001 0000 0106 0003 0001 0000 0002
0000 0111 0004 0001 0000 1014 0000 0115
0003 0001 0000 0003 0000 0116 0003 0001
```

Fig. 1. Hexadecimal Values of a TIFF file [2]

The interesting data to be extracted from the file is noted with red colour in the figure above. In order to show the image a TIFF viewer software first validates whether the file he is asked to open is structured conform the TIFF file format. In this case the TIFF format specification notes that the first two bytes of the TIFF file note the endianness and can have values 49 in case of little endian or 4d in case of big endian. Furthermore, the specification says that the third byte in case of little endian should have value 00 and the fourth one 2a. The first four bytes of a TIFF file together make what is called the magic number. By checking the magic number if the values, and their position is conform the specification the software determines the file format. In addition, in the picture there is some more data marked red, from the information that they are carrying (Image height, Image width, and Colour Model) one can conclude that this data is used by the software to interpret the true meaning of the file to user [2].

2.3 Verification

Verification is the third step that Digital Preservation conducts towards a file format. This step tries to figure out whether a given file is structured in compliance with its file format specifications, and whether a software designed to open this file format will be able to successfully open it. By successfully is meant to interpret the real meaning of the file. One strict approach might be that the achievement of both goals of verification is mutual inclusive. Therefore if one goal is achieved then the other is also achieved. But in practice this assumption does not stand, and by not standing it creates some trouble towards verification. This comes due