Digital Media Preservation Based on Change History

Byoung-Dai Lee[1], Sungryeul Rhyu[2], Kyungmo Park[2], Jaeyeon Song[2]

[1]Department of Computer Science, Kyonggi University, Suwon, Korea
blee@kug.ac.kr
[2]Multimedia Global Standard Group, Samsung Electronics, Co., Ltd., Suwon, Korea
{suzz.rhyu, kyungmo.park, jy_song}@samsung.com

Abstract. Recently, with accessible media, tools and applications, a user can create multimedia files without professional knowledge. From the perspective of digital preservation and version management, however, most existing multimedia file formats have several shortcomings. For instance, there is no support for recovering damaged multimedia using its derived files or systematic comparison of content among derived files. In this paper, we propose a method to preserve multimedia files in such a way that individual files maintain preservation information along with multimedia data to keep track of change history. In order to show the feasibility of our approach, we extended the ISO base media file format that various well-known media formats such as MP4 are based on.

Keywords: ISO base media file format, Context and Provenance Information, Digital Preservation, Multimedia.

1 Introduction

With accessible media, tools and applications, a user can create media files without professional knowledge. Existing multimedia file formats focus mainly on effective representation of multimedia content and efficient packaging of data. As a result, no information for media preservation, especially context and provenance information [1], is included in the multimedia files. Context and provenance information describe the relationship of the multimedia file and/or the contents of the file to other information objects. Examples of information include the origin of the multimedia file, any changes that may have taken place since it was created, and how it relates to other multimedia files. Therefore, multimedia files without embedded context and provenance information can have several shortcomings. First, when a multimedia file is damaged, it is not possible to recover the file using those multimedia files that have been derived from it. Replicating the multimedia file to several possible locations is the only solution to address the problem. If derived files can also be used for restoration, preservation of multimedia files can be further enhanced by employing both strategies. Second, there is no systematic way to compare multimedia content stored in related files. For instance, in order to find differences between a multimedia file and its modified version, a user must watch both programs and spot the differences by eye.

To address the abovementioned shortcomings, we present a method to preserve multimedia files in such a way that individual files maintain preservation information based on change history to describe relationships to other multimedia files. Our approach is applicable to a wide range of multimedia file formats as it does not need to consider the internal semantics of individual formats due to bitstream level manipulation of files. In order to show the feasibility of our approach, we extended the ISO base media file format [2]. ISO base media file format defines a general structure for time-based multimedia files. Several well-known formats such as MP4 and 3GP are extended from it.

The paper is organized as follows: Section 2 presents the related work. Introduction to ISO base media file format is presented in Section 3. Section 4 describes the proposed media preservation method in detail. Finally, Section 5 gives the conclusions and future work.

2 Related Work

The OAIS reference model [1] is a conceptual framework for long-term preservation of digital information and; as such, it addresses all the major activities of an information-preserving archive in order to define a consistent and useful set of terms and concepts. Many existing digital preservation systems are based on the OAIS reference model. In this paper, however, we focus on standard-based file formats to deal with media preservation and version management.

MPEG-A Professional Archival Application Format (PA-AF) [3] provides a standardized packaging format for digital files. It specifies metadata formats 1) to describe the original structure and attributes of digital files archived in a PA-AF file, 2) to describe context information related to a PA-AF file and digital files archived in it, and 3) to describe necessary information to reverse the pre-processing process applied to digital files prior to archiving them in a PA-AF file [4]. It also specifies a file format for carriage of the metadata formats and digital files.

Self-contained Information Retention Format (SIRF) [5] is a logical container format for a storage subsystem appropriate for long-term storage of digital information. A SIRF container consists of three components: a magic object; preservation objects; and a catalog. The magic object identifies whether this is a SIRF container and its version. The preservation object is a digital information object that includes the raw data to be preserved and additional embedded or linked metadata. The container may include multiple versions of preservation objects and multiple copies of each version. The catalog contains metadata needed to make the container and its preservation objects portable into future storage systems without relying on functions external to the storage subsystem.

Packaging is the common characteristic of the abovementioned approaches. That is, different versions of a media file are packaged into a container and metadata is used to provide information to locate and relate individual media files within the container. In particular, metadata is present as an independent entity from the stored media files. This approach is flexible in that it requires no knowledge of the internal semantics of the media files to be preserved. For the same reason, only limited sets of relationship information are provided.

3 Overview of ISO Base Media File Format

ISO base media file format [2] was specified as ISO/IEC 14496 (MPEG-4 Part 12). It defines a general structure for time-based multimedia files, such as audio and video, that facilitates interchange, management, editing and presentation of the media. Due to its flexibility and extensibility, ISO base media file format is used as the basis for other formats in the family, such as MP4 and 3GP.

The ISO base media file format consists of three logical components: header, metadata, and media data. The header contains the general information for the media contained in the file. Examples of the information are a content identifier, content providers, and the creation date of the content. If the media file consists of multiple tracks, each of which represents a timed sequences of media (e.g., frames of video), the header also contains the track configuration information. The primary information contained in metadata includes the information about the placement and timing of the media components and the profile information required for decoding the media components. A media component, called samples, represents the timed unit within each track; it might be a frame of video or audio. Media data contains the actual media data. It may be in the same file or can be in other files.

Files conforming to the ISO base media file format are formed as a series of objects, called boxes, which are defined by a unique type identifier and length. Therefore, all data is contained in the boxes and there is no other data within the file. For instance, individual tracks are represented by the track boxes. As a container box, the track box only contains several sub boxes for storing the track header information, the layout of the media data represented by the track, and the time ordering of the media. The sub boxes, in turn, may contain other sub boxes, if necessary.

4 A Method for Media File Preservation

4.1 Change History Based Media File Preservation

In order to provide capabilities to recover a damaged multimedia file using its derived files and to support systematic comparison of multimedia content among derived files, the approach that we have taken is that each multimedia file contains change history and the minimum data to reproduce the source file from which the multimedia file is derived. The integral parts of our approach are as follows: 1) Multimedia files are compared in the bitstream level and the differences are maintained as change history. Therefore, there is no need to consider the internal semantic of individual multimedia file formats. 2) Each multimedia file embeds as preservation information only the direct relationship to its source file that it has modified. Therefore, the file size can be significantly reduced. Other related multimedia files can be restored by incremental restoration. In other words, an older version of a multimedia file restored using a current multimedia file also contains preservation information for its multimedia file. This process continues until the designated version of the multimedia file is restored.

Fig. 1 illustrates the differences in bitstream level between a multimedia file (e.g., Version #1) and a modified version (e.g., Version #2). For instance, the content of Version #1 corresponding to the block from file offset 0x50 to file offset 0xA0 is not present in Version #2 due to deletion. Similarly, the content from file offsets 0xA0 to 0xD0 in Version #1 has been updated and is located in the block from 0x70 to 0xA0 in Version #2. Two new blocks from offsets 0x10 to 0x30 and from 0xC0 to 0xE0 are newly inserted into Version #2.

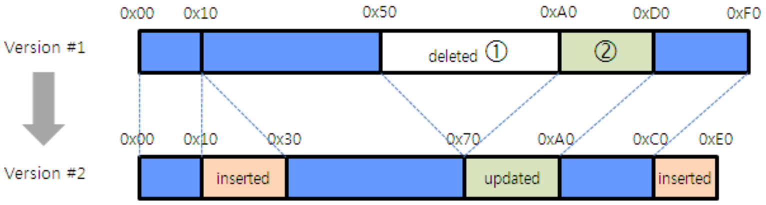
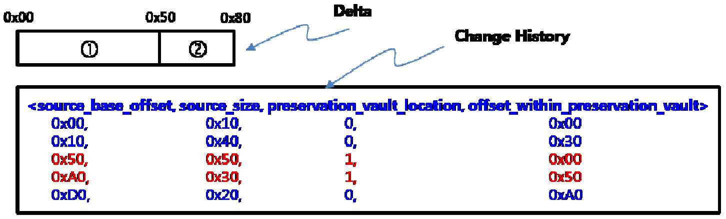


Fig. 1. An example of bitstream level comparison of multimedia files.



<source_base_offset,	source_size,	preservation_vault_location,	offset_within_preservation_vault>
0x00,	0x10,	0,	0x00
0x10,	0x40,	0,	0x30
0x50,	0x50,	1,	0x00
0xA0,	0x30,	1,	0x50
0xD0,	0x20,	0,	0xA0

Fig. 2. Required metadata and media data for preservation.

Fig. 3 shows the metadata for change history and minimum media data, which we call the delta, to be embedded into Version #2. The delta contains media data that is required to reproduce Version #1 from Version #2. Therefore, the blocks that are not present in Version #2 due to deletion or update are stored in the delta. The change history contains four different types of information required to restore the source media file: 1) "source_base_offset"_and "source_size" represent the file offset and size of the data block starting from the offset in the source multimedia file; 2) "preservation_vault_location" and "offset_within_preservation_vault" represent where bitstream data should be retrieved to restore the data block indicated by "source_base_offset" and "source_size". Note that if the contents of the data blocks of Version #1 were not modified, then those blocks would be present in Version #2, potentially with different locations. On other hand, if the data blocks of Version #1 were deleted or modified, those blocks would not be present in Version #2; therefore, they must be stored in the delta. 3) "preservation_vault_location" indicates which data source must be used for data retrieval. For instance, the second row in the change history tells that the data block corresponding to file offsets 0x30 to 0x70 of Version #2 must be used to restore the data block corresponding to 0x10 to 0x50 of Version #1. The fourth row of the change history tells that the data block corresponding to block offsets 0x50 to 0x80 in the delta must be used to restore the data block corresponding to file offsets 0xA0 to 0xD0 of Version #1.

4.1 Implementation

In order to implement our approach, we extend the ISO base media file format. Fig. 3 shows additional boxes proposed in this paper, **prsv (Preservation Box)** and **pdat (Preservation Data Box)**, and their relationship to existing boxes. The prsv box is located in the top-level and contains the pdat as its sub-box. Table 1 shows the detailed syntaxes for both boxes.

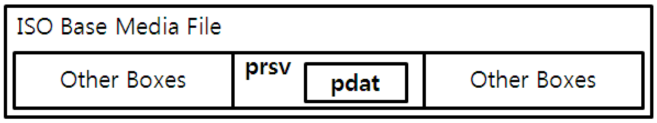


Fig. 3. Logical view of ISO base media file extension.

Both boxes are conformant to the box format specified by ISO/IEC 14496 standard. Explanations on data types used in the paper can be found in [2]. The primary role of the prsv box is to indicate whether the current multimedia file contains preservation information for its source file, whereas the pdat box contains the actual data for the change history and the delta required to restore the source file. Fig. 4 illustrates an example of the ISO base media file after preservation information is embedded.

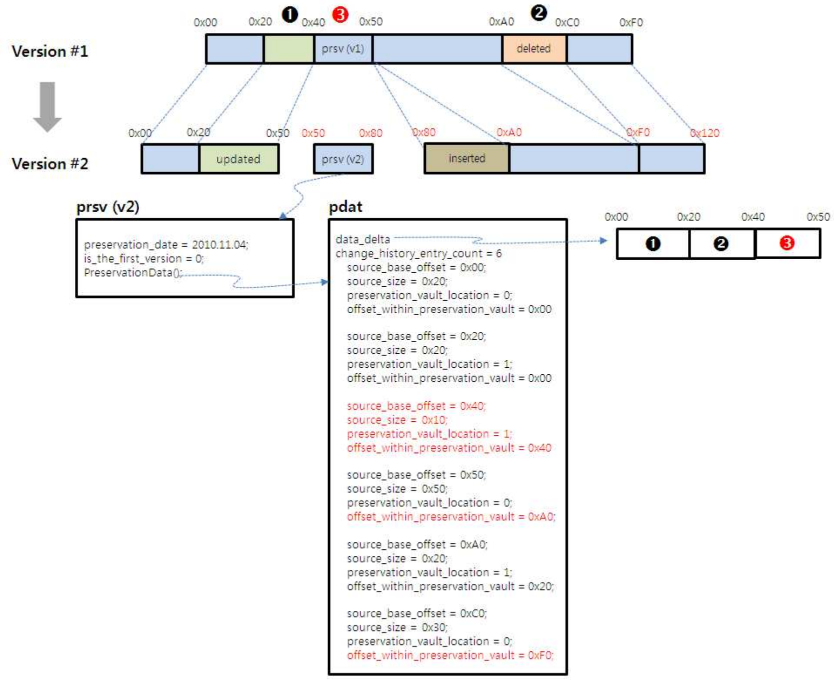
Table 1. Detailed syntaxes for proposed boxes.

Box	Syntax

prsv	`aligned (8) class Preservation extends Box(‘prsv’) {`
	`    unsigned ing (32) preservation_date;`
	`    bit (1) is_dirty;`
	`    if (is_dirty = ‘1’)`
	`        PreservationData();`
	`}`

pdat	`aligned (8) class PreservationData extends Box(‘pdat’) {`
	`    bit (8) data_delta[];`
	`    unsigned int (4) offset_size;`
	`    unsigned int (4) length_size;`
	`    unsigned int (4) change_history_entry_count;`
	`    for (int i = 0; change_history_entry_count; ++i) {`
	`      unsigned int (offset_size*8) source_base_offset;`
	`      unsigned int (length_size*8) source_size;`
	`      unsigned int (8) preservation_vault_location;`
	`      unsigned int (offset_size*8)`
	`                    offset_within_preservation_vault;`

```
        }
}
```



```
preservation_date = 2010.11.04;
is_the_first_version = 0;
PreservationData();
```

```
data_delta
change_history_entry_count = 6
   source_base_offset = 0x00;
   source_size = 0x20;
   preservation_vault_location = 0;
   offset_within_preservation_vault = 0x00

   source_base_offset = 0x20;
   source_size = 0x20;
   preservation_vault_location = 1;
   offset_within_preservation_vault = 0x00

   source_base_offset = 0x40;
   source_size = 0x10;
   preservation_vault_location = 1;
   offset_within_preservation_vault = 0x40

   source_base_offset = 0x50;
   source_size = 0x50;
   preservation_vault_location = 0;
   offset_within_preservation_vault = 0xA0;

   source_base_offset = 0xA0;
   source_size = 0x20;
   preservation_vault_location = 1;
   offset_within_preservation_vault = 0x20;

   source_base_offset = 0xC0;
   source_size = 0x30;
   preservation_vault_location = 0;
   offset_within_preservation_vault = 0xF0;
```

Fig. 4. Example of an ISO base media file with preservation information.

Acknowledgments. This work was supported by Samsung Electronics, Co., Ltd.

References

1. ISO 14721:2003, Pink Book, Issue 1.1, CCSDS 650.0-P-1.1, Reference Model for an Open Archival Information System (OAIS), CCSDS (2009)
2. ISO/IEC 14496-12:2008(E), Information Technology – Coding of Audio-Visual Objects-Part 12: ISO Base Media File Format, ISO/IEC, (2008)
3. ISO/IEC 23000-6:2009(E), Information Technology-Multimedia Application Format (MPEG-A) – Professional Archival Application Format, ISO/IEC, (2009)
4. Harada, N., Hendry, Sabirin, H., Kim, M., Kamamoto, Y., Moriya, T., Introduction of MPEG-A Professional Archival Application Format (PA-AF), In: 1st International Digital Preservation Incomparability Framework Symposium, Dresden (2010)
5. Self-contained Information Retention Format (SIRF) – Use Cases and Functional Requirements, SNIA (2010)