(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
3 November 2005 (03.11.2005)

PCT

(10) International Publication Number
WO 2005/104127 A1

(51) International Patent Classification[7]: G11B 27/00

(21) International Application Number:
PCT/EP2005/004377

(22) International Filing Date: 22 April 2005 (22.04.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
04009912.9 27 April 2004 (27.04.2004) EP

(71) Applicant *(for all designated States except US)*: **THOMSON LICENSING S.A.** [FR/FR]; 46 Quai A. le Gallo, 92100 Boulogne-Billancourt (FR).

(72) Inventors; and

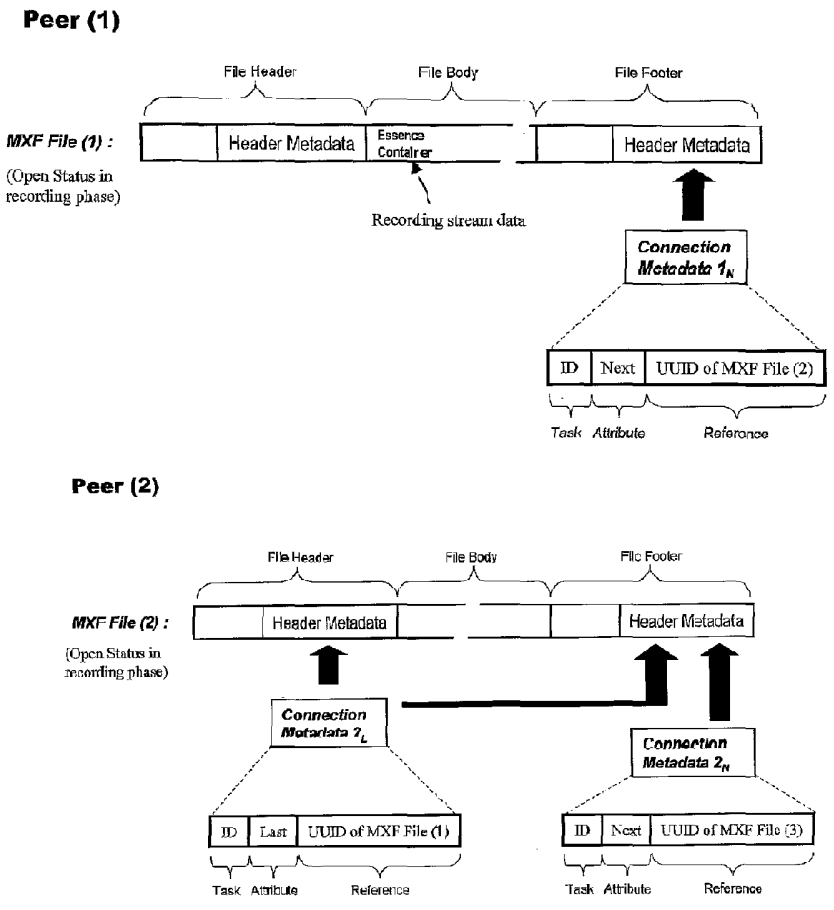
(75) Inventors/Applicants *(for US only)*: **LI, Hui** [DE/DE]; Haltenhoffstr. 221, 30419 Hannover (DE). **BLAWAT, Meinolf** [DE/DE]; Sutelstrasse 54 C, 30659 Hannover (DE). **KLAUSBERGER, Wolfgang** [DE/DE]; Am Uhrturm 26, 30519 Hannover (DE). **KUBSCH, Stefan** [DE/DE]; Im Ohr 2, 31559 Hohnhorst (DE). **HEPPER, Dietmar** [DE/DE]; Wernigeroder Weg 41, 30419 Hannover (DE).

(74) Agent: **RITTNER, Karsten**; Deutsche Thomson-Brandt GmbH, European Patent Operations, Karl-Wiechert-Allee 74, 30625 Hannover (DE).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

[Continued on next page]

(54) Title: METHOD AND SREAMS IN DISTRIBUTED STORAGE SYSTEMS



(57) Abstract: A distributed storage system able to support real-time re-cording of streaming d ata shall be provided. Therefore, on the basis of MXF files a data stream is pac ked into at least a first MXF file (l) and a second MXF file (2). First connection data 1-N)are inserted into the first MXF file (l), wherein this first connection data (1-N) points to the second MXF file (2) Fur-thermore, second connection data (2-L) are inserted into the sec-ond MXF file (2), wherein the second connection dat a (2-L) points to the first MXF file (1). These connection data inserted as meta-da to into the MXF files enable both seamless real-time stream recording and stream -able data playback.

WO 2005/104127 A1

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

RECORDING DATA SREAMS IN DISTRIBUTED STORAGE SYSTEMS

Technical field

The present invention relates to a method or a device for recording a data stream on a distributed storage system. Furthermore, the present invention relates to a method or device for playing back such a data stream recorded on a distributed storage system.

Background

Based on the peer-to-peer (P2P) technology, a distributed storage system can make use of all storage capacity of connected peers, i.e. nodes, to store data, such as duplicating important data in different peers, or splitting large-size data and sharing or storing them in different peers.

The present document relates to the specific Material Exchange Format (MXF). MXF is an open file format targeted at the interchange of audio-visual material with associated data and metadata. It has been designed and implemented with the aim of improving file based interoperability between servers, workstations and other content creating devices.

Basic MXF structure

As shown in Fig. 1 taken form the MXF standard specification, the MXF Format consists of a File Header, a File Body and a File Footer.

The File Header contains Header Metadata, which provides information about the file as a whole, including labels for the early determination of decoder compliance.

The File Body comprises picture, sound and data essence stored in Essence Containers. Essence Containers from different tracks may be interleaved or separated as shown in Fig. 2. Body Partition Packs of File Body may optionally contain repetition of Header Metadata.

The File Footer terminates the file and may optionally contain Header Metadata. The File Footer may further include some information not available at the time of writing the File Header (such as the duration of the file). In certain specialized cases, the File Footer may be omitted.

The File Header is designed to be small enough that it can easily be isolated and sent to a microprocessor for parsing. The bulk of the file will usually be the File Body - this is the picture, sound and data essence. The File Footer provides capacity to put the Header Metadata at the end of the file. The reason is: in certain applications, such as recording a stream to an MXF file, there will be Header Metadata values that will not be known until the recording is finished. The File Footer provides a mechanism for doing this. It also provides clear indication that the file has terminated.

The basic MXF structure supports embedded metadata throughout the MXF file.

MXF interoperation with stream Interfaces

MXF files will often be processed in streaming environments. This will include streaming to and from videotapes and data

tapes, and transmission over unidirectional links or links with a narrow-band return-channel.

Sequential writing is necessary when the source or the link or the destination operates only in streaming mode. Random access writing is permissible before or after data transfer, for example, to optimize downstream access performance.

MXF's Operational Patterns have a special qualifier bit that indicates that the file has been created for streaming.

MXF files may be directly created from standardized formats such as MPEG2 systems and elementary streams, AES3 data streams (Audio Engineering Society Standard) and Digital Video (DV) DIF packet streams. These formats may be mapped from one of several real-time interfaces such as SMPTE 259M (Serial Digital Interface, SDI), SMPTE 305M (Serial Digital Transport Interface, SDTI), SMPTE 292M (High-Definition Serial Digital Interface, HD-SDI), or transport interfaces with real-time protocols such as IEEE-1394, ATM, IEEE802 (ethernet), ANSI Fibre Channel and so on.

When a streaming file is captured, a File Header is created and the essence is KLV (Key Length Value) wrapped on the fly. The data rate increases due to the KLV wrapping and addition of headers. Real time streaming devices must ensure that any buffering requirements of a streaming interface are catered for with this change of data rate.

The basic MXF structure supports stream recording and playback.

File splitting requirement of a Distributed Storage system

The behaviour of a distributed storage home network system will in the future probably be characterized by a convenient

automatically "storekeeping" of all kind of data. The user will not know in any case where the content is stored. So this kind of distributed storage system can be seen as a monolithic storage block and the system has to look by its own for a storage device that matches capacity of free storage without any administrative effort by the user, and with a transparent behaviour for the requiring application.

The distributed storage system has to deal with different lengths of application packets coming along with the streaming data of different standards (e.g. MPEG-2, DV, etc.). When streaming data has to be handled by the distributed storage system, it is not generally possible to get the size of the data in advance. So seamless real-time splitting of data could be necessary when the maximum capacity of a peer, or node, will be reached. This invention proposes methods for the processing of streaming data in the distributed storage environment for the record and playback condition.

Invention

It is the object of the present invention to provide a distributed storage system which is able to support real-time recording of streaming data.

According to the present invention this object is solved by a method for recording a data stream on a distributed storage system by packing said data stream into at least a first MXF file and a second MXF file, inserting first connection data into said first MXF file, said first connection data pointing to said second MXF file, and inserting second connection data into said second MXF file, said second connection data pointing to said first MXF file.

Furthermore, there is provided a corresponding device for recording a data stream on a distributed storage system in-

cluding packing means for packing said data stream into at least a first MXF file and a second MXF file, data processing means for inserting first connection data into said first MXF file, said first connection data pointing to said second MXF file, and for inserting second connection data into said second MXF file, said second connection data pointing to said first MXF file.

Simultaneously, there is provided a method for playing back a data stream recorded on a distributed storage system by reading connection data from a first MXF file, said connection data pointing to a second MXF file, building an MXF file chain on the basis of said connection data and playing back the essence data of said first MXF file and said second MXF file according to said MXF file chain.

Moreover, there is provided a corresponding device for playing back a data stream recorded on a distributed storage system including reading means for reading connection data from a first MXF file, said connection data pointing to a second MXF file, data processing means for building a MXF file chain on the basis of said connection data and playing means for playing back the essence data of said first MXF file and said second MXF file according to said MXF file chain.

The advantage of such methods and devices is, that while stream recording, if a peer reaches its storage-capacity, the input stream may be rerouted to another peer or other peers, in order to record the input stream continuously. The file splitting can be performed in real-time. Consequently, the present invention enables such seamless real-time stream recording and stream-able data play back for MXF files.

Preferably, the inventive method of recording a data stream further includes the step of inserting third connection data into said second MXF file, wherein said third connection

data pointing to a third MXF file. Thus, a current MXF file points to the last MXF file and to the next MXF file.

The step or steps of inserting connection data into the MXF file may be performed while terminating said recording of an MXF file. At that moment the last MXF file and the next MXF file are available, or known respectively.

While recording, the data stream may be split into a specific number of MXF files, wherein the specific number is not known in advance, and connection data pointing to all of these MXF files may be stored in each of said specific number of MXF files. This enables to obtain the complete MXF file chain by reading only one MXF file.

The connection data may be inserted as metadata into the header and/or the footer of an MXF file. If the connection data are stored both in the header and the footer, the MXF file can be split later without problems.

When MXF files are in Open Status, in order to built up the MXF file chain the step of reading connection data may be repeated for each MXF file of said MXF file chain. Thus, the MXF file chain can be built up step by step.

When MXF files are in Closed Status, i.e. finished, the step of reading connection data needs to be performed only once for one MXF file chain, wherein connection data pointing to all MXF files of said MXF file chain are included in each of said MXF files of said MXF file chain. In this case it is only necessary to read any one of the MXF files in order to obtain the complete MXF file chain.

Preferably, the connection data include task data, attribute data and reference data. The task data contains a task ID value, the attribute data contains a status value like "next" (for a following MXF file) or "last" (for a preceding

MXF file) and the reference value contains an ID value, e.g. UUID, of the other MXF file. By inserting these three different data as connection metadata into the MXF header, it is possible to realize seamless real-time stream splitting for recording and play back.

Drawings

The present invention will now be described in more detail in connection with the attached drawings showing in :

Fig. 1 an overall data structure of a simple MXF file according to the standard;

Fig. 2 an overall data structure of an MXF file with some optional components;

Fig. 3 a scheme of splitting a metadata according to the present invention;

Fig. 4 the insertion of connection metadata to an MXF file 1 after stream recording on a peer 1 of a distributed storage system;

Fig. 5 the insertion of connection metadata to an MXF file 2 before and after stream recording on a peer 2 of the distributed storage system;

Fig. 6 the insertion of connection metadata to an MXF file k before and after stream recording on a peer k [k= 2,3, … (n-1)] of the distributed storage systems;

Fig. 7 the insertion of connection metadata to an MXF file n before and after stream recording on a peer n;

Fig. 8 the connection metadata of the MXF file 1 in a closed and completed status;

Fig. 9 the connection metadata in the MXF file k in the closed and completed status;

Fig. 10 the connection metadata in the MXF file n in the closed and completed status; and

Fig. 11 connection metadata packages in an MXF file.

Exemplary embodiments

In the following, preferred embodiments of the present invention are described in more detail.

Setting up special "Connection Metadata" into MXF file

In order to realize real-time file splitting for stream recording and playback, a special "Connection Metadata" is created. This Connection Metadata is inserted in the Header Metadata of the MXF file. The Header Metadata may be located in the File Header, the File Body and/or the File Footer.

As shown in Fig. 3, a Connection Metadata according to the present embodiment consists of at least three parts: Task, Attribute and Reference. These parts may e.g. be fields.

The Task field contains a task ID value. For example, each stream recording action has an own task ID. The Attribute field value can represent at least two statuses: "next" and "last". The "next" attribute means that the current MXF file is followed by a subsequent MXF file, because of the split

stream recording. The subsequent MXF file can be found by the UUID (Universal Unique Identifier) included in the Reference field. The "last" attribute means that the current file follows a preceding MXF file and is a part of a split stream. The preceding MXF file can be found by the UUID included in the Reference field.

For representing different kinds of splitting scenarios of stream recording, a Header Metadata may contain multiple Connection Metadata. The Connection Metadata may be firstly generated by both the MXF File Header and the File Footer.

In the following, application scenarios of the Connection Metadata are pointed out in detail.

Recording a stream into a sequence of MXF files by using Connection Metadata

To handle concatenated stream objects in a distributed storage system, all associated MXF files will be labeled by a UUID, and the complete sequence of connected streamable objects (MXF files) can be generated with the help of Connection Metadata embedded in each of the MXF files.

One requirement of the distributed storage system is the self-organizing of all functions that are not significant for the user. For these tasks (e.g. file splitting when the capacity of one node is achieved), a self-controlled exchange of control messages between the nodes is necessary.

In the distributed storage system an input stream shall be real-time recorded continuously into n MXF files of n peers. Connection Metadata in all these MXF files have the same task ID value. Every peer communicates with each other through control metadata designed by the distributed storage

system. The recording procedure is described in detail in the following (compare Fig. 4 to 8):

1. At first, the input stream is packed into MXF file (1) in peer (1) of the distributed storage system.

2. Before peer (1) reaches its maximum storage capacity, peer (1) should send a Control Message to other peers to ask if any peer can continue to record the stream.

3. If peer (2) replies that it can continue to record the stream, peer (1) will tell the receiving device of the input stream to prepare for rerouting the incoming stream to peer (2). Peer (2) will be noticed to prepare for recording the input stream.

4. As shown in Fig. 4, while terminating the current stream recording, peer (1) generates a new Connection Metadata (1_N) and inserts it into the Header Metadata in the File Footer of the current MXF file (1). In this case, the File Footer of MXF file (1) must include Header Metadata, though usually the Header Metadata in MXF File Footer is optional. The current stream recording task ID is set as the Task value of the Connection Metadata (1_N). The Attribute value of the Connection Metadata (1_N) is set to be "next", and the Reference contains the UUID of MXF file (2) in peer (2), which records the following stream.

5. As shown in Fig. 5, when peer (2) starts to record the incoming stream, at first it generates the File Header of the MXF file (2) and inserts a Connection Metadata (2_L) in the Header Metadata of the File Header. The value of the Attribute of the Connection Metadata (2_L) is "last" and the UUID of MXF file (1) is put in its Reference.

6. If peer (2) will reach its maximum storage capacity, the same action as described in steps 2 to 4 are repeated.

The input stream will be rerouted to another peer after terminating the recording task in peer (2), a Connection Metadata (2_N) will be inserted in the Header Metadata of the File Footer of peer (2). The Attribute of Connection Metadata (2_N) is "next" and its Reference includes the UUID of the next MXF file (3), see Fig. 5.

7. Further peers (k) may repeat steps 5 and 6 until the whole stream recording processing is finished with MXF file (n) of peer (n), wherein k= 2,3, …, n-1. As shown in Fig. 6, in the Header Metadata of the File Header, all previous "last" Connection Metadata ((2_L) …(k_L)) are accumulated and inserted into the Header Metadata according to the natural order of created MXF files. This structure enables each MXF file (k) to know all its prior concatenated MXF files.

8. When the whole stream recording processing will be finished in peer (n), as shown in Fig. 7, the MXF file (n) generates a Header Metadata in the File Header, which accumulates all "last" Connection Metadata ((2_L) …(n_L)), and arranges them in their natural order. Before terminating the whole stream recording task, the same sequence of Connection Metadata ((2_L) …(n_L)) may be optionally repeated in the Header Metadata of the File Footer of the MXF file (n).

According to the above described stream recording processing, the input stream is recorded into a sequence of MXF files (1,2,..n). These real-time generated "last" and "next" Connection Metadata in the Header Metadata of the File Header or the File Footer provides enough messages for playing back the sequence of MXF files.

Finalizing Connection Metadata in Closed MXF File

However, it must be kept in mind that these n MXF files generated from the input stream according to the above procedure are still in an "Open" status. According to the MXF standard specification, MXF files shall be closed before playback. Therefore, a finalizing process is necessary to make the Connection Metadata partitions in the Header Metadata of the above MXF files to be "Closed". All associated peers may send control metadata to each other to search and collect the Connection Metadata messages in these concatenated MXF files. The Connection Metadata in "Closed" MXF files may be described as following (compare Fig. 8 to 10):

1. As shown in Fig. 8, after the finalizing-process, Header Metadata of the first MXF file (1) contains no "last", but all "next" Connection Metadata ($(1_N)\ldots((n-1)_N)$), as indicated by the index "N". The Header Metadata may optionally be repeated in the File Footer and the Body Partitions of the File Body as indicated by dashed arrows in Fig. 8.

2. As shown in Fig. 9, after the finalizing-process, the Header Metadata of any following MXF file (k) contains all "last" Connection Metadata ((2_L) …(k_L)) created before the current MXF file (k), including that of the current file, and all "next" Connection Metadata ((k_N) …$((n-1)_N)$) created after the current MXF file (k), including that of the current file.

3. As shown in Fig. 10, after the finalizing-process, the Header Metadata of the end MXF file (n) contains all "last" Connection Metadata ((2_L) … (n_L)) created in the sequence of MXF files. Like in MXF file (1) the Header Metadata in all MXF files (k) and (n) may optionally be

repeated in the File Footer and the Body Partitions of the File Body as indicated by dashed arrows in Fig. 10.

Concatenated MXF files playback by using Connection Metadata

MXF features enable stream-able playback of its essence in the MXF Essence Container.

By using control metadata that are designed and generated by the distributed storage system, it is possible to set up stream-able playback among a group of peers.

Playback of Open MXF files

Although MXF decoders are not required to be able to decode Open MXF files, it shall be assumed herein to be possible to playback these concatenated Open MXF files with Connected Metadata. As shown in Fig. 4-7, usually these embedded Connection Metadata in each MXF file provide reference to all prior concatenated MXF files and only the next MXF file. For playback of these MXF files, it may be started from any MXF file of any peer (1,2,…,n) to set up the playback task. At first, all UUIDs of all prior MXF files can be extracted from these accumulated "last" Connection Metadata in the Header Metadata from the File Header or the File Footer of each MXF file, except MXF file (1), which has no "last" Connection Metadata. Then, the "next" Connection Metadata in the File Footer of the current MXF file is parsed to find its next reference MXF file. The "next" Connection Metadata in the File Footer of each next MXF file is parsed one by one. Afterwards, a playback task is arranged by using control metadata among related peers.

Playback of Closed MXF files

As shown in Fig. 8 to 10, in closed MXF files, each MXF file contains compact Connection Metadata, which link to all other concatenated MXF files. In this case, a playback task may start from any MXF file of any peer (1,…,n). Only Connection Metadata in the current MXF file need to be parsed, since it includes all reference messages about all concatenated MXF files from any one selected MXF file. Based on the reference messages from one MXF file in one peer, the stream-able playback task in the system can be set up.

Multiple tasks supported by Connection Metadata Packages

As shown in Fig. 11, while recording or packing a stream into MXF files, it is allowed to create different Connection Metadata Packages in the Header Metadata. Connection Metadata in different packages are distinguished with their Task ID. Different Connection Metadata Packages direct to different sequences of concatenated MXF files. This design may be useful in the following scenarios:

While recording a stream into an MXF file of a peer, before the peer reaches its maximum storage capacity, it will ask other peers if they can continue to record the input stream. If more than one peer can do it, and the current network capacity permits to do synchronized stream recording, and the task manager would like to do so, the current peer will generate multiple Connection Metadata packages. Connection Metadata in the same package will have the same Task ID, which represents a sequence of concatenated MXF files. Different packages direct to different sequences of MXF files. In this application, multiple sequences of concatenated MXF files are generated.

When a stream is split recorded into concatenated MXF files, sometimes the user would like to duplicate some MXF files for making safe copies. A duplicated MXF file may stay in the same or different peer from its origin. In this case, the system should generate separated Connection Metadata Package(s) in associated Header Metadata of all related MXF file(s). Based on such alternative Connection Metadata Package(s), it is possible to repair defect files and arrange optimal playback.

The invention is usable for all kinds of stream recording devices that can be used in distributed storage networks, particularly audio and/or video stream recorders in decentralized networks, such as peer-to-peer networks.

Claims

1. Method for recording a data stream on a distributed storage system
characterized by
- packing said data stream into at least a first MXF file (1) and a second MXF file (2),
- inserting first connection data (1_N) into said first MXF file (1), said first connection data (1_N) pointing to said second MXF file (2), and
- inserting second connection data (2_L) into said second MXF file (2), said second connection data (2_L) pointing to said first MXF file (1).

2. Method according to claim 1, further including the step of inserting third connection data (2_N) into said second MXF file (2), said third connection data (2_N) pointing to a third MXF file (3).

3. Method according to claim 1 or 2, wherein said step or steps of inserting connection data is/are performed while terminating said recording of the respective MXF file.

4. Method according to one of the preceding claims, wherein said data stream is split into a specific number of MXF files and connection data pointing to all of these MXF files are stored in each of said specific number of MXF files.

5. Method according to one of the preceding claims, wherein connection data are inserted into header and/or footer data of an MXF file.

6. Device for recording a data stream on a distributed storage system
characterized by

- packing means for packing said data stream into at least a first MXF file (1) and a second MXF file (2),
- data processing means for inserting first connection data (1_N) into said first MXF file (1), said first connection data (1_N) pointing to said second MXF file (2), and for inserting second connection data (2_L) into said second MXF file (2), said second connection data (2_L) pointing to said first MXF file (1).

7. Device according to claim 6, wherein said data processing means is suitable for inserting third connection data (2_N) into said second MXF file (2), said third connection data (2_N) pointing to a third MXF file (3).

8. Device according to claim 6 or 7, wherein said packing means enables splitting of said data stream into a specific number of MXF files and connection data pointing to all these MXF files are storable in each of said specific number of MXF files.

9. Device according to one of the claims 6 to 8, wherein said data processing means are suitable for inserting connection data into header and/or footer data of a MXF file.

10. Method for playing back a data stream recorded on a distributed storage system
characterized by
- reading connection data (1_N) from a first MXF file (1), said connection data (1_N) pointing to a second MXF file (2),
- building a MXF file chain on the basis of said connection data (1_N) and
- playing back the essence data of said first MXF file (1) and said second MXF file (2) according to said MXF file chain.

11. Method according to claim 10, wherein said step of reading connection data is repeated for each MXF file of said MXF file chain.

12. Method according to claim 10, wherein said step of reading connection data is performed only once for one MXF file chain, wherein connection data pointing to all MXF files of said MXF file chain are included in each of said MXF files of said MXF file chain.

13. Device for playing back a data stream recorded on a distributed storage system
characterized by
- reading means for reading connection data (1_N) from a first MXF file (1), said connection data pointing to a second MXF file (2),
- data processing means for building a MXF file chain on the basis of said connection data (1_N) and
- playing means for playing back the essence data of said first MXF file (1) and said second MXF file (2) according to said MXF file chain.

14. Device according to claim 13, wherein said reading means is able to read connection data from each MXF file of said MXF file chain.

15. Device according to claim 13, wherein said reading means is able to read all connection data of a MXF file chain from one MXF file of said MXF file chain.

16. Method or device according to one of the preceding claims, wherein said connection data includes task data, attribute data and reference data, said task data contains a task ID value, said attribute data contains a

status value like "next" or "last" and said reference data contains an ID value of another MXF file.

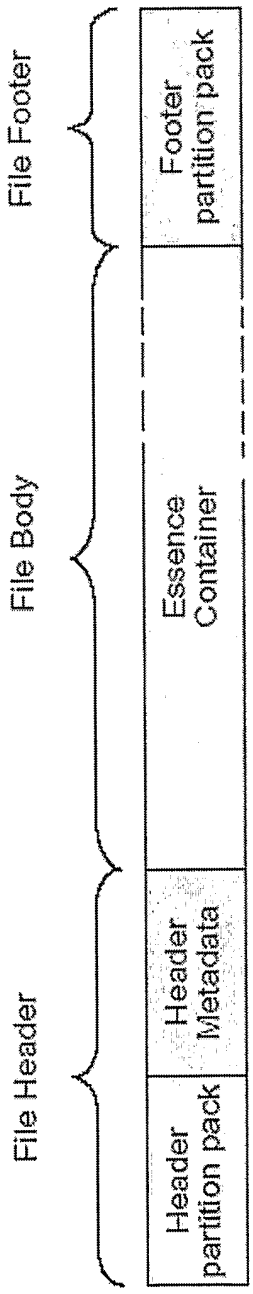

File Header
File Body
File Footer
Header partition pack
Header Metadata
Essence Container
Footer partition pack

Fig. 1

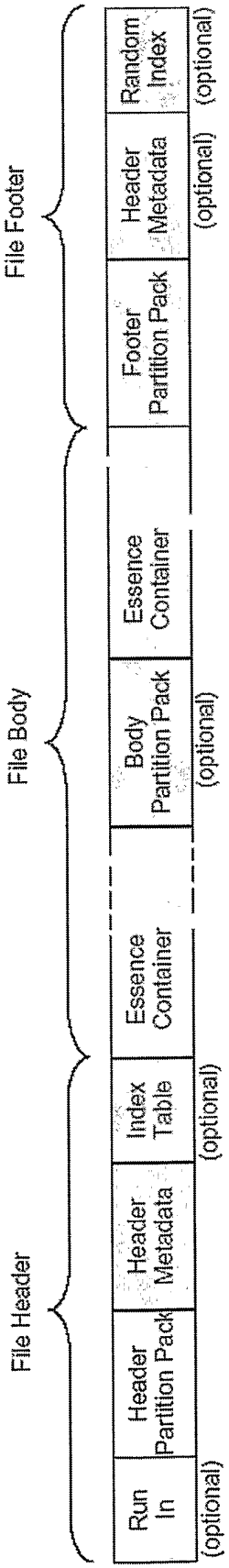

File Header
File Body
File Footer
Run In
(optional)
Header Partition Pack
Header Metadata
Index Table
(optional)
Essence Container
Body Partition Pack
(optional)
Essence Container
Footer Partition Pack
Header Metadata
(optional)
Random Index
(optional)

Fig. 2



Fig. 3

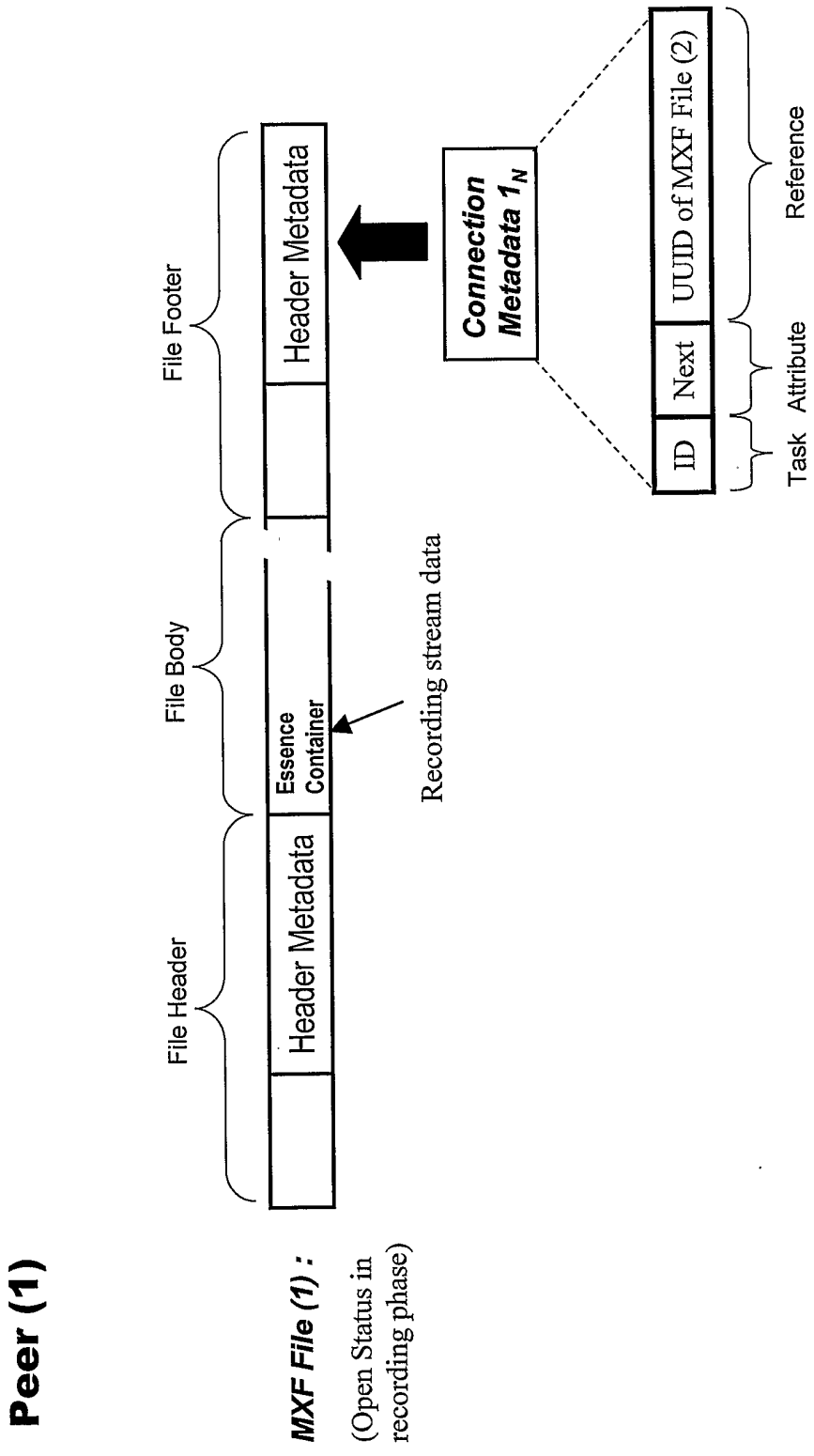

Peer (1)
MXF File (1) :
(Open Status in recording phase)
File Header
File Body
File Footer
Header Metadata
Essence Container
Header Metadata
Recording stream data
Connection Metadata 1N
ID
Next
UUID of MXF File (2)
Task
Attribute
Reference

Fig. 4

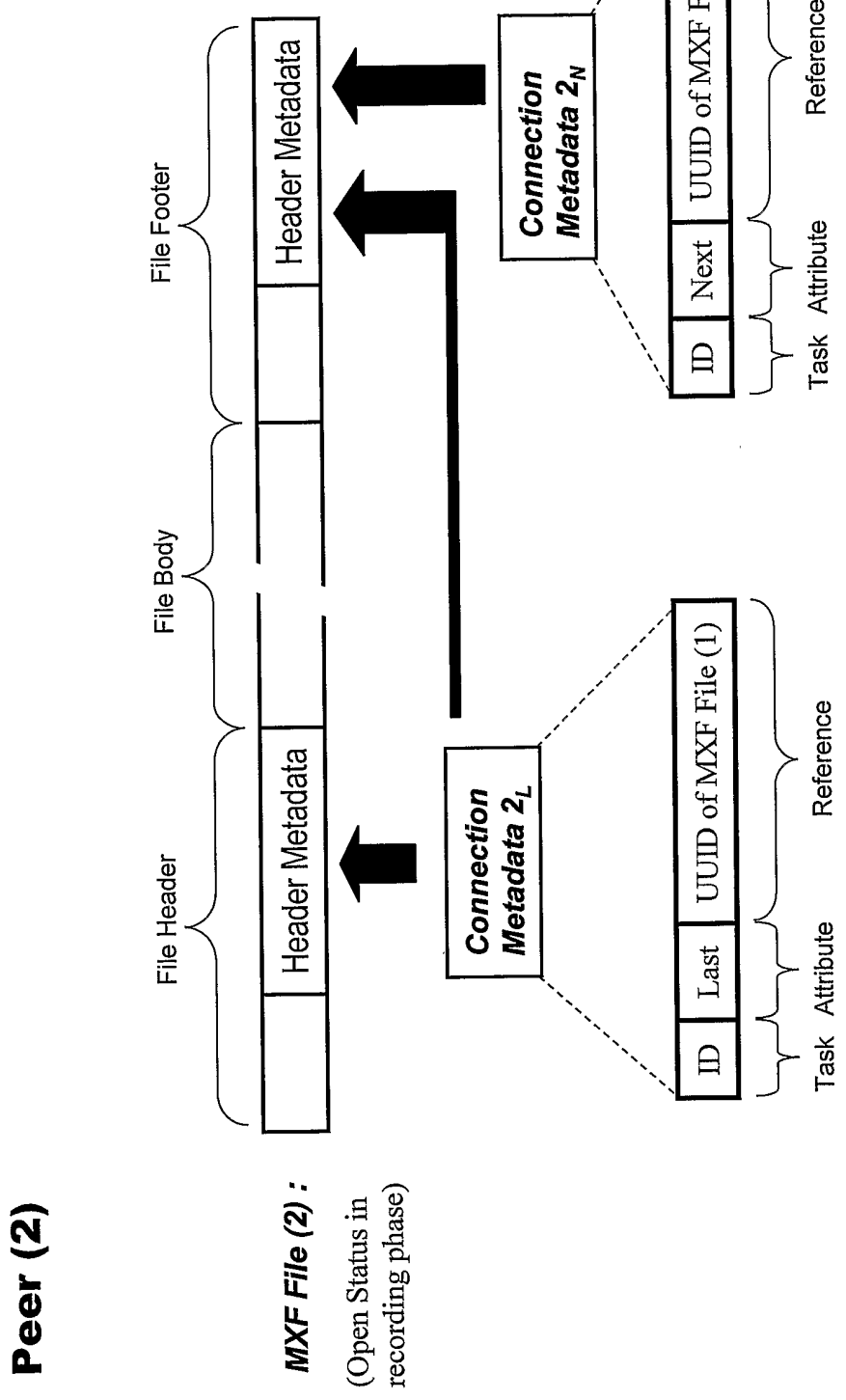

Peer (2)
MXF File (2) :
(Open Status in recording phase)
File Header
File Body
File Footer
Header Metadata
Header Metadata
Connection Metadata 2L
Connection Metadata 2N
ID
Last
UUID of MXF File (1)
ID
Next
UUID of MXF File (3)
Task Attribute
Reference
Task Attribute
Reference

Fig. 5

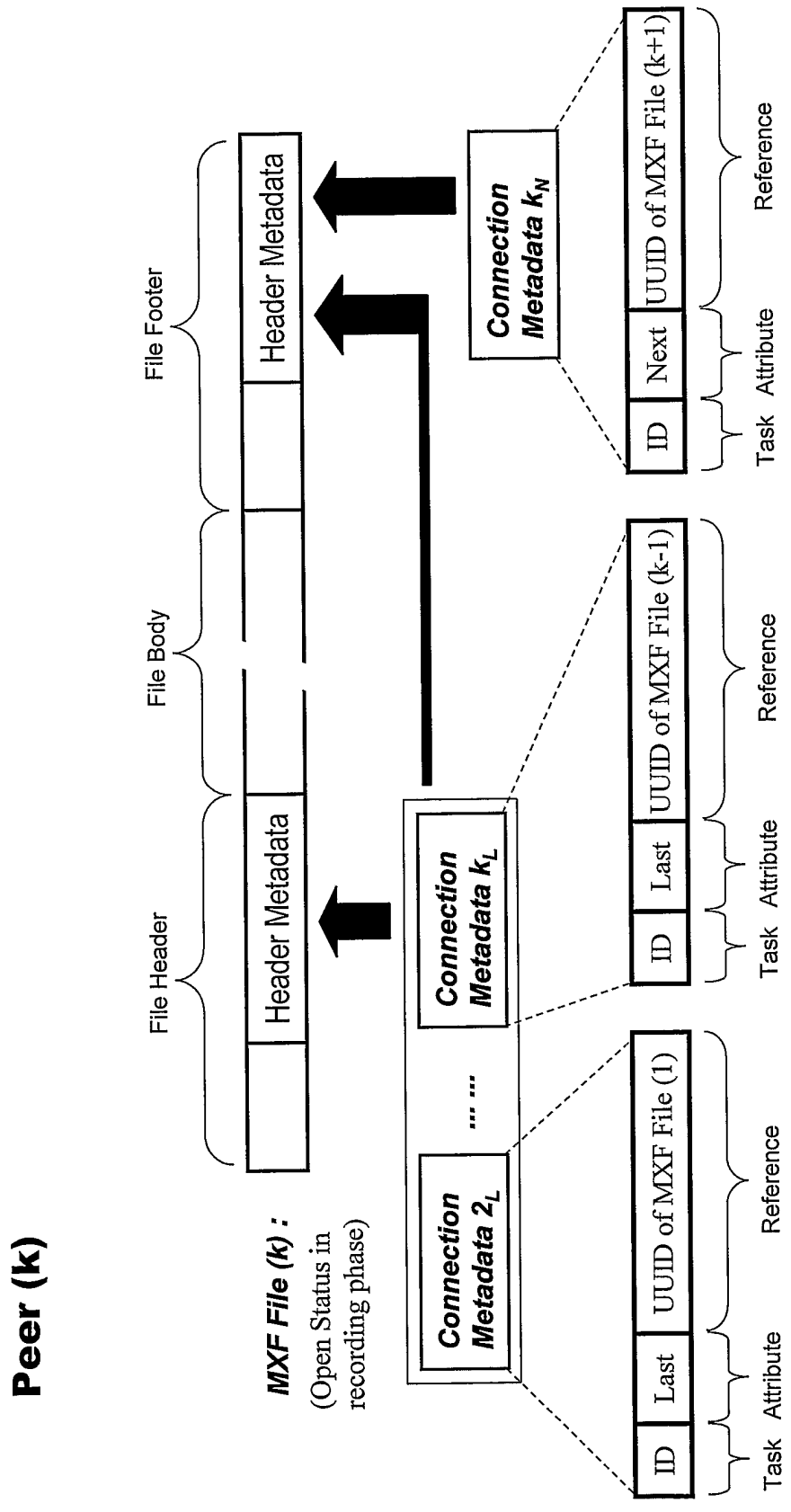

Peer (k)
MXF File (k) :
(Open Status in recording phase)
File Header
File Body
File Footer
Header Metadata
Header Metadata
Connection Metadata k_N
Connection Metadata k_L
Connection Metadata 2_L
ID
Next
UUID of MXF File (k+1)
ID
Last
UUID of MXF File (k-1)
ID
Last
UUID of MXF File (1)
Task Attribute
Reference

Fig. 6

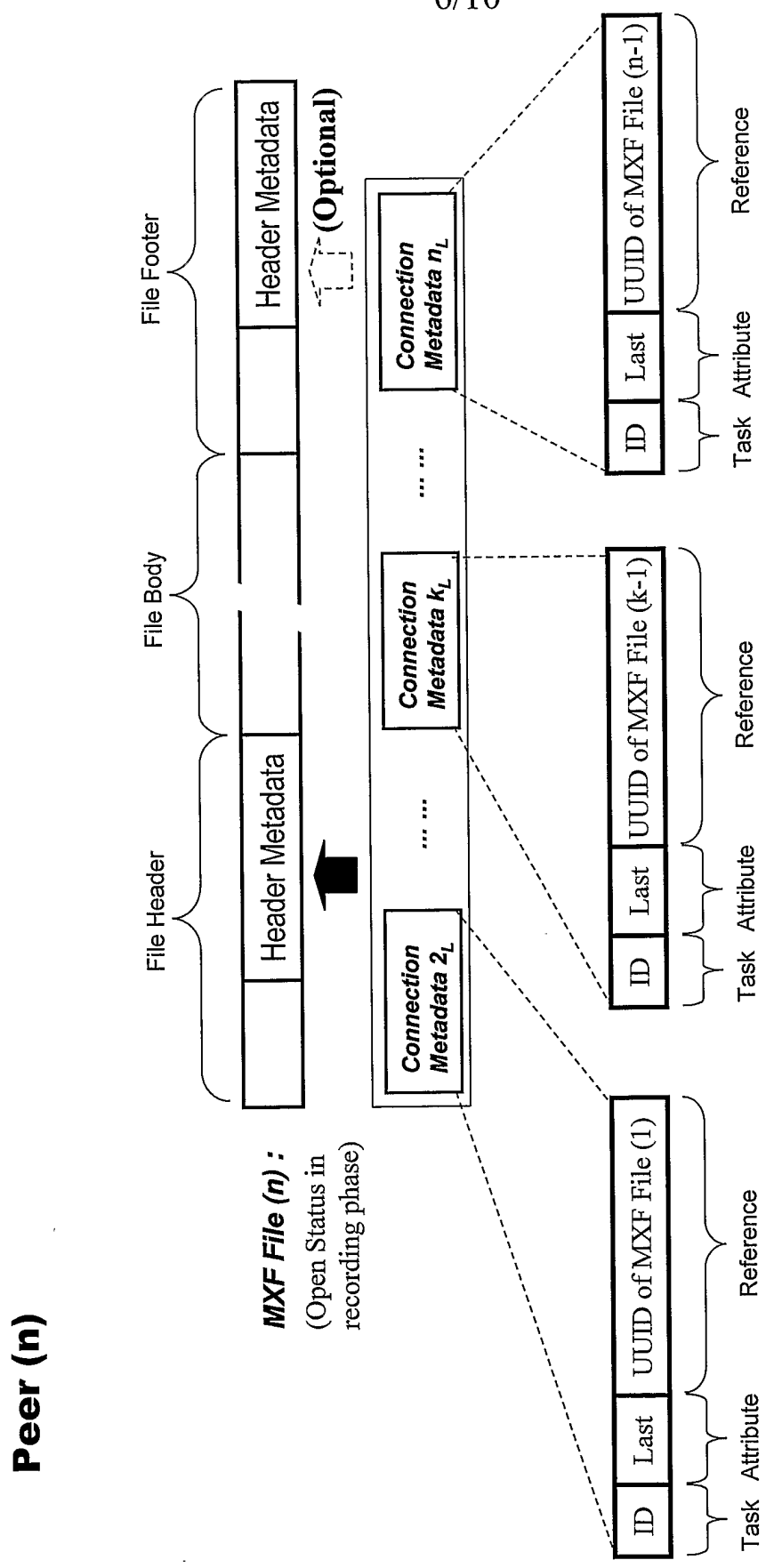

Peer (n)
MXF File (n) :
(Open Status in recording phase)
File Header
File Body
File Footer
Header Metadata
Header Metadata
(Optional)
Connection Metadata 2L
Connection Metadata kL
Connection Metadata nL
... ...
... ...
ID
Last
UUID of MXF File (1)
ID
Last
UUID of MXF File (k-1)
ID
Last
UUID of MXF File (n-1)
Task Attribute
Reference
Task Attribute
Reference
Task Attribute
Reference

Fig. 7

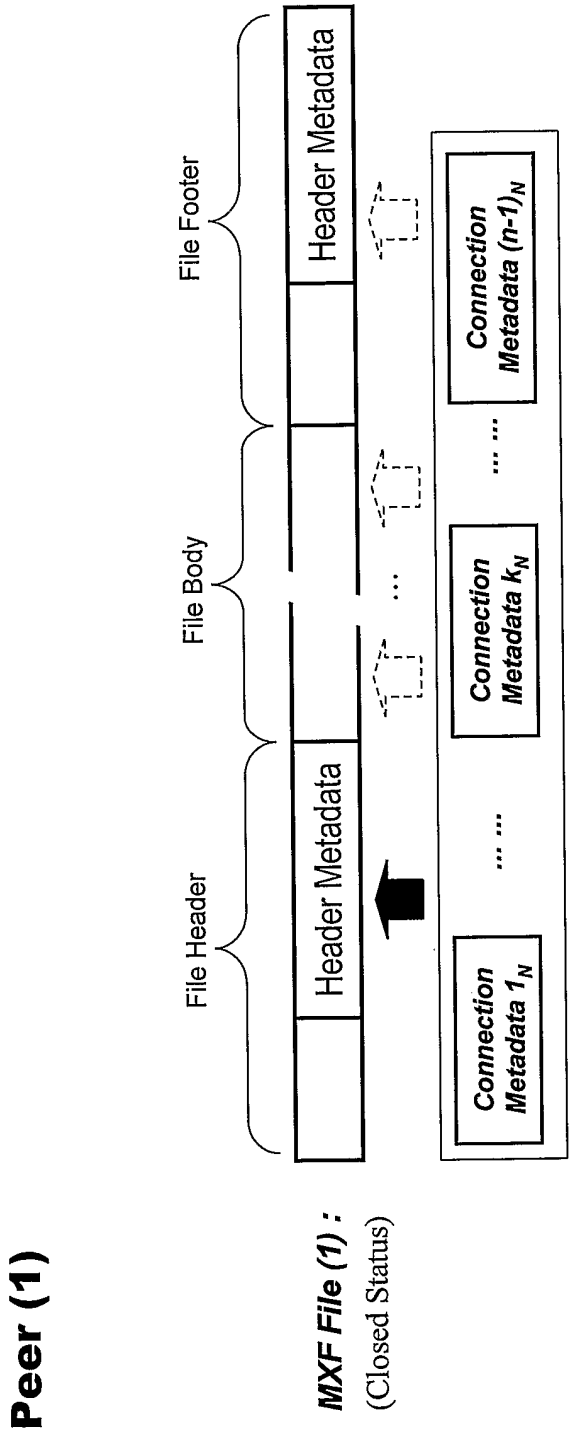

Peer (1)
MXF File (1) :
(Closed Status)
File Header
File Body
File Footer
Header Metadata
Header Metadata
Connection Metadata 1N
Connection Metadata kN
Connection Metadata (n-1)N

Fig. 8

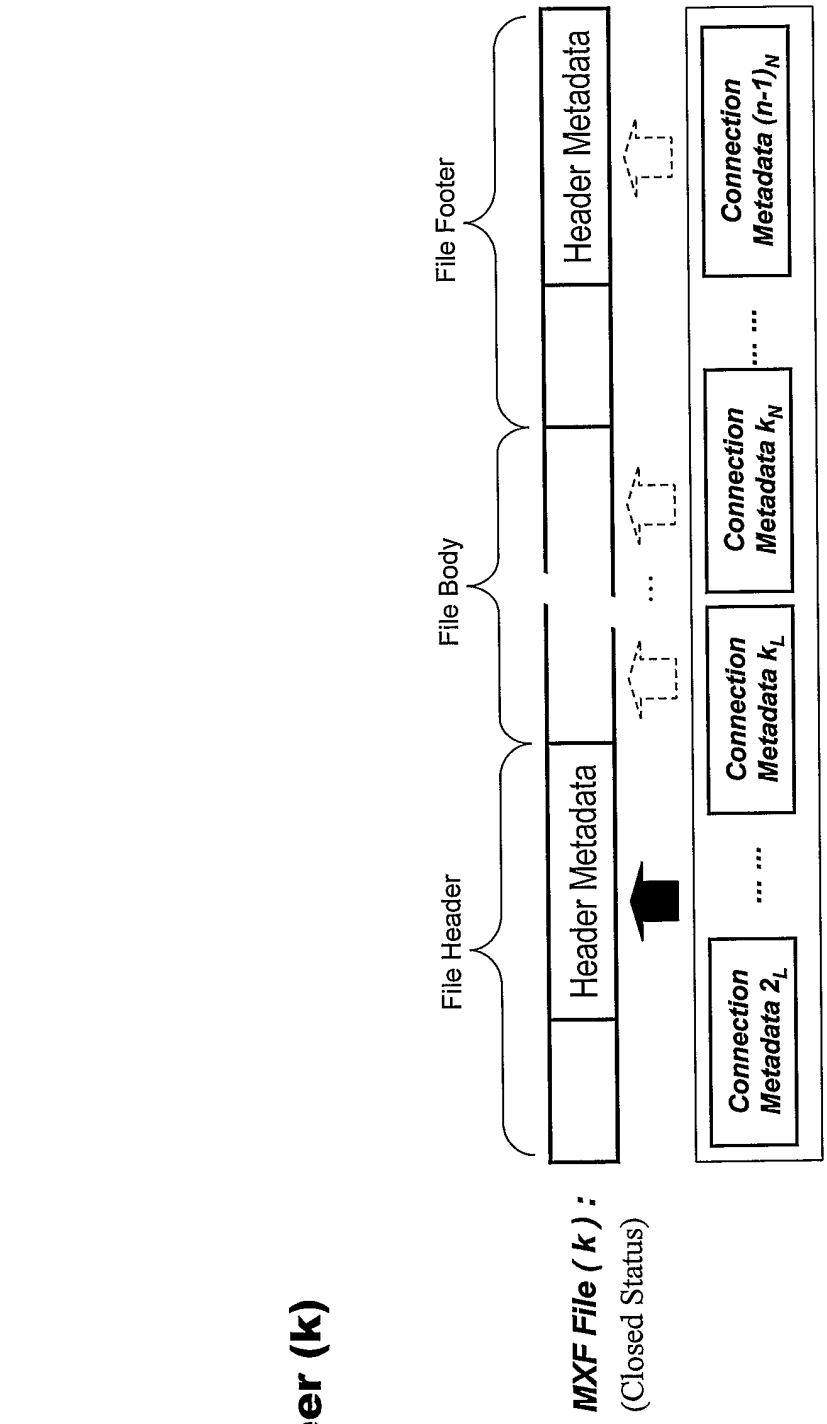

Peer (k)
MXF File (k) :
(Closed Status)
File Header
File Body
File Footer
Header Metadata
Header Metadata
Connection Metadata 2_L
Connection Metadata k_L
Connection Metadata k_N
Connection Metadata $(n-1)_N$

Fig. 9

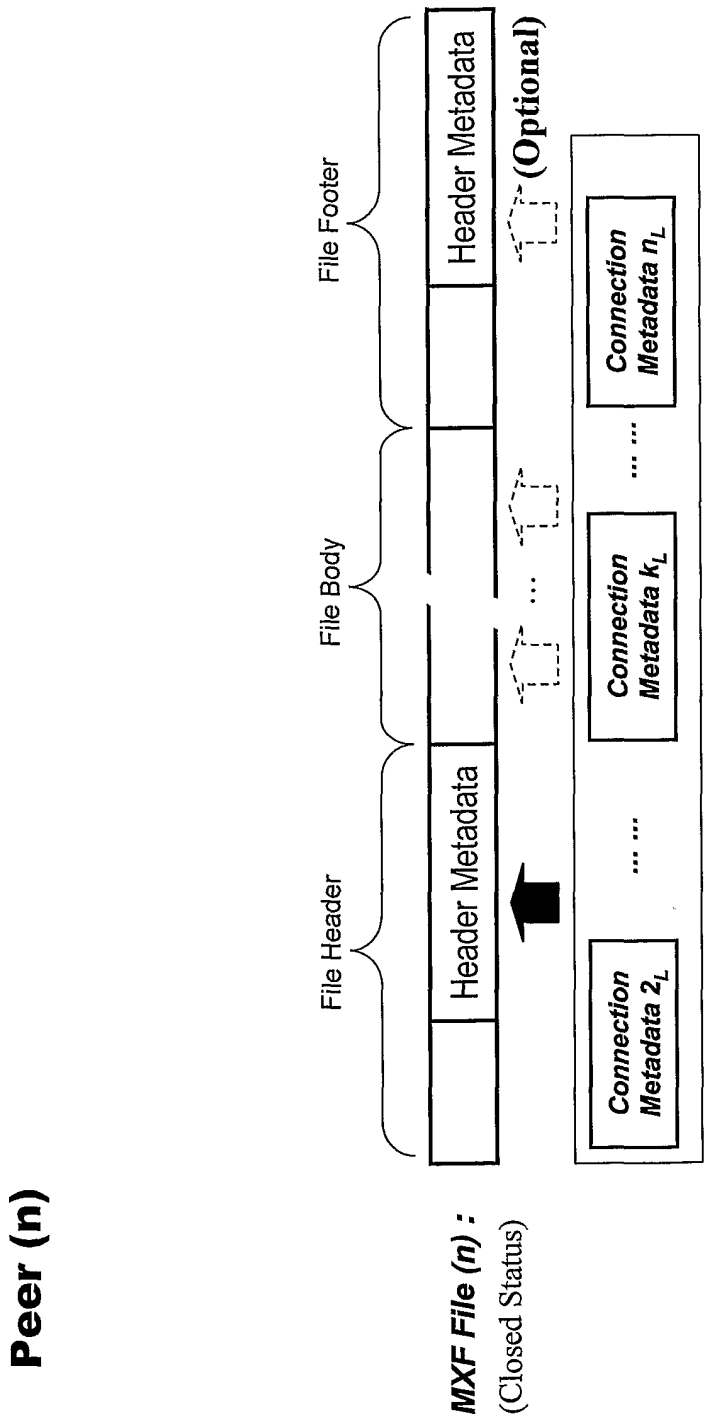

Peer (n)
MXF File (n) :
(Closed Status)
File Header
File Body
File Footer
Header Metadata
Header Metadata
(Optional)
Connection Metadata 2L
Connection Metadata kL
Connection Metadata nL

Fig. 10

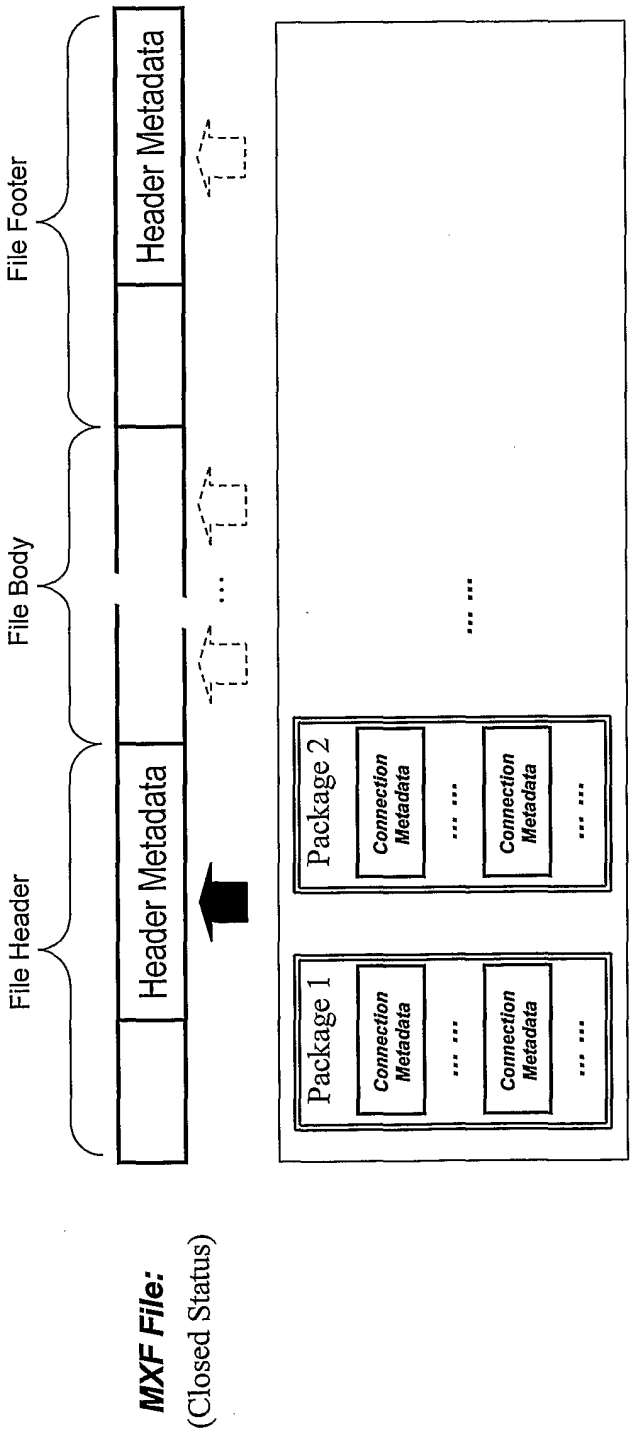

File Header
File Body
File Footer
Header Metadata
Header Metadata
Package 1
Connection Metadata
Connection Metadata
Package 2
Connection Metadata
Connection Metadata
MXF File:
(Closed Status)

Fig. 11

INTERNATIONAL SEARCH REPORT

International Application No
PCT/EP2005/004377

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G11B27/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G11B H04N G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 6 499 083 B1 (HAMLIN CHRISTOPHER L) 24 December 2002 (2002-12-24) column 3, lines 18-25 column 11, line 65 - column 12, line 3 column 14, line 60 - column 15, line 49 column 19, line 35 - column 21, line 9 figures 5B,10,14-17	1-16
Y	SMPTE: "Material Exchange Format (MXF)" PROPOSED SMPTE ENGINEERING GUIDELINE, vol. EG41, 22 July 2003 (2003-07-22), XP002312809 pages 1-10 pages 53,54	1-16
A	GB 2 371 889 A (* SONY UNITED KINGDOM LIMITED) 7 August 2002 (2002-08-07) page 11, lines 3-21 page 13, line 28 - page 14, line 19	1-16

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
22 July 2005	02/08/2005

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Hermes, L

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/EP2005/004377

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6499083 B1	24-12-2002	NONE	
GB 2371889 A	07-08-2002	NONE	