



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2013년04월03일
(11) 등록번호 10-1249224
(24) 등록일자 2013년03월25일

(51) 국제특허분류(Int. Cl.)
G11B 20/10 (2006.01) H04N 5/76 (2006.01)
H04N 5/93 (2006.01)
(21) 출원번호 10-2012-7012703(분할)
(22) 출원일자(국제) 2005년05월10일
심사청구일자 2012년05월17일
(85) 번역문제출일자 2012년05월17일
(65) 공개번호 10-2012-0056899
(43) 공개일자 2012년06월04일
(62) 원출원 특허 10-2012-7000874
원출원일자(국제) 2005년05월10일
심사청구일자 2012년01월12일
(86) 국제출원번호 PCT/JP2005/008488
(87) 국제공개번호 WO 2005/109434
국제공개일자 2005년11월17일
(30) 우선권주장
JP-P-2004-141560 2004년05월11일 일본(JP)
(56) 선행기술조사문헌
JP2003308645 A
JP2002369154 A
전체 청구항 수 : 총 9 항

(73) 특허권자
파나소닉 주식회사
일본 오오사카후 가도마시 오오아자 가도마 1006
반치
(72) 발명자
다나카 게이이치
일본국 오오사카후 가도마시 오오아자가도마 1006
파나소닉 주식회사
오아시 마사히로
일본국 오오사카후 가도마시 오오아자가도마 1006
파나소닉 주식회사
가나마루 도모카즈
일본국 오오사카후 가도마시 오오아자가도마 1006
파나소닉 주식회사
(74) 대리인
홍승규, 김영철

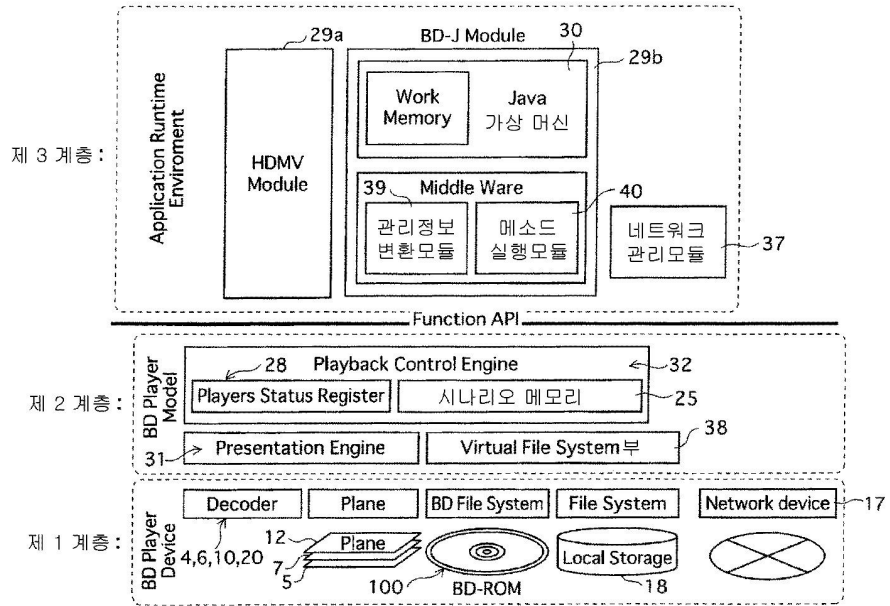
심사관 : 이진익

(54) 발명의 명칭 재생장치, 재생방법, 프로그램

(57) 요약

BD-ROM 및 로컬 기억장치(18)에 기록되어 있는 복수의 AVClip이 하나의 스트림 시퀀스를 구성하고 있는 경우, 재생제어엔진(32)은 스트림 시퀀스를 구성하는 일부의 AVClip이 인식할 수 있는 상태이면, 나머지 AVClip이 인식할 수 없는 disable 상태라도 재생을 개시한다. 가상 파일 시스템부(38)는 스트림 시퀀스 중 일부의 AVClip에 대한 재생처리가 계속되고 있는 동안, 애플리케이션에 의한 메소드 호출에 따라서, 나머지 AVClip의 상태를 재생제어 엔진(32)이 인식할 수 있는 enable 상태, 또는 인식할 수 없는 disable 상태로 변화시킨다.

대표도



특허청구의 범위

청구항 1

스트림 시퀀스(stream sequence)를 재생하는 재생장치로,

재생수단과,

상기 스트림 시퀀스를 구성하는 개개의 디지털 스트림을 저장하기 위한 저장 장소가 복수의 기록매체 각각에 나뉘어져 있는 경우, 각 디지털 스트림의 상태를 상기 재생수단이 인식할 수 있는 상태, 또는 상기 재생수단이 인식할 수 없는 상태로 설정하는 패키지 관리수단과,

애플리케이션을 실행하여 상기 기록매체에 대한 액세스 제어를 실행하는 애플리케이션 실행수단과,

상기 애플리케이션 실행수단에 의해 실행되는 애플리케이션으로부터의 호출에 따라, 상기 스트림 시퀀스를 구성하는 디지털 스트림 중 상기 재생수단이 인식할 수 없는 상태의 디지털 스트림에 관한 정보를 상기 애플리케이션에 되돌리는 API를 구비하고,

상기 재생수단은,

상기 스트림 시퀀스를 구성하는 일부의 디지털 스트림이 인식할 수 있는 상태이면 나머지 디지털 스트림이 인식할 수 없는 상태라도 재생을 개시하고,

상기 패키지 관리수단은,

상기 스트림 시퀀스 중 일부의 디지털 스트림에 대한 재생처리가 계속되고 있는 동안, 상기 애플리케이션 실행수단에 의해 실행되는 애플리케이션으로부터의 요구에 따라서, 상기 나머지 디지털 스트림의 상태를 상기 재생수단이 인식할 수 있는 상태, 또는 상기 재생수단이 인식할 수 있는 디지털 스트림 중 상기 재생처리가 이루어지고 있는 일부의 디지털 스트림 이외의 디지털 스트림을 인식할 수 없는 상태로 변화시키고,

상기 애플리케이션 실행수단에 의해 실행되는 애플리케이션은 상기 API의 호출에 의해 상기 재생수단이 인식할 수 없는 상태의 디지털 스트림에 관한 정보를 얻을 수 있도록 구성하는 것을 특징으로 하는 재생장치.

청구항 2

제 1 항에 있어서,

네트워크를 개재하여 상기 재생수단이 인식할 수 없는 상태의 디지털 스트림을 상기 복수의 기록매체 중의 기입 가능한 기록매체에 다운로드 하는 다운로드수단을 구비하고,

상기 다운로드수단이 상기 기입 가능한 기록매체에 상기 재생수단이 인식할 수 없는 상태의 디지털 스트림을 다운로드 한 후, 상기 패키지 관리수단은 상기 애플리케이션 실행수단에 의해 실행되는 애플리케이션으로부터 상기 다운로드 한 디지털 스트림을 상기 재생수단이 인식할 수 있는 상태로 변화시키는 요구를 수신하도록 한 재생장치.

청구항 3

제 1 항에 있어서,

상기 재생수단이 인식할 수 있는 상태의 디지털 스트림의 재생이 종료하면, 상기 애플리케이션 실행수단에 의해 실행되는 애플리케이션에 통지하도록 구성한 재생장치.

청구항 4

제 3 항에 있어서,

상기 패키지 관리수단은 상기 애플리케이션 실행수단에 의해 실행되는 애플리케이션으로부터의 요구에 따라서 상기 재생이 종료한 상기 재생수단이 인식할 수 있는 상태의 디지털 스트림을 다시 상기 재생수단이 인식할 수 없는 상태로 변화시키는 재생장치.

청구항 5

스트림 시퀀스를 재생하는 재생방법으로,

재생스텝과,

상기 스트림 시퀀스를 구성하는 개개의 디지털 스트림을 저장하기 위한 저장 장소가 복수의 기록매체 각각에 나뉘어져 있는 경우, 각 디지털 스트림의 상태를 상기 재생스텝이 인식할 수 있는 상태, 또는 상기 재생스텝이 인식할 수 없는 상태로 설정하는 패키지 관리스텝과,

애플리케이션을 실행하여 상기 기록매체에 대한 액세스 제어를 실행하는 애플리케이션 실행스텝과,

상기 애플리케이션 실행스텝에 의해 실행되는 애플리케이션으로부터의 호출에 따라, 상기 스트림 시퀀스를 구성하는 디지털 스트림 중 상기 재생스텝이 인식할 수 없는 상태의 디지털 스트림에 관한 정보를 상기 애플리케이션에 되돌리는 API 응답스텝을 포함하고,

상기 재생스텝은,

상기 스트림 시퀀스를 구성하는 일부의 디지털 스트림이 인식할 수 있는 상태이면 나머지 디지털 스트림이 인식할 수 없는 상태라도 재생을 개시하고,

상기 패키지 관리스텝은,

상기 스트림 시퀀스 중 일부의 디지털 스트림에 대한 재생처리가 계속되고 있는 동안, 상기 애플리케이션 실행스텝에 의해 실행되는 애플리케이션으로부터의 요구에 따라서, 상기 나머지 디지털 스트림의 상태를 상기 재생스텝이 인식할 수 있는 상태, 또는 상기 재생스텝이 인식할 수 있는 디지털 스트림 중 상기 재생처리가 이루어지고 있는 일부의 디지털 스트림 이외의 디지털 스트림을 인식할 수 없는 상태로 변화시키며,

상기 애플리케이션 실행스텝에 의해 실행되는 애플리케이션은 상기 API 호출스텝에 의해 상기 재생스텝이 인식할 수 없는 상태의 디지털 스트림에 관한 정보를 얻을 수 있도록 한 것을 특징으로 하는 재생방법.

청구항 6

제 5 항에 있어서,

네트워크를 개재하여 상기 재생스텝이 인식할 수 없는 상태의 디지털 스트림을 상기 복수의 기록매체 중의 기입 가능한 기록매체에 다운로드 하는 다운로드스텝을 구비하고,

상기 다운로드스텝이 상기 기입 가능한 기록매체에 상기 재생스텝이 인식할 수 없는 상태의 디지털 스트림을 다운로드 한 후, 상기 패키지 관리스텝은 상기 애플리케이션 실행스텝에 의해 실행되는 애플리케이션으로부터 상기 다운로드 한 디지털 스트림을 상기 재생스텝이 인식할 수 있는 상태로 변화시키는 요구를 수신하도록 한 재생방법.

청구항 7

제 5 항에 있어서,

상기 재생스텝이 인식할 수 있는 상태의 디지털 스트림의 재생이 종료하면, 상기 애플리케이션 실행스텝에 의해 실행되는 애플리케이션에 통지하도록 구성하는 재생방법.

청구항 8

제 7 항에 있어서,

상기 패키지 관리스텝은 상기 애플리케이션 실행스텝에 의해 실행되는 애플리케이션으로부터의 요구에 따라서 상기 재생이 종료한 상기 재생스텝이 인식할 수 있는 상태의 디지털 스트림을 다시 상기 재생스텝이 인식할 수 없는 상태로 변화시키는 재생방법.

청구항 9

스트림 시퀀스를 재생하는 처리를 컴퓨터에 실행시키는 프로그램을 기록한 컴퓨터 판독 가능한 기록매체로,

재생스텝과,

상기 스트림 시퀀스를 구성하는 개개의 디지털 스트림을 저장하기 위한 저장장소가 복수의 기록매체 각각에 나눠져 있는 경우, 각 디지털 스트림의 상태를 상기 재생스텝이 인식할 수 있는 상태, 또는 상기 재생스텝이 인식할 수 없는 상태로 설정하는 패키지 관리스텝과,

애플리케이션을 실행하여 상기 기록매체에 대한 액세스 제어를 실행하는 애플리케이션 실행스텝과,

상기 애플리케이션 실행스텝에 의해 실행되는 애플리케이션으로부터의 호출에 따라, 상기 스트림 시퀀스를 구성하는 디지털 스트림 중 상기 재생스텝이 인식할 수 없는 상태의 디지털 스트림에 관한 정보를 상기 애플리케이션에 되돌리는 API 응답스텝을 컴퓨터에 실행시키고,

상기 재생스텝은,

상기 스트림 시퀀스를 구성하는 일부의 디지털 스트림이 인식할 수 있는 상태이면 나머지 디지털 스트림이 인식할 수 없는 상태라도 재생을 개시하고,

상기 패키지 관리스텝은,

상기 스트림 시퀀스 중 일부의 디지털 스트림에 대한 재생처리가 계속되고 있는 동안, 상기 애플리케이션 실행스텝에 의해 실행되는 애플리케이션으로부터의 요구에 따라서, 상기 나머지 디지털 스트림의 상태를 상기 재생스텝이 인식할 수 있는 상태, 또는 상기 재생스텝이 인식할 수 없는 디지털 스트림 중 상기 재생처리가 이루어지고 있는 일부의 디지털 스트림 이외의 디지털 스트림을 인식할 수 없는 상태로 변화시키고,

상기 애플리케이션 실행스텝에 의해 실행되는 애플리케이션은 상기 API 호출 스텝에 의해 상기 재생스텝이 인식할 수 없는 상태의 디지털 스트림에 관한 정보를 얻을 수 있도록 한 것을 특징으로 하는 프로그램을 기록한 컴퓨터 판독 가능한 기록매체.

명세서

기술분야

[0001] 본 발명은 가상 패키지(Virtual Package)의 기술분야에 속하는 발명이다.

배경기술

[0002] 가상 패키지란, BD-ROM 등의 판독전용(read-only)의 기록매체에 기록되어 있는 디지털 스트림과, 하드 디스크 등의 재기록이 가능(re-writable)한 기록매체에 기록되어 있는 디지털 스트림을 동적으로 조합하여 가상적인 패키지를 구축함으로써, 판독전용의 기록매체의 내용 확장을 도모하는 기술이다. 여기서, BD-ROM에 기록되어 있는 디지털 스트림이 영화작품의 본편을 구성하는 것이고, 하드 디스크에 기록되어 있는 디지털 스트림이 영화작품의 속편을 구성하는 것인 경우, 상술한 가상 패키지를 구축함으로써, BD-ROM 상의 본편과 하드 디스크 상의 속편을 하나의 장편의 영화작품으로 취급하여 재생에 제공할 수 있다.

[0003] 또한, 가상 패키지에 대한 선행기술로는 이하의 특허문헌에 기재되어 있는 것이 있다.

[0004] 특허문헌 1 : 일본국 특허공개공보 2002-369154호

발명의 내용

해결하려는 과제

[0005] 그런데 BD-ROM에 기록된 디지털 스트림과 하드 디스크에 기록된 디지털 스트림을 하나의 스트림 시퀀스(stream sequence)로 재생해 가는 경우, 스트림 시퀀스를 구성하는 디지털 스트림 중에 불안정한 상태의 것이 포함되어 있을 가능성이 있다. 여기서, 불안정한 상태의 디지털 스트림이란, 하드 디스크로의 다운로드가 완료되지 않았거나, 기록이 불완전한 디지털 스트림을 말한다. 불안정한 상태의 디지털 스트림이 포함되어 있음에도 불구하고 상술한 스트림 시퀀스를 재생하려고 하면, 재생수단에 의한 재생시점이 불안정한 스트림에 이르자마자 재생이 불안정해지거나, 재생이 도중에 끊기는 현상이 있을 수 있다.

[0006] 여기서, 재생하고자 하는 스트림이 불안정한 상태인가 여부를 재생장치 내의 재생수단(디코더 등)이 올바르게 체크 할 수 있으면, 이와 같은 재생의 불안정이나 도중에 끊기는 문제를 회피할 수 있다. 그러나 재생수단은

1/24초, 1/29.94초라고 하는 프레임 기간마다 영상신호를 출력하는 처리에 전념하고 있으며, 그와 같은 체크를 위한 노력은 존재하지 않는 것이 일반적이다. 그런 이유에서, 상술한 바와 같은 재생의 불안정이나 도중에 끊기는 문제의 발생은 피할 수 없다. 한편, BD-ROM 재생장치를 제조하는 민생기기의 산업분야에서는 영상의 재생에 대하여 엄격한 품질기준이 요구되고 있고, 디지털 스트림을 체크 할 여력이 재생수단 측에 없다고 해도, 상술한 재생의 불안정이나 도중에 끊기는 문제가 용인될 수는 없다.

[0007] 본 발명의 목적은, 불안정한 상태의 디지털 스트림을 체크 할 여력이 재생수단 측에 없는 경우에도, BD-ROM에 기록된 디지털 스트림과 하드 디스크에 기록된 디지털 스트림을 하나의 스트림 시퀀스로 재생해 가는 경우의 재생의 불안정이나 도중에 끊기는 문제를 방지할 수 있는 재생장치를 제공하는 것이다.

과제의 해결 수단

[0008] 상기 목적을 달성하기 위하여, 스트림 시퀀스(stream sequence)를 재생하는 재생장치로, 재생수단과, 상기 스트림 시퀀스를 구성하는 개개의 디지털 스트림을 저장하기 위한 저장 장소가 복수의 기록매체 각각에 나뉘어져 있는 경우, 각 디지털 스트림의 상태를 상기 재생수단이 인식할 수 있는 상태, 또는 상기 재생수단이 인식할 수 없는 상태로 설정하는 패키지 관리수단과, 애플리케이션을 실행하여 상기 기록매체에 대한 액세스 제어를 실행하는 애플리케이션 실행수단과, 상기 애플리케이션 실행수단에 의해 실행되는 애플리케이션으로부터의 호출에 따라, 상기 스트림 시퀀스를 구성하는 디지털 스트림 중 상기 재생수단이 인식할 수 없는 상태의 디지털 스트림에 관한 정보를 상기 애플리케이션에 되돌리는 API를 구비하고, 상기 재생수단은, 상기 스트림 시퀀스를 구성하는 일부의 디지털 스트림이 인식할 수 있는 상태이면 나머지 디지털 스트림이 인식할 수 없는 상태라도 재생을 개시하고, 상기 패키지 관리수단은, 상기 스트림 시퀀스 중 일부의 디지털 스트림에 대한 재생처리가 계속되고 있는 동안, 상기 애플리케이션 실행수단에 의해 실행되는 애플리케이션으로부터의 요구에 따라서, 상기 나머지 디지털 스트림의 상태를 상기 재생수단이 인식할 수 있는 상태, 또는 상기 재생수단이 인식할 수 있는 디지털 스트림 중 상기 재생처리가 이루어지고 있는 일부의 디지털 스트림 이외의 디지털 스트림을 인식할 수 없는 상태로 변화시키고, 상기 애플리케이션 실행수단에 의해 실행되는 애플리케이션은 상기 API의 호출에 의해 상기 재생수단이 인식할 수 없는 상태의 디지털 스트림에 관한 정보를 얻을 수 있도록 구성하는 것을 특징으로 한다.

발명의 효과

[0009] 애플리케이션이 디지털 스트림을 인식할 수 있는 상태로 하는가, 인식할 수 없는 상태로 하는가를 변화시킴으로써, 디지털 스트림을 재생수단이 재생하도록 하는가, 재생하지 않도록 하는가의 선택을 실현할 수 있다. 디지털 스트림을 재생수단이 재생하도록 하는가 재생하지 않도록 하는가의 선택권을 기록매체의 액세스를 실현하는 애플리케이션이 장악하므로, 애플리케이션이 디지털 스트림이 기록매체에 올바르게 기록되었음을 확인할 수 없는 경우에, 그 디지털 스트림을 인식할 수 없는 상태로 하면, 그 디지털 스트림은 없었던 것으로 취급된다. 이에 의해, 불안정한 디지털 스트림을 잘못 재생하는 경우는 발생하지 않는다. 따라서, 영상의 불안정이나 도중에 끊기는 문제가 발생하지 않는다.

[0010] 여기서, 복수의 기록매체에는 판독전용형의 기록매체(read-only recording medium)와 재기록 가능형의 기록매체(re-writable recording medium)가 있고, 상기 재생수단은 판독전용형 기록매체의 파일 레이아웃(file layout)만을 인식할 수 있는 것이며, 재기록 가능형 기록매체에 기록되어 있는 디지털 스트림은 파일에 저장된 상태로 재기록 가능형 기록매체에 기록되어 있으며, 상기 패키지 관리수단에 의한 인식할 수 있는 상태로의 변화는, 판독전용형 기록매체 상의 파일관리정보와 재기록 가능형 기록매체에서의 파일관리정보로부터 가상적인 볼륨 관리정보(virtual volume management information)를 생성하여 재생수단에 인계함으로써 이루어지며, 상기 인식할 수 없는 상태로의 변화는, 상기 가상적인 볼륨 관리정보에서 재기록 가능형 기록매체에서의 파일관리정보를 제거함으로써 이루어진다.

[0011] 패키지 관리수단에 의한 인식할 수 있는 상태로의 변화는, 재기록 가능형 기록매체에서 디지털 스트림을 저장한 파일을 판독전용 속성(read-only attribute)으로 변화시키는 처리를 더 포함하고, 인식할 수 없는 상태로의 변화는 재기록 가능형 기록매체에서 디지털 스트림을 저장한 파일을 재기록가능 속성(rewritable attribute)으로 변화시키는 처리를 더 포함한다.

[0012] 여기서, 일부의 디지털 스트림의 재생 중에 나머지의 디지털 스트림을 기록매체에 다운로드 하도록 애플리케이션이 제어를 행하고, 그 일부의 디지털 스트림의 재생이 완료되면 그 디지털 스트림을 인식할 수 있는 상태로 설정함으로써, 기록된 디지털 스트림의 재생을 재생수단에 행하게 할 수 있다.

[0013] 또, 나머지 디지털 스트림의 기록이 일부의 디지털 스트림의 재생기간 내에 완료되지 않으면, 그 디지털 스트림을 인식할 수 없는 상태로 설정함으로써, 이와 같은 디지털 스트림을 재생수단에 의한 재생으로부터 제외할 수 있다. 디지털 스트림의 재생 여부의 설정을, 다운로드 처리의 진척을 연동시킬 수 있으므로, 스트림 시퀀스를 구성하는 디지털 스트림의 재생과 스트림 시퀀스를 구성하는 디지털 스트림의 다운로드를 병렬로 실행할 수 있다. 여기서 BD-ROM에 기록되어 있는 디지털 스트림이 영화의 본편을 구성하는 것이고, 하드 디스크에 다운로드되는 디지털 스트림이 영화의 예고편인 경우, 본편 디지털 스트림의 재생 중에 예고편 디지털 스트림을 다운로드해 두고, 본편 디지털 스트림의 재생이 완료되면, 그 직후에 예고편 디지털 스트림의 재생을 개시시킬 수 있다.

도면의 간단한 설명

[0014] 도 1은 본 발명에 관한 재생장치의 사용행위에 대한 형태를 나타내는 도면이다.

도 2는 BD-ROM의 내부 구성을 나타내는 도면이다.

도 3은 AVClip이 어떻게 구성되어 있는가를 모식적으로 나타내는 도면이다.

도 4는 PL 정보의 구성을 나타내는 도면이다.

도 5는 AVClip 시간 축과 PL 시간 축의 관계를 나타내는 도면이다.

도 6은 4개의 Clip_Information_file_name에 의해 이루어진 일괄지정을 나타내는 도면이다.

도 7은 PLmark 정보의 내부 구성을 나타내는 도면이다.

도 8은 PLmark에 의한 챕터 정의를 나타내는 도면이다.

도 9는 SubPath 정보의 내부 구성을 나타내는 도면이다.

도 10은 SubPlayItem 시간 축 상의 재생구간 정의와 동기지정을 나타내는 도면이다.

도 11(a)은 Java(등록상표) 아카이브 파일(archive file)에 수용되어 있는 프로그램 및 데이터를 나타내는 도면이다. (b)는 클래스 파일의 내부 구성을 나타내는 도면이다.

도 12는 본 발명에 관한 재생장치의 내부 구성을 나타내는 도면이다.

도 13 (a)는 로컬 기억장치(Local Storage, 18)에서의 디렉터리 구조를 나타내는 도면이다. (b)는 스트림 저장 위치정보 파일의 내부 구성을 나타내는 도면이다.

도 14는 ROM(24)에 저장된 소프트웨어와 하드웨어로 이루어지는 부분을 계층 구성으로 치환하여 묘사한 도면이다.

도 15는 Java(등록상표) 가상 머신(virtual machine, 30)의 내부 구성을 나타내는 도면이다.

도 16은 가상 파일 시스템부(38)에 의한 가상 패키지 정보 구축의 일례를 나타내는 도면이다.

도 17은 가상 파일 시스템부(38)에 의한 상태 관리의 일례를 나타내는 도면이다.

도 18은 상태 관리를 반영한 가상 패키지 정보의 작성을 나타내는 도면이다.

도 19는 네트워크 관리모듈(37)에 의한 다운로드가 진행중인 경우의 가상 파일 시스템부(38)에 의한 가상 패키지 구축을 나타내는 도면이다.

도 20(a)은 로컬 기억장치에 저장된 PL 정보에 의해, 어떠한 Playlist 재생시간 축이 정의되는가를 나타내는 도면이다.

(b)는 Progressive Playlist 정보의 개념을 나타내는 도면이다.

도 21은 관리정보 변환모듈(39)에 의한 인스턴스의 생성을 나타내는 도면이다.

도 22는 Playlist 오브젝트에서의 Play 메소드의 호출을 나타내는 도면이다.

도 23은 getMissingStreams 메소드에 의한 문의의 결과를 모식적으로 나타내는 도면이다.

도 24는 enableClip 메소드의 호출을 모식적으로 나타내는 도면이다.

도 25는 disableClip 메소드의 호출을 모식적으로 나타내는 도면이다.

도 26은 가상 패키지 구축 시의 처리를 나타내는 플로차트이다.

도 27(a)은 enableClip 메소드의 호출 시의 처리순서를 나타내는 플로차트이다. (b)는 disableClip 메소드의 호출 시의 처리순서를 나타내는 플로차트이다.

도 28은 재생제어엔진(Playback Control Engine, 32)에 의한 재생순서를 나타내는 흐름도이다.

도 29는 애플리케이션의 처리순서를 나타내는 플로차트이다.

도 30(a), (b)는 네트워크 관리모듈(37)에 의한 다운로드의 진척과 재생제어엔진(32)에 의한 재생의 진척을 나타내는 도면이다.

도 31(a), (b)는 네트워크 관리모듈(37)에 의한 다운로드의 진척과 재생제어엔진(32)에 의한 재생의 진척을 나타내는 도면이다.

도 32(a), (b)는 네트워크 관리모듈(37)에 의한 다운로드의 진척과 재생제어엔진(32)에 의한 재생의 진척을 나타내는 도면이다.

도 33은 네트워크 관리모듈(37)에 의한 다운로드의 진척과 재생제어엔진(32)에 의한 재생의 진척을 나타내는 도면이다.

발명을 실시하기 위한 구체적인 내용

(제 1 실시 예)

이하, 본 발명에 관한 기록매체의 실시 예에 대하여 설명한다. 먼저, 본 발명에 관한 재생장치의 실시행위 중 사용행위에 대한 형태를 설명한다. 도 1은 본 발명에 관한 재생장치의 사용행위에 대한 형태를 나타내는 도면이다. 도 1에서, 본 발명에 관한 재생장치는 재생장치 200이다. 이 재생장치(200)는 리모컨(300) 및 텔레비전(400)에 의해 형성되는 홈시어터 시스템에 영화작품을 공급하는 용도로 제공된다.

이상이 본 발명에 관한 재생장치의 사용형태에 대한 설명이다. 이어서 본 발명에 관한 재생장치가 재생의 대상으로 하고 있는 기록매체에 대하여 설명한다. 본 발명에 관한 재생장치에 의해 재생되는 것은 BD-ROM이다. 도 2는 BD-ROM의 내부 구성을 나타내는 도면이다.

본 도면의 제 4단계에 BD-ROM을 나타내고, 제 3단계에 BD-ROM 상의 트랙을 나타낸다. 본 도면의 트랙은 BD-ROM의 내주에서 외주에 걸쳐서 나선형상으로 형성되어 있는 트랙을 횡 방향으로 확대하여 묘사하고 있다. 이 트랙은 리드인 영역과 볼륨영역과 리드 아웃 영역으로 이루어진다. 본 도면의 볼륨영역은 물리 층, 파일 시스템 층, 응용 층이라는 계층모델(layer model)을 갖는다. 디렉터리 구조를 사용하여 BD-ROM의 응용 층 포맷(애플리케이션 포맷)을 실현하면 도면 중의 제 1단계와 같이 된다. 이 제 1단계에서 BD-ROM에는 루트 디렉터리의 아래에 BDMV 디렉터리가 있다.

BDMV 디렉터리의 하위에는 PLAYLIST 디렉터리, CLIPINF 디렉터리, STREAM 디렉터리, BDDA 디렉터리라고 불리는 4개의 서브 디렉터리가 존재한다.

STREAM 디렉터리는, 소위 디지털 스트림 본체가 되는 파일을 저장하고 있는 디렉터리이고, 확장자 M2TS가 부여된 파일(00001.M2TS)이 존재한다.

PLAYLIST 디렉터리에는 확장자 MPLS가 부여된 파일(00001.MPLS)이 존재한다.

CLIPINF 디렉터리에는 확장자 CLPI가 부여된 파일(00001.CLPI)이 존재한다. 이하에 이들 파일에 대하여 설명한다.

<AVClip의 구성>

먼저, 확장자 .M2TS가 부여된 파일에 대하여 설명한다. 도 3은 확장자 .M2TS가 부여된 파일이 어떻게 구성되어 있는가를 모식적으로 나타내는 도면이다. 확장자 .M2TS가 부여된 파일(00001.M2TS, 00002.M2TS, 00003.M2TS...)은 AVClip을 저장하고 있다. AVClip은(중간 단) 복수의 비디오 프레임(픽처(picture) pj 1, 2, 3)으로 이루어지는 비디오 스트림 및 복수의 오디오 프레임으로 이루어지는 오디오 스트림을(상 1단계) PES 패킷 열로 변환하고(상 2단계), 다시 TS 패킷으로 변환하며(상 3단계), 동일한 자막계의 프레젠테이션 그래픽스 스트림

(Presentation Graphics stream, 하 1단계의 PG 스트림) 및 대화 계의 인터랙티브 그래픽스 스트림 (Interactive Graphics stream, 하 2단계의 IG 스트림)을 TS 패킷으로 변환하여(하 3단계), 이들을 다중화함으로써 구성된다.

- [0025] AVClip은, 도 3에 도시한 바와 같이, 다중화가 이루어져 있는 스트림 외에 다중화가 이루어져 있지 않은 것도 있다. 이와 같은 AVClip은 SubClip이라고 불리며, 오디오 스트림, 그래픽스 스트림, 텍스트 자막 스트림(TextSTStream) 등을 구성하는 AVClip이 존재한다.
- [0026] <Clip 정보>
- [0027] 확장자 "CLPI"가 부여된 파일(00001.CLPI)은 AVClip의 각각에 1대 1로 대응하는 Clip 정보이다. 관리정보이므로, Clip 정보는 AVClip에서의 스트림의 부호화 형식, 프레임 비율, 비트 비율, 해상도 등의 정보나 GOP의 선두위치를 나타내는 EP_map을 가지고 있다.
- [0028] <PlayList 정보>
- [0029] 확장자 "MPLS"가 부여된 파일(00001.MPLS)은 PlayList(PL) 정보를 저장한 파일이다. PL 정보는 AVClip을 참조하여 플레이 리스트를 정의하는 정보이다. 도 4는 PL 정보의 구성을 나타내는 도면이며, 본 도면의 좌측에 도시하는 바와 같이, PL 정보는 「MainPath 정보」, 「PLMark 정보」, 「SubPath 정보」로 구성된다.
- [0030] MainPath 정보(MainPath())는, 점선 화살표 mp 1에 도시하는 바와 같이, 복수의 PlayItem 정보(PlayItem())로 이루어진다. PlayItem은 하나 이상의 AVClip 시간 축 상에서 In_Time, Out_Time을 지정함으로써 지정되는 재생 구간이다. PlayItem 정보를 복수 배치함으로써 복수의 재생구간으로 이루어지는 플레이 리스트(PL)가 정의된다. 도면 중의 점선 mp 2는 PlayItem 정보의 내부 구성을 클로즈업하고 있다. 본 도면에 도시하는 바와 같이, PlayItem 정보는 대응하는 AVClip을 나타내는 「Clip_information_file_name」과 「In_time」 및 「Out_time」으로 이루어진다. 도 5는 AVClip과 PL의 관계를 나타내는 도면이다. 제 1단계는 AVClip이 갖는 시간 축을 나타내고, 제 2단계는 PL이 갖는 시간 축을 나타낸다. PL 정보는 PlayItem #1, #2, #3이라는 3개의 PlayItem 정보를 포함하고 있고, 이들 PlayItem #1, #2, #3의 In_time, Out_time에 의해 3개의 재생구간이 정의되게 된다. 이들 재생구간을 배열시키면 AVClip 시간 축과는 다른 시간 축이 정의되게 된다. 이것이 제 2단계에 나타내는 PL 시간 축이다. 이와 같이 PlayItem 정보의 정의에 의해 AVClip과는 다른 시간 축의 정의가 가능해진다.
- [0031] AVClip에 대한 지정은 원칙적으로 하나이나, 복수의 AVClip에 대한 일괄지정도 있을 수 있다. 이 일괄지정은 PlayItem 정보에서의 복수의 Clip_Information_file_name에 의해 이루어진다. 도 6은 4개의 Clip_Information_file_name에 의해 이루어진 일괄지정을 나타내는 도면이다. 본 도면에서, 제 1단계 ~ 제 4단계는 4개의 AVClip 시간 축(AVClip #1, #2, #3, #4의 시간 축)을 나타내고, 제 5단계는 PL 시간 축을 나타낸다. PlayItem 정보가 갖는 4개의 Clip_Information_file_name에 의해서 이들 4개의 시간 축이 지정되어 있다. 이렇게 함으로써, PlayItem이 갖는 In_time, Out_time에 의해 택일적으로 재생 가능한 4개의 재생구간이 정의되게 된다. 이에 의해, PL 시간 축에는 전환(switching) 가능한 복수의 앵글 영상(angle video)으로 이루어지는 구간(소위, 멀티앵글구간)이 정의되게 된다.
- [0032] PLmark 정보(PLmark())는 PL 시간 축 중 임의의 구간을 챕터로 지정하는 정보이다. 도 7은 PLmark 정보의 내부 구성을 나타내는 도면이며, 본 도면의 인출선 pm 1에 도시하는 바와 같이, PLmark 정보는 「ref_to_PlayItem_Id」와 「Mark_time_stamp」를 포함한다. 도 8은 PLmark에 의한 챕터 정의를 나타내는 도면이다. 본 도면에서 제 1단계는 AVClip 시간 축을 나타내고, 제 2단계는 PL 시간 축을 나타낸다. 도면 중의 화살표 pk 1, 2는 PLmark에서의 PlayItem 지정(ref_to_PlayItem_Id)과 일 시점의 지정(mark_time_stamp)을 나타낸다. 이들 지정에 의해 PL 시간 축에는 3개의 챕터(Chapter #1, #2, #3)가 정의되게 된다. 이상이 PLmark에 대한 설명이다. 이어서 SubPath 정보에 대하여 설명한다.
- [0033] SubPath 정보(SubPath())는, SubClip의 시간 축 상에 In_Time, Out_Time을 지정함으로써 하나 이상의 재생구간을 정의하는 정보이며, 도 9에 도시한 내부 구성을 가지고 있다. 본 도면에 도시하는 바와 같이 SubPath 정보는, 점선의 인출선 sh 1에 나타내는 바와 같이, 복수의 SubPlayItem 정보(SubPlayItem())로 이루어진다. 점선 sh 2를 사용하여 클로즈업하고 있는 바와 같이, SubPlayItem 정보는 「Clip_information_file_name」과, 「In_time」 및 「Out_time」과, 「Sync_PlayItem_Id」와, 「Sync_start_Pts_of_PlayItem」으로 이루어진다. SubClip의 시간 축 상에 대한 In_Time, Out_time의 지정은 「Clip_information_file_name」, 「In_time」, 「Out_time」에 의해 이루어진다. 「Sync_PlayItem_Id」 및 「Sync_start_Pts_of_PlayItem」은 SubClip 시간 축 상의 재생구간과 PL 시간 축을 동기화(synchronization)시키는 동기지정을 이룬다. 이 동기지정에 의해 PL 시간

축과 SubClip 시간 축은 동기 하여 진행하게 된다.

- [0034] 도 10은 SubClip 시간 축 상의 재생구간 정의와 동기지정을 나타내는 도면이다. 본 도면에서 제 1단계는 PL 시간 축을 나타내고, 제 2단계는 SubClip 시간 축을 나타낸다. 도면 중의 SubPlayItem.IN_time은 재생구간의 시작점을, SubPlayItem.Out_time은 재생구간의 종점을 각각 나타낸다. 이에 의해 SubClip 시간 축 상에도 재생구간이 정의되고 있음을 알 수 있다. 화살표 Sn 1에서 Sync_PlayItem_Id는 PlayItem에 대한 동기지정을 나타내고, 화살표 Sn 2에서 sync_start_PTS_of_PlayItem은 PL 시간 축에서의 PlayItem 상의 일시점의 지정을 나타낸다.
- [0035] 복수의 AVClip의 전환을 가능하게 하는 멀티앵글구간이나 AVClip-SubClip을 동기시킬 수 있는 동기구간의 정의를 가능하게 하는 것이 BD-ROM에서의 PL 정보의 특징이다. 이상의 Clip 정보 및 PL 정보는 "정적 시나리오"로 분류된다.
- [0036] 이어서, "동적 시나리오"에 대하여 설명한다. "동적"이라는 것은 재생장치(200)에서의 상태변화나 사용자로부터의 키 이벤트(key event)에 의해 재생제어의 내용이 변하는 것을 말한다. BD-ROM에서는 Java(등록상표) 애플리케이션과 동일한 기술(description)에 의해 이 재생제어를 기술할 수 있다. 즉 BD-ROM에서는 Java(등록상표) 애플리케이션이 동적 시나리오로서의 역할을 담당하게 된다.
- [0037] <Java(등록상표) 애플리케이션>
- [0038] Java(등록상표) 애플리케이션에 대하여 설명한다. Java(등록상표) 애플리케이션은 가상 머신의 힙 영역(heap area)(워크 메모리(work memory)라고도 한다)에 로드된 하나 이상의 xlet 프로그램으로 이루어진다. 이 워크 메모리에 로드된 xlet 프로그램 및 데이터로 애플리케이션이 구성되게 된다. 이상이 Java(등록상표) 애플리케이션의 구성이다.
- [0039] 이 Java(등록상표) 애플리케이션의 실체에 해당하는 것이 도 2에서의 BDMV 디렉터리의 하위의 BDJA 디렉터리에 저장된 Java(등록상표) 아카이브 파일(00001.jar, 00002.jar)이다. 이하, Java(등록상표) 아카이브 파일에 대하여, 도 11을 참조하면서 설명한다.
- [0040] <Java(등록상표) 아카이브 파일>
- [0041] *Java(등록상표) 아카이브 파일(도 9의 00001.JAR)은 하나 이상의 클래스 파일 및 하나 이상의 데이터 파일 등을 하나로 집약함으로써 얻을 수 있는 파일이다. 도 11(a)은 아카이브 파일에 의해 집약되어 있는 프로그램 및 데이터를 나타내는 도면이다. 본 도면에서의 데이터는 테두리(frame) 내에 도시된 디렉터리 구조가 배치된 복수 파일을 java(등록상표) 아카이버(archiver)에 의해 집약한 것이다. 테두리 내에 도시된 디렉터리 구조는 루트 디렉터리(root directory), java(등록상표) 디렉터리, 이미지 디렉터리(image directory)로 이루어지며, 루트 디렉터리에 common.pkg가, java(등록상표) 디렉터리에 클래스 파일(aaa.class, bbb.class)이, 이미지 디렉터리에 menu.jpg가 배치되어 있다. java(등록상표) 아카이브 파일은 이들을 java(등록상표) 아카이버로 정리함으로써 얻을 수 있다. 이와 같은 클래스 파일 및 데이터는 BD-ROM에서 캐시(cache)에 판독됨으로써 전개되고, 캐시 상에서 디렉터리에 배치된 복수 파일로 취급된다. Java(등록상표) 아카이브 파일의 파일명에서의 "zzzzz"이라는 5자리의 수치는 애플리케이션의 ID(applicationID)를 나타낸다. 본 Java(등록상표) 아카이브 파일이 캐시에 판독된 때, 이 파일명에서의 수치를 참조함으로써 임의의 Java(등록상표) 애플리케이션을 구성하는 프로그램 및 데이터를 인출할 수 있다.
- [0042] 본 도면에서의 클래스 파일(도면 중의 aaa.class, bbb.class)은 상술한 xlet 프로그램에 대응하는 클래스 파일이다. Java(등록상표) 동작환경에 의한 동작 모드(BD-J 모드)에서의 재생순서는 이 클래스 파일의 인스턴스(instance)에 해당하는 xlet 프로그램에 의해 규정된다. xlet 프로그램이란, JMF(Java(등록상표) Media FrameWork) 방식의 인터페이스를 이용할 수 있는 Java(등록상표) 프로그램이며, JMF 등의 방식에 따라서 키 이벤트(key event)에 의거한 처리를 행한다. xlet 프로그램은, JMF 방식의 처리가 가능하므로, MPLS 파일에 대한 JMF 방식의 인스턴스(Playlist Object)를 생성함으로써 플레이 리스트 재생을 재생장치(200)에 명령할 수 있다. 그 외에도, xlet 프로그램에서는 기능 API(function API)의 콜(call)을 기술함으로써 BD-ROM 재생장치(200) 특유의 처리를 실행할 수 있다.
- [0043] 또한, xlet 프로그램은 WWW 사이트를 액세스하여 콘텐츠를 다운로드 하는 순서를 실행할 수도 있다. 이에 의해 다운로드 콘텐츠와 플레이 리스트 재생을 접목한 참신한 작품을 재생하도록 할 수 있다.
- [0044] xlet 프로그램의 클래스 파일에 대하여 설명한다. 도 11(b)은 클래스 파일의 내부 구성을 나타내는 도면이다.

본 도면에 도시하는 바와 같이, 클래스 파일은 통상의 클래스 파일과 마찬가지로, 「컨스턴트 풀(constant pool)」, 「인터페이스」 및 「메소드(method) 1, 2, 3 ... n」으로 이루어진다. 클래스 파일에서의 메소드에는 거동의 트리거(trigger)가 되는 키 이벤트가 미리 등록되어 있는 메소드(EventListner)와 BD-ROM 재생장치(200) 측의 기능 API를 콜하는 메소드가 있다. 이들 메소드는 자신에게 할당된 로컬 변수나 자신을 콜할 때의 인수를 사용함으로써 연산 등의 순서가 기술되어 있다. 이상이 Java(등록상표) 아카이브 파일에 대한 설명이다. 이것으로 BD-ROM에 대한 설명을 마친다. 이어서 본 발명에 관한 재생장치의 내부 구성에 대하여 설명한다.

[0045] <재생장치>

[0046] 도 12는 본 발명에 관한 재생장치의 내부 구성을 나타내는 도면이다. 본 발명에 관한 재생장치는 본 도면에 도시된 내부에 의거하여 공업적으로 생산된다. 본 발명에 관한 재생장치는 주로 시스템 LSI와 드라이브 장치라는 2개의 파트로 이루어지고, 이들 파트를 장치의 캐비닛 및 기판에 실장 함으로써 공업적으로 생산할 수 있다. 시스템 LSI는 재생장치의 기능을 달성하는 다양한 처리부를 집적한 집적회로이다. 이렇게 하여 생산되는 재생장치는 BD-ROM 드라이브(1), 리드 버퍼(Read Buffer, 2), 디 멀티플렉서(De-multiplexor, 3), 비디오 디코더(4), 비디오 플레인(video plane, 5), P-Graphics 디코더(6), 프레젠테이션 그래픽스 플레인(Presentation Graphics Plane, 7), 폰트 제네레이터(Font Generator, 9), I-Graphics 디코더(10), 스위치(11), 인터랙티브 그래픽스 플레인(Interactive Graphics Plane, 12), 합성부(13), CLUT부(14), CLUT부(15), 오디오 디코더(16), 네트워크 장치(Network Device, 17), 로컬 기억장치(Local Storage, 18), 리드 버퍼(19), 디 멀티플렉서(20), 명령 ROM(21), 사용자 이벤트 처리부(22), PSR 세트(23), CPU(24), 시나리오 메모리(25), 로컬 메모리(26), 스위치(27)로 이루어진다.

[0047] 먼저, BD-ROM에 기록된 AVClip 재생에 관한 구성요소(BD 드라이브(1) ~ 오디오 디코더(16))에 대하여 설명한다.

[0048] BD-ROM 드라이브(1)는 BD-ROM의 로딩/이젝트를 행하며, BD-ROM에 대한 액세스를 실행한다.

[0049] 리드 버퍼(2)는 FIFO 메모리로, BD-ROM으로부터 판독된 TS 패킷이 선입 선출 방식으로 저장된다.

[0050] 디 멀티플렉서(De-MUX, 3)는 리드 버퍼(2)에서 TS 패킷을 인출하여, 이 TS 패킷을 구성하는 TS 패킷을 PES 패킷으로 변환한다. 그리고 변환에 의해 얻어진 PES 패킷 중 CPU(24)에서 설정된 PID를 갖는 것을 비디오 디코더(4), P-Graphics 디코더(6), I-Graphics 디코더(10), 오디오 디코더(16) 중 어느 하나에 출력한다.

[0051] 비디오 디코더(4)는 디 멀티플렉서(3)로부터 출력된 복수의 PES 패킷을 복호 하여 비 압축형식의 픽처를 얻어서 비디오 플레인(5)에 기록한다.

[0052] 비디오 플레인(5)은 비 압축형식의 픽처를 저장해 두기 위한 플레인이다. 플레인은 재생장치에서 한 화면 분의 화소 데이터를 저장해 두기 위한 메모리 영역이다. 비디오 플레인(5)에서의 해상도는 1920×1080 이고, 이 비디오 플레인(5)에 저장된 픽처 데이터는 16비트의 YUV 값으로 표현된 화소 데이터에 의해 구성된다. 비디오 플레인(5)에서는 비디오 스트림에서의 1 프레임 단위의 재생 영상을 스케일링(scaling)할 수 있다. 스케일링이란, 1 프레임 단위의 재생 영상을 비디오 플레인(5) 전체의 1/4(쿼터라고 한다) 또는 1/1(풀 스케일이라고 한다) 중 어느 하나로 변화시키는 것이다. 이와 같은 스케일링을 BD-J 모드에서 CPU(24)로부터의 지시에 따라서 실행하므로, 비디오 스트림의 재생 영상을 화면의 구석으로 내보내거나, 전면에 표시하는 화면 연출을 할 수 있게 된다.

[0053] P-Graphics 디코더(6)는 BD-ROM으로부터 판독된 프레젠테이션 그래픽스 스트림을 디코드하여, 비 압축 그래픽스를 프레젠테이션 그래픽스 플레인(7)에 기록한다. 그래픽스 스트림의 디코드에 의해 자막이 화면상에 나타나게 된다.

[0054] 프레젠테이션 그래픽스 플레인(7)은 한 화면 분의 영역을 갖는 메모리이며, 한 화면 분의 비 압축 그래픽스를 저장할 수 있다. 본 플레인에서의 해상도는 1920×1080 이고, 프레젠테이션 그래픽스 플레인(7) 중의 비 압축 그래픽스의 각 화소는 8비트의 인덱스 컬러(index color)로 표현된다. CLUT(Color Lookup Table)를 사용하여 이와 같은 인덱스 컬러를 변환함으로써 프레젠테이션 그래픽스 플레인(7)에 저장된 비 압축 그래픽스는 표시에 제공된다.

[0055] 합성부(8)는 비디오 플레인(5)에 저장된 비 압축상태의 픽처 데이터(i)를 프레젠테이션 그래픽스 플레인(7)의 저장내용과 합성한다.

[0056] 폰트 제네레이터(9)는 문자 폰트를 사용하여 textST 스트림에 포함되는 텍스트 코드를 비트맵으로 전개하여 프레젠테이션 그래픽스 플레인(7)에 기록한다.

- [0057] I-Graphics 디코더(10)는 HDMV 모드에서 BD-ROM 또는 로컬 기억장치(18)로부터 관독된 IG 스트림을 디코드하여 비 압축 그래픽스를 인터랙티브 그래픽스 플레인(12)에 기록한다.
- [0058] 스위치(11)는 폰트 제네레이터(9)가 생성한 폰트 열 또는 P-Graphics 디코더(6)의 디코드에 의해 얻어진 그래픽스 중 어느 하나를 선택적으로 프레젠테이션 그래픽스 플레인(7)에 기록하는 스위치이다.
- [0059] 인터랙티브 그래픽스 플레인(12)은 I-Graphics 디코더(10)에 의한 디코드에 의해 얻어진 비 압축 그래픽스가 기록된다. 또, 인터랙티브 그래픽스 플레인(12)에는 BD-J 모드에서 애플리케이션에 의해 묘사된 문자나 그래픽스가 기록된다.
- [0060] 합성부(13)는 인터랙티브 그래픽스 플레인(12)의 저장내용과 합성부(8)의 출력인 합성화상(비 압축상태의 픽처 데이터와 프레젠테이션 그래픽스 플레인(7)의 저장내용을 합성한 것)을 합성한다. 이와 같은 합성에 의해 애플리케이션이 I-Graphics 디코더(10)에 기록한 문자 및 그래픽스를 비 압축상태의 픽처 데이터 상에 오버레이하여 표시할 수 있다.
- [0061] CLUT부(14)는 비디오 플레인(5)에 저장된 비 압축 그래픽스에서의 인덱스 컬러를 Y, Cr, Cb 값으로 변환한다.
- [0062] CLUT부(15)는 인터랙티브 그래픽스 플레인(12)에 저장된 비 압축 그래픽스에서의 인덱스 컬러를 Y, Cr, Cb 값으로 변환한다.
- [0063] 오디오 디코더(16)는 디 멀티플렉서(3)로부터 출력된 PES 패킷을 복호 하여 비 압축형식의 오디오 데이터를 출력한다.
- [0064] 이상이 AVClip 재생에 관한 구성요소이다. 이어서 BD-J 모드에서의 동작에 관한 구성요소(네트워크 장치(17) ~ De-mux(20))에 대하여 설명한다.
- [0065] 네트워크 장치(17)는 재생장치에서의 통신기능을 실현하는 것으로, BD-J 모드에서 URL 지정이 Java(등록상표) 애플리케이션으로부터 주어지면, 그 URL에 해당하는 웹 사이트와의 TCP 커넥션, FTP 커넥션 등을 확립한다. 이와 같은 커넥션의 확립에 의해 웹 사이트로부터의 다운로드를 Java(등록상표) 애플리케이션이 행하도록 한다.
- [0066] 로컬 기억장치(18)는 네트워크 장치(17)에 의해 확립된 커넥션을 통하여 웹 사이트로부터 다운로드 된 콘텐츠 등, BD-ROM 이외의 기록매체나 통신매체로부터 공급된 콘텐츠를 메타데이터(metadata)와 함께 저장해 두기 위한 하드 디스크이다. 이 메타데이터는 다운로드 콘텐츠를 로컬 기억장치(18)에 바인드하여(binding) 관리하기 위한 정보로, 이 로컬 기억장치(18)를 액세스함으로써, BD-J 모드에서의 애플리케이션은 다운로드 콘텐츠를 이용한 다양한 처리를 행할 수 있다.
- [0067] 리드 버퍼(19)는 FIFO 메모리이며, 로컬 기억장치(18)에 저장된 다운로드 콘텐츠에 SubClip이 포함되어 있는 경우에, 이 SubClip을 구성하는 TS 패킷을 선입 선출 방식으로 저장한다.
- [0068] 디 멀티플렉서(De-MUX, 20)는 리드 버퍼(19)에서 TS 패킷을 인출하여 TS 패킷을 PES 패킷으로 변환한다. 그리고 변환에 의해 얻어진 PES 패킷 중 원하는 PID를 갖는 것을 폰트 제네레이터(9), I-Graphics 디코더(10), 오디오 디코더(16)에 출력한다.
- [0069] *이상의 네트워크 장치(17) ~ De-mux(20)에 의해, Java(등록상표) 애플리케이션이 네트워크를 통하여 다운로드 한 콘텐츠를 BD-ROM에 기록된 콘텐츠와 마찬가지로 재생시킬 수 있다. 이어서 재생장치에서의 통합제어를 실현하는 구성요소(명령 ROM(21) ~ 스위치(27))에 대하여 설명한다.
- [0070] 명령 ROM(21)은 재생장치의 제어를 규정하는 소프트웨어를 기억하고 있다.
- [0071] 사용자 이벤트 처리부(22)는 리모컨이나 재생장치의 프런트 패널에 대한 키 조작에 따라서 그 조작을 행하는 사용자 이벤트를 CPU(24)에 출력한다.
- [0072] PSR 세트(23)는 재생장치에 내장되는 레지스터이며, 64개의 플레이어 스테이터스 레지스터(Player Status Register, PSR)와 4096개의 제너럴 퍼포즈 레지스터(General Purpose Register, GPR)로 이루어진다. 플레이어 스테이터스 레지스터의 설정 값(PSR) 중 PSR 4 ~ PSR 8은 현재의 재생시점을 표현하는 것에 사용된다.
- [0073] PSR 4는, 1 ~ 100의 값으로 설정됨으로써 현재의 재생시점이 속하는 타이틀을 나타내고, 0으로 설정됨으로써 현재의 재생시점이 톱 메뉴임을 나타낸다.

- [0074] PSR 5는, 1 ~ 999의 값으로 설정됨으로써 현재의 재생시점이 속하는 챕터번호를 나타내고, 0xFFFF로 설정됨으로써 재생장치에서 챕터번호가 무효임을 나타낸다.
- [0075] *PSR 6은 0 ~ 999의 값으로 설정됨으로써 현재의 재생시점이 속하는 PL(현재 PL)의 번호를 나타낸다.
- [0076] PSR 7은 0 ~ 255의 값으로 설정됨으로써 현재의 재생시점이 속하는 PlayItem(현재 Play Item)의 번호를 나타낸다.
- [0077] PSR 8은 0 ~ 0xFFFFFFFF의 값으로 설정됨으로써 45KHz의 시간 정밀도를 사용하여 현재의 재생시점(현재 PTM(Presentation Time))을 나타낸다. 이상의 PSR 4 ~ PSR 8에 의해 도 21(a)에서의 BD-ROM 전체의 시간 축에서 현재의 재생시점이 어디인가를 특정할 수 있다.
- [0078] CPU(24)는 명령 ROM(21)에 저장되어 있는 소프트웨어를 실행하여 재생장치 전체의 제어를 실행한다. 이 제어의 내용은 사용자 이벤트 처리부(22)로부터 출력된 사용자 이벤트 및 PSR 세트(23)에서의 각 PSR의 설정 값에 따라서 동적으로 변화한다.
- [0079] 시나리오 메모리(25)는 현재의 PL 정보나 현재의 Clip 정보를 저장해 두기 위한 메모리이다. 현재 PL 정보는 BD-ROM에 기록되어 있는 복수 PL 정보 중 현재 처리대상이 되어 있는 것을 말한다. 현재 Clip 정보는 BD-ROM에 기록되어 있는 복수 Clip 정보 중 현재 처리대상이 되어 있는 것을 말한다.
- [0080] 로컬 메모리(26)는, BD-ROM으로부터의 판독이 지속이므로, BD-ROM의 기록내용을 일시적으로 저장해 두기 위한 캐시 메모리이다. 이와 같은 로컬 메모리(26)가 존재함으로써 BD-J 모드에서의 애플리케이션의 실행이 효율화되게 된다.
- [0081] 스위치(27)는 BD-ROM 및 로컬 기억장치(18)로부터 판독된 각종 데이터를 리드 버퍼(2), 리드 버퍼(19), 시나리오 메모리(25), 로컬 메모리(26) 중 어느 하나에 선택적으로 투입하는 스위치이다.
- [0082] 이상이 본 실시 예에 관한 재생장치의 하드웨어 구성이다. 이어서 본 실시 예에 관한 재생장치에서의 파일 시스템의 구조에 대하여 설명한다.
- [0083] 도 13(a)은 로컬 기억장치(18)에서의 디렉터리 구조를 나타내는 도면이다.
- [0084] 본 도면의 디렉터리 구조에서, 루트 디렉터리의 하위에는 「DLDAT」라는 서브 디렉터리가 있고, 그 하위에 「PKG00001」, 「PKG00002」라는 서브 디렉터리가 있다. 「PKG00001」, 「PKG00002」는 각각이 BD-ROM의 볼륨 라벨(volume label)이며, 「PKG00001」이라는 볼륨 라벨을 가진 BD-ROM에 대응하는 디렉터리, 및 「PKG00002」라는 볼륨 라벨의 BD-ROM에 대응한 디렉터리가 로컬 기억장치(18)에 존재하는 것을 의미한다. 그 중, 서브 디렉터리 「PKG00001」는 볼륨 라벨 「PKG00001」의 BD-ROM에 대응하는 다운로드 데이터가 놓이는 장소라는 것을 나타낸다. 각 BD-ROM에 대응하는 디렉터리에 서브 디렉터리를 배치함으로써 각 BD-ROM에 대한 다운로드 데이터가 개별로 저장된다. 이 서브 디렉터리의 하위에, BD-ROM에 저장하고 있었던 것과 마찬가지로, Playlist 정보, Clip 정보 및 AVClip이 저장되어 있고, 이 외에 Java(등록상표) 아카이브 파일이나 스트림 저장위치정보 파일이 존재한다.
- [0085] 여기서, BD-ROM에 기록되어 있었던 AVClip이 영화작품의 본편에 해당하는 것에 대하여, 로컬 기억장치(18)상에 기록되어 있는 AVClip은 현재 극장에서 상영되고 있는 영화작품의 예고편인 것으로 한다. 예고편의 내용은 시대의 변천에 따라서 변화하므로, 현재 현재극장에서 상영되고 있는 영화작품의 예고편이 로컬 기억장치(18)상에 AVClip으로 기록되어 있는 것이다.
- [0086] 후자의 스트림 저장위치정보 파일에 대하여 설명한다. 이 스트림 저장위치정보 파일은 가상 패키지를 구성하는 파일의 일람을 나타내는 정보이다. 도 13(b)은 스트림 저장위치정보 파일의 내부 구성을 나타내는 도면이며, 스트림 저장위치정보 파일은 가상 패키지를 구성하는 각 AVClip 및 Clip 정보의 저장위치정보로 이루어진다. 각 저장위치정보는 AVClip 및 Clip 정보의 "식별자"와, 로컬 기억장치(18)에서의 그 AVClip 및 Clip 정보의 저장위치를 나타내는 "파일 패스(file path)"로 이루어진다. 이와 같은 저장위치정보를 참조함으로써, 가상 패키지를 구축할 때에 가상 패키지의 구성요소가 되는 AVClip 및 Clip 정보가 구비되어 있는가를 확인할 수 있다.
- [0087] 이상이 본 실시 예에 관한 재생장치의 파일 시스템의 구성이다. 이어서 본 실시 예에 관한 재생장치에서의 소프트웨어 구조에 대하여 설명한다.

- [0088] 도 14는 ROM(24)에 저장된 소프트웨어와 하드웨어로 이루어지는 부분을 계층 구성으로 치환하여 묘사한 도면이다. 본 도면에 도시하는 바와 같이, 재생장치의 계층 구성은 이하의 a), b), c)로 이루어진다. 즉,
- [0089] a) BD Player Device의 제 1 계층,
- [0090] b) BD Player Model의 제 2 계층, 및
- [0091] c) Application Runtime Environment의 제 3 계층으로 이루어진다.
- [0092] 이들 계층 중, 도 32에 도시한 재생장치의 하드웨어 구성은 제 1 계층에 속하게 된다. 본 도면의 제 1 계층인 "BD Player Device"에는, 도 12에 도시된 하드웨어 구성 중 비디오 디코더(4), P-Graphics 디코더(6), I-Graphics 디코더(10), 오디오 디코더(16)로 이루어지는 "디코더"와, 비디오 플레인(5), 프레젠테이션 그래픽스 플레인(7), 인터랙티브 그래픽스 플레인(12)으로 이루어지는 "플레인", BD-ROM 및 그 파일 시스템, 로컬 기억장치(18) 및 그 파일 시스템을 포함한다.
- [0093] 제 2 계층인 "BD Player Model"은 이하의 b1), b2)의 층으로 이루어진다. 즉,
- [0094] b2) 재생제어엔진(32)의 층
- [0095] b1) 가상 파일 시스템(38) 및 프레젠테이션 엔진(presentation engine, 31)의 층
- [0096] 으로 이루어지며, 자신보다 상위의 계층에 대하여 기능 API를 제공한다.
- [0097] 이 중, 도 12에 도시한 PSR 세트(23)나 시나리오 메모리(25)는 재생제어엔진(32) 내에 존재하게 된다.
- [0098] 제 3 계층인 "Application Runtime Environment"는 이하의 c1)의 스택 계층으로 이루어진다. 즉,
- [0099] c1) HDMV 모듈(29a), BD-J 모듈(29b)이 존재하는 층
- [0100] 으로 이루어진다.
- [0101] 이하, 이 소프트웨어 구조에서의 각 구성요소에 대하여 설명한다.
- [0102] <HDMV 모듈(29a), BD-J 모듈(29b)>
- [0103] HDMV 모듈(29a)은 내비게이션 커맨드(navigation command)를 해독하고, 해독결과에 의거하여 재생제어엔진(32)에 대한 기능 콜(function call)을 실행한다.
- [0104] BD-J 모듈(29b)은 소위 Java(등록상표) 플랫폼으로,
- [0105] d1-1) Java(등록상표) 가상 머신(30),
- [0106] d1-2) Java(등록상표) 가상 머신이 동작하기 위한 미들웨어(middle-ware)
- [0107] 를 계층화한 구성으로 되어 있다.
- [0108] <Java(등록상표) 가상 머신(30)>
- [0109] Java(등록상표) 가상 머신(30)은 애플리케이션을 구성하는 xlet 프로그램을 워크 메모리에 로드하여 xlet 프로그램을 해독하고, 해독결과에 따라서 하위층에 대한 제어를 행한다. 하위층에 대한 제어는, 메소드를 미들웨어에 발행하여, BD 재생장치가 대응하고 있는 기능 콜로 치환하도록 하고, 치환 후의 기능 콜을 재생제어엔진(32)에 발행함으로써 이루어진다.
- [0110] (Java(등록상표) 가상 머신(30)의 내부 구성)
- [0111] 여기서 Java(등록상표) 가상 머신(30)의 내부 구성에 대하여 설명한다. 도 15는 Java(등록상표) 가상 머신(30)의 내부 구성을 나타내는 도면이다. 본 도면에 도시하는 바와 같이, Java(등록상표) 가상 머신(30)은, 도 12에 도시한 CPU(24)와, 사용자 클래스 로더(user class loader, 52), 메소드 영역(53), 워크 메모리(54), 스레드(thread) (55a, b ... n), Java(등록상표) 스택(56a, b ... n)으로 구성된다.
- [0112] 사용자 클래스 로더(52)는 BDJA 디렉터리의 Java(등록상표) 아카이브 파일에서의 클래스 파일을 로컬 메모리(26) 등으로부터 판독하여 메소드 영역(53)에 저장한다. 이 사용자 클래스 로더(52)에 의한 클래스 파일의 판독은, 애플리케이션 매니저(36)가 사용자 클래스 로더(52)에 파일 패스를 지정한 판독을 지시함으로써 이루어진다. 파일 패스가 로컬 메모리(26)를 가리키고 있으면, 사용자 클래스 로더(52)는 애플리케이션을 구성하는 Java(등록상표) 아카이브 파일에서의 클래스 파일을 로컬 메모리(26)에서 워크 메모리(54)로 판독한다. 파

일 패스가 파일 시스템상의 디렉터리를 가리키고 있으면, 사용자 클래스 로더(52)는 애플리케이션을 구성하는 Java(등록상표) 아카이브 파일에서의 클래스 파일을 BD-ROM 또는 로컬 기억장치(18)에서 워크 메모리(54)로 판독한다.

[0113] 메소드 영역(53)은 사용자 클래스 로더(52)에 의해 로컬 메모리(26)로부터 판독된 클래스 파일이 저장된다.

[0114] 워크 메모리(54)는, 소위 힙 영역이며, 다양한 클래스 파일의 인스턴스가 저장된다. 워크 메모리(54)에는 상주형의 레지던트 애플리케이션이나 메소드 영역(53)에 판독된 클래스 파일에 대응하는 인스턴스가 저장된다. 이 인스턴스가 애플리케이션을 구성하는 xlet 프로그램이다. 이와 같은 xlet 프로그램을 워크 메모리(54)에 배치함으로써 애플리케이션은 실행 가능한 상태가 된다.

[0115] 스레드(55a, b ... n)는 워크 메모리(54)에 저장된 메소드를 실행하는 논리적인 실행 주체로, 로컬 변수(local variable)나 오퍼랜드 스택에 저장된 인수(argument)를 오퍼랜드(operand)로 하여 연산을 행하고, 연산결과를 로컬 변수 또는 오퍼랜드 스택에 저장한다. 도면 중의 화살표 ky 1, ky 2, ky n은 워크 메모리(54)에서 스레드(55a, b ... n)로의 메소드의 공급을 추상적으로 나타내고 있다. 물리적인 실행 주체가 CPU 하나뿐인데 대하여, 논리적인 실행 주체인 스레드는 최대 64개가 Java(등록상표) 가상 머신(30) 내에 존재할 수 있다. 이 64개라는 수치 내에서 스레드를 신규로 작성하는 것도, 기존의 스레드를 삭제하는 것도 가능하며, 스레드의 동작 수는 Java(등록상표) 가상 머신(30)의 동작 중에서 증감할 수 있다. 스레드의 수는 적의 증가시킬 수 있으므로, 복수의 스레드에 의해 하나의 인스턴스의 병렬 실행을 행하여 인스턴스의 고속화를 도모할 수도 있다.

[0116] Java(등록상표) 스택(56a, b, ... n)은 스레드(55a, b, ... n)와 1대 1의 비율로 존재하고 있고, 프로그램 카운터(도면 중의 PC)와 하나 이상의 프레임의 내부에 갖는다. "프로그램 카운터"는 인스턴스에서 현재 어느 부분이 실행되고 있는가를 나타낸다. "프레임"은 메소드에 대한 1회의 콜에 대하여 할당된 스택타입의 영역(stack-type area)이며, 그 1회의 콜 시의 인수가 저장되는 "오퍼랜드 스택"과 콜 된 메소드가 사용하는 "로컬 변수 스택(도면 중의 로컬 변수)"으로 이루어진다. 프레임은 콜이 1회 이루어질 때마다 Java(등록상표) 스택(56a, b, ... n) 상에 적층되는 것이므로, 어떤 메소드가 자신을 재귀적으로 호출하는 경우에도 이 프레임은 하나 적층되게 된다.

[0117] 이상이 Java(등록상표) 가상 머신의 내부 구성이다. 이와 같이 구성된 Java(등록상표) 가상 머신은 이벤트에 의해 구동되는 실행 주체(이벤트 드리븐(event-driven)의 실행 주체)가 된다. 이상이 Java(등록상표) 가상 머신에 대한 설명이다.

[0118] <프레젠테이션 엔진(31)>

[0119] 프레젠테이션 엔진(31)은 AV 재생기능을 실행한다. 재생장치의 AV 재생기능이란, DVD 플레이어나 CD 플레이어로부터 답습한 전통적인 기능 군이며, 재생개시(Play), 재생정지(Stop), 일시정지(Pause On), 일시정지의 해제(Pause Off), 스틸 기능의 해제(still off), 속도지정부 빨리 감기(Forward Play(speed)), 속도지정부 되감기(Backward Play(speed)), 음성전환(Audio Change), 부 영상전환(Subtitle Change), 앵글전환(Angle Change)이라는 기능이다. AV 재생기능을 실현하기 위해, 프레젠테이션 엔진(31)은, 리드 버퍼(2) 상에 판독된 AVClip 중 원하는 시각에 해당하는 부분의 디코드를 행하도록, 비디오 디코더(4), P-Graphics 디코더(6), I-Graphics 디코더(10), 오디오 디코더(16)를 제어한다. 원하는 시각으로 PSR 8(현재 PTM)에 지시되는 개소의 디코드를 행하게 함으로써, AVClip에서 임의의 시점의 재생을 가능하게 할 수 있다.

[0120] <재생제어엔진(32)>

[0121] 재생제어엔진(Playback Control Engine(PCE), 32)은 플레이 리스트에 대한 재생제어 기능(i), PSR 세트(23)에서의 상태 취득/설정 기능(ii)이라는 제반 기능을 실행한다. PL에 대한 재생제어 기능이란, 프레젠테이션 엔진(31)이 행하는 AV 재생기능 중, 재생개시나 재생정지를 현재 PL 정보 및 Clip 정보에 따라서 행하게 하는 것을 말한다. 이들 기능(i) ~ (ii)는 HDMV 모듈(29a) ~ BD-J 모듈(29b)로부터의 기능 콜에 따라서 실행한다.

[0122] 여기서, 재생제어엔진(32)에 의한 처리와 Java(등록상표) 가상 머신에 의한 처리의 동기에 대하여 설명한다. 재생제어엔진(32)은 기능 콜이 이루어지면 PL 정보에 의거한 처리순서를 실행한다. 재생하고자 하는 AVClip이 15분, 30분이라는 재생시간을 가지면, 이들 시간 동안 상술한 처리는 계속되게 된다. 여기서 문제가 되는 것은, Java(등록상표) 가상 머신(30)이 성공 응답(success response)을 반송하는 시간과 재생제어엔진(32)이 실제로 처리를 종료하는 시간과의 간격이다. Java(등록상표) 가상 머신(30)은 이벤트 드리븐의 처리 주체이므로, 콜 직후에 재생성공인가 재생실패인가를 나타내는 응답을 반송하나, 재생제어엔진(32)이 AVClip 및 PlayItem의 재생을 종료하는 것은 15분, 30분 경과 후이므로, 성공 응답을 애플리케이션에 반송하는 시간을 기준으로 하는 경우

에는 15분, 30분 경과 후에 해당하는 처리의 종결을 감지할 수 없다. PL 재생에서 빨리 감기, 되감기가 행해지면, 이 15분, 30분이라는 재생기간은 전후로 변동하게 되어, 처리 종결의 감지는 더 곤란해진다. 그래서 재생제어엔진(32)은 PlayItem의 재생종료 시 및 AVClip의 재생종료 시에 PlayItem이나 AVClip의 재생이 종료되었음을 나타내는 이벤트를 애플리케이션에 출력한다. 이와 같은 출력에 의해, 재생제어엔진(32)이 PlayItem이나 AVClip의 재생을 종결한 시점을 애플리케이션이 알 수 있다.

[0123] 이상이 Java(등록상표) 가상 머신, 네트워크 장치(31), 재생제어엔진(32)의 설명이다. 재생제어엔진(32)에 대한 Java(등록상표) 가상 머신에 의한 제어는 가상 패키지를 통하여 이루어진다. 가상 패키지를 통한 재생제어엔진(32)에 대한 제어를 실현하기 위해, 재생장치에는 이하에 나타내는 구성요소(네트워크 관리모듈(37), 가상 파일 시스템부(38), 관리정보 변환모듈(39), 메소드 실행모듈(40))가 존재한다. 이하, 이들 구성요소에 대하여 설명한다.

[0124] <네트워크 관리모듈(37)>

[0125] 네트워크 관리모듈(37)은 애플리케이션으로부터의 메소드 호출에 따라서 가상 패키지 구축에 필요한 데이터를 영화작품의 공급자가 운영하고 있는 WWW 사이트에서 다운로드한다. 가상 패키지의 구축에 필요한 데이터에는 스트림 저장위치정보 파일, Playlist 정보, Clip 정보, AVClip이 있다. 워크 메모리(54) 상의 애플리케이션에 의한 다운로드 요구가 이루어지면, 네트워크 관리모듈(37)은 네트워크를 통하여 가상 패키지 구축에 필요한 데이터를 다운로드 하여, 로컬 기억장치(18)에 기록해 간다.

[0126] 이 다운로드 시의 기록순서는, 스트림 저장위치정보 파일 → Playlist 정보 → 복수 Clip 정보 → 복수 AVClip 이 된다.

[0127] 여기서 PlayItem 정보 - 복수의 Clip 정보의 기록이 완료된 단계에서 가상 패키지의 구축이 가능해진다.

[0128] <가상 파일 시스템부(38)>

[0129] 가상 파일 시스템부(38)는 제 2 계층에 속하는 구성요소의 하나이며, 애플리케이션으로부터의 메소드 호출에 따라서 가상 패키지를 구축한다. 가상 패키지의 구축 처리에는 가상 패키지를 구성하는 각 AVClip의 상태 관리와 가상 패키지 정보의 생성처리가 포함된다.

[0130] 1. 가상 패키지 정보

[0131] 가상 패키지 정보는 BD-ROM에서의 볼륨 관리정보를 확장한 정보이다. 여기서 볼륨 관리정보는 어떤 기억매체 상에 존재하는 디렉터리 - 파일 구조를 규정하는 정보이며, 디렉터리에 대한 디렉터리 관리정보와 파일에 대한 파일관리정보로 이루어진다.

[0132] 가상 패키지 정보란, BD-ROM의 디렉터리 - 파일 구조를 나타내는 볼륨 관리정보에 새로운 파일관리정보를 추가함으로써, BD-ROM에서의 디렉터리 - 파일 구조의 확장을 도모한 것이다. 여기서 BD 볼륨 관리정보에 추가되는 파일관리정보란, 로컬 기억장치(18)에 존재하는 Playlist 정보, Clip 정보, AVClip에 대한 파일관리정보이며, 이와 같은 파일관리정보를 추가한 가상 패키지 정보를 작성하여 재생제어엔진(32)에 인계함으로써, 재생제어엔진(32)은 로컬 기억장치(18)상에 있는 Playlist 정보, Clip 정보, AVClip이 BD-ROM에 존재하는 것으로 인식할 수 있다. 도 16은 가상 파일 시스템부(38)에 의한 가상 패키지 정보 구축의 일례를 나타내는 도면이다. 본 도면의 좌측 상단은 BD-ROM에서의 디렉터리 - 파일 구조이며, 도 2에 도시한 것과 동일하다. 본 도면의 좌측 하단은 로컬 기억장치(18)에서의 디렉터리 - 파일 구조이며, 도 13(a)에 도시한 것과 동일하다. 이 로컬 기억장치(18)상의 Playlist 정보, Clip 정보, AVClip에 대한 파일관리정보를 BD-ROM의 볼륨 관리정보에 추가한다.

[0133] 구체적으로는,

[0134] i) 로컬 기억장치(18) 상의 Playlist(00002.MPLS)에 대한 파일관리정보를 BD 볼륨 구성정보에서의 MPLS 디렉터리의 디렉터리 관리정보에 추가하고,

[0135] ii) 로컬 기억장치(18) 상의 Clip 정보 #2, Clip 정보 #3, #4(00002.CLPI, 00003.CLPI, 00004.CLPI)에 대한 파일관리정보를 BD 볼륨 구성정보에서의 CLPI 디렉터리의 디렉터리 관리정보에 추가하며,

[0136] iii) 로컬 기억장치(18) 상의 AVClip #2, AVClip #3, #4(00002.M2TS, 00003.M2TS, 00004.M2TS)에 대한 파일관리정보를 BD 볼륨 구성정보에서의 STREAM 디렉터리의 디렉터리 관리정보에 추가한다.

[0137] 이렇게 함으로써 가상 패키지 정보를 얻을 수 있다. 이 추가에 의해 얻어진 볼륨 관리정보가 가상 패키지 정보

이다.

- [0138] 이렇게 하여 작성된 가상 패키지 정보를 재생제어엔진(32)에 인계한다. 이에 의해, 재생제어엔진(32)은 로컬 기억장치(18) 상의 Playlist 정보, Clip 정보, AVClip을 BD-ROM 상의 Playlist 정보, Clip 정보, AVClip과 대응하게 취급할 수 있다. 이상이 가상 패키지 정보생성에 대한 설명이다. 이어서 가상 파일 시스템부(38)에 의한 상태 관리에 대하여 설명한다.
- [0139] 3. enable 및 disable
- [0140] "상태 관리"란 가상 패키지를 구축해야 하는 AVClip이 enable인가, disable인가의 관리이다. 여기서 Playlist 정보에서의 하나의 PlayItem 정보가 멀티앵글대상인 경우에 멀티앵글구간을 구성하는 각각의 AVClip이 enable인가/disable인가라는 상태 관리의 대상이 된다. 또, PlayItem 정보에 SubPlayItem 정보가 존재하는 경우, 그 SubPlayItem 정보의 Clip_information_file_name에 기술되어 있는 AVClip도 enable인가/disable인가라는 상태 관리의 대상이 된다.
- [0141] 도 17(a)은 가상 파일 시스템부(38)에 의한 상태 관리의 일례를 나타내는 도면이다. 본 도면에 도시하는 바와 같이, AVClip의 상태에는 "enable", "disable"이 있다. "enable"이란, Clip 정보와 이에 대응하는 AVClip이 로컬 기억장치(18) 상에 존재하고, 그 AVClip이 판독전용 모드(Read Only Mode)로 설정되어 있는 상태를 말한다.
- [0142] 한편, "disable"이란, Clip 정보가 로컬 기억장치(18) 상에 존재하는 상태, 이에 대응하는 AVClip이 로컬 기억장치(18)에 존재하지 않거나(Missing Stream), 존재한다고 해도 삭제될 수 있는 상태에 있음을 말한다.
- [0143] disable은 삭제될 수 있는 것으로 취급되므로, 로컬 기억장치(18) 상의 용량이 부족했을 때 disable의 AVClip은 삭제의 대상이 된다.
- [0144] *따라서 로컬 기억장치(18)에 AVClip이 다운로드 되었을 경우에, 가상 파일 시스템부(38)가 이 AVClip을 enable로 변화시키지 않는 한, 재생제어엔진(32)은 이 다운로드 된 AVClip을 존재하지 않는 것으로 취급하게 된다. 가상 파일 시스템부(38)를 변화시키지 않는 한 재생제어엔진(32)은 로컬 기억장치(18) 상의 AVClip을 인식하지 않으므로, 불안정한 상태인 체로 재생제어엔진(32)이 재생을 진행하는 경우는 있을 수 없다.
- [0145] 3. 판독전용 모드 및 그 취소
- [0146] 이어서 판독전용 모드에 대하여 설명한다. 판독전용 모드로의 변화는 로컬 기억장치(18)에서의 볼륨 관리정보의 파일관리정보를 고쳐씀으로써(rewriting) 이루어진다. 이 고쳐쓰기의 대상은 로컬 기억장치(18)에서의 DLDAT 디렉터리 하위의 PACKAGE 00001 디렉터리에서의 파일관리정보이며, 이와 같은 파일관리정보에서의 파일속성을 전부 판독전용(Read-only) 속성으로 고쳐씀으로써, 로컬 기억장치(18) 상의 AVClip은 고쳐쓸 수 없게 된다. 그런 이유에서, 재생제어엔진(32)에 의한 재생 중 애플리케이션이 로컬 기억장치(18) 상의 AVClip을 액세스하려고 해도, AVClip을 고쳐쓸 수는 없으므로, 재생 중의 AVClip의 동일성이 유지되게 된다.
- [0147] 로컬 기억장치(18) 상의 AVClip을 재생제어엔진(32)에서 인식할 수 있는 상태로 했을 때, 로컬 기억장치(18)에서 그 AVClip을 저장하고 있는 파일을 판독전용 속성으로 하므로, Java(등록상표) 가상 머신 상의 애플리케이션이 직접적으로 로컬 기억장치(18)를 액세스했다고 해도, 그 AVClip의 내용이 도중에서부터 변화하는 경우는 없다.
- [0148] 판독전용 모드의 취소에 대하여 설명한다. 판독전용 모드의 취소는, 판독전용 모드로의 변화와 마찬가지로, 로컬 기억장치(18)에서의 볼륨 관리정보의 파일관리정보를 고쳐씀으로써 이루어진다. 이 고쳐쓰기의 대상은 로컬 기억장치(18)에서의 DLDAT 디렉터리 하위의 PACKAGE 00001 디렉터리에서의 파일관리정보이며, 이와 같은 파일관리정보에서의 파일관리정보를 판독전용 속성에서 고쳐쓰기 가능(Re-writable) 속성으로 고쳐씀으로써, 로컬 기억장치(18) 상의 AVClip은 언제나 고쳐쓸 수 있게 된다. 그런 이유에서, 재생제어엔진(32)에 의한 재생이 종료된 직후, 로컬 기억장치(18)에서의 파일관리정보를 고쳐쓰기 가능 속성으로 고쳐씀으로써, 로컬 기억장치(18) 상의 AVClip은 삭제의 대상이 된다. 이상이 상태 관리에 대한 설명이다.
- [0149] 4. 상태 관리와 가상 패키지 정보의 관계
- [0150] 이 상태 관리의 내용은 가상 패키지 정보의 작성에 반영될 수 있다. 즉 상태 관리에서 disable로 되어 있는 AVClip에 대한 파일관리정보는 BD-ROM의 볼륨 관리정보에 추가되지 않으며, enable로 되어 있는 AVClip에 대한 파일관리정보만이 BD-ROM의 볼륨 관리정보에 추가되는 것이다. 이렇게 하여 작성한 가상 패키지 정보를 재생제

어엔진(32)에 인계함으로써 재생제어엔진(32)은 enable로 설정된 AVClip만을 인식하게 된다. BD-ROM 상에 존재하는 AVClip #1 및 로컬 기억장치(18) 상에 존재하는 AVClip #2, AVClip #3, AVClip #4 중 AVClip #1, AVClip #2는 enable이나, AVClip #3, AVClip #4는 disable이라고 하면, 가상 파일 시스템부(38)는, 도 18에 도시하는 바와 같이, 가상 패키지 정보의 작성을 행한다. 도 18은 상태 관리를 반영한 가상 패키지 정보의 작성을 나타내는 도면이다. 본 도면에서 AVClip #2에 대한 파일관리정보는 BD 볼륨구성정보에서의 STREAM 디렉터리의 디렉터리 관리정보에 추가되나, AVClip #3, AVClip #4에 대한 파일관리정보는 BD 볼륨 구성정보에서의 STREAM 디렉터리의 디렉터리 관리정보에 추가되지 않는다. 도 18 좌측 하단에서의 4개의 AVClip 중, AVClip #2에 해당하는 00002.M2TS가 enable이고, AVClip #3, AVClip #4에 해당하는 00003.M2TS, 00004.M2TS가 disable인 것으로 한다. 그렇게 하면, 본 도면에 도시하는 바와 같이, 로컬 기억장치(18)에서 AVClip #2에 해당하는 00002.M2TS의 파일속성은 판독전용 속성으로 설정된다. 또, AVClip #3, AVClip #4에 해당하는 00003.M2TS, 00004.M2TS의 파일속성은 고쳐쓰기 가능 속성으로 설정된다. 이렇게 하여 작성한 가상 패키지 정보를 재생제어엔진(32)에 인계함으로써, 재생제어엔진(32)은 disable에 해당하는 AVClip #3, AVClip #4를 인식하지 않게 된다.

[0151] 여기서, Clip 정보는 존재하나 대응하는 AVClip이 로컬 기억장치(18) 상에 없는 경우, 그 AVClip은 disable로 설정되었다. 그러나 Clip 정보에 대한 파일 패스가 스트림 저장위치정보 파일에 있음에도, AVClip은 물론 Clip 정보조차 로컬 기억장치(18)에 없는 경우에는 가상 패키지의 구축은 불가능해진다. 즉 가상 패키지에 본래 있어야 하는 Clip 정보가 존재하지 않는다는 것은 가상 패키지 전체에 파급되는 것을 의미한다.

[0152] 5. 가상 패키지 정보의 인계

[0153] 이상의 1. ~ 4.에 의해서 작성된 가상 패키지 정보가 어떻게 인계되는가에 대하여 설명한다.

[0154] 여기서, 도 18에 도시한 바와 같이, Playlist 정보, Clip 정보 #2 ~ Clip 정보 #4, AVClip #2는 다운로드가 완료되어 있는 경우를, AVClip #3, AVClip #4의 다운로드는 미완료 상태인 경우를 상정한다. 이와 같은 상태에서 가상 패키지 구축의 요구가 있으면, 가상 파일 시스템부(38)는 도 19에 도시하는 것과 같은 동작을 행한다. 도 19는 네트워크 관리모듈(37)에 의한 다운로드가 진행중인 경우의 가상 파일 시스템부(38)에 의한 가상 패키지의 구축을 나타내는 도면이다. 본 도면의 상황에서 가상 패키지 구축이 요구되면(◎1 VPCreate), BD-ROM 및 로컬 기억장치(18)로부터 디렉터리 관리정보 및 파일관리정보를 판독하여(◎2, 3) 가상 패키지의 구축을 행하고, 가상 패키지 정보를 재생제어엔진(32)에 인계하는 동시에(◎4), 애플리케이션에 인계한다(◎5). 이렇게 함으로써, 재생제어엔진(32)은 가상 패키지를 통하여 BD-ROM 및 로컬 기억장치(18) 상의 AVClip으로부터 하나의 스트림 시퀀스를 구축할 수 있다.

[0155] *또, 애플리케이션에 인계한 가상 패키지 정보에서 Clip 정보 #3, Clip 정보 #4가 존재하고, AVClip #3, AVClip #4는 존재하지 않으므로, 애플리케이션은 AVClip #3, AVClip #4가 스트림 열을 구성하며, 이들이 존재하지 않는 스트림(Missing Stream)이라고 인식할 수 있다.

[0156] 이상이 가상 파일 시스템부(38)에 대한 설명이다.

[0157] <관리정보 변환모듈(39)>

[0158] 관리정보 변환모듈(39)은, BD-J 모듈(29b) 내의 미들웨어의 하나이며, 가상 패키지 중에 존재하는 Playlist 정보의 인스턴스 생성을 애플리케이션이 요구했을 때, Java(등록상표) 가상 머신의 워크 메모리(54)에 JMF 형식에 의한 Playlist 정보의 인스턴스(도 16의 워크 메모리(54) 내에 나타난 Playlist 오브젝트)를 생성한다.

[0159] 이 인스턴스(Playlist 오브젝트)는 Playlist 정보인 "데이터부"와 복수의 "메소드부"를 갖는다.

[0160] Playlist 오브젝트에서의 복수의 메소드는 재생개시(Play), 재생정지(Stop), 일시정지(Pause On), 일시정지의 해제(Pause Off), 스틸 기능의 해제(still off), 속도지정부 빨리 감기(Forward Play(speed)), 속도지정부 되 감기(Backward Play(speed)), 음성전환(Audio Change), 부 영상전환(Subtitle Change), 앵글전환(Angle Change)이라는 다양한 재생처리를 재생제어엔진(32)이 행하도록 하는 것이다. 따라서 애플리케이션은 이 메소드를 호출함으로써, 그 Playlist에 의해 규정되는 스트림 시퀀스를 재생하거나, 또는 이와 같은 재생을 정지하거나 할 수 있다.

[0161] 여기서 Playlist 오브젝트에 의해 재생되는 Playlist 정보에 대하여 설명한다. 이 Playlist 정보는 로컬 기억장치(18) 상에 존재하는 Playlist 정보이다.

- [0162] BD-ROM 상의 PlayList 정보는 BD-ROM 상의 AVClip만을 참조하고 있는 것에 대하여, 로컬 기억장치(18) 상의 PlayList 정보는 "Progressive PlayList 정보"라고 불리며, BD-ROM 상의 AVClip과 로컬 기억장치(18) 상의 AVClip을 참조한다.
- [0163] 여기서 Progressive PlayList 정보가 PlayItem 정보 #1 ~ PlayItem 정보 #5라는 5개의 PlayItem 정보로 구성되어 있는 것으로 한다. 이 중 3개의 PlayItem 정보 #1 ~ PlayItem 정보 #3은 BD-ROM 상의 Clip 정보를 참조하는 것이고, 나머지 2개(PlayItem 정보 #4, PlayItem 정보 #5)가 로컬 기억장치(18) 상의 Clip 정보를 참조하는 것인 경우, 이 Progressive PlayList 정보는, 도 20(a)에 도시하는 바와 같이, BD-ROM 상의 AVClip과 로컬 기억장치(18) 상의 AVClip으로부터 하나의 스트림 시퀀스를 정의할 수 있다.
- [0164] 도 20(a)은 로컬 기억장치에 저장된 PL 정보에 의해 어떤 PlayList 재생시간 축이 정의되는가를 나타내는 도면이다. 제 1단계는 BD-ROM에 기록된 AVClip에서의 재생시간 축을 나타내고, 제 2단계는 로컬 기억장치에 저장된 PL 정보에 의해 정의되는 PlayList 재생시간 축을 나타낸다. 제 3단계는 로컬 기억장치에 저장된 AVClip #2에서의 재생시간 축을, 제 4단계는 로컬 기억장치에 저장된 AVClip #3에서의 재생시간 축을, 제 5단계는 로컬 기억장치에 저장된 AVClip #4에서의 재생시간 축을 나타낸다.
- [0165] PlayList 정보에서의 PlayItem 중 PlayItem 정보 #4, PlayItem 정보 #5, PlayItem 정보 #6이 AVClip #2, AVClip #3, AVClip #4를 재생구간으로 지정하면, Progressive PlayList 정보는 BD-ROM 상의 AVClip 및 로컬 기억장치(18) 상의 AVClip을 하나의 스트림 시퀀스로 규정할 수 있다.
- [0166] 여기서 Progressive PlayList 정보의 "Progressive"란, PlayList 정보를 구성하는 PlayItem 정보에 의해 참조되는 AVClip 중 일부가 enable이면, 나머지가 disable이었다고 해도 스트림 시퀀스의 재생을 개시할 수 있음을 의미한다.
- [0167] 도 20(b)은 Progressive PlayList 정보의 개념을 나타내는 도면이다. 본 도면의 제 2단계 ~ 제 5단계는 도 20(a)의 제 2단계, 제 3단계, 제 4단계, 제 5단계를 나타내는 것이다. 도 20(a)의 제 3단계, 도 20(b)의 제 3단계의 AVClip은 enable이나, 제 4단계 ~ 제 5단계의 AVClip이 disable인 경우, Progressive PlayList 정보에 의해 참조 되는 4개의 AVClip 중 AVClip #3, AVClip #4는 재생할 수 없고, AVClip #1, AVClip #2는 재생할 수 있으므로, 스트림 시퀀스의 재생은 개시할 수 있게 된다. 이와 같이, 일부의 AVClip이면 투기적으로(speculatively) 재생을 개시할 수 있는 것이 Progressive PlayList 정보의 특징이다.
- [0168] 가상 패키지 구축이 가상 파일 시스템부(38)에 의해 이루어진 단계에서, 도 20과 같은 Progressive PlayList 정보에 대한 인스턴스의 생성을 워크 메모리(54) 상의 애플리케이션이 요구한 경우, 관리정보 변환모듈(39)은 도 21에 도시한 바와 같은 처리를 행한다. 도 21은 관리정보 변환모듈(39)에 의한 인스턴스의 생성을 나타내는 도면이다.
- [0169] 가상 패키지에 존재하는 PlayList 정보의 인스턴스 생성을 요구하는 생성 메소드(generation method)를 애플리케이션 호출(application call)하면(◎1), 관리정보 변환모듈(39)은 PlayList 정보를 시나리오 메모리(25)에서 판독해서(◎2), 이 Progressive PlayList 정보를 데이터부로 하여, 메소드부를 부가한 PlayList 오브젝트를 워크 메모리(54)에 생성한다(◎3). 이와 같은 PlayList 오브젝트가 워크 메모리(54)에 얻어지므로, 애플리케이션은 이 PlayList 오브젝트의 메소드를 호출함으로써 PlayList 정보에 의해 규정되는 스트림 시퀀스의 재생을 재생제어엔진(32)에 명령할 수 있다.
- [0170] 도 22는 PlayList 오브젝트에서의 Play 메소드의 호출을 나타내는 도면이다. 이 Play 메소드의 호출에 의해(◎4), 재생제어엔진(32)에 대해 PL 재생의 기능 콜이 이루어진다(◎5). 그렇게 하면, 재생제어엔진(32)은 재생이 요구된 PlayList 정보에 의해 참조 되고 있는 AVClip(BD-ROM 상의 AVClip #1, 로컬 기억장치(18) 상의 AVClip #2)을 판독하여(◎6, 7), 순차 디코더에 공급한다. 이상이 관리정보 변환모듈(39)에 대한 설명이다.
- [0171] <메소드 실행모듈(40)>
- [0172] 메소드 실행모듈(40)은 BD-J 모듈(29b) 내의 미들웨어에서의 구성요소의 하나이며, Java(등록상표) 가상 머신(30)에 존재하는 애플리케이션으로부터의 메소드 호출에 따라서 각종 메소드를 실행한다. 메소드 실행모듈(40)에 의해 실행되는 메소드에는 다양한 것이 있으나, getMissingStreams 메소드, enableClip 메소드, disableClip 메소드, getDisableClips 메소드라는 것이 본원에서의 특징적인 메소드이다. 이하, 이들 메소드에 대하여 설명한다.
- [0173] 1. getMissingStreams 메소드

- [0174] getMissingStreams 메소드란, 네트워크 관리모듈(37)에 의한 다운로드가 계속되고 있을 때에, 아직 Missing Stream으로 되어 있는 AVClip이 존재하는가를 가상 파일 시스템부(38)에 문의하도록 하는 메소드이다.
- [0175] 여기서, 도 19와 같은 가상 패키지 정보가 워크 메모리(54) 상의 애플리케이션에 인계된 것으로 한다. 가상 패키지 구축 시에 판명되어 있었던 Missing Stream이 AVClip #3, AVClip #4이고, 그 후, 애플리케이션이 getMissingStreams 메소드를 호출함으로써 최신의 Missing Stream에 대하여 가상 파일 시스템부(38)에 문의한 것으로 한다. 이 경우의 메소드 실행모듈(40)에 의한 처리는 도 22와 같이 된다.
- [0176] 도 23은 이 문의의 결과를 모식적으로 나타내는 도면이다. 본 도면에서의 ㉠에 도시하는 바와 같이, getMissingStreams 메소드의 호출이 이루어지고, 이 호출에 의해, 가상 파일 시스템부(38)가 AVClip #4를 복귀값(return value)으로 반송하는 것으로 한다. 그렇게 하면, getMissingStreams 메소드를 호출하기 전에 Missing Stream이 AVClip #3, AVClip #4이었던 것에 대해, getMissingStreams 메소드의 호출 후에 AVClip #4가 되었으므로, AVClip #3은 Missing Stream이 아닌 것이 되었다(다운로드가 완료되었다)고 판정할 수 있다. 이렇게 함으로써, 다운로드의 진행에 의해 Missing Stream이 어느 정도 해소되었는가를 알 수 있다.
- [0177] 2. enableClip 메소드
- [0178] enableClip 메소드는, 가상 패키지의 구축 시에 Missing Stream이었으나, 그 후, Missing Stream이 아니게 된 AVClip이 존재하는 경우에, 그 AVClip을 enable로 설정하기 위한 메소드이다.
- [0179] 여기서, 도 23에서 AVClip #3이 Missing Stream이 아니게 되었음이 명백해졌으므로, 이 AVClip #3을 대상으로 한 enableClip 메소드를 호출한 것으로 한다. 이 경우, 메소드 실행모듈(40)은 도 24에 도시하는 바와 같은 처리를 행한다. 도 24는 enableClip 메소드의 호출을 모식적으로 나타내는 도면이다. 이 AVClip #3을 대상으로 한 enableClip 메소드의 호출이 이루어지면(㉠), 애플리케이션은 AVClip #3을 enable로 변화시키고, 성공 응답을 애플리케이션으로 반송한다(㉡). 이렇게 함으로써, 가상 패키지의 구축에 의해서 Missing Stream이 아니게 된 AVClip을 enable로 변화시켜 갈 수 있다. AVClip을 enable로 했을 때, 메소드 실행모듈(40)은 새로 enable로 한 AVClip을 포함하는 가상 패키지 정보를 애플리케이션 및 재생제어엔진(32)에 인계하도록 한다. 이에 의해, 재생제어엔진(32) 및 애플리케이션은 새로 enable이 된 AVClip을 인식하여 액세스할 수 있다.
- [0180] 3. disableClip 메소드
- [0181] disableClip 메소드는 enable에 있는 AVClip을 disable로 설정하는 메소드이다. 여기서, 도 20에 도시된 Progressive Playlist 정보의 재생 중에 AVClip #2의 재생이 완료되었다는 취지의 이벤트가 재생제어엔진(32)에 의해 통지된 것으로 한다. 이 경우, AVClip #2의 재생이 완료되었음이 통지되면 도 25와 같은 처리를 행한다. 도 25는 disableClip 메소드의 호출을 모식적으로 나타내는 도면이다. 본 도면에서 재생제어엔진(32)에 의한 재생의 진행에 수반하여 AVClip #2의 재생이 완료되었음이 end of Clip 이벤트에 의해 통지되면(㉠), 이 AVClip #2를 대상으로 한 disableClip 메소드를 호출함으로써(㉡), 애플리케이션은 AVClip #3을 disable로 변화시킬 수 있다. 이렇게 함으로써, 재생이 종료된 AVClip을 disable로 변화시켜 갈 수 있다. 이렇게 함으로써, AVClip #2가 disable로 설정되므로, AVClip #2는 삭제의 대상이 된다. 예고편을 구성하는 AVClip이 시시각각으로 다운로드 되는 경우에, Progressive Playlist 정보에 의해 규정되는 스트림 시퀀스 중 재생이 종료된 것을 순차적으로 disable로 해 가면, 재생이 종료된 AVClip을 disable로 하여 삭제의 대상으로 해 갈 수 있다. AVClip을 disable로 했을 때, 메소드 실행모듈(40)은 새로 disable로 한 AVClip을 제외한 가상 패키지 정보를 애플리케이션 및 재생제어엔진(32)에 인계하도록 한다. 이에 의해, 재생제어엔진(32)은 disable이 된 AVClip을 인식하지 않는다. 또한, 로컬 기억장치(18) 상의 AVClip을 삭제할 때에, 미들웨어는 삭제하고자 하는 애플리케이션의 기록 권한을 체크 한 후에 그 삭제를 실행한다. 이는 본의가 아닌 삭제에 의한 AVClip의 소실을 방지하기 위함이다.
- [0182] 4. getDisableClips 메소드
- [0183] getDisableClips 메소드는 Disable로 되어 있는 AVClip이 존재하는가를 문의하도록 하는 메소드이다. 이 메소드의 호출에 의해 애플리케이션은 이 메소드를 사용하여 현재 Disable로 되어 있는 AVClip이 있는가 여부의 정보를 얻을 수 있다.
- [0184] 사용 예로는, 예를 들어, Progressive Playlist 재생 중 사용자로부터의 챕터 스킵 요구(chapter skip request)를 수신한 때에, 애플리케이션은 점프 처리 AVClip이 Disable인가 여부를 이 메소드를 이용하여 판단한다. 그리고 Disable이면 챕터 스킵을 행하지 않고, 「현재 다운로드 중이므로, 그 챕터로는 아직 점프할 수 없

습니다」 등의 주의를 환기하는 표시를 행한다. 이렇게 함으로써, 챕터 스킵이라는 특수재생을 Progressive Playlist의 재생 중에 실행할 수 있다.

[0185] 이상으로 메소드 실행모듈(40)에 의해 실행되는 메소드에 대한 설명을 마친다.

[0186] 지금까지 설명한 각 구성요소에 대하여 플로차트를 참조하면서, 그 처리순서의 상세를 설명한다.

[0187] 도 26은 가상 패키지 구축 시의 처리를 나타내는 플로차트이다. 본 플로차트에서는, 스트림 저장위치정보 파일에 저장된 AVClip에 대응하는 Clip 정보가 로컬 기억장치(18)에 완비되어 있는가 여부를 먼저 판정하고(스텝 S 1), Clip 정보에 미완인 것이 있으면 애플리케이션으로부터의 가상 패키지 호출에 대해 False 리턴을 행한다(스텝 S 10). Clip 정보에 대한 파일 패스가 스트림 저장위치정보 파일에 있음에도, AVClip은 물론 Clip 정보도 로컬 기억장치에 없는 경우에는 가상 패키지의 구축은 불가능해진다.

[0188] 한편, 스트림 저장위치정보 파일에 기술된 모든 Clip 정보가 로컬 기억장치(18)에 존재하면(스텝 S 1에서 Yes), 로컬 기억장치(18)에서의 Playlist 정보의 파일관리정보를 BD 볼륨 관리정보에서의 PLAYLIST 디렉터리의 디렉터리 관리정보에 추가하고(스텝 S 2), 로컬 기억장치(18)에서의 Clip 정보의 파일관리정보를 BD 볼륨 관리정보에서의 CLPINF 디렉터리의 디렉터리 관리정보에 추가한다(스텝 S 3). 그 후, 스텝 S 4 ~ 스텝 S 8의 루프처리를 실행한다. 이 루프처리는 로컬 기억장치(18)에 존재하는 각각의 Clip 정보에 대하여 스텝 S 6 ~ 스텝 S 8의 처리를 반복하는 것이다(스텝 S 4, 스텝 S 5). 여기서 루프처리의 대상이 되는 하나의 Clip 정보를 Clip 정보 x로 한다.

[0189] 본 루프처리에서, Clip 정보 x에 대응하는 AVClip을 특정하고(스텝 S 6), 이 AVClip은 enable인가, disable인가를 판정한다(스텝 S 7). enable이면 로컬 기억장치(18)에서의 AVClip x의 파일관리정보를 BD 볼륨 관리정보에서의 STREAM 디렉터리의 디렉터리 관리정보에 추가한다(스텝 S 8). 한편, AVClip이 disable이면 스텝 S 8을 스킵 한다. 이상의 처리를 모든 Clip 정보 및 AVClip에 대하여 반복하면, enable의 AVClip에 대한 파일관리정보가 BD 볼륨 관리정보에 추가되게 된다. 이와 같은 추가에 의해 얻어진 BD 볼륨 관리정보가 가상 패키지 정보가 된다. 이 가상 패키지 정보를 가상 패키지 호출의 호출 원의 애플리케이션 및 재생제어엔진(32)에 인계하고(스텝 S 9) 처리를 종료한다.

[0190] 도 27(a)은 enableClip 메소드의 호출 시의 처리순서를 나타내는 플로차트이다. 이 메소드의 호출이 이루어지면, 호출 시의 인수로부터 enable로 해야하는 AVClip y를 특정하고(스텝 S 14), 스텝 S 15, 스텝 S 16의 판정을 순차적으로 행한다. 스텝 S 15는 AVClip y가 로컬 기억장치(18)에 기억되어 있는가 여부의 판정이며, 스텝 S 16은 AVClip y의 파일 패스가 스트림 저장위치정보 파일에 기재되어 있는가의 판정이다. 이들 판정스텝 중 어느 하나가 No이면, 호출 원에 False를 반송하고, 본 플로차트의 처리를 종료한다(스텝 S 19). 한편, 스텝 S 15 ~ 스텝 S 16의 쌍방이 Yes이면, AVClip y를 enable로 하고, 로컬 기억장치(18)에서 AVClip y를 저장한 파일의 파일속성을 판독전용 속성으로 하며, AVClip y를 판독전용 모드로 한다(스텝 S 17). 그 후 성공(success)을 복귀 값으로 리턴한다(스텝 S 18).

[0191] 도 27(b)은 disableClip 메소드의 호출 시의 처리순서를 나타내는 플로차트이다. 이 메소드의 호출이 이루어지면, 호출 시의 인수로부터 disable로 해야하는 AVClip y를 특정하고(스텝 S 21), 스텝 S 22, 스텝 S 23의 판정을 순차적으로 행한다. 스텝 S 22는 AVClip y가 재생 중인가 여부의 판정이며, 스텝 S 23은 AVClip y의 파일 패스가 스트림 저장위치정보 파일에 기재되어 있지 않은가의 판정이다. 이들 판정스텝 중 어느 하나가 Yes이면 호출 원에 false를 반송하고, 본 플로차트의 처리를 종료한다(스텝 S 26). 한편, 스텝 S 22 ~ 스텝 S 23의 쌍방이 Yes이면, AVClip y를 disable로 하고, 로컬 기억장치(18)에서 AVClip y를 저장하고 있는 파일의 파일속성을 Re-writable 속성으로 하며, AVClip y의 판독전용 모드를 취소한다(스텝 S 24). 그 후 성공을 복귀 값으로 리턴한다(스텝 S 25).

[0192] 도 28은 재생제어엔진(32)에 의한 재생순서를 나타내는 플로차트이다. 본 플로차트는 Playlist 정보에서의 선두의 PlayItem 정보를 PlayItem으로 하여 스텝 S 33 ~ 스텝 S 42의 루프처리를 실행하는 것이다. 스텝 S 33 ~ 스텝 S 42의 루프처리에서의 제어변수는 변수 i이며, 스텝 S 33 ~ 스텝 S 40의 처리를 실행하고, 그 후 제어변수 i를 인크리먼트 하는 처리를 변수 i가 PlayItem의 수(Number Of PlayItem)를 넘을 때까지 실행하는 것이다(스텝 S 41).

[0193] 이 스텝 S 32 ~ 스텝 S 40의 처리에 대하여 설명한다. 이들 처리는, PlayItem 정보 i의 Clip_information_file_name에 기술되어 있는 AVClip을 AVClip j로 하여(스텝 S 33), 재생의 대상으로 한다. 스텝 S 34는 AVClip j가 enable인가 여부의 판정이며, 만일 AVClip j가 enable이 아니면 본 플로차트의 처리를

그 시점에서 중지한다. 만일 enable이면 AVClip j 중 PlayItem.In_time에서 PlayItem.Out_time까지의 부분을 재생하도록 드라이브 장치 및 디코더에 지시한다(스텝 S 35).

- [0194] 스텝 S 36은 PlayItem 정보 i를 Sync_PlayItem_id로 지정하고 있는 SubPlayItem k가 존재하는가 여부의 판정이다. 만일 존재하지 않으면 그대로 스텝 S 41로 이행한다.
- [0195] 만일 존재하면, SubPlayItem k의 Clip_information_file_name에 기술되어 있는 AVClip을 AVClip h로 하고(스텝 S 38), 그 AVClip h가 enable인가 여부를 판정한다(스텝 S 39). 만일 enable이면, 그 AVClip h 중 Sync_Start_PTS_of_PlayItem에서 Out_time까지를 재생하도록 드라이브 장치 및 디코더에 지시하고 스텝 S 41로 이행한다(스텝 S 40).
- [0196] 만일 그 AVClip h가 enable이 아닌 경우에는 스텝 S 40을 스킵한다. 이상의 처리를 PlayList 정보를 구성하는 모든 PlayItem 정보에 대하여 반복함으로써 PlayList 정보에 의해 규정되는 스트림 시퀀스의 재생이 이루어진다. 도 29는 애플리케이션의 처리순서를 나타내는 플로차트이다. 먼저, 스트림 저장위치정보 파일과 PlayList 정보, Clip 정보, AVClip의 다운로드를 요구한다(스텝 S 51). 그리고 가상 패키지의 구축요구를 행한다(스텝 S 52). 스텝 S 53은 구축요구가 성공했는가 여부의 판정이며, 가상 패키지의 구축이 성공이 될 때까지 가상 패키지의 구축요구는 몇 회라도 반복된다.
- [0197] 가상 패키지의 구축이 성공하면 스텝 S 55를 실행한다. 스텝 S 55는 Missing Stream이 가상 패키지 정보 내에 존재하는가 여부의 판정이다. 구체적으로는 Clip 정보는 존재하는 것에 대응하는 AVClip이 없는 경우에 그 AVClip을 Missing Stream으로 취득한다(스텝 S 56).
- [0198] 그 후, 스텝 S 57에서 Progressive PlayList 정보의 Play 메소드를 호출하고, 그리고 Play 메소드가 성공했는가 여부의 판정을 행한다(스텝 S 58). 만일 False이면 본 플로차트의 처리를 종료한다. 만일 성공이면 스텝 S 59 ~ 스텝 S 61로 이루어지는 루프처리를 실행한다.
- [0199] *스텝 S 59는 getMissingStreams 메소드의 호출이다. 스텝 S 60은 스텝 S 59에서의 호출에 의해 Missing Stream이 통지되면, 이전에 취득한 Missing Stream 중에 다운로드가 완료된 것이 존재하는가 여부를 판정한다.
- [0200] 스텝 S 61은 AVClip의 재생완료를 나타내는 이벤트가 발생하였는가 여부의 판정이다. 애플리케이션은 재생제어 엔진(32)에 의한 재생이 이루어지고 있는 사이에 이 스텝 S 59 ~ 스텝 S 61의 처리를 반복하여 실행한다.
- [0201] 상술한 처리의 반복에서 다운로드가 완료된 AVClip이 있으면(스텝 S 60에서 Yes), 그 다운로드가 완료된 Missing Stream에 대해 enableClip 메소드를 호출한다(스텝 S 62). 이렇게 함으로써 다운로드가 완료된 AVClip을 조기에 enable로 할 수 있다.
- [0202] 또, 재생이 종료된 AVClip이 있으면(스텝 S 61에서 Yes), 그 AVClip이 예고편인가 여부를 판정하여, 만일 예고편이면 그 AVClip에 대해 disableClip 메소드를 호출한다. 이렇게 함으로써 재생이 완료된 예고편의 AVClip을 조기에 disable로 할 수 있다.
- [0203] <동작>
- [0204] 도 20에 나타낸 PlayList 정보를 재생하는 경우의 재생장치의 동작에 대하여 설명한다. 도 30 ~ 도 33은 네트워크 관리모듈(37)에 의한 다운로드의 진척과 재생제어엔진(32)에 의한 재생의 진척을 나타내는 도면이다. 제 1단계는 네트워크 관리모듈(37)에 의한 다운로드의 진척을 나타내고, 제 2 단계는 재생제어엔진(32)에 의한 재생의 진척을 나타낸다. 제 3단계는 현상의 시간 축이다. 제 4단계는 애플리케이션에 의한 메소드의 호출을 나타낸다.
- [0205] 제 1단계에 도시된 바와 같이, 네트워크 관리모듈(37)에 의한 다운로드도 도 20에 도시한 PlayList 정보의 다운로드, Clip 정보 #1 ~ Clip 정보 #4의 다운로드, AVClip #1 ~ AVClip #4의 다운로드로 이루어진다.
- [0206] 도 30(a) 현재시점 = 시점 t 0
- [0207] 시점 t 0은 PlayList 정보, Clip 정보 #1 ~ Clip 정보 #4의 다운로드가 완료된 시점이다. 따라서 가상 패키지의 구축이 가능해진다. 또, 가상 패키지에서의 PlayList 정보는 AVClip #1을 참조하고 있고, 이 AVClip #1은 적어도 enable이므로 스트림 시퀀스의 재생은 t 0부터 개시하고 있다.
- [0208] 도 30(b) 현재시점 = 시점 t 1
- [0209] 시점 t 1은 PlayList #1에 의해 참조 되는 AVClip #1의 재생도중이지만, AVClip #2의 다운로드가 완료된 시점이

다. AVClip #2의 다운로드가 완료되었으므로, AVClip #2는 disable이 아닌 것으로 되어 있다. 그런 이유에서, 애플리케이션이 enableClip 메소드를 호출함으로써 AVClip #2를 enable로 할 수 있다. 이 호출에 의해 AVClip #2에 해당하는 PlayItem #4는 재생의 대상이 된다.

[0210] 도 31(a) 현재시점 = 시점 t 2

[0211] 시점 t 2는 PlayList #1에 의해 참조 되는 AVClip #2의 재생도중이지만, AVClip #3의 다운로드가 완료된 시점이다. AVClip #3의 다운로드가 완료되었으므로, AVClip #3은 disable이 아닌 것으로 되어 있다. 그런 이유에서, 애플리케이션이 enableClip 메소드를 호출함으로써 AVClip #3을 enable로 할 수 있다. 이 호출에 의해 AVClip #3에 해당하는 PlayItem #5는 재생의 대상이 된다.

[0212] 도 31(b) 현재시점 = 시점 t 3

[0213] 시점 t 3은 AVClip #2의 재생이 완료된 시점이다. AVClip #2의 재생이 완료되어 있고, AVClip #2는 예고편이므로, PlayItem에 의해 참조 되는 AVClip #2는 삭제의 대상이 될 수 있다. 그래서 이 AVClip #2를 대상으로 한 disableClip 메소드를 호출함으로써 AVClip #2를 삭제 대상으로 한다. 이렇게 함으로써, 재생제어엔진(32)은 AVClip #2가 마치 존재하지 않았던 것과 같이 취급할 수 있다.

[0214] 도 32(a) 현재시점 = 시점 t 4

[0215] 시점 t 4는 PlayList #1에 의해 참조 되는 AVClip #3의 재생도중이지만, AVClip #4의 다운로드가 완료된 시점이다. AVClip #4의 다운로드가 완료되었으므로, AVClip #4는 disable이 아닌 것으로 되어 있다. 그런 이유에서, 애플리케이션이 enableClip 메소드를 호출함으로써 AVClip #4를 enable로 할 수 있다. 이 호출에 의해 AVClip #4에 해당하는 PlayItem #6은 재생의 대상이 된다.

[0216] 도 32(b) 현재시점 = 시점 t 5

[0217] 시점 t 5는 AVClip #3의 재생이 완료된 시점이다. AVClip #3의 재생이 완료되어 있고, AVClip #3은 예고편이므로, PlayItem #5에 의해 참조 되는 AVClip #3은 삭제의 대상이 될 수 있다. 그래서 이 AVClip #3을 대상으로 한 disableClip 메소드를 호출함으로써 AVClip #3을 삭제 대상으로 한다. 이렇게 함으로써, 재생제어엔진(32)은 AVClip #3이 마치 존재하지 않았던 것과 같이 취급할 수 있다.

[0218] 도 33 현재시점 = 시점 t 6

[0219] 시점 t 6은 AVClip #4의 재생이 완료된 시점이다. AVClip #4의 재생이 완료되어 있고, AVClip #4는 예고편이므로, PlayItem #6에 의해 참조 되는 AVClip #4는 삭제의 대상이 될 수 있다. 그래서 이 AVClip #4를 대상으로 한 disableClip 메소드를 호출함으로써 AVClip #4를 삭제 대상으로 한다. 이렇게 함으로써, 재생제어엔진(32)은 AVClip #4가 마치 존재하지 않았던 것과 같이 취급할 수 있다.

[0220] 이상과 같이 본 실시 예에 의하면, 네트워크 관리모듈(37)에 의한 다운로드와 재생제어엔진(32)에 의한 재생을 병렬로 행하고, 다운로드가 이루어진 AVClip을 disable에서 enable로 설정함으로써, 재생제어엔진(32)에 의한 재생에 제공할 수 있다. 이에 의해, 다운로드의 진척을 따라가도록 재생처리를 진행하는 "추적 재생(follow-up playback)"을 실현할 수 있다.

[0221] 또, 다운로드 된 디지털 스트림 중 재생이 종료된 것을 enable에서 disable로 변화시킴으로써, 재생이 종료된 디지털 스트림을 삭제의 대상으로 하여 제공할 수 있으므로, 재생처리의 진척을 따라가도록 삭제처리를 진행하는 "추적 삭제(follow-up deletion)"를 실현할 수 있다.

[0222] 다운로드의 진척을 따라가도록 재생처리를 진행하는 "추적 재생"이나 재생처리의 진척을 따라가도록 삭제처리를 진행하는 "추적 삭제"가 가능해지므로, 재생장치와 재생장치를 연결하는 통신회선의 통신율이 낮고, 또 재생장치에 내장되어 있는 로컬 기억장치(18)의 용량이 작은 경우에도, BD-ROM에 기록되어 있는 디지털 스트림과 로컬 기억장치(18)에 다운로드를 하나의 스트림 시퀀스로 취급하여, 연속적으로 재생해 갈 수 있다.

[0223] 이상의 설명은 본 발명의 모든 실시행위의 형태를 제시하고 있는 것은 아니다. 하기 (A) (B) (C) (D) ...의 변경을 한 실시행위의 형태에 의해서도 본 발명의 실시는 가능하다. 본원의 청구항에 기재된 각 발명은 이상에 기재한 복수의 실시 예 및 그들의 변형 예를 확장한 기재 내지 일반화한 기재로 하고 있다. 확장 내지 일반화의 정도는 본 발명의 기술분야의 출원 당시의 기술수준의 특성에 의거한다.

[0224] (A) 상술한 가상 파일 시스템부(38)의 설명에서, 디렉터리 관리정보 및 파일관리정보의 구체적인 데이터 구조에

대해서는 언급하지 않았으나, BD-ROM에서의 파일 시스템이 ISO/IEC13346에 규정된 것인 경우, 디렉터리 관리정보는 "디렉터리 파일" 및 "파일 엔트리"로 구성되어도 된다. 디렉터리 파일은 자신의 하위에 있는 디렉터리의 디렉터리 명을 기술한 파일식별 기술자나 자신의 하위에 있는 파일의 파일명을 기술한 파일식별 기술자로 이루어진다. 이 "파일 엔트리"는 기술자 태그(descriptor tag)와, ICB 태그와, 할당 기술자 길이(allocation descriptor length)와, 확장속성(extension attribute) 및 할당 기술자(allocation descriptor)로 이루어진다.

- [0225] 파일관리정보는 파일 엔트리로 구성되어도 된다. 할당 기술자는 파일을 구성하는 각 익스텐트(extent) 별 기록 위치를 나타낸다.
- [0226] 이 경우, 가상 패키지 정보의 작성은 BD-ROM의 볼륨 관리정보에서의 STREAM 디렉터리에 상응하는 디렉터리 파일에 로컬 기억장치상의 AVClip의 파일명의 파일식별 기술자를 추가하고, BD-ROM의 볼륨 관리정보에서의 STREAM 디렉터리의 하위에 로컬 기억장치상의 AVClip에 대한 파일 엔트리를 추가함으로써 이루어진다.
- [0227] (B) 모든 실시 예에서는 기록매체를 BD-ROM으로 실시하였으나, AVClip 및 Playlist 정보를 기록할 수 있는 기록매체라면 어떤 기록매체라도 좋다. 예를 들어, DVD-ROM, DVD-RAM, DVD-RW, DVD-R, DVD+RW, DVD+R, CD-R, CD-RW 등의 광 디스크, PD, MO 등의 광자기 디스크라도 된다. 또, 콤팩트 플래시(등록상표) 카드, 스마트 미디어, 메모리 스틱, 멀티미디어 카드, PCM-CIA 카드 등의 반도체 메모리 카드라도 된다. 플렉시블 디스크, SuperDisk, Zip, Clif! 등의 자기기록 디스크(i), ORB, Jaz, SparQ, SyJet, EZFley, 마이크로 드라이브 등의 이동 가능한 하드디스크 드라이브(ii)라도 된다. 또한, 기기 내장형의 하드 디스크라도 된다.
- [0228] (C) 모든 실시 예에서의 재생장치는 BD-ROM에 기록된 AVClip을 디코딩한 후에 TV에 출력하고 있으나, 재생장치를 BD-ROM 드라이브만으로 하고, 이 이외의 구성요소를 TV가 구비하도록 해도 된다. 이 경우, 재생장치와 TV를 IEEE1394로 접속된 홈 네트워크에 포함할 수 있다. 또, 실시 예에서의 재생장치는 TV와 접속하여 이용되는 타입이었으나, 디스플레이와 일체형으로 된 재생장치라도 된다. 또한, 각 실시 예의 재생장치에서 처리의 본질적 부분을 이루는 부분만을 재생장치로 해도 된다. 이들 재생장치는 모두 본원 명세서에 기재된 발명이므로, 이들의 어떠한 형태라도 각 실시 예에 제시된 재생장치의 내부 구성을 토대로 재생장치를 제조하는 행위는 본원의 명세서에 기재된 발명의 실시행위가 된다. 각 실시 예에 제시한 재생장치의 유상·무상에 의한 양도(유상의 경우는 판매, 무상의 경우는 증여가 된다), 대여, 수입하는 행위도 본 발명의 실시행위이다. 점포 전시, 카탈로그 권유, 팸플릿 배포에 의해 이들의 양도나 대여를 일반 사용자에게 제의하는 행위도 본 재생장치의 실시행위이다.
- [0229] (D) 각 플로차트에 제시된 프로그램에 의한 정보처리는 하드웨어 자원을 이용하여 구체적으로 실현되어 있으므로, 상기 플로차트에 처리순서를 제시한 프로그램은 단일체로 발명으로 성립한다. 모든 실시 예는 재생장치에 포함된 형태로, 본 발명에 관한 프로그램의 실시행위에 대한 실시 예를 제시하였으나, 재생장치로부터 분리하여, 각 실시 예에 제시한 프로그램 단일체를 실시해도 된다. 프로그램 단일체의 실시행위에는 이들 프로그램을 생산하는 행위(1)나, 유상·무상에 의해 프로그램을 양도하는 행위(2), 대여하는 행위(3), 수입하는 행위(4), 쌍방향 전자통신 회선을 경유하여 공중에 제공하는 행위(5), 점포 전시, 카탈로그 권유, 팸플릿 배포에 의해 프로그램의 양도나 대여를 일반 사용자에게 제의하는 행위(6)가 있다.
- [0230] (E) 각 플로차트에서 시계열로 진행되는 각 단계의 「시(time)」의 요소를 발명을 특정하기 위한 필수 사항이라고 생각한다. 그렇게 하면, 이들 플로차트에 의한 처리 순서는 재생방법의 사용 예를 개시하고 있음을 알 수 있다. 각 단계의 처리를 시계열로 행함으로써 본 발명의 본래의 목적을 달성하고, 작용 및 효과가 있도록 이들의 플로차트의 처리를 행하는 것이라면, 본 발명에 관한 기록방법의 실시행위에 해당하는 것은 말할 필요도 없다.
- [0231] (F) 각 실시 예에서 기록매체에 기록되는 디지털 스트림은 AVClip이었으나, DVD-Video 규격, DVD-Video Recording 규격의 VOB(Video Object)라도 된다. VOB는 비디오 스트림, 오디오 스트림을 다중화함으로써 얻어진 ISO/IEC13818-1 규격 준거의 프로그램 스트림이다. 또한, AVClip에서의 비디오 스트림은 MPEG4나 WMV 방식이어도 된다. 또한, 오디오 스트림은 Linear-PCM 방식, Dolby-AC 3 방식, MP3 방식, MPEG-AAC 방식, Dts, WMA(Windows(등록상표) media audio)라도 된다.
- [0232] (G) 각 실시 예에서의 영화 작품은 아날로그 방송으로 방송된 아날로그 영상신호를 인코드함으로써 얻어진 것이어도 된다. 디지털 방송으로 방송된 트랜스 포트 스트림으로 구성되는 스트림 데이터라도 된다.
- [0233] 또한, 비디오 테이프에 기록되어 있는 아날로그/디지털의 영상신호를 인코드하여 콘텐츠를 얻어도 된다. 또한, 비디오 카메라로부터 직접 취득한 아날로그/디지털 영상신호를 인코드하여 콘텐츠를 얻어도 된다. 이 외에도, 배신(distribution) 서버에 의해 배신되는 디지털 저작물이라도 된다.

[0234] (H) BD-J 모듈(29b)은 위성방송 송신을 위해 기기에 내장된 Java(등록상표) 플랫폼이어도 된다. BD-J 모듈(29b)이 이와 같은 Java(등록상표) 플랫폼이면, 본 발명에 관한 재생장치는 MHP용 STB로서의 처리를 겸용하게 된다.

[0235] 또한, 휴대전화의 처리제어를 위해 기기에 내장된 Java(등록상표) 플랫폼이어도 된다. 이와 같은 BD-J 모듈(29b)이 이와 같은 Java(등록상표) 플랫폼이면, 본 발명에 관한 재생장치는 휴대전화로서의 처리를 겸용하게 된다.

[0236] (I) 계층 모델에서 BD-J 모드 상에 HDMV 모드를 배치해도 된다. 특히, HDMV 모드에서의 동적 시나리오의 해석이나 동적 시나리오에 의거한 제어순서의 실행은 재생장치에 대한 부담이 가벼우므로, HDMV 모드를 BD-J 모드 상에서 실행시켜도 하등의 문제는 발생하지 않기 때문이다. 또, 재생장치나 영화작품의 개발 시에 동작보증이 하나의 모드로 종료되기 때문이다.

[0237] 또한, BD-J 모드만으로 재생처리를 실행해도 된다. BD-J 모드에서도 PL의 재생과 동기한 재생제어를 할 수 있게 되므로, 역지로 HDMV 모드를 설치하지 않아도 된다는 이유에 의한다.

[0238] (J) 로컬 기억장치(18)에 다운로드 되는 AVClip은 특전 영상(bonus video)이나, 또는, 제작자나 배우의 인터뷰라도 된다.

[0239] (K) 어떤 Clip이 Missing Stream 혹은 고쳐쓰기 가능 중 어느 상태인가를 애플리케이션으로부터 아는 방법으로, 상기 실시 예에서는 getMissingStream 메소드를 호출하는 것으로 하였으나, java(등록상표).net, java(등록상표).io 등의 java(등록상표) 라이브러리를 사용하여 실현해도 된다. 이 경우, 이하의 a), b), c)의 방법을 사용하는 것이 현실적이다.

[0240] a) Binding Unit Manifest File(스트림 위치 저장정보)의 내용을 XML Parser를 이용하여 해석한다.

[0241] Binding Unit Manifest File의 예 :

[0242] <Progressive>

```
[0243] <ProgressiveAsset VPFilename = "BDMV/STREAM/01000.m2ts">
```

[0244] ← 식별자

[0245] <BUDAFFile name = "1/1/streams/0.m2ts">/BUDAFFile>

[0246] ← 파일 패스

[0247] </ProgressiveAsset>

```
[0248] <ProgressiveAsset VPFilename = "BDMV/STREAM/01001.m2ts">
```

[0249] <BUDAFFile name = "1/1/streams/1.m2ts">/BUDAFFile>

[0250] </ProgressiveAsset>

[0251] <PorgressiveAsset VPFilename = "BDMV/STREAM/01002.m2ts">

[0252] <BUDAFFile name = "1/1/streams/2.m2ts"></BUDAFFile>

[0253] </ProgressiveAsset>

[0254] </Progressive>

[0255] * "BUDA"는 Binding Unit Data Area의 약자이며, Binding Unit Data Area는 가상 패키지에 이용하는 파일을 저장하는 로컬 기억장치 내의 영역을 말한다.

[0256] b) java(등록상표).io를 사용하여 Binding Unit Manifest File에 기재되어 있는 파일 패스(스트림 저장위치)에 xxxxx.m2ts가 존재하는가 확인한다.

[0257] 예 : new java(등록상표).io.File("1/1/streams/0.m2ts").exists() ;

```
[0258] // false의 경우는 Missing Stream
```

[0259] c) 다운로드가 완료되었는가 여부의 판정은 java(등록상표).net, java(등록상표).io를 사용하여 서버상의 파일

의 사이즈와 로컬 기억장치상의 파일 사이즈가 일치하고 있는가 여부를 확인한다.

예 :

```
new java(등록상표).net.URL(http://xxx.com/vp_data/0.m2ts).openConnection().getContentLength()
```

```
== new java(등록상표).io.File("1/1/streams/0.m2ts").length();
```

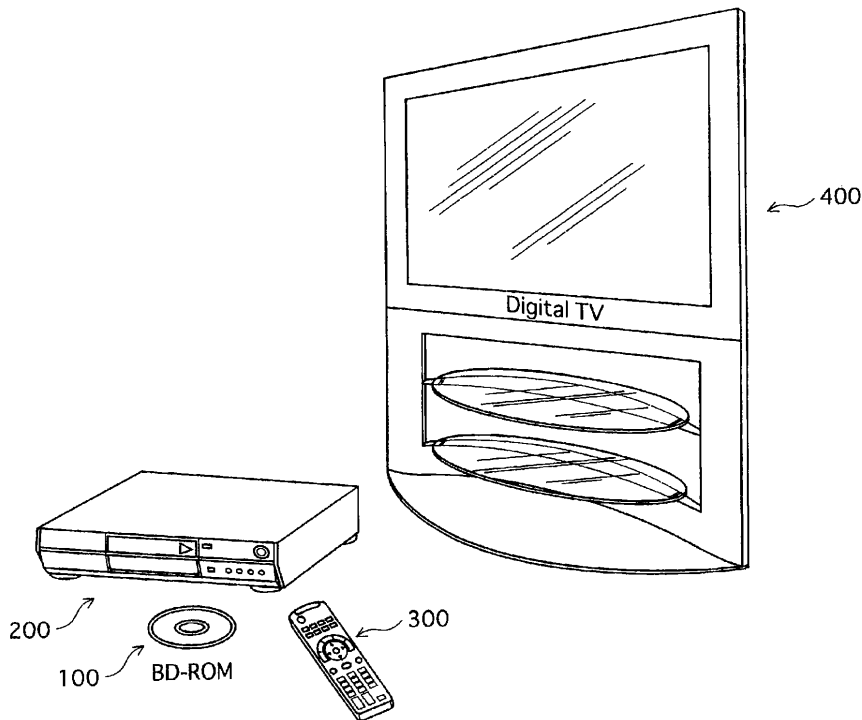
```
// 일치하고 있으면 다운로드가 완료되었다고 간주하는 것이 가능하다.
```

산업상 이용가능성

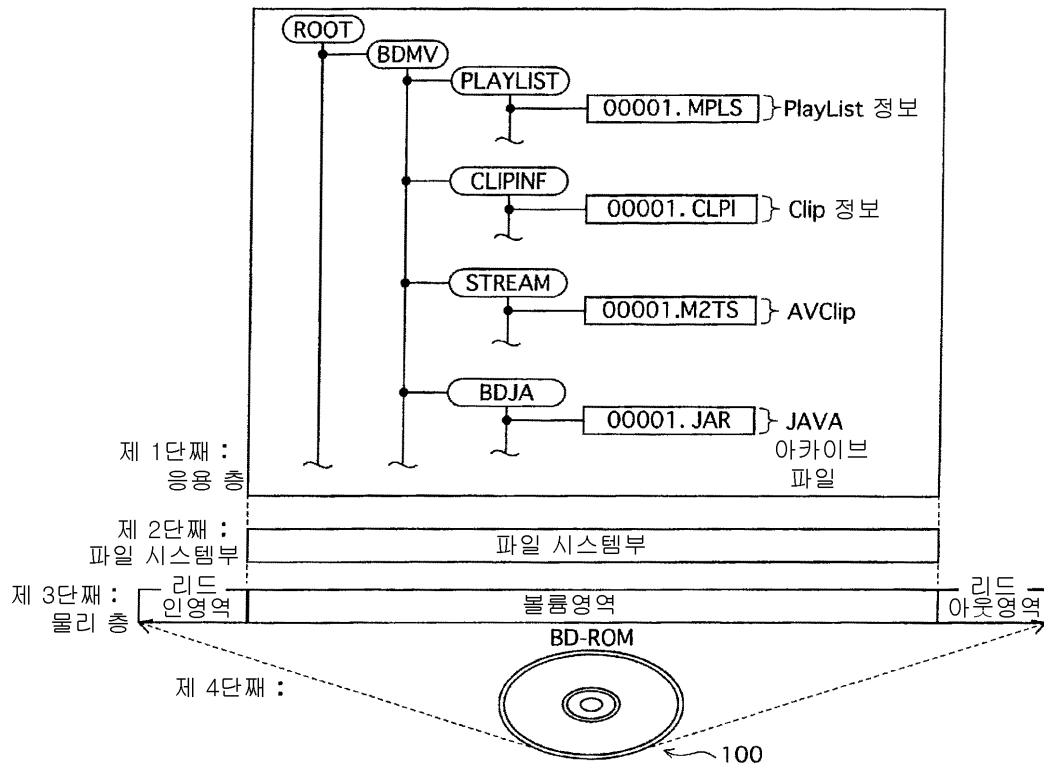
본 발명에 관한 재생장치는 홈시어터 시스템에서의 이용과 같이, 개인적인 용도로 이용되는 경우가 있을 수 있다. 그러나 본 발명은 상기 실시 예에 내부 구성이 개시되어 있고, 이 내부 구성에 의거하여 양산하는 것이 명백하므로, 자질에 있어서 공업상 이용할 수 있다. 그러므로 본 발명에 관한 재생장치는 산업상의 이용가능성을 갖는다.

도면

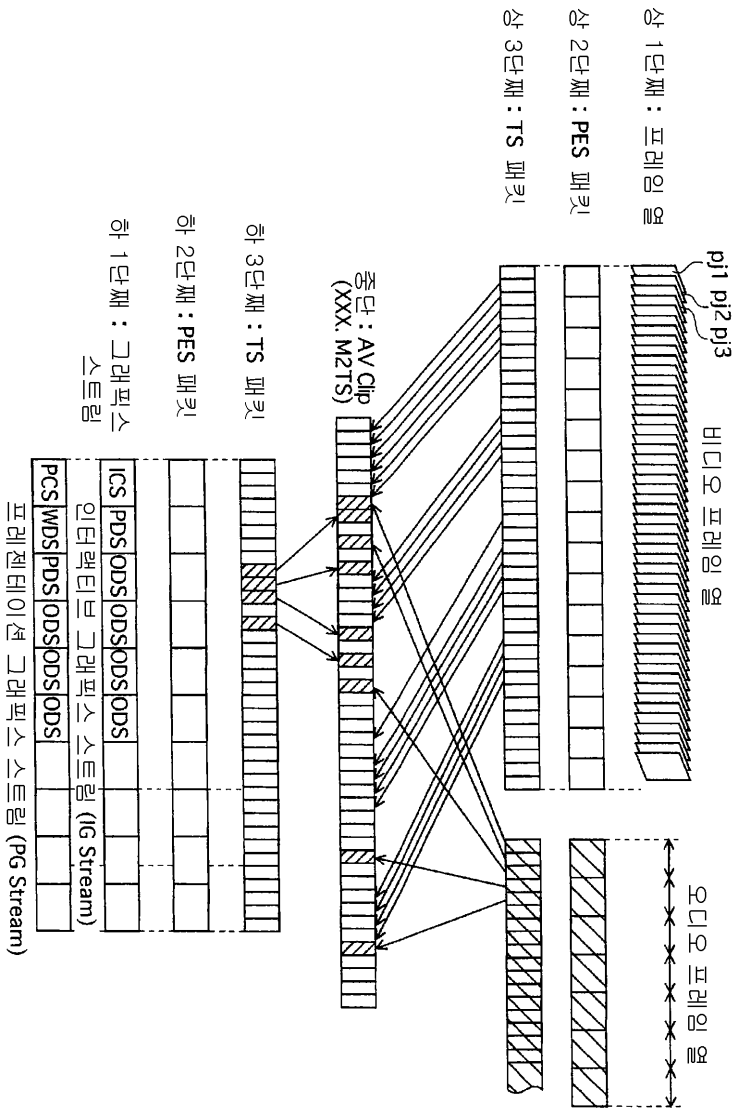
도면1



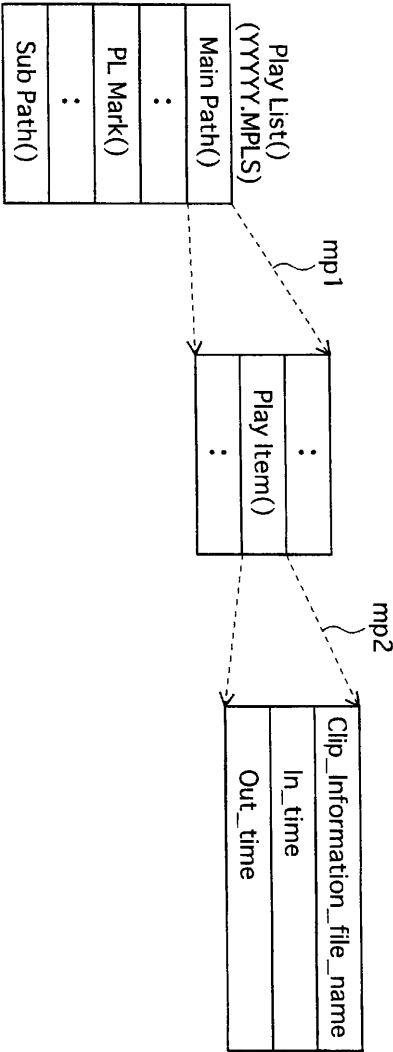
도면2



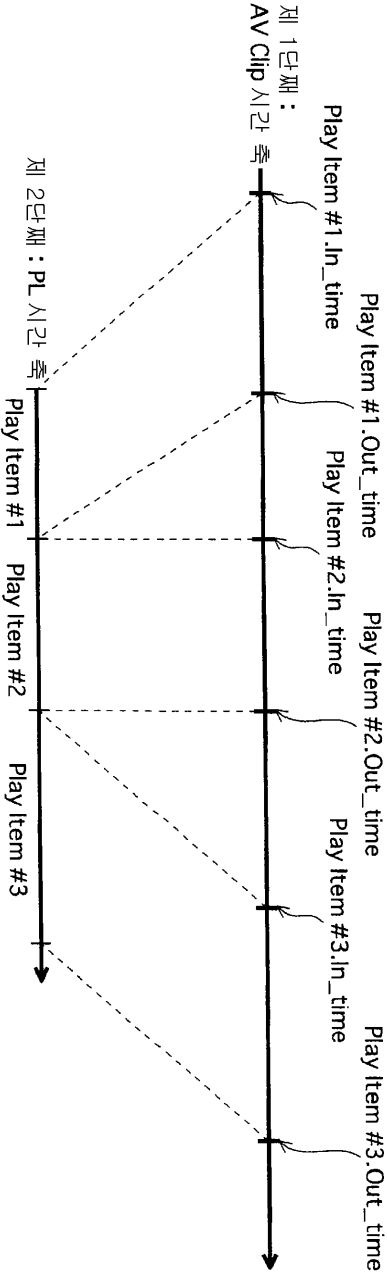
도면3



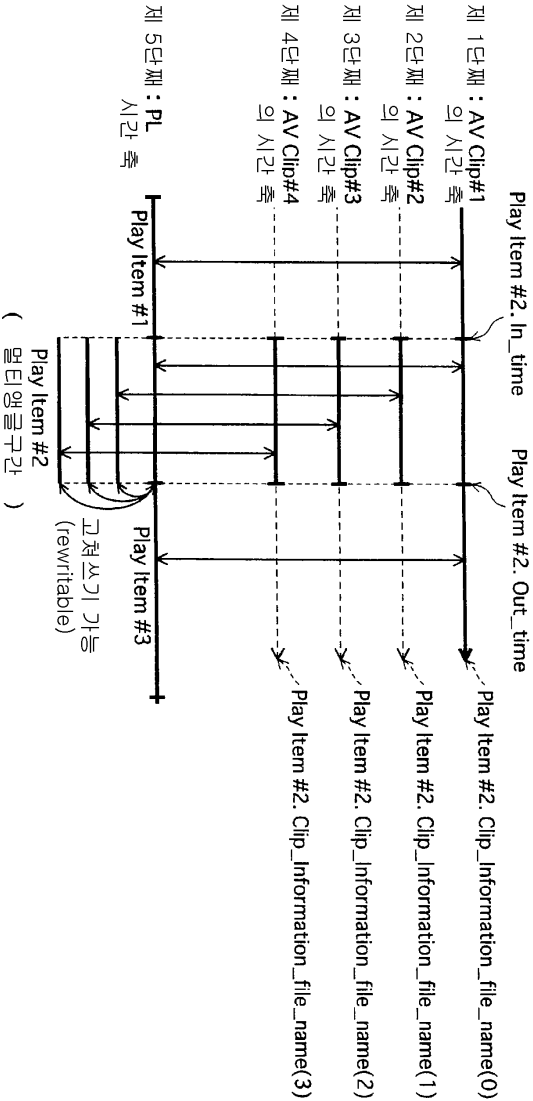
도면4



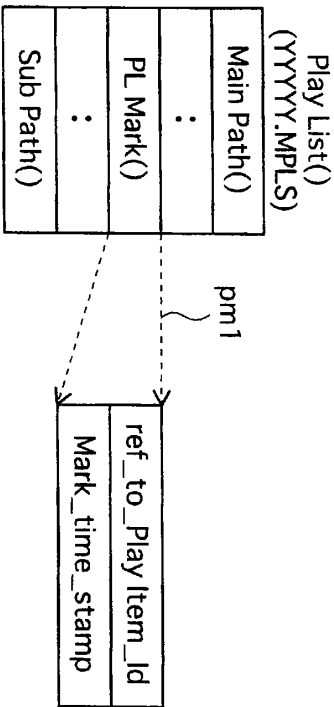
도면5

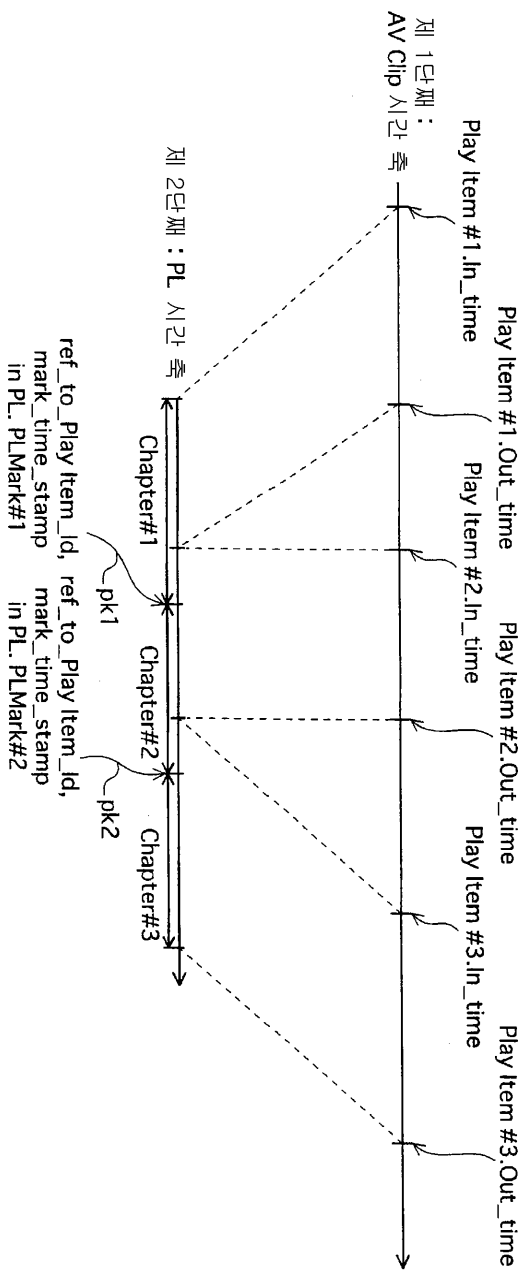


도면6



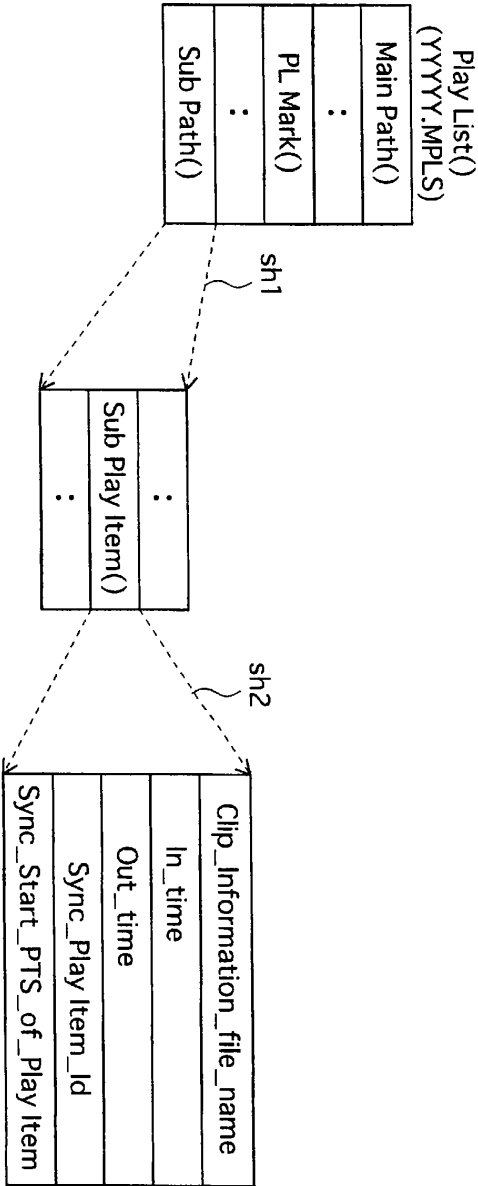
도면7



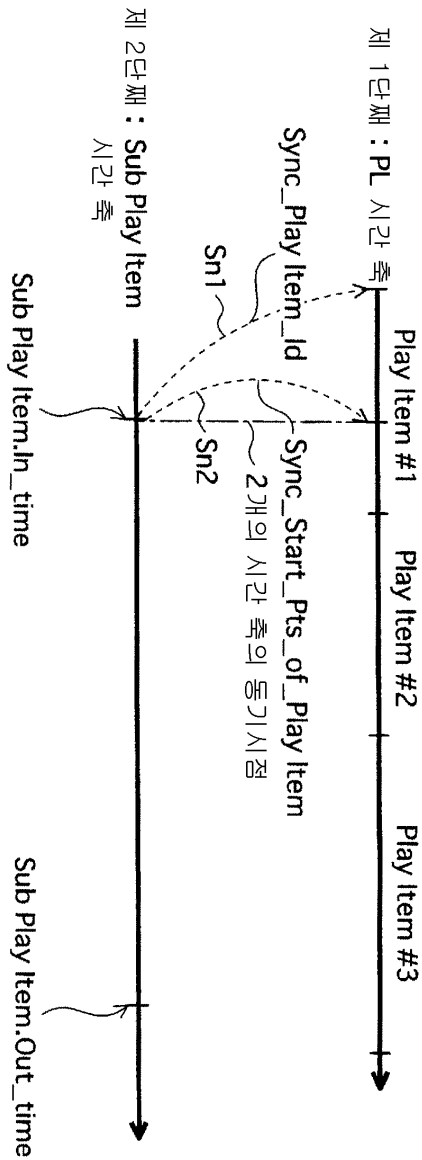


도면8

도면9



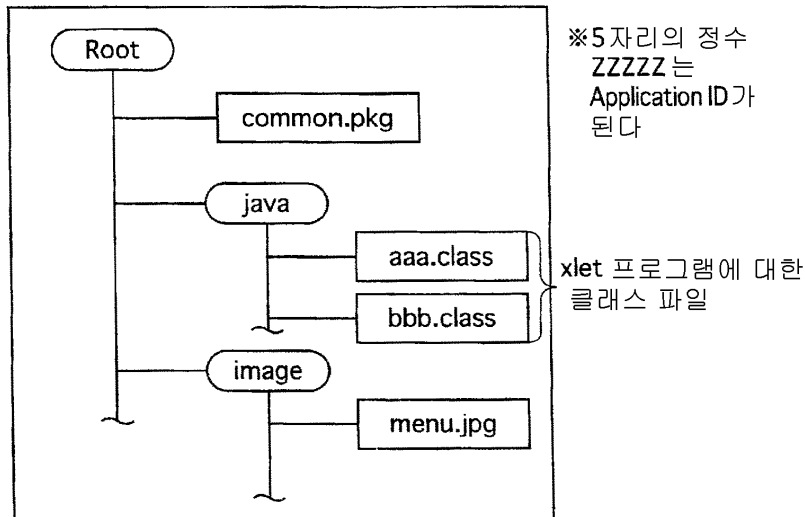
도면10



도면11

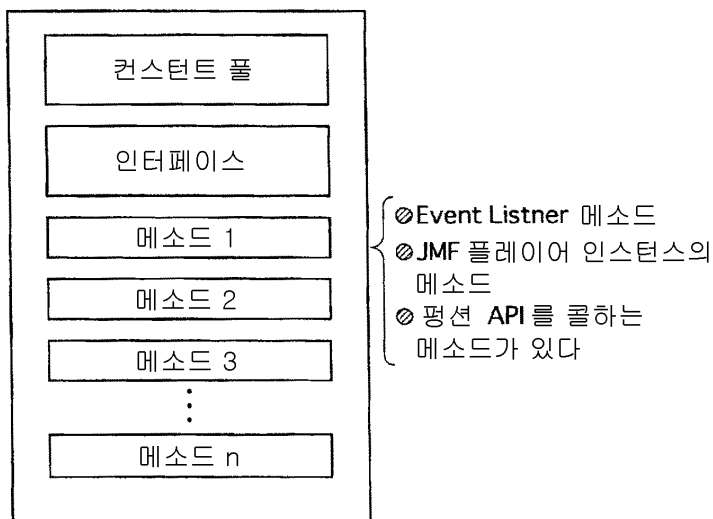
(a)

JAR 파일 (ZZZZZ.JAR)

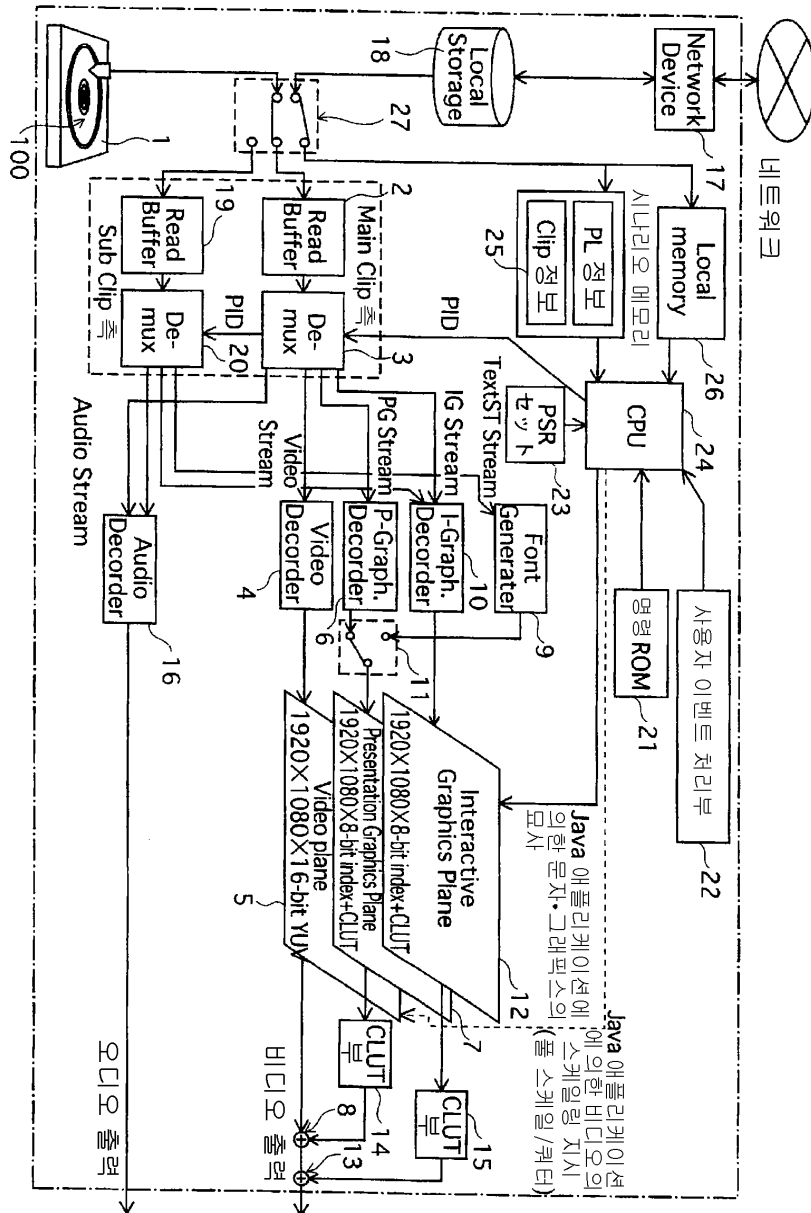


(b)

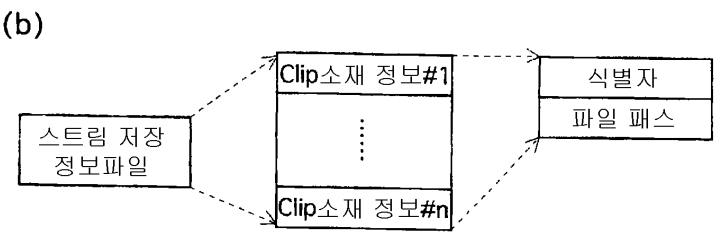
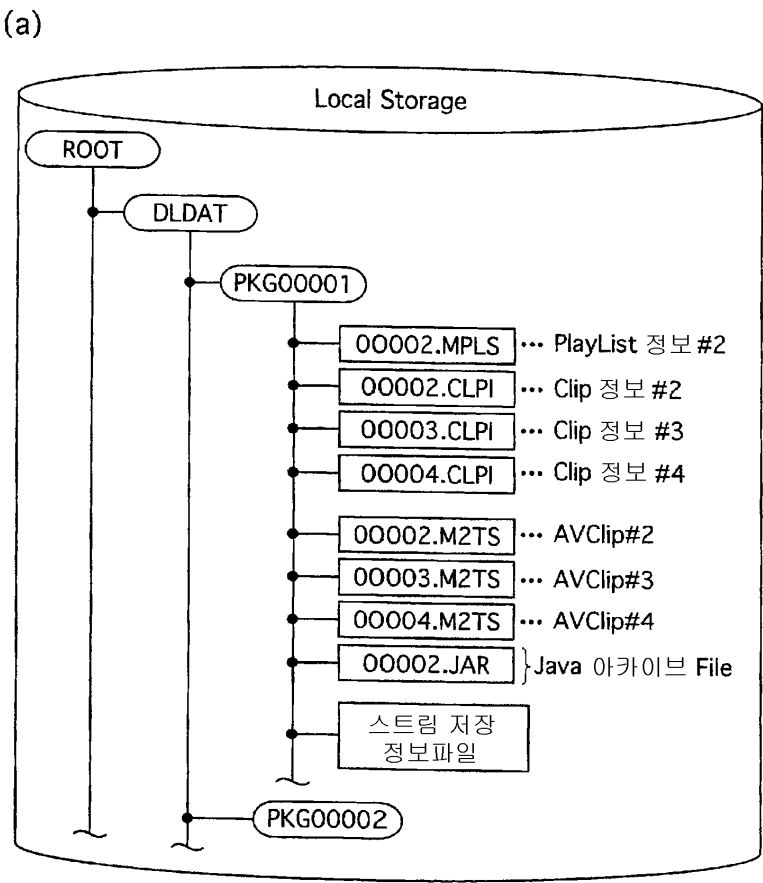
xlet 프로그램에 대한 클래스 파일



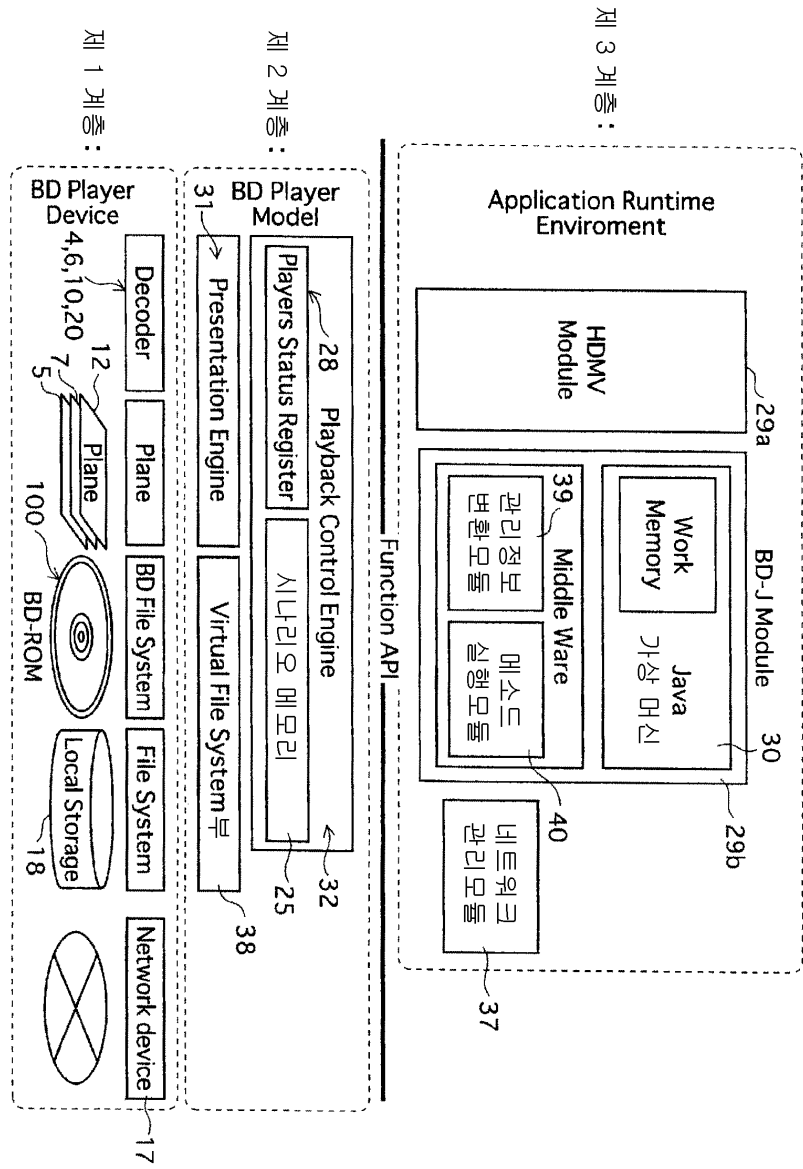
도면12



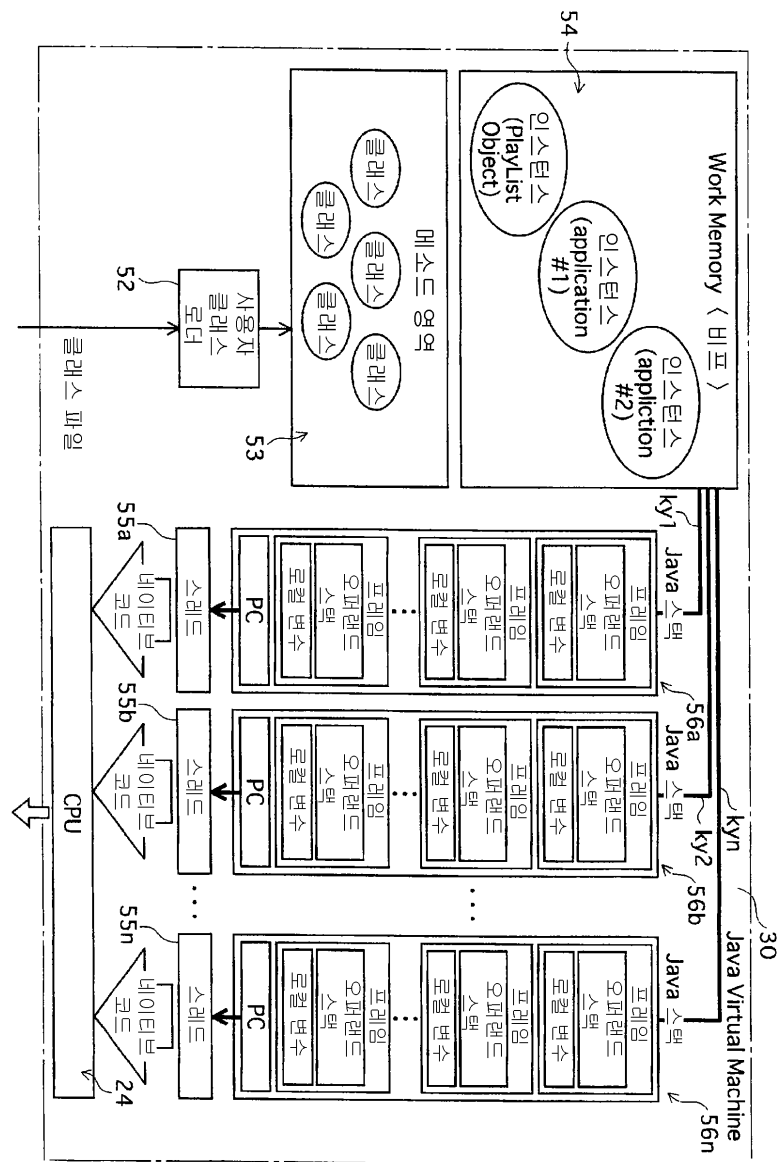
도면13



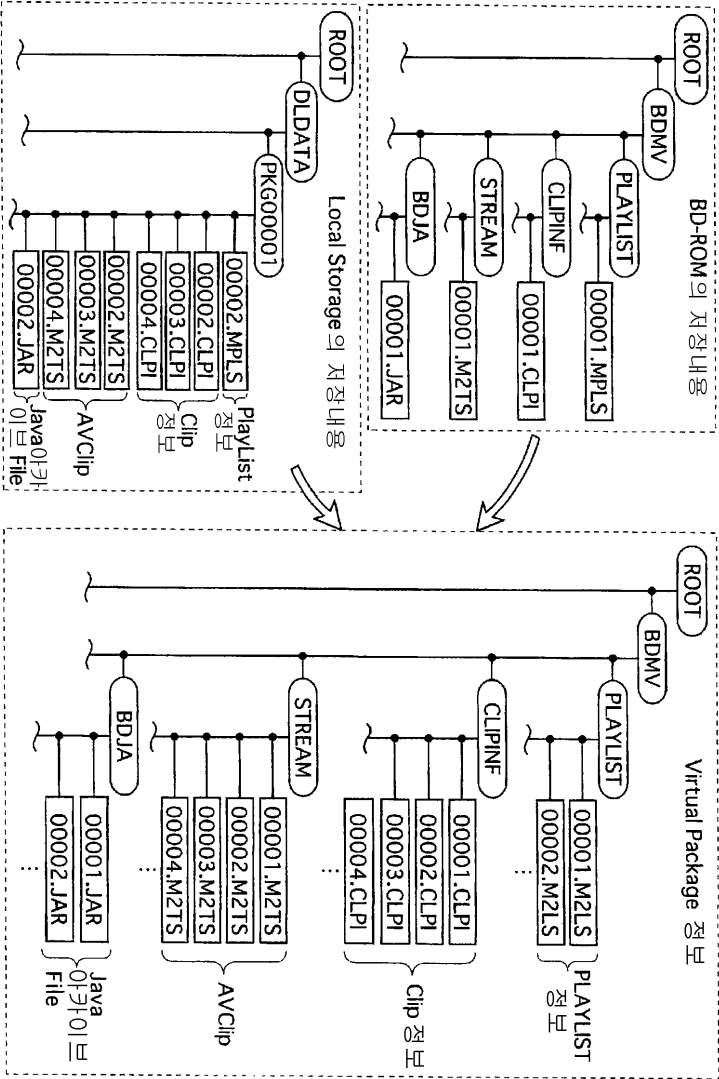
도면14



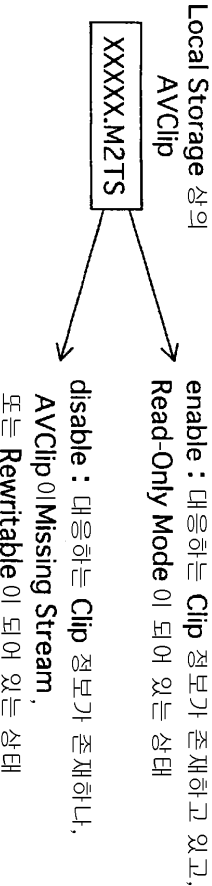
도면15



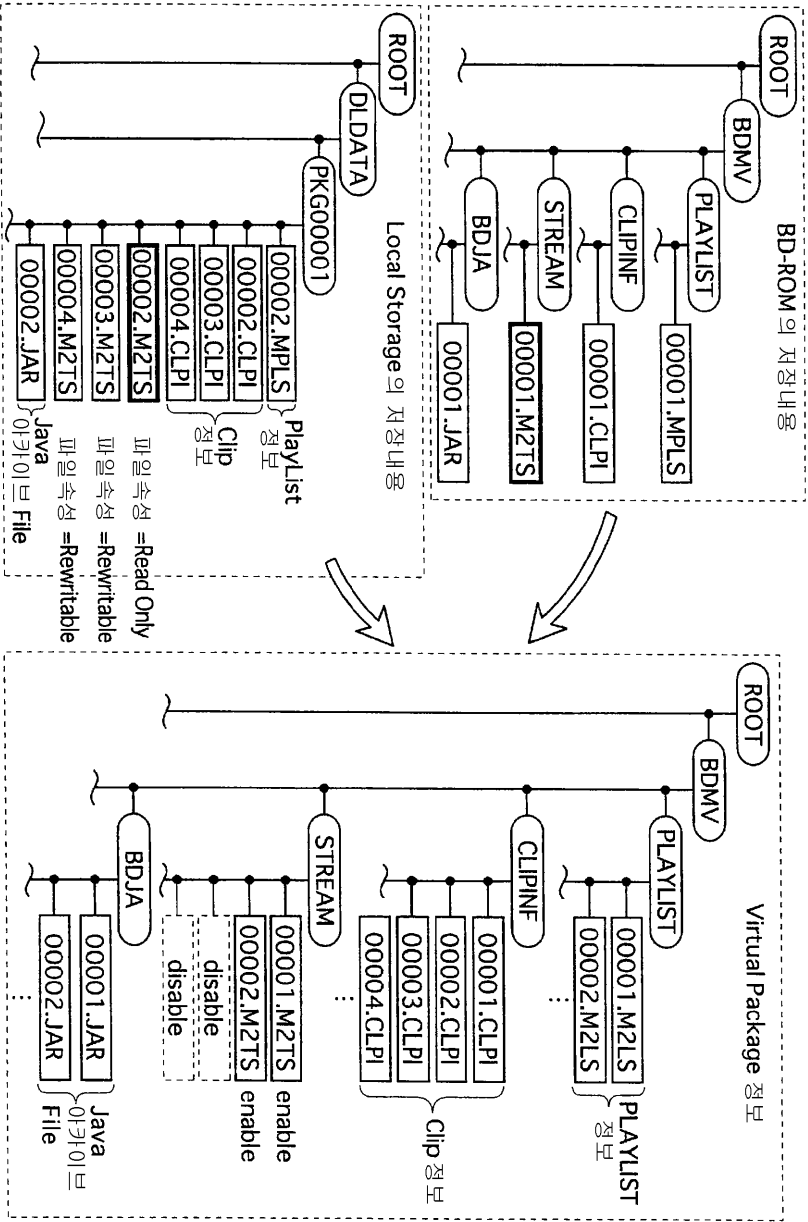
도면16



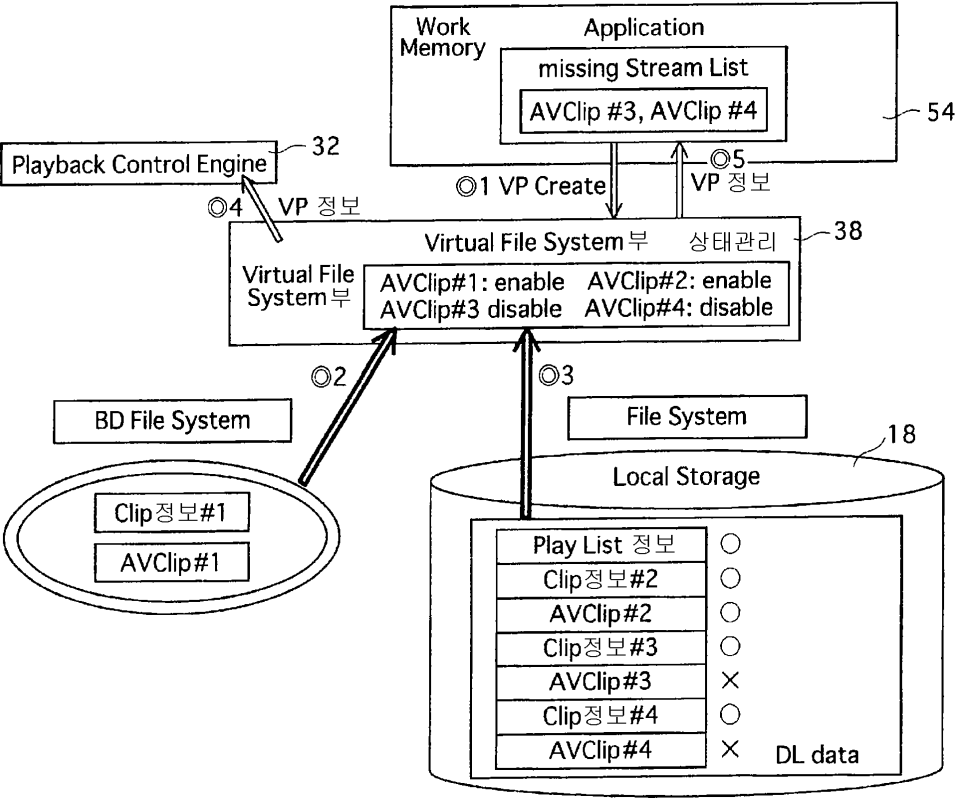
도면17

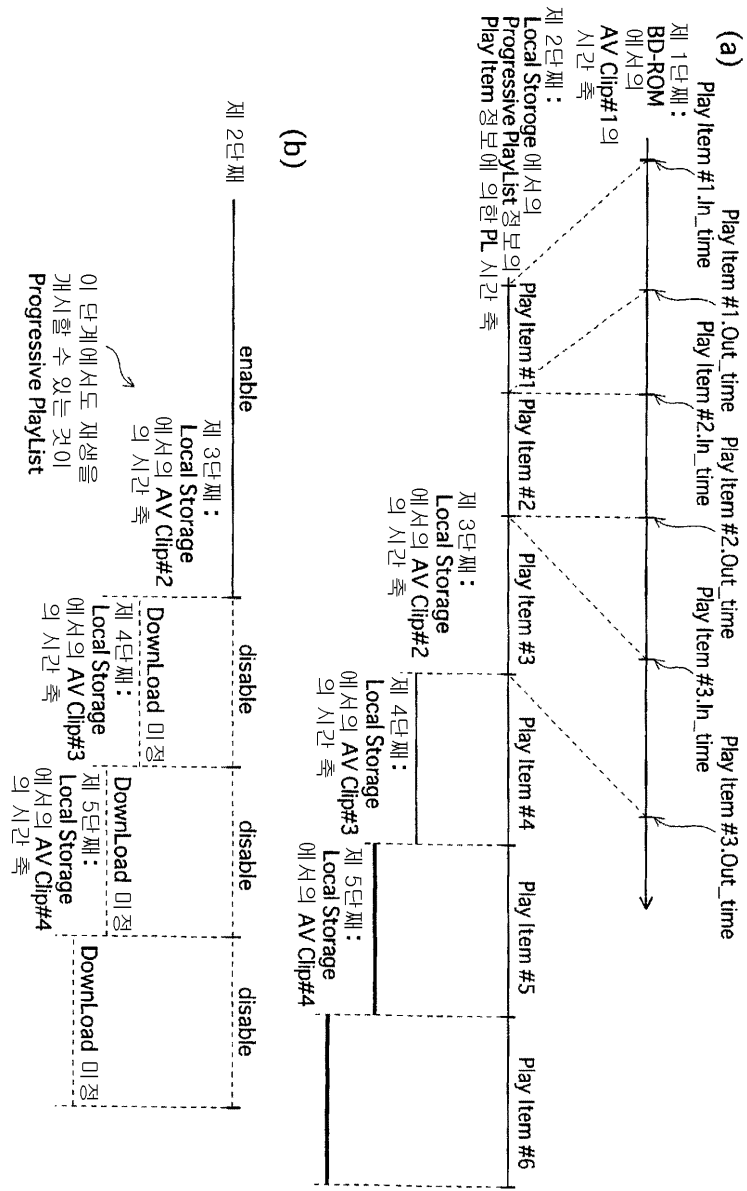


도면18

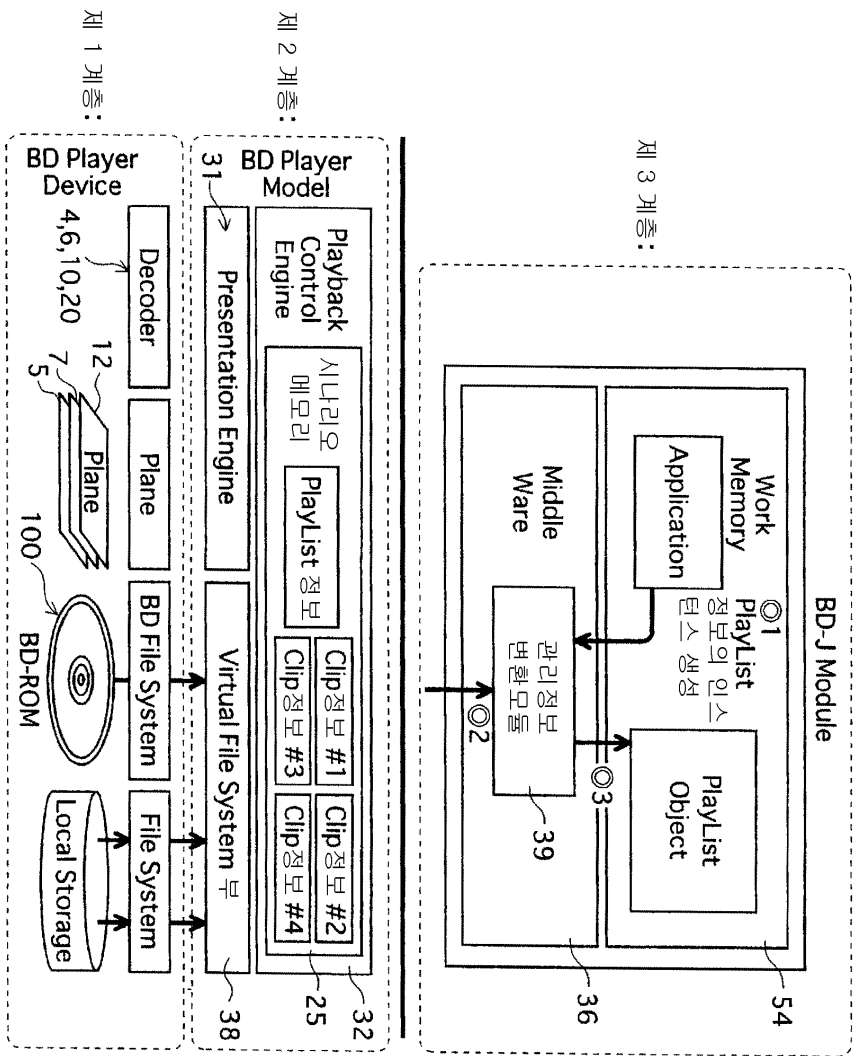


도면19

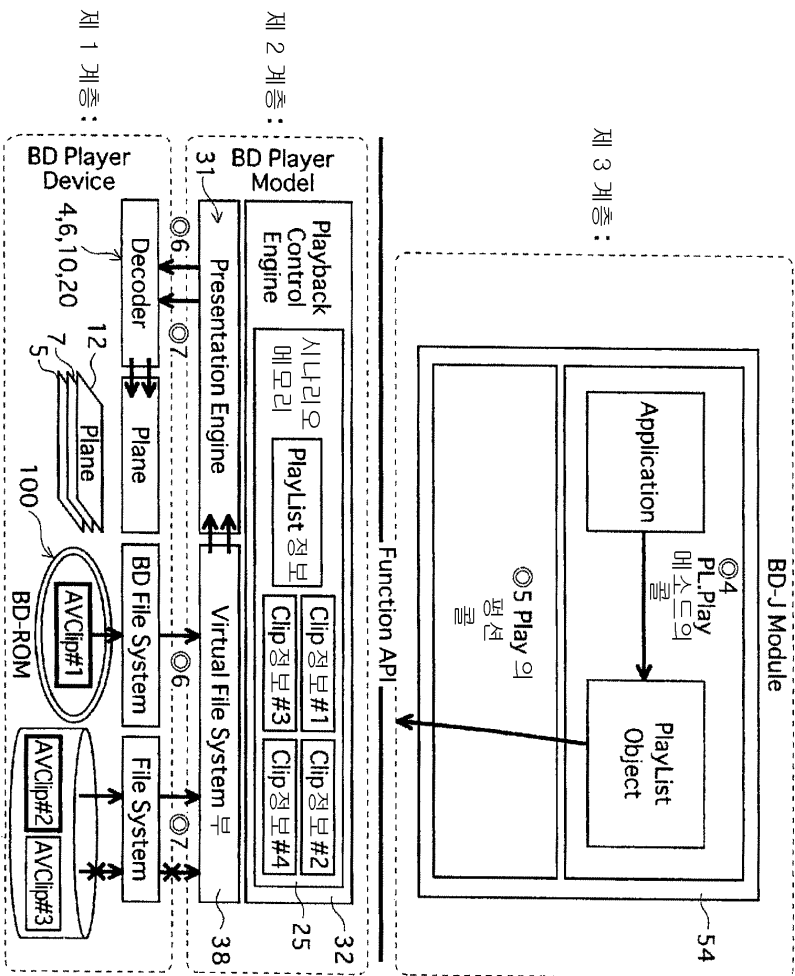




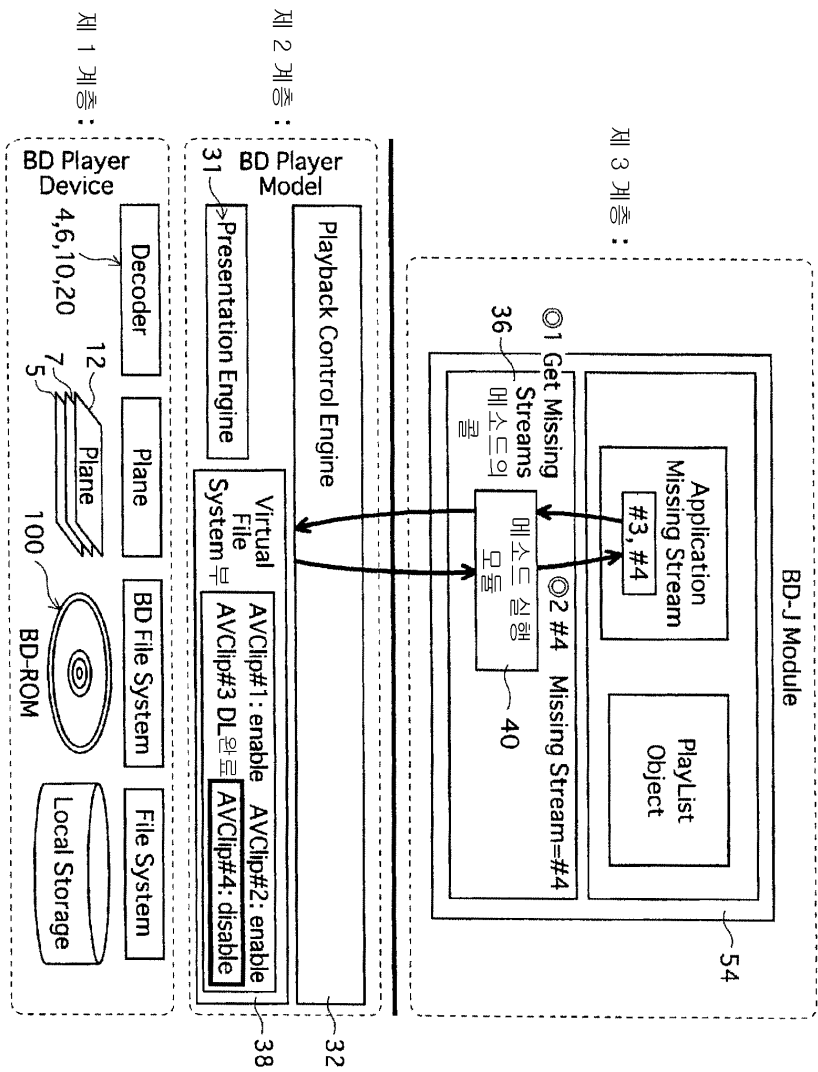
도면21



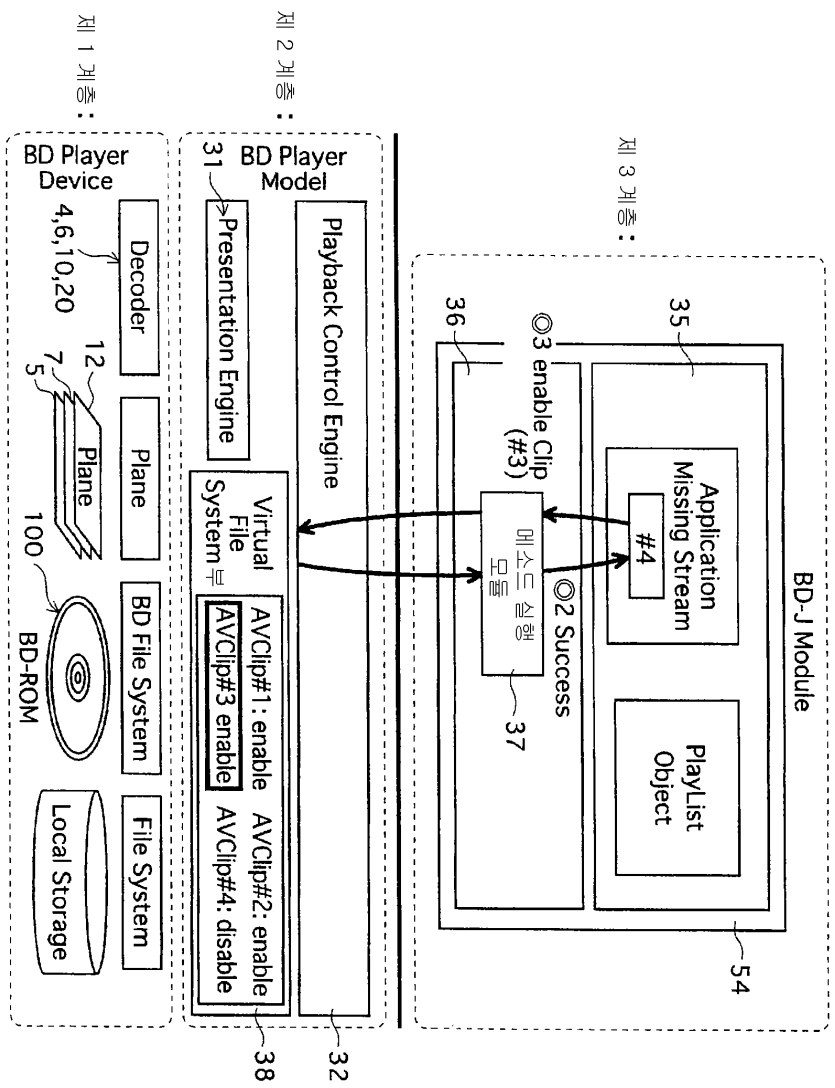
도면22



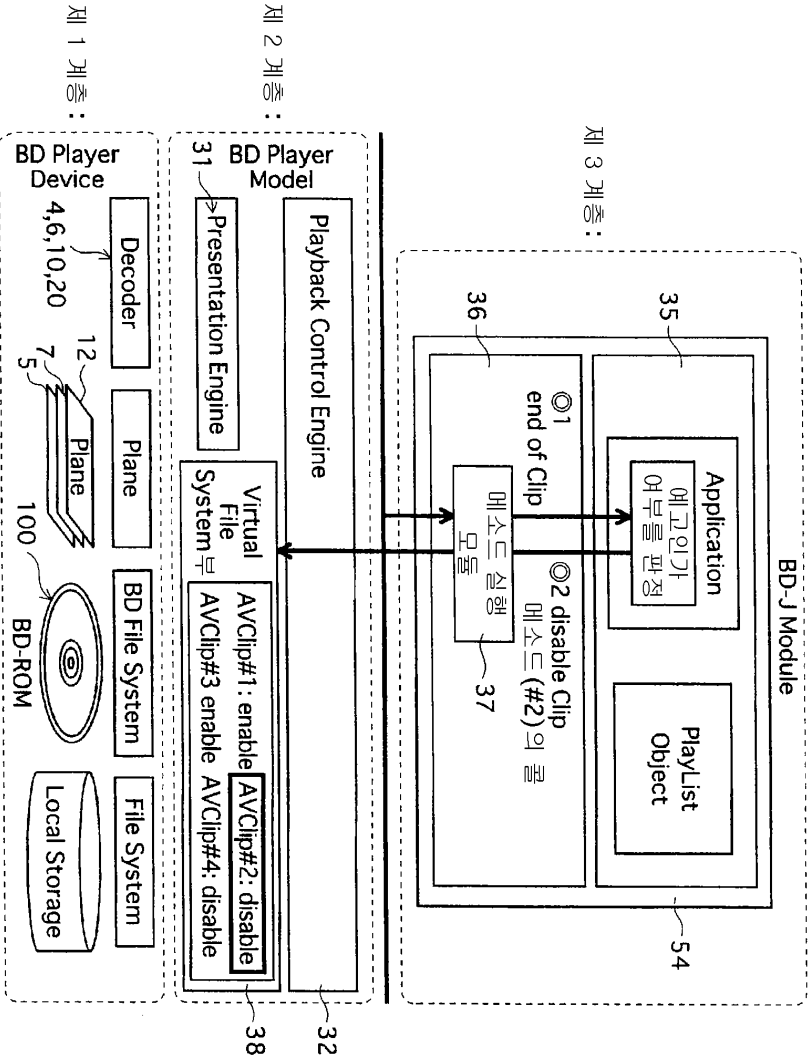
도면23



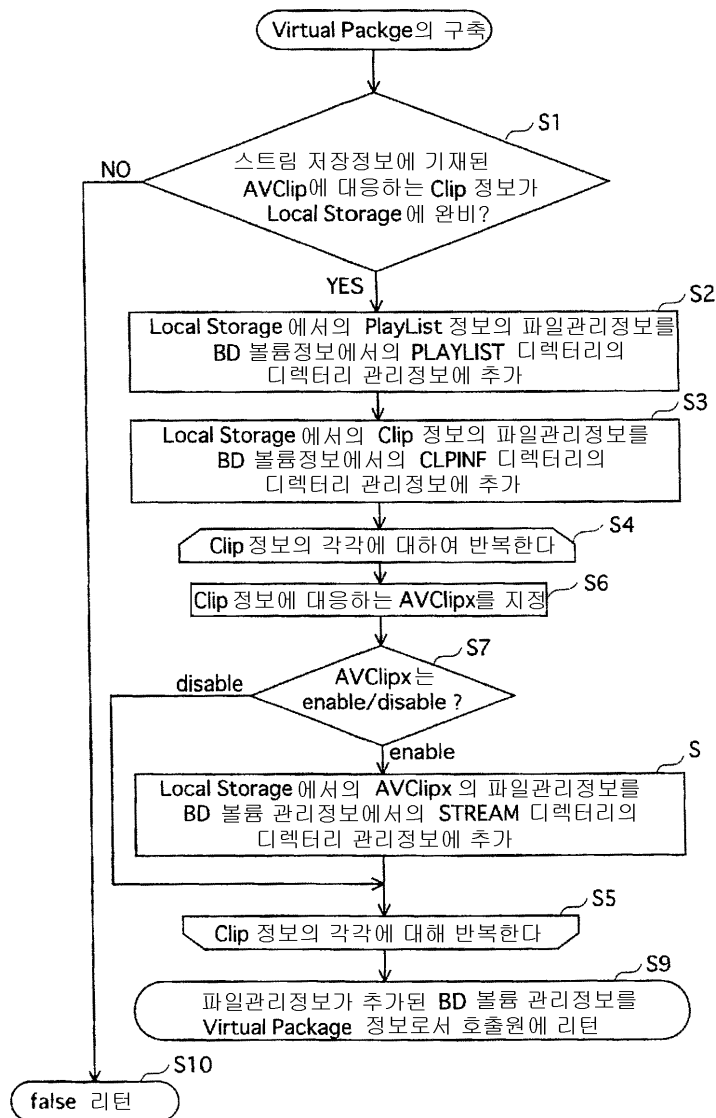
도면24



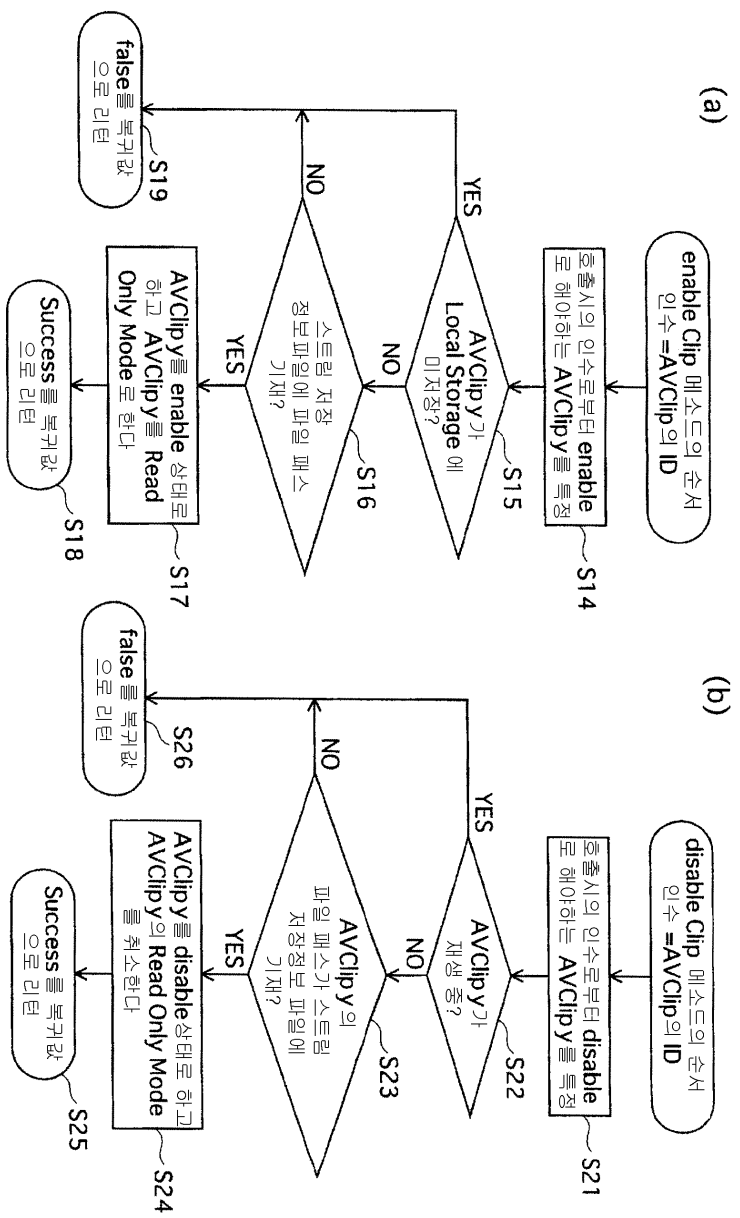
도면25



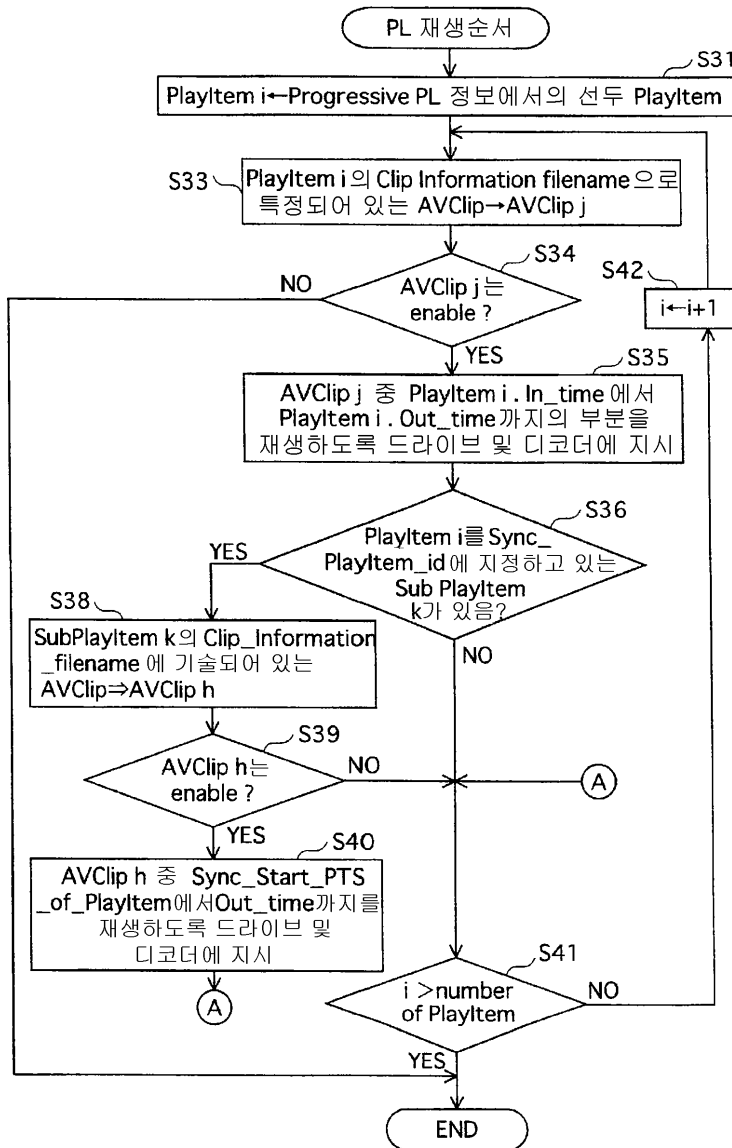
도면26



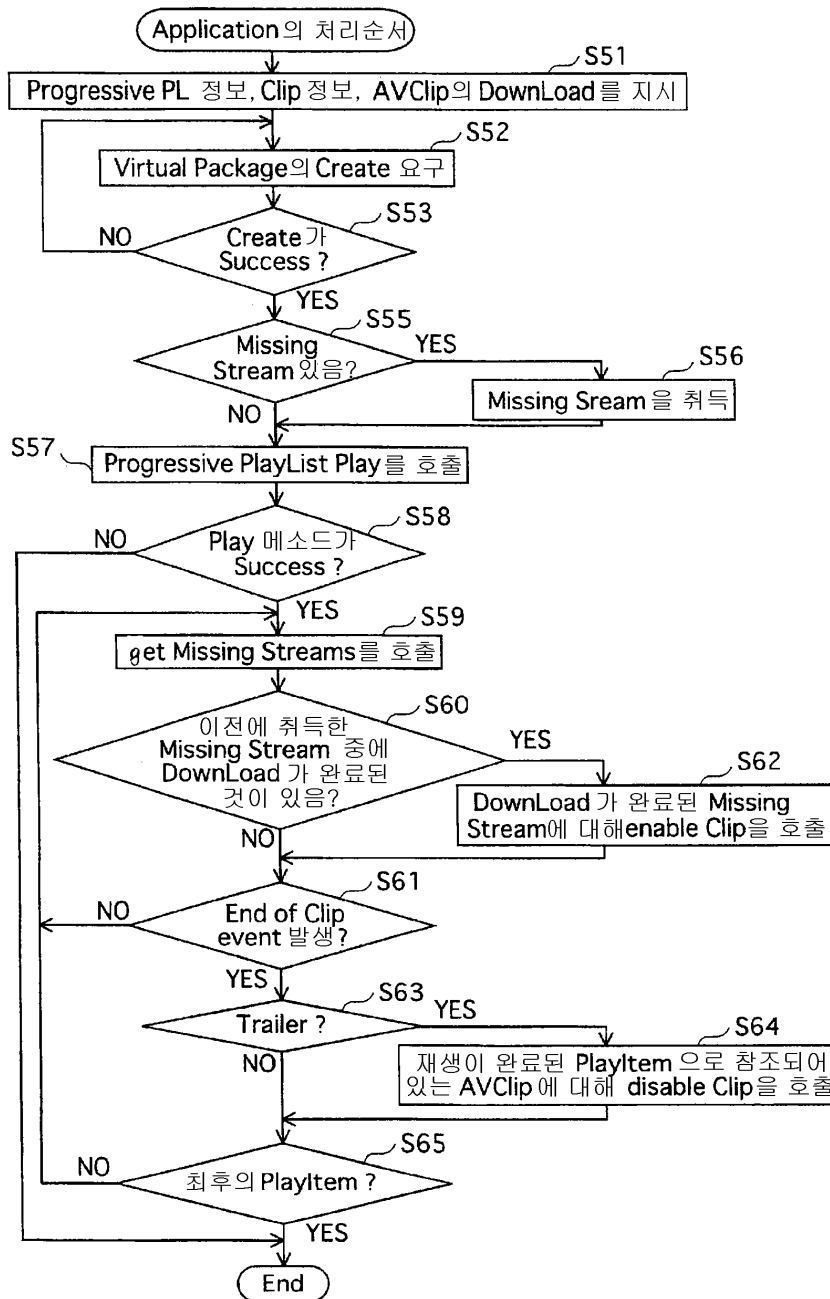
도면27



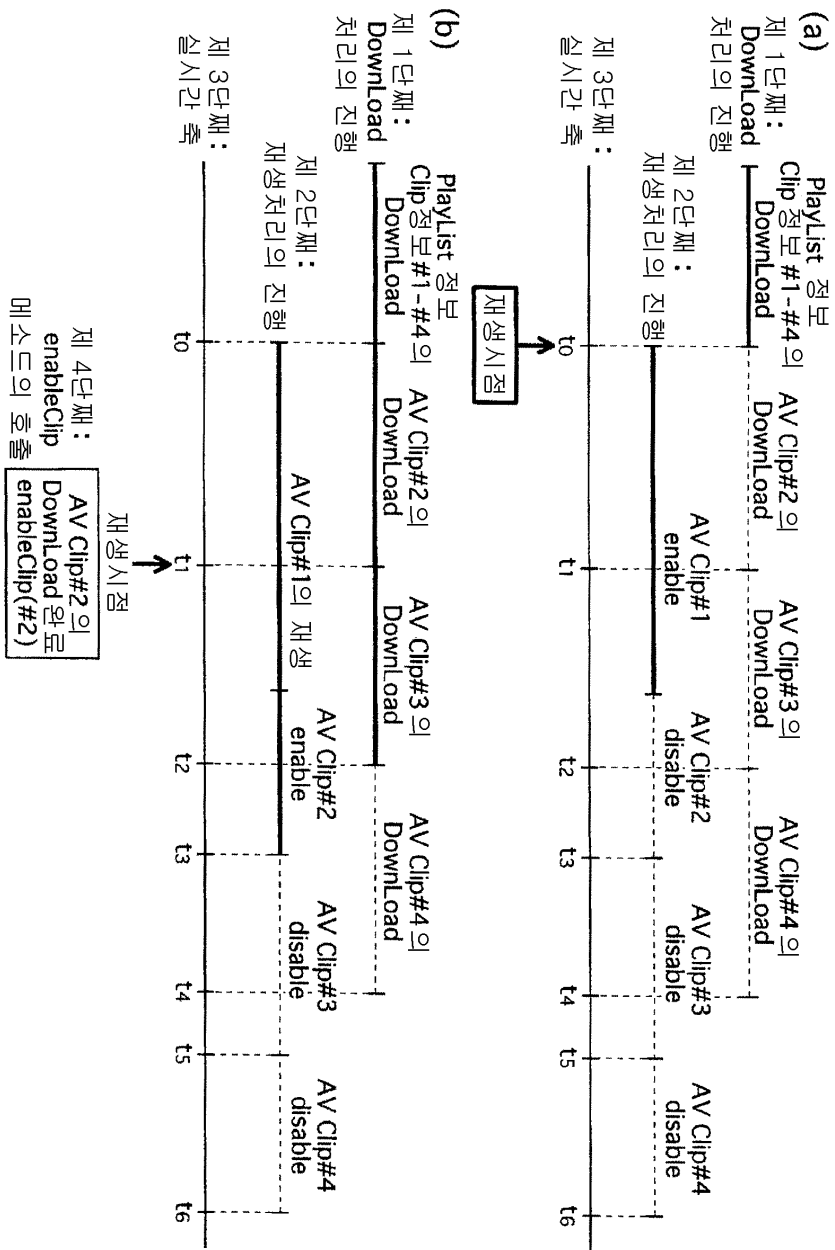
도면28

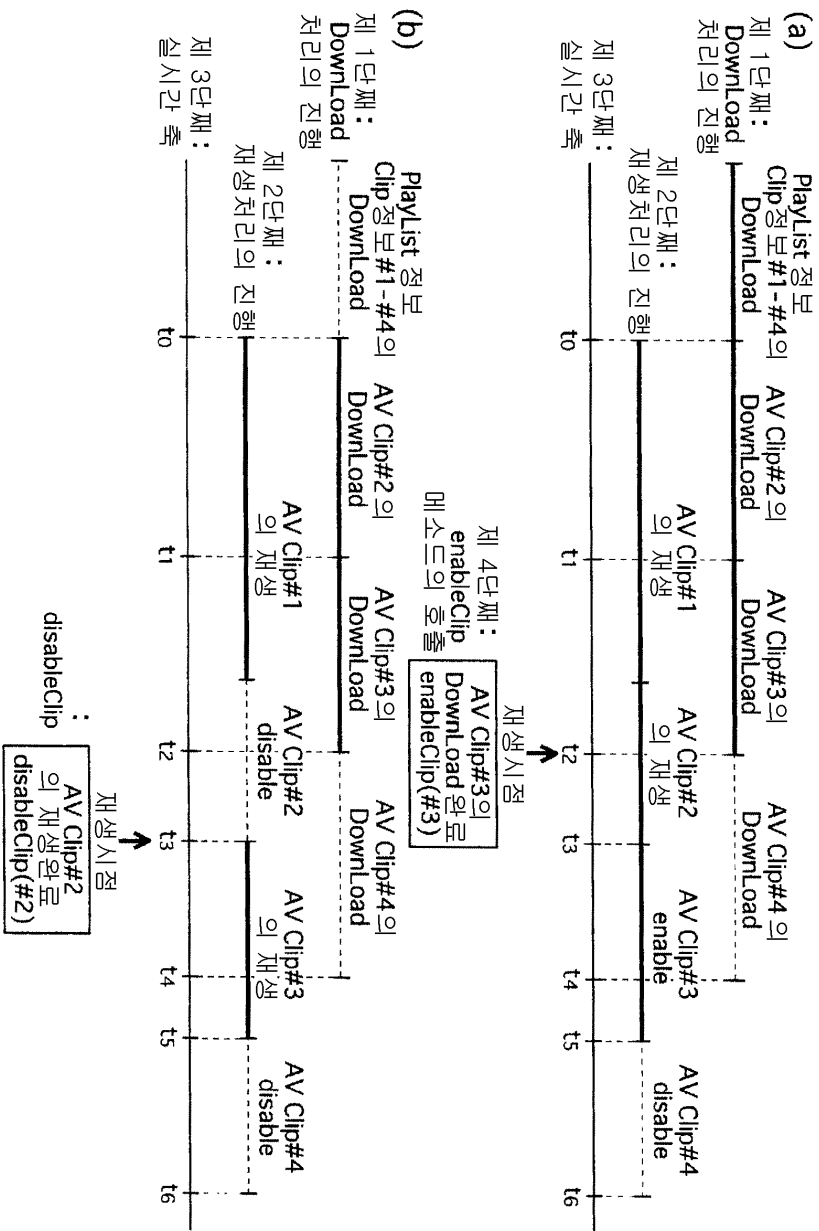


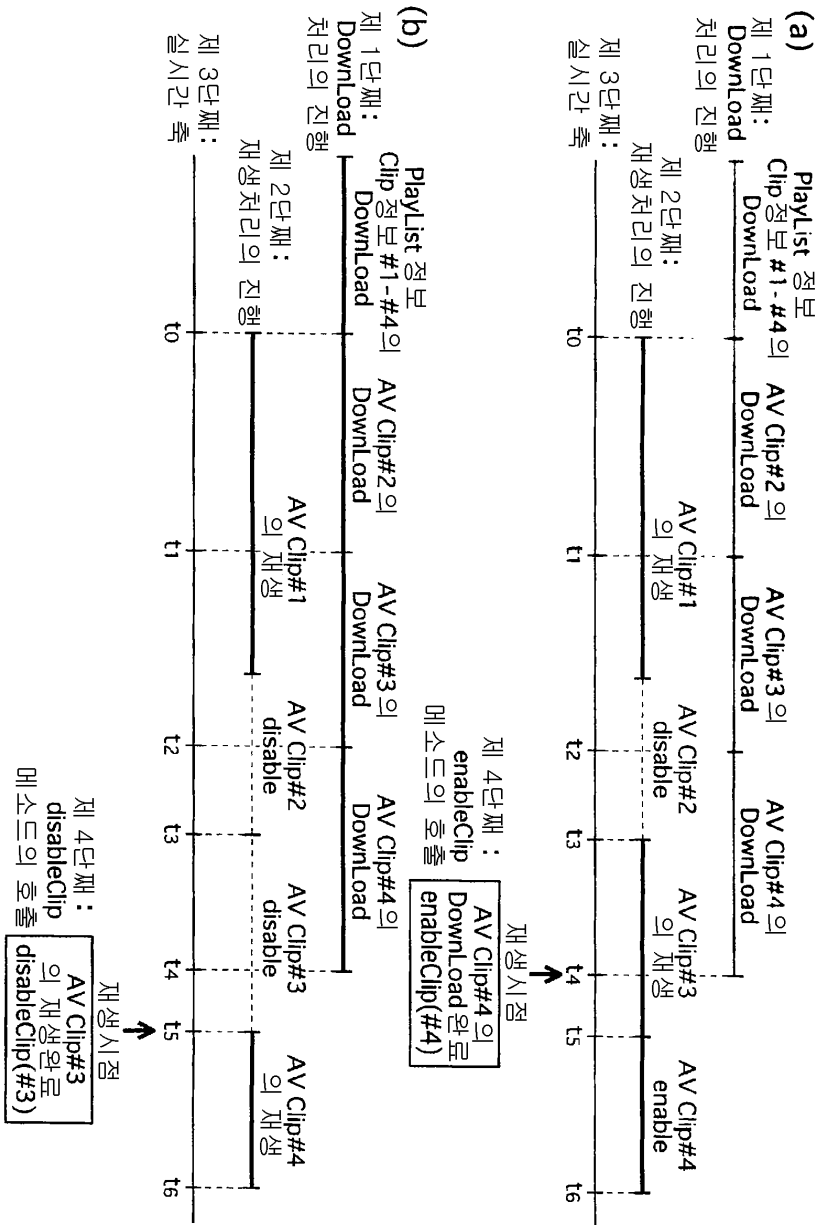
도면29



도면30







도면33

