(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。Int. Cl.
G11B 27/10 (2006.01)
G11B 20/10 (2006.01)
G11B 27/00 (2006.01)
G11B 20/12 (2006.01)

(11) 공개번호 10-2007-0053270
(43) 공개일자 2007년05월23일

(21) 출원번호 10-2007-7006148
(22) 출원일자 2007년03월16일
심사청구일자 없음
번역문 제출일자 2007년03월16일
(86) 국제출원번호 PCT/JP2005/014490
국제출원일자 2005년08월02일
(87) 국제공개번호 WO 2006/018999
국제공개일자 2006년02월23일

(30) 우선권주장 JP-P-2004-00239346 2004년08월19일 일본(JP)

(71) 출원인 소니 가부시끼 가이샤
일본국 도쿄도 미나토쿠 코난 1-7-1
가부시키가이샤 소니 컴퓨터 엔터테인먼트
일본국 도쿄도 107-0062 미나토구 미나미-아오야마 2-6-21

(72) 발명자 하마다, 도시야
일본 141-0001 도쿄도 시나가와꾸 기따시나가와 6쪼메 7-35 소니가부시끼 가이샤 내
후지나미, 야스시
일본 141-0001 도쿄도 시나가와꾸 기따시나가와 6쪼메 7-35 소니가부시끼 가이샤 내
가꾸무, 다쯔야
일본 107-0062 도쿄도 미나또꾸 미나미 아오야마 2쪼메 6반 21고가부시키가이샤 소니 컴퓨터 엔터테인먼트 내
오시마, 다께노리
일본 107-0062 도쿄도 미나또꾸 미나미 아오야마 2쪼메 6반 21고가부시키가이샤 소니 컴퓨터 엔터테인먼트 내

(74) 대리인 장수길
구영창
이중희

전체 청구항 수 : 총 16 항

(54) 재생 장치, 재생 방법, 재생 프로그램, 기록 매체, 및데이터 구조

(57) 요약

컨텐츠를 재생할 때의 유저 오퍼레이션을 용이하게 제한 가능하게 한다. 컨텐츠의 재생을 제어하는 유저 오퍼레이션을 제한하는 제한 모드를 미리 설정하고, 제한 모드를 나타내는 값을 플레이 리스트마다 속성 정보로서 디스크에 기록한다. 플레이어는, 디스크의 재생시에, 제한 모드에 기초하는 테이블을 플레이 리스트마다 생성하고, 플레이어에 대한 유저 오퍼레이션에 의해 생성된 컨텐츠의 재생 제어를 행하는 제어 커맨드를, 테이블을 참조하여 커맨드 필터에 의해 필터링하여 제한한다. 제한 모드는, 예를 들면 플레이 리스트를 반드시 선두로부터 예를 들면 1배속이라고 하는 소정속도로 재생해야만 하는 모드와, 플레이 리스트를 재생 중에 점프하는 것이 금지되는 모드가 준비된다. 컨텐츠 제작측에서의 유저 오퍼레이션 제한에 대한 검증의 부하가 경감됨과 함께, 플레이어측에서도, 동작 검증의 부하가 경감된다.

대표도

도 39

특허청구의 범위

청구항 1.

원반 형상 기록 매체에 기록된 컨텐츠 데이터를 재생하는 재생 장치로서,

적어도 컨텐츠 데이터와, 그 컨텐츠 데이터에 대한 재생 경로를 지정하고, 속성 정보로서 상기 컨텐츠 데이터의 재생 제어 지시에 대한 제한 모드를 나타내는 값을 포함하는 재생 지시 정보와, 그 컨텐츠 데이터의 재생을 제어하는 재생 제어 프로그램이 기록된 기록 매체로부터 데이터를 읽어내는 읽어내기 수단과,

상기 재생 제어 프로그램에 따라 상기 컨텐츠 데이터를 재생하는 플레이어 수단과,

상기 컨텐츠 데이터의 상기 재생 제어 지시를 부여하기 위한 유저 오퍼레이션에 따라서 상기 플레이어 수단에 대한 제어 커맨드를 생성하는 제어 커맨드 생성 수단

을 갖고,

상기 플레이어 수단은, 상기 기록 매체로부터 상기 재생 지시 정보마다 상기 제한 모드를 나타내는 값을 읽어내고, 읽어내어진 그 제한 모드를 나타내는 값에 기초하여 상기 재생 지시 정보마다 테이블을 생성하고, 상기 제어 커맨드 생성 수단에 의해 생성된 상기 제어 커맨드의 실행을 허가할지의 여부를, 상기 테이블에 기초하여 상기 재생 지시 정보마다 제어하도록 한 것을 특징으로 하는 재생 장치.

청구항 2.

제1항에 있어서,

상기 제한 모드는, 상기 플레이어 수단에 대하여 상기 컨텐츠 데이터의 재생 정지를 지시하는 상기 제어 커맨드만을 허가하는 모드인 것을 특징으로 하는 재생 장치.

청구항 3.

제1항에 있어서,

상기 제한 모드는, 상기 제어 커맨드가, 상기 제한 모드를 나타내는 값이 포함되는 상기 속성 정보를 갖는 상기 재생 지시 정보에 대응한 재생 구간 내의 임의의 시각으로부터의 재생 개시를 지시하는 경우에, 그 재생 구간의 선두로부터 소정의 재생 속도로 재생을 행하도록 상기 플레이어 수단을 제어하는 모드인 것을 특징으로 하는 재생 장치.

청구항 4.

제1항에 있어서,

상기 제한 모드는,

상기 제한 모드가 나타내는 값이 포함되는 상기 속성 정보를 갖는 상기 재생 지시 정보에 대응하는 상기 컨텐츠 데이터를 재생 중에 상기 제어 커맨드가 있었던 경우에, 상기 플레이어 수단에 대하여 상기 컨텐츠 데이터의 재생 정지를 지시하는 상기 제어 커맨드만을 허가하고,

상기 제한 모드를 나타내는 값이 포함되는 상기 속성 정보를 갖는 상기 재생 지시 정보에 대응한 재생 구간 내의 임의의 시각으로부터의 재생 개시를 지시하는 상기 제어 커맨드가 있는 경우에, 그 재생 구간의 선두로부터 소정의 재생 속도로 재생을 행하도록 상기 플레이어 수단을 제어하는

모드인 것을 특징으로 하는 재생 장치.

청구항 5.

제1항에 있어서,

상기 제한 모드는, 상기 제한 모드가 나타내는 값이 포함되는 상기 속성 정보를 갖는 상기 재생 지시 정보에 대응하는 상기 컨텐츠 데이터를 재생 중에, 상기 제어 커맨드가 상기 재생 지시 정보에 대응하는 그 컨텐츠 데이터의 재생을 중지하여 그 재생 지시 정보에 대응한 재생 구간의 말미에 점프하는 것을 지시하는 경우, 그 제어 커맨드의 실행을 금지하는 모드인 것을 특징으로 하는 재생 장치.

청구항 6.

제1항에 있어서,

상기 플레이어 수단은, 상기 제어 커맨드가 있었을 때에, 현재 재생 중인 상기 컨텐츠 데이터에 대응하는 상기 재생 지시 정보의 상기 속성 정보에 포함되는 상기 제한 모드를 나타내는 값에 기초하는 상기 테이블에 의해 그 제어 커맨드의 실행이 허가되어 있는지의 여부를 판단하고,

허가되어 있다고 판단된 경우, 상기 제어 커맨드가 상기 재생 지시 정보에 대응한 제1 재생 구간 내에서 실행되는 것인지의 여부를 더 판단하고,

상기 제어 커맨드가 상기 제1 재생 구간 내에서 실행되는 것이 아니라고 판단된 경우에, 상기 제어 커맨드에 의해 새롭게 재생 개시되려고 하고 있는 제2 재생 구간에 대응하는 상기 재생 지시 정보의 상기 속성 정보에 포함되는 상기 제한 모드를 나타내는 값에 기초하는 상기 테이블에 의해 상기 제2 재생 구간의 선두로부터의 재생만이 허가되어 있는지의 여부를 더 판단하도록 한

것을 특징으로 하는 재생 장치.

청구항 7.

원반 형상 기록 매체에 기록된 컨텐츠 데이터를 재생하는 재생 방법으로서,

적어도 컨텐츠 데이터와, 그 컨텐츠 데이터에 대한 재생 경로를 지정하고, 속성 정보로서 상기 컨텐츠 데이터의 재생 제어 지시에 대한 제한 모드를 나타내는 값을 포함하는 재생 지시 정보와, 그 컨텐츠 데이터의 재생을 제어하는 재생 제어 프로그램이 기록된 기록 매체로부터 데이터를 읽어내는 읽어내기 스텝과,

상기 재생 제어 프로그램에 따라 상기 컨텐츠 데이터를 재생하는 컨텐츠 재생 스텝과,

상기 컨텐츠 데이터의 상기 재생 제어 지시를 부여하기 위한 유저 오퍼레이션에 따라서 상기 플레이어 수단에 대한 제어 커맨드를 생성하는 제어 커맨드 생성 스텝

을 갖고,

상기 컨텐츠 재생 스텝은, 상기 기록 매체로부터 상기 재생 지시 정보마다 상기 제한 모드를 나타내는 값을 읽어내고, 읽어내어진 그 제한 모드를 나타내는 값에 기초하여 상기 재생 지시 정보마다 테이블을 생성하고, 상기 제어 커맨드 생성의 스텝에서 생성된 상기 제어 커맨드의 실행을 허가할지의 여부를, 상기 테이블에 기초하여 상기 재생 지시 정보마다 제어하도록 한 것을 특징으로 하는 재생 방법.

청구항 8.

원반 형상 기록 매체에 기록된 컨텐츠 데이터를 재생하는 재생 방법을 컴퓨터 장치에 실행시키는 재생 프로그램으로서,

상기 재생 방법은,

적어도 컨텐츠 데이터와, 그 컨텐츠 데이터에 대한 재생 경로를 지정하고, 속성 정보로서 상기 컨텐츠 데이터의 재생 제어 지시에 대한 제한 모드를 나타내는 값을 포함하는 재생 지시 정보와, 그 컨텐츠 데이터의 재생을 제어하는 재생 제어 프로그램이 기록된 기록 매체로부터 데이터를 읽어내는 읽어내기 스텝과,

상기 재생 제어 프로그램에 따라 상기 컨텐츠 데이터를 재생하는 컨텐츠 재생 스텝과,

상기 컨텐츠 데이터의 상기 재생 제어 지시를 부여하기 위한 유저 오퍼레이션에 따라서 상기 플레이어 수단에 대한 제어 커맨드를 생성하는 제어 커맨드 생성 스텝

을 갖고,

상기 컨텐츠 재생 스텝은, 상기 기록 매체로부터 상기 재생 지시 정보마다 상기 제한 모드를 나타내는 값을 읽어내고, 읽어내어진 그 제한 모드를 나타내는 값에 기초하여 상기 재생 지시 정보마다 테이블을 생성하고, 상기 제어 커맨드 생성 스텝에서 생성된 상기 제어 커맨드의 실행을 허가할지의 여부를, 상기 테이블에 기초하여 상기 재생 지시 정보마다 제어하도록 한 것을 특징으로 하는 재생 프로그램.

청구항 9.

적어도

컨텐츠 데이터와,

상기 컨텐츠 데이터에 대한 재생 경로를 지정하고, 속성 정보로서 상기 컨텐츠 데이터의 재생 제어 지시에 대한 제한 모드를 나타내는 값을 포함하는 재생 지시 정보와,

상기 컨텐츠 데이터의 재생을 제어하는 재생 제어 프로그램이 기록되는 것을 특징으로 하는 기록 매체.

청구항 10.

제9항에 있어서,

재생 장치에 의해 읽어내어진 상기 재생 제어 프로그램에 따라 상기 컨텐츠 데이터가 재생되고, 상기 컨텐츠 데이터의 상기 재생 제어 지시를 부여하기 위하여 상기 재생 장치에 부여되는 유저 오퍼레이션에 따라서 상기 컨텐츠 데이터의 재생을 제어하는 제어 커맨드가 상기 재생 장치 위에서 생성되고,

상기 제한 모드를 나타내는 값에 기초하여 상기 재생 장치에 의해 상기 재생 지시 정보마다 테이블이 생성되고, 상기 제어 커맨드의 실행을 상기 재생 장치 위에서 허가할지의 여부가, 상기 테이블에 기초하여 상기 재생 지시 정보마다 제어되도록 한 것을 특징으로 하는 기록 매체.

청구항 11.

제9항에 있어서,

상기 제한 모드는, 상기 재생 장치에 대한 상기 컨텐츠 데이터의 재생 정지의 지시만을 허가하는 모드인 것을 특징으로 하는 기록 매체.

청구항 12.

제9항에 있어서,

상기 제한 모드는, 상기 제한 모드를 나타내는 값이 포함되는 상기 속성 정보를 갖는 상기 재생 지시 정보에 대응한 재생 구간 내의 임의의 시각으로부터의 재생 개시가 지시되는 경우에, 그 재생 구간의 선두로부터 소정의 재생 속도로 재생을 행하도록 상기 재생 장치에 의한 상기 컨텐츠 데이터의 재생을 제어하는 모드인 것을 특징으로 하는 기록 매체.

청구항 13.

제9항에 있어서,

상기 제한 모드는,

상기 재생 장치에 대하여, 재생 중인 상기 제한 모드가 나타내는 값이 포함되는 상기 속성 정보를 갖는 상기 재생 지시 정보에 대응하는 상기 컨텐츠 데이터의 재생 정지의 지시가 있었던 경우에, 그 지시만을 허가하고,

상기 제한 모드를 나타내는 값이 포함되는 상기 속성 정보를 갖는 상기 재생 지시 정보에 대응한 재생 구간 내의 임의의 시각으로부터의 재생 개시의 지시가 있었던 경우에, 그 재생 구간의 선두로부터 소정의 재생 속도로 재생을 행하도록 상기 재생 장치에 의한 상기 컨텐츠 데이터의 재생을 제어하는

모드인 것을 특징으로 하는 기록 매체.

청구항 14.

제9항에 있어서,

상기 제한 모드는, 상기 제한 모드가 나타내는 값이 포함되는 상기 속성 정보를 갖는 상기 재생 지시 정보에 대응하는 상기 컨텐츠 데이터를 상기 재생 장치가 재생 중에, 상기 제어 커맨드가 상기 재생 지시 정보에 대응하는 그 컨텐츠 데이터의 재생을 중지하여 그 재생 지시 정보에 대응한 재생 구간의 말미에 점프하는 지시가 있는 경우, 그 지시의 실행을 금지하는 모드인 것을 특징으로 하는 기록 매체.

청구항 15.

제10항에 있어서,

상기 재생 장치 상에서 상기 제어 커맨드가 생성되었을 때에, 현재 재생 중인 상기 컨텐츠 데이터에 대응하는 상기 재생 지시 정보의 상기 속성 정보에 포함되는 상기 제한 모드를 나타내는 값에 기초하는 상기 테이블에 의해 그 제어 커맨드의 실행이 허가되어 있는지의 여부가 판단되고,

허가되어 있다고 판단된 경우, 상기 제어 커맨드가 상기 재생 지시 정보에 대응한 제1 재생 구간 내에서 실행되는 것인지의 여부가 더 판단되고,

상기 제어 커맨드가 상기 제1 재생 구간 내에서 실행되는 것이 아니라고 판단된 경우에, 상기 제어 커맨드에 의해 새롭게 재생 개시되려고 하는 제2 재생 구간에 대응하는 상기 재생 지시 정보의 상기 속성 정보에 포함되는 상기 제한 모드를 나타내는 값에 기초하는 상기 테이블에 의해 상기 제2 재생 구간의 선두로부터의 재생만이 허가되어 있는지의 여부가 더 판단되도록 한

것을 특징으로 하는 기록 매체.

청구항 16.

적어도

컨텐츠 데이터와,

상기 컨텐츠 데이터에 대한 재생 경로를 지정하고, 속성 정보로서 상기 컨텐츠 데이터의 재생 제어 지시에 대한 제한 모드를 나타내는 값을 포함하는 재생 지시 정보와,

상기 컨텐츠 데이터의 재생을 제어하는 재생 제어 프로그램

을 갖는 것을 특징으로 하는 데이터 구조.

명세서

기술분야

본 발명은, 대용량의 기록 매체에 기록된 프로그램에 대한 유저에 의한 인터랙티브한 조작을 가능하게 함과 함께, 소정의 유저 조작을 제한하는 것이 용이한 재생 장치, 재생 방법, 재생 프로그램, 기록 매체, 및, 데이터 구조에 관한 것이다.

배경기술

랜덤 액세스 및 착탈이 가능한 기록 매체로서, DVD(Digital Versatile Disc)가 출현한지 오래지만, 최근에는, 이 DVD보다도 대용량의 디스크 형상 기록 매체나, DVD보다 휴대하기 편리한 소형의 디스크 형상 기록 매체의 개발이 진행되고 있다.

한편, 종래부터 존재하는, 재생 전용의 DVD 비디오 규격에서는, 메뉴 화면 위에 배치된 버튼 화상 등을 이용하여, 인터랙티브한 기능을 실현하고 있다. 예를 들면, DVD 비디오에 의해 동화상을 재생 중에, 리모트 컨트롤 커맨더 등을 이용하여 메뉴 화면을 호출하고, 메뉴 화면 위에 배치된 버튼 화상을 선택하여 재생 장면을 변경하는 등의 처리가 가능하였다.

이러한 DVD 비디오 규격에서는, 인터랙티브한 기능을 실현하기 위한 제어 프로그램은, DVD 비디오 규격으로 정의된 특유한 커맨드로 기술된다. 또한, 제어 프로그램은, 복수의 파일이나 데이터 파일 중의 복수의 개소, 나아가서는, AV 스트림 파일 중에도 분산되어 매립된다. 이들이 실행되는 조건이나 순서도, DVD 비디오 규격으로 정해져 있다.

그 때문에, 종래에는, 범용적인 컨텐츠 제작 시스템을 만드는 것이 어렵고, 미리 정해진 각본에 적용시켜서 스토리를 만드는, 소위 템플릿을 이용한 컨텐츠 제작이 행해지고 있었다. 템플릿에서는 대응할 수 없는, 복잡한 구성의 컨텐츠를 만드는 경우에는, 우선 컨텐츠 제작 시스템 그 자체를 커스텀 메이드로서 작성하고 있었다.

그런데, DVD 비디오 등의 재생에서, 통상적으로, 영화 본편의 재생 중 등에서는, 챕터 간의 점프 등과 같은, 유저에 의한 재생 제어 조작(이하, 유저 오퍼레이션이라고 함)은, 자유롭게 행할 수 있게 되어 있다. 한편, 스토리가 특정한 조건에서 분기되는 멀티 스토리나, 유저가 회답을 선택함으로써 시나리오가 진행되어 가는 퀴즈 게임 등과 같이, 재생 제어가 복잡한 컨텐츠에서는, 유저 오퍼레이션을 제한하고자 하는 장면이 발생한다.

예를 들면, 멀티 스토리이면, 과거의 재생 이력에 의해 다음의 스토리가 결정되는 것 같은 구성으로 되어 있는 경우가 있다. 이러한 경우에, 챕터 점프 등의 유저 조작에 의해, 설정된 스토리로부터 벗어나는 재생을 할 수 없도록, 유저 오퍼레이션을 제한할 필요가 있다.

다른 예로서, 퀴즈 게임의 경우에는, 대답을 선택할 수 있는 시간이 제한되어 있는 경우가 있다. 이 경우, 그 제한 시간 동안, 유저가 재생의 일시 정지를 할 수 없도록, 유저 오퍼레이션을 제한할 필요가 있다. 퀴즈 게임의 경우, 또한, 대답을 선택하지 않고, 정답이 표시된 씬으로 점프할 수 있는 것을 방지할 필요가 있다.

이와 같이, 유저와 컨텐츠 사이에서 쌍방향의 교환이 발생하는 경우, 컨텐츠 제작자의 의도대로의 재생을 실현하기 위하여, 유저 오퍼레이션을 제한할 필요가 있다.

또한, 본편의 재생 전에, 소정의 경고 화면 등을 유저에게 반드시 제시해야만 하는 경우가 있다. 이 경우에도, 경고 화면을 스킵하거나, 빨리 감기할 수 없도록, 유저 오퍼레이션을 제한할 필요가 있다.

종래의 DVD 비디오 규격에서는, 도 1에 일례가 도시된 바와 같이, 재생, 챕터 점프 등과 같은, 유저 오퍼레이션의 각각에 대하여, 그 조작을 허가할지의 여부를 나타내는 플래그를 설치하고, 그 플래그를 이용하여, 유저 오퍼레이션을 허가할지의 여부를 설정하고 있었다. 일본 특개 2003-203433호 공보에는, DVD 비디오 규격에서의 PGC(Program Chain)의 단위로 정보 재생 장치의 특정 동작을 허가 또는 금지하는 금지 플래그를, PGCI(Program Chain Infomation) 내 및 PCI (Presentation Control Information) 내에 각각 구축한 기술이 기재되어 있다.

그런데, 전술한 바와 같은, 유저에 의해 가능한 유저 오퍼레이션의 각각에 대하여 플래그를 설치하는 방법은, 컨텐츠 제작자측에서 상당히 사용하기 어렵다고 하는 문제점이 있었다.

예를 들면, 임의의 유저 오퍼레이션을 허가하고 싶지 않은 경우에는, 해당 조작에 관련하는 다른 유저 오퍼레이션도, 동시에 불허가로 하는 경우가 많다고 예상된다. 일례로서, 「순방향 빨리 감기」라고 하는 유저 오퍼레이션을 불허가로 하고자 하는 경우, 「역방향 빨리 감기」라고 하는 유저 오퍼레이션도 함께 불허가로 하고자 하는 것이 생각된다. 이러한 경우, 종래의 DVD 비디오 규격에서는, 「순방향 빨리 감기」를 나타내는 플래그와, 「역방향 빨리 감기」를 나타내는 플래그가 각각 독립되어 있기 때문에, 이 유저 오퍼레이션에 대하여 각각 플래그의 설정을 행하지 않으면 안 되었다.

이러한 종래의 설정 방법에서는, 유저 오퍼레이션에 대한 허가 및 불허가에 대하여, 다수의 조합이 발생되게 되어, 유저 오퍼레이션에 대한 제한에, 누락이나 모순이 발생하기 쉽다고 하는 문제점이 있었다.

예를 들면, 유저에게 반드시 제시하고자 하는 경고 화면 등을 표시하는 장면에서, 「순방향 빨리 감기」 및 「역방향 빨리 감기」를 지시하는 유저 오퍼레이션을 금지하도록 하는 한편, 「챕터 점프」를 지시하는 유저 오퍼레이션을 허가한 그대로 하고 있는 경우, 유저는, 「챕터 점프」를 지시하는 조작을 행함으로써, 경고 화면을 스킵시킬 수 있다.

한편, 컨텐츠 제작자가 의도하는, 불허가로 하는 유저 오퍼레이션의 조합은, 자주 이용되고 있는 몇 개의 조합으로 한정되어 있는 경우가 많다고 생각된다. 그 때문에, 유저 오퍼레이션마다 허가 및 불허가를 설정하는 방법은, 과잉의 자유도를 갖고, 그 결과, 설정을 잊어버리는 등에 의한 누락이나 모순이 발생하기 쉬운 것으로 생각된다.

또한, DVD 비디오 규격에서는, 유저 오퍼레이션을 제한하는 플래그가, AV 스트림에 가까운 하위의 계층으로부터 어플리케이션에 가까운 상위의 계층까지, 복수의 계층에 존재하고 있다. 따라서, 플래그를 설정할 때에는, 그들 계층 간의 조합도 고려할 필요가 있어, 난해한 것으로 되어 있다고 하는 문제점이 있었다.

또한, 플래그의 설정에 의해 컨텐츠 제작자가 의도한 바와 같은 동작이 실현 되어 있는지의 여부는, 컨텐츠 제작자측에서 검증할 필요가 있다. 이것은 컨텐츠 제작자측의 부담으로 되게 된다고 하는 문제점이 있었다.

또한, 유저 오퍼레이션마다 허가 및 불허가를 설정할 수 있기 때문에, 플레이어 제조자측에서는, 플레이어가 정확하게 동작하는지의 여부를, 그 모든 조합에 대하여 검증할 필요가 있으며, 플레이어 검증의 부담이 크다고 하는 문제점이 있었다.

<발명의 개시>

따라서, 본 발명의 목적은, 대용량의 기록 매체에 기록된 프로그램을 재생할 때의 유저 오퍼레이션을 용이하게 제한할 수 있는 재생 장치, 재생 방법, 재생 프로그램, 기록 매체, 및 데이터 구조를 제공하는 것에 있다.

본 발명은, 전술한 과제를 해결하기 위해, 원반 형상 기록 매체에 기록된 컨텐츠 데이터를 재생하는 재생 장치로서, 적어도 컨텐츠 데이터와, 컨텐츠 데이터에 대한 재생 경로를 지정하고, 속성 정보로서 컨텐츠 데이터의 재생 제어 지시에 대한 제한 모드를 나타내는 값을 포함하는 재생 지시 정보와, 컨텐츠 데이터의 재생을 제어하는 재생 제어 프로그램이 기록된 기록 매체로부터 데이터를 읽어내는 읽어내기 수단과, 재생 제어 프로그램에 따라 컨텐츠 데이터를 재생하는 플레이어 수단과, 컨텐츠 데이터의 재생 제어 지시를 부여하기 위한 유저 오퍼레이션에 따라서 플레이어 수단에 대한 제어 커맨드를 생성하는 제어 커맨드 생성 수단을 갖고, 플레이어 수단은, 기록 매체로부터 재생 지시 정보마다 제한 모드를 나타내는 값을 읽어내고, 읽어내어진 제한 모드를 나타내는 값에 기초하여 재생 지시 정보마다 테이블을 생성하고, 제어 커맨드 생성 수단에 의해 생성된 제어 커맨드의 실행을 허가할지의 여부를, 테이블에 기초하여 재생 지시 정보마다 제어하도록 한 것을 특징으로 하는 재생 장치이다.

또한, 본 발명은, 원반 형상 기록 매체에 기록된 컨텐츠 데이터를 재생하는 재생 방법으로서, 적어도 컨텐츠 데이터와, 컨텐츠 데이터에 대한 재생 경로를 지정하고, 속성 정보로서 컨텐츠 데이터의 재생 제어 지시에 대한 제한 모드를 나타내는 값을 포함하는 재생 지시 정보와, 컨텐츠 데이터의 재생을 제어하는 재생 제어 프로그램이 기록된 기록 매체로부터 데이터를 읽어내는 읽어내기 스텝과, 재생 제어 프로그램에 따라 컨텐츠 데이터를 재생하는 컨텐츠 재생 스텝과, 컨텐츠 데이터의 재생 제어 지시를 부여하기 위한 유저 오퍼레이션에 따라서 플레이어 수단에 대한 제어 커맨드를 생성하는 제어 커맨드 생성 스텝을 갖고, 컨텐츠 재생 스텝은, 기록 매체로부터 재생 지시 정보마다 제한 모드를 나타내는 값을 읽어내고, 읽어내어진 제한 모드를 나타내는 값에 기초하여 재생 지시 정보마다 테이블을 생성하고, 제어 커맨드 생성 스텝에서 생성된 제어 커맨드의 실행을 허가할지의 여부를, 테이블에 기초하여 재생 지시 정보마다 제어하도록 한 것을 특징으로 하는 재생 방법이다.

또한, 본 발명은, 원반 형상 기록 매체에 기록된 컨텐츠 데이터를 재생하는 재생 방법을 컴퓨터 장치에 실행시키는 재생 프로그램으로서, 재생 방법은, 적어도 컨텐츠 데이터와, 컨텐츠 데이터에 대한 재생 경로를 지정하고, 속성 정보로서 컨텐츠 데이터의 재생 제어 지시에 대한 제한 모드를 나타내는 값을 포함하는 재생 지시 정보와, 컨텐츠 데이터의 재생을 제어하는 재생 제어 프로그램이 기록된 기록 매체로부터 데이터를 읽어내는 읽어내기 스텝과, 재생 제어 프로그램에 따라 컨텐츠 데이터를 재생하는 컨텐츠 재생 스텝과, 컨텐츠 데이터의 재생 제어 지시를 부여하기 위한 유저 오퍼레이션에 따라서 플레이어 수단에 대한 제어 커맨드를 생성하는 제어 커맨드 생성 스텝을 갖고, 컨텐츠 재생 스텝은, 기록 매체로부터 재생 지시 정보마다 제한 모드를 나타내는 값을 읽어내고, 읽어내어진 제한 모드를 나타내는 값에 기초하여 재생 지시 정보마다 테이블을 생성하고, 제어 커맨드 생성 스텝에서 생성된 제어 커맨드의 실행을 허가할지의 여부를, 테이블에 기초하여 재생 지시 정보마다 제어하도록 한 것을 특징으로 하는 재생 프로그램이다.

또한, 본 발명은, 적어도 컨텐츠 데이터와, 컨텐츠 데이터에 대한 재생 경로를 지정하고, 속성 정보로서 컨텐츠 데이터의 재생 제어 지시에 대한 제한 모드를 나타내는 값을 포함하는 재생 지시 정보와, 컨텐츠 데이터의 재생을 제어하는 재생 제어 프로그램이 기록되는 것을 특징으로 하는 기록 매체이다.

또한, 본 발명은, 적어도 컨텐츠 데이터와, 컨텐츠 데이터에 대한 재생 경로를 지정하고, 속성 정보로서 컨텐츠 데이터의 재생 제어 지시에 대한 제한 모드를 나타내는 값을 포함하는 재생 지시 정보와, 컨텐츠 데이터의 재생을 제어하는 재생 제어 프로그램을 갖는 것을 특징으로 하는 데이터 구조이다.

전술한 바와 같이, 본 발명은, 적어도 컨텐츠 데이터와, 컨텐츠 데이터에 대한 재생 경로를 지정하고, 속성 정보로서 컨텐츠 데이터의 재생 제어 지시에 대한 제한 모드를 나타내는 값을 포함하는 재생 지시 정보와, 컨텐츠 데이터의 재생을 제어하는 재생 제어 프로그램이 기록된 기록 매체로부터 데이터를 읽어내고, 재생 장치는, 컨텐츠 데이터의 재생을 재생 제어 프로그램에 따라 행하고, 재생 지시 정보마다 읽어내어진 제한 모드를 나타내는 값에 기초하여 재생 지시 정보마다 테이블을 생성하고, 컨텐츠 데이터의 재생 제어 지시를 부여하기 위한 유저 오퍼레이션에 따라서 생성된 제어 커맨드의 실행을 허가할지의 여부를, 테이블에 기초하여 재생 지시 정보마다 제어하도록 하고 있기 때문에, 컨텐츠 제작 시에, 유저 오퍼레이션에 대한 제한을, 제한 모드에 기초하여 재생 지시 정보 단위로 용이하게 설정할 수 있음과 함께, 재생 장치측에서도, 유저 오퍼레이션에 대한 제한을 제한 모드에 기초하여 용이하게 검증할 수 있다.

또한, 본 발명은, 적어도 컨텐츠 데이터와, 컨텐츠 데이터에 대한 재생 경로를 지정하고, 속성 정보로서 컨텐츠 데이터의 재생 제어 지시에 대한 제한 모드를 나타내는 값을 포함하는 재생 지시 정보와, 컨텐츠 데이터의 재생을 제어하는 재생 제어 프로그램이 기록 매체에 기록되어 있기 때문에, 컨텐츠 제작 시에, 기록 매체를 재생하는 재생 장치에 대한 유저 오퍼레이션에 의해 이루어지는, 컨텐츠 데이터에 대한 재생 제어 지시를, 제한 모드에 기초하여 용이하게 설정할 수 있음과 함께, 재생 장치측에서도, 유저 오퍼레이션에 대한 제한을 제한 모드에 기초하여 용이하게 검증할 수 있다.

또한, 본 발명에 의한 데이터 구조는, 적어도 컨텐츠 데이터와, 컨텐츠 데이터에 대한 재생 경로를 지정하고, 속성 정보로서 컨텐츠 데이터의 재생 제어 지시에 대한 제한 모드를 나타내는 값을 포함하는 재생 지시 정보와, 컨텐츠 데이터의 재생을 제어하는 재생 제어 프로그램을 갖기 때문에, 컨텐츠 제작 시에, 이 데이터 구조를 갖는 데이터를 재생하는 재생 장치에 대한 유저 오퍼레이션에 의해 이루어지는, 컨텐츠 데이터에 대한 재생 제어 지시를, 제한 모드에 기초하여 용이하게 설정할 수 있음과 함께, 재생 장치측에서도, 유저 오퍼레이션에 대한 제한을 제한 모드에 기초하여 용이하게 검증할 수 있다.

본 발명은, 유저 오퍼레이션의 제한의 조합을 모드로서 정의하고, 빈번하게 사용되는 유저 오퍼레이션의 세트를 미리 플레이어측에서 준비하고, 컨텐츠 제작자측에서는, 제공된 유저 오퍼레이션의 조합의 모드를 선택함으로써, 유저 오퍼레이션에 대한 제어를 실현하도록 하고 있다.

그 때문에, 컨텐츠 제작자측은, 미리 플레이어 측에서 준비된 모드를 선택 하는 것만으로, 유저 오퍼레이션에 대하여 제한을 가할 수 있기 때문에, 보다 용이하게 유저 오퍼레이션의 제어가 가능하게 됨과 함께, 컨텐츠 제작자측에 의한 제작 및 검증 시의 부담이 경감된다고 하는 효과가 있다.

도면의 간단한 설명

도 1은 종래의 DVD 비디오 규격에 의한 유저 오퍼레이션 제어를 설명하기 위한 개략 선도.

도 2는 UMD 비디오 규격의 레이어 구성을 도시하는 개략 선도.

도 3은 본 발명의 일 실시 형태에 의한 일례의 플레이어 모델을 모식적으로 도시하는 개략 선도.

도 4는 무비 플레이어의 일례의 내부 구성을 도시하는 개략 선도.

도 5는 무비 플레이어의 3상태를 설명하기 위한 도면.

도 6은 본 발명의 일 실시 형태에 의한 무비 플레이어의 이벤트 모델을 모식적으로 도시하는 개략 선도.

도 7은 플레이 리스트의 재생 중에 발생하는 일례의 이벤트를 도시하는 개략 선도.

도 8은 무비 플레이어 오브젝트가 갖는 일례의 프로퍼티를 일람하여 도시하는 개략 선도.

도 9는 무비 플레이어 오브젝트가 갖는 일례의 메소드를 일람하여 도시하는 개략 선도.

도 10은 유저 입력에 의한 일례의 키 입력을 도시하는 개략 선도.

도 11은 유저 입력에 의한 일례의 키 입력을 도시하는 개략 선도.

도 12A, 도 12B 및 도 12C는 키 입력에 따른 일례의 제어 커맨드를 도시하는 개략 선도.

도 13은 키 입력에 대응하는 일례의 이벤트를 도시하는 개략 선도.

도 14는 일례의 이벤트 핸들러를 도시하는 개략 선도.

도 15는 일례의 이벤트 핸들러를 도시하는 개략 선도.

도 16은 유저 입력 이벤트를 계기로 하여, 준비된 프로그램이 실행되는 일례의 처리를 도시하는 플로우차트.

도 17은 UMD 비디오 플레이어에 디스크가 로드되고 나서 이젝트 될 때까지의 처리를 개략적으로 도시하는 플로우차트.

도 18은 스크립트 파일의 구성예를 도시하는 개략 선도.

도 19는 이벤트 핸들러 onAutoPlay()를 실행하는 일례의 수순을 도시하는 플로우차트.

도 20은 이벤트 핸들러 onContinuePlay()를 실행하는 일례의 수순을 도시하는 플로우차트.

도 21은 재생 종료 시의 일례의 처리를 도시하는 플로우차트.

도 22는 스크립트 프로그램의 예에 대하여 설명하기 위한 도면.

도 23은 일례의 스크립트 프로그램을 도시하는 개략 선도.

도 24는 UMD 비디오 규격에 적용되는 파일의 관리 구조를 설명하기 위한 개략 선도.

도 25는 파일 "PLAYLIST.DAT"의 전체 구조를 나타내는 일례의 신택스를 도시하는 개략 선도.

도 26은 블록 PlayItem()의 일례의 내부 구조를 도시하는 개략 선도.

도 27은 블록 PlayListMark()의 일례의 내부 구조를 도시하는 개략 선도.

도 28은 블록 Mark() 내의 필드 mark_type에 대하여 설명하기 위한 도면.

도 29는 클립 AV 스트림 파일 내에서의 마크 시각의 지정에 대하여 설명하기 위한 도면.

도 30은 클립 AV 스트림 파일 "XXXXX.CLP"의 전체 구조를 나타내는 일례의 신택스를 도시하는 개략 선도.

도 31은 블록 StreamInfo()의 엘리멘터리 스트림에 대한 관련지음을 설명하기 위한 도면.

도 32는 블록 StaticInfo()의 일례의 내부 구조를 도시하는 개략 선도.

도 33은 블록 DynamicInfo()의 일례의 내부 구조를 도시하는 개략 선도.

도 34는 블록 EP_map()의 일례의 내부 구조를 도시하는 개략 선도.

도 35는 본 발명을 적용 가능한 디스크 재생 장치의 일례의 구성을 개략적으로 도시하는 블록도.

도 36A 및 도 36B는 디스크 재생 장치에서의 동작을 보다 상세하게 설명하기 위한 기능 블록도.

도 37은 본 발명의 일 실시 형태에 의한 파일 "PLAYLIST.DAT"의 일례의 신택스를 도시하는 개략 선도.

도 38은 필드 UOP_mask_mode가 나타내는 값의 의미를 예시하는 개략 선도.

도 39는 무비 플레이어 내에서 유저 오퍼레이션 제한 기능을 실현하기 위한 일례의 기능 블록도.

도 40은 커맨드 필터 테이블의 일례의 작성 수순을 도시하는 플로우차트.

도 41은 유저 오퍼레이션 마스크 모드 「1」에 대응하는 일례의 커맨드 필터 테이블을 도시하는 개략 선도.

도 42는 유저 오퍼레이션 마스크 모드 「2」에 대응하는 일례의 커맨드 필터 테이블을 도시하는 개략 선도.

도 43은 커맨드 필터 테이블을 이용하여 유저 오퍼레이션을 제한하는 일례의 처리를 나타내는 플로우차트.

<부호의 설명>

101 디스크

112 CPU

113 메모리

115 입력 인터페이스

116 비디오 디코더

117 오디오 디코더

118 비디오 출력 인터페이스

119 오디오 출력 인터페이스

201 오퍼레이션 시스템

210 비디오 컨텐츠 재생부

211 스크립트 제어 모듈

212 플레이어 제어 모듈

214 디코드 제어 모듈

215 버퍼 제어 모듈

216 비디오 디코더 제어 모듈

217 오디오 디코더 제어 모듈

218 자막 디코더 제어 모듈

219 그래픽스 제어 모듈

241 비디오 출력 모듈

242 오디오 출력 모듈

250 불휘발성 메모리 제어 모듈

300 무비 플레이어

301 네이티브 실장 플랫폼

302 스크립트 레이어

310 유저 입력

311 제어 커맨드

312 이벤트

313 메소드

320 데이터베이스

321 플레이 백 모듈

322 디코더 엔진

323 프로퍼티

324 리쥼 인포메이션

S10 재생 중에 유저가 "next"키를 눌렀다

S11 uo_playNextChapter()가 발생

S12 다음의 챕터 마크의 위치를 플레이 리스트의 데이터베이스로부터 알 수 있다

S13 다음의 챕터 마크가 존재하는가?

S14 현재의 재생을 중지한다

S15 다음의 챕터 마크가 가리키는 위치로 점프하여 비디오 재생을 개시한다.

S16 마크 이벤트 발생

S17 마크 이벤트에 대응한 이벤트 핸들러를 실행 개시

S18 이벤트 발생 시에 통지된 정보로부터 챕터 번호를 알 수 있다

S19 챕터 선두인 것을 나타내는 메시지를 표시

S20 이벤트 핸들러 실행 종료

S30 디스크의 로드

S31 계속 재생 정보의 로드

S32 계속 재생 정보가 있는가?

S33 onContinuePlay

S34 onAutoPlay

S35 이벤트 접수 이벤트 핸들러 실행

S36 onExit 실행

S37 무비 플레이어 정지(계속 재생 정보 보유)

S38 재생 종료

S39 다시 한번 보겠는가?

S40 디스크 이젝트

S50 유저로부터 무비 플레이어에의 재생 지시(처음부터 재생)

S51 onAutoPlay 이벤트 핸들러가 존재하는가?

S52 네이티브 실장 플랫폼은 autoPlay 이벤트를 스크립트에 통지

S53 네이티브 실장 플랫폼은 Exit 이벤트를 스크립트에 통지

S54 onAutoPlay 이벤트 핸들러 실행

S60 유저로부터 무비 플레이어에의 재생 지시(계속 재생)

S61 리쥼 인포메이션이 존재하는가?

S62 처음부터 재생

S63 스크립트로 onContinuePlay 이벤트 핸들러가 준비되어 있는가?

S64 onContinuePlay 이벤트 핸들러 실행

S65 디폴트 onContinuePlay 이벤트 핸들러 실행

S70 유저로부터의 종료 지시

S71 유저 조작을 받은 네이티브 실장 플랫폼은 종료 처리를 개시한다

(1)새로운 이벤트 발생의 억지

(2)큐에 모인 이벤트 핸들러의 파기

(3)무비 플레이어에 대해서 uo_stop() 커맨드를 발행

S72 실행 중인 이벤트 핸들러가 종료한다

S73 네이티브 실장 플랫폼으로부터 스크립트 레이어에 Exit 이벤트를 통지한다

S74 onExit 이벤트 핸들러의 실행(후처리, setUserData 메소드의 실행 등)

S75 네이티브 실장 플랫폼에 의한 종료 처리(불휘발 메모리에의 계속 정보 보존, 시스템 메뉴에의 천이 등)

S80 디스크가 로드되었다

S81 플레이 리스트의 속성 정보로부터 UOP_mask_code를 읽어들인다

S82 플레이 리스트마다, 모드에 대응한 커맨드 필터 테이블을 생성한다.

S100 유저 조작이 발생했다

S101 네이티브 실장 플랫폼이 유저 조작을 접수한다

S102 네이티브 실장 플랫폼이 유저 조작을 제어 커맨드로 변환하여 무비 플레이어에 통지

S103 재생 중인 플레이 리스트의 커맨드 필터 테이블을 참조

S104 실행하고자 하는 제어 커맨드는 허가되어 있는가?

S105 제어 커맨드를 실행하지 않는다

S106 제어 커맨드는 현재 재생 중인 플레이 리스트 내에서 실행하는 것인가?

S107 제어 커맨드를 실행한다

S108 새롭게 재생 개시하려고 하고 있는 플레이 리스트의 커맨드 필터 테이블을 참조

S109 선두로부터의 재생만 허가되어 있는가?

S11O 플레이 리스트 선두로부터의 재생으로서 제어 커맨드 실행

S111 지정된 시각(또는 챕터)으로부터의 플레이 리스트 재생으로서 제어 커맨드를 실행

<발명을 실시하기 위한 최량의 형태>

이하, 본 발명의 일 실시 형태에 대하여, 하기의 순서에 따라 설명한다.

1.UMD 비디오 규격에 대하여

2.UMD 비디오 규격의 플레이어 모델에 대하여

3.무비 플레이어의 이벤트 모델에 대하여

4.무비 플레이어 오브젝트에 대해서

5.스크립트 프로그램의 예

6.파일의 관리 구조에 대하여

7.디스크 재생 장치에 대해서

8.유저 오퍼레이션의 제어에 대하여

1.UMD 비디오 규격에 대해서

우선, 이해를 용이하게 하기 위해, 이 일 실시 형태에 적용 가능한 시스템에 대하여 개략적으로 설명한다. 본 발명의 일 실시 형태에서는, ECMA 스크립트라고 하는 스크립트 언어를 이용하여 플레이어 모델을 기술하고 있다. ECMA 스크립트는, ECMA(European Computer Manufacturers Association)에 의해 정해진, Java Script(등록 상표)에 기초한 크로스 플랫폼용의 스크립트 언어이다. ECMA 스크립트는, HTML 문서와의 친화성이 높은 것과, 독자적인 오브젝트의 정의가 가능하기 때문에, 본 발명에 의한 플레이어 모델에 이용하기에 바람직하다.

또한, 이하에서는, 이 ECMA 스크립트를 바탕으로 한 스크립트 언어를 이용한, 본 발명의 일 실시 형태에 기초하는 규격을, UMD(Universal Media Disc:등록 상표) 비디오 규격이라고 한다. 또한, UMD 비디오 규격 중, 특히 스크립트에 관한 부분을 UMD 비디오 스크립트 규격이라고 한다.

UMD 비디오 규격에 대하여, 개략적으로 설명한다. 도 2는, UMD 비디오 규격의 레이어 구성을 나타낸다. UMD 비디오 규격에서는, 스크립트 레이어, 플레이 리스트 레이어 및 클립 레이어의 3층의 레이어 구조가 정의되고, 이 구조에 기초하여 스트림 관리가 이루어진다.

UMD 비디오 규격에서는, 디지털 부호화된 비디오, 오디오 및 자막을, MPEG2(Moving Pictures Experts Group2)의 엘리멘터리 스트림으로서 다중화한 MPEG2 스트림으로서 취급한다. 이 비디오, 오디오 및 자막의 엘리멘터리 스트림이 다중화된 MPEG2 스트림을, 클립 AV 스트림(Clip AV Stream)이라고 한다. 클립 AV 스트림은, 클립 AV 스트림 파일에 저장된다. 클립 AV 스트림 파일의 기록 시에, 그 클립 AV 스트림 파일에 1대1로 대응하고, 클립 인포메이션 파일(Clip Information File)이 동시에 작성된다. 이들 클립 인포메이션 파일과, 대응하는 클립 AV 스트림 파일로 이루어지는 조를, 클립(Clip)이라고 한다.

클립은, 디스크에의 기록의 단위라고도 하며, 재생 시에 어떤 순서로 클립을 재생할지는, 클립의 상위의 레이어인 플레이 리스트 레이어에서 관리한다. 플레이 리스트 레이어는, 클립의 재생 경로를 지정하는 레이어이며, 1 또는 복수의 플레이 리스트(PlayList)를 포함한다. 플레이 리스트는, 플레이 아이템의 (PlayItem)의 집합으로 이루어진다. 플레이 아이템에는, 클립의 재생 범위를 나타낸 1조의 인(In) 점 및 아웃(Out) 점이 포함되어 있어, 플레이 아이템을 이어서 나란히 함으로써, 임의의 순서로 클립을 재생할 수 있게 된다. 플레이 아이템은, 클립을 중복하여 지정할 수 있다. 클립 AV 스트림 파일의 인점 및 아웃점은, 타임 스탬프(클립 내 시각)로 지정되고, 타임 스탬프는, 클립 인포메이션 파일의 정보에 의해 클립 AV 스트림 파일 상의 바이트 위치로 변환된다.

플레이 리스트는, 클립의 전부 또는 일부를 가리키는 플레이 아이템을 순서에 따라 재생해 가는 구조만을 갖고 있어, 플레이 리스트만을 이용하여 재생 순서의 분기나, 유저와의 쌍 방향성을 실현하는 것은 불가능하다. 본 발명의 일 실시 형태에서는, 복수의 플레이 리스트가 1개의 파일 "PLAYLIST.DAT"에 정리되어 있다.

스크립트 레이어는, 언어 사양의 ECMA 스크립트를 확장한, UMD 비디오 스크립트에 의해 구축되는 레이어이다. UMD 비디오 스크립트는, ECMA 스크립트를 기본으로 하여, UMD 비디오에 특유한 기능을 실현하기 위한 확장을 가한 스크립트이다.

스크립트 레이어는, 플레이 리스트 레이어의 상위의 레이어로서, 플레이 리스트의 재생 지시나, 플레이어 설정을 행하는 커맨드열로 구성된다. 스크립트 레이어의 커맨드에 의해, 복수의 언어용으로 준비된 스트림 중 어느 것을 선택할지, 임의의 조건에 따라서 선택되는 플레이 리스트에 재생의 흐름이 변화될지, 와 같은 조건 분기를 수반하는 플레이 리스트 재생을 실현할 수 있다. 이러한 조건 분기를 수반하는 플레이 리스트 재생이 이용되는 어플리케이션의 예로서는, 멀티스토리를 들 수 있다. 이 스크립트 레이어에 의해, 유저와의 쌍 방향성 기능(인터랙티브 기능)이 도입되게 된다.

또한, 본 발명의 일 실시 형태에서는, 스크립트 레이어는, 1개의 파일 "SCRIPT.DAT"으로 구성되어, 리소스로서 관리된다. 파일 "SCRIPT.DAT"은, 실제의 ECMA 스크립트에 기초하여 기술되는 스크립트 데이터, 버튼 조작 시의 효과음 등을 출력하기 위한 사운드 데이터, 메뉴 화면의 배경 화상 등에 이용하는 화상 데이터로 이루어지는 스크린 디자인, 및, 버튼 화상 등의 GUI 부품을 표시시키기 위한 화상 데이터(비트 맵 데이터)가 포함된다.

2.UMD 비디오 규격의 플레이어 모델에 대하여

다음으로, UMD 비디오 규격에 따른 데이터를 재생하는 재생 장치(플레이어)의 모델, 즉, 플레이어 모델에 대하여 설명한다. 플레이어는, 우선, 디스크로부터 스크립트 프로그램, 플레이 리스트 및 클립 인포메이션 파일을 읽어내고, 다음으로, 이들에 의해 정해져 있는 재생 순서에 따라서, 클립 AV 스트림 파일을 읽어내고, 비디오, 오디오 및 자막 등을 재생한다.

스크립트 프로그램의 언어사양에서는, 플레이 리스트를 재생하는 기능 블록을, 스크립트 프로그램 내의 오브젝트로서 실장한다. 이 플레이 리스트 재생을 행하는 오브젝트를, UMD 비디오 규격에서는, 무비 플레이어(Movie Player) 오브젝트라고 한다. 플레이 리스트의 재생 지시나, 플레이어 설정을 행하는 커맨드는, 이 무비 플레이어 오브젝트가 갖는 메소드로 된다. 무비 플레이어 오브젝트는, 스크립트 레이어로부터의 메소드에 의해 제어된다. 이 때, 무비 플레이어 오브젝트로부터 스크립트 레이어에 대하여, 상태의 변화나 재생 위치 등을 통지하는 기능이 필요하게 된다. 이것은, 무비 플레이어 오브젝트가 스크립트 프로그램에 대하여 이벤트를 발하는 것에 대응하고, 그 이벤트에 대응한 처리는, 이벤트 핸들러로서 기술된다.

이와 같이, 무비 플레이어 오브젝트로부터 스크립트 프로그램에의 정보 전달은, 이벤트에 의해 행하여, 스크립트 프로그램으로부터 무비 플레이어 오브젝트에 대한 제어를 메소드에 의해 행하는 모델을 구축함으로써, 클립 AV 스트림의 재생을 스크립트 프로그램으로 제어할 수 있게 된다.

도 3은, 전술한, 본 발명의 일 실시 형태에 의한 일례의 플레이어 모델을 모식적으로 나타낸다. 무비 플레이어(300)는, UMD 비디오 규격에서 비디오, 오디오 및 자막의 재생을 담당하는 모듈이다. 전술한 무비 플레이어 오브젝트는, 무비 오브젝트를 스크립트 프로그램으로부터 조작하기 위해 스크립트 프로그램 내의 오브젝트로 한 것이다. 환언하면, 무비 플레이어 오브젝트는, 무비 플레이어의 기능을 실현하기 위한 스크립트 프로그램 그 자체이다.

또한, 무비 플레이어(300)와 무비 플레이어 오브젝트는, 실질적으로 동일한 대상을 나타낸다고 생각되므로, 이하, 이들을 동일한 부호를 붙여서 설명한다.

도 3에서, 무비 플레이어(300)는, 유저 입력(310) 등에 의해 야기되는 하위 레이어(도 3의 예에서는 네이티브 실장 플랫폼(301))나, 상위 레이어인 스크립트 레이어(302)로부터의 메소드에 따라, 플레이 리스트 및 클립 인포메이션의 데이터베이스에 기초하여, 클립 AV 스트림 파일을 읽어내고, 읽어내어진 클립 AV 스트림의 디코드 및 표시를 행한다.

무비 플레이어 오브젝트(300)의 내부는, UMD 비디오를 재생하는 UMD 비디오 플레이어의 실장에 의존하는 것으로서, 스크립트 레이어(302)로부터는, 블랙박스화된 오브젝트로서, 메소드나 프로퍼티와 같은 API(Application Programming Interface)가 제공된다. 여기에서, UMD 비디오 플레이어는, 무비 플레이어를 실장한 실제의 기기를 가리킨다. 모든 UMD 비디오 플레이어는, UMD 비디오 규격의 제약을 지켜서 무비 플레이어를 실장하고 있고, 재생 호환을 갖는다.

도 3에 도시된 바와 같이, 무비 플레이어 오브젝트(300)는, 네이티브 실장 플랫폼(301)으로부터의 제어 커맨드(311)를 접수하는 패스, 스크립트 레이어(302)에 대하여 이벤트(312)를 통지하는 패스, 스크립트 레이어(302)로부터의 메소드(313)를 접수하는 패스의, 3개의 입출력 패스를 갖는다.

제어 커맨드(311)는, 네이티브 실장의 플랫폼(301)으로부터의, 무비 플레이어 오브젝트(300)의 동작을 제어하는 커맨드이다. 네이티브 실장 플랫폼(301)은, 예를 들면, 실제의 기기로서의 UMD 비디오 플레이어에서의, 기기에 고유의 부분과 무비 플레이어(300)의 인터페이스이다. 이벤트(312)는, 무비 플레이어(300)로부터 스크립트 레이어(302)에 대한 스크립트 이벤트이다. 메소드(313)는, 스크립트 레이어(302)의 스크립트 프로그램으로부터 무비 플레이어(300)에 지시되는 메소드이다.

무비 플레이어 오브젝트(300)는, 내부에, UMD 비디오 규격의 플레이 리스트 및 클립 인포메이션의 데이터베이스(320)를 갖는다. 무비 플레이어 오브젝트(300)는, 이 데이터베이스(320)를 이용하여, 유저 입력(310)에 대한 무효화(mask)나, 시각으로 지정된 재생 위치를 클립 AV 스트림 내의 바이트 위치로 변환하는 처리 등을 행한다.

무비 플레이어 오브젝트(300) 내의 플레이 백 모듈(321)은, 비디오, 오디오 및 자막이 다중된 MPEG2 PS(Program Stream)인 클립 AV 스트림의 디코드를 행한다. 플레이 백 모듈(321)은, 플레이, 스톱 및 포즈의 3상태를 갖고, 제어 명령이나 메소드에 의해, 이 3상태 사이를 천이한다(도 4 참조).

스크립트 레이어(302)는, UMD 비디오 스크립트 규격에 기초하는 스크립트 프로그램을 실행하고, 무비 플레이어 오브젝트(300)의 제어나, 화면 표시를 행하는 레이어이다. 이 스크립트 레이어(302)는, 컨텐츠 제작자측이 의도한 시나리오를 실

현하는 역할을 완수한다. 스크립트 레이어(302)는, 무비 플레이어 오브젝트(300)에 대하여 메소드(313)를 발행하고, 무비 플레이어 오브젝트(300)로부터는, 이벤트(312)를 수취한다. 스크립트 레이어(302)는, 네이티브 실장 플랫폼(301)과의 사이에서, 유저 입력(310)에 따른 키 이벤트(314)나, 화면 묘화 등을 네이티브 실장 플랫폼(301)에 대하여 지시하는 메소드(315) 등의 교환을 행한다.

예를 들면, 메뉴 화면 위에 배치되는 버튼은, 스크립트 레이어(302)의 스크립트 프로그램으로부터 네이티브 실장 플랫폼(301)에 건네지는 메소드(315)에 기초하여, 네이티브 실장 플랫폼(301)에 의해 묘화된다. 유저가 그 버튼에 대하여 선택이나 결정 등의 조작을 행하였을 때에는, 유저 입력(310)에 따른 키 이벤트(314)가 네이티브 실장 플랫폼(301)으로부터 스크립트 레이어(302)에 통지되고, 스크립트 레이어(302) 내의 스크립트 프로그램은, 키 이벤트(314)에 기초하여 키 입력(310)에 따른 처리를 행한다.

이와 같이, 비디오, 오디오 및 자막의 디코드나 표시 제어는 무비 플레이어(300)가 담당하고, 버튼 등의 GUI(Graphical User Interface)를 구성하기 위한 부품 화상(이하, GUI 부품이라고 함)의 배치나 표시, 및, GUI 부품에 대하여 선택이나 결정 등의 조작이 이루어졌을 때의 처리는, 스크립트 레이어(302)가 행하는 등과 같이, 역할 분담이 이루어져 있다.

네이티브 실장 플랫폼(301)은, 무비 플레이어 오브젝트(300)나 스크립트 프로그램이 동작하기 위한 기반으로 되는 플랫폼으로서, 예를 들면, 실제의 UMD 비디오 플레이어가 하드웨어인 경우, 하드웨어와 플레이어 모델 사이의 처리를 중개하는 역할을 완수하도록, 하드웨어에 고유하게 실장된다.

예를 들면, 네이티브 실장 플랫폼(301)은, 유저로부터의 유저 입력(310)을 접수하고, 접수한 유저 입력(310)이 무비 플레이어(300)에 대한 명령인지, 스크립트 레이어(302)에서 묘화 및 표시하고 있는 버튼에 대한 명령인지를 판정한다. 네이티브 실장 플랫폼(301)은, 유저 입력(310)이 무비 플레이어(300)에 대한 명령이라고 판정되면, 유저 입력(310)을 무비 플레이어(300)에 대한 내부 제어 명령인 제어 커맨드(311)로 변환하고, 무비 플레이어(300)에 대하여 제어 명령을 발한다.

한편, 네이티브 실장 플랫폼(301)은, 유저 입력(310)이 스크립트 레이어(302)에서 묘화 및 표시하고 있는 GUT 부품에 대한 명령이라고 판정되면, 유저 입력(310)에 따른 키 이벤트(314)를 스크립트 레이어(302)에 통지한다. 그리고, 이 키 이벤트(314)에 따라서 스크립트 레이어(302)로부터 지시된 메소드(315)에 기초하여, 예를 들면 화면 상에 버튼 화상을 표시시킬 수 있다. 즉, 네이티브 실장 플랫폼(301)과 스크립트 레이어(302)는, 무비 플레이어(300)를 통하지 않고서, 직접적으로 이벤트 및 메소드의 교환을 행할 수 있다.

다음으로, 무비 플레이어(300)에 대하여 보다 상세하게 설명한다. 도 4는, 무비 플레이어(300)의 일례의 내부 구성을 나타낸다. 전술한 바와 같이, 무비 플레이어(300)는, 데이터베이스(320) 및 플레이 백 모듈(321)로 구성된다. 데이터베이스(320)는, 디스크로부터 판독한 플레이 리스트의 정보와, 클립의 정보 즉 클립 인포메이션을 저장하는 영역이다.

플레이 백 모듈(321)은, 디코더 엔진(322)과, 플레이 백 모듈(321)의 상태를 나타내는 값인 프로퍼티(323)로 이루어진다. 프로퍼티(323)는, 예를 들면 언어 코드와 같이, 무비 플레이어(300)의 초기 설정으로 값이 결정되는 프로퍼티(323A)(리드 온리 파라미터)와, 플레이 백 모듈(321)의 상태에 의해 값이 변화하는 프로퍼티(323B)(플레이어 스테이터스)의 2종류가 있다.

초기 설정으로 값이 결정되는 프로퍼티(323A)는, 네이티브한 시스템, 예를 들면 실제의 기기에 의해 값이 세트되어, 플레이 리스트나 클립 인포메이션, 스크립트 프로그램으로부터 값이 변경되지 않는다. 프로퍼티(323A)는, 스크립트 프로그램으로부터 값을 읽어내는 것만이 가능하게 된다. 한편, 플레이 백 모듈(321)의 상태를 나타내는 프로퍼티(323B)는, 스크립트 프로그램으로부터 값을 읽어낼 수 있음과 함께, 일부의 스크립트 프로그램으로부터 값을 기입하는 것이 가능하게 된다.

또한, 이 동작 모델에서는, 플레이 리스트 및 클립 인포메이션은, 클립 AV 스트림의 재생 전에 디스크로부터 프리 로드되어 있는 것을 상정하고 있다. 이에 한하지 않고, 다른 실장이어도, 무비 플레이어 모델에서 정한 동작을 실현할 수 있으면 된다.

무비 플레이어 오브젝트(300)는, 스크립트 레이어(302) 또는 네이티브 실장 플랫폼(301)으로부터의 지시에 따라, 지정된 플레이 리스트를 재생한다. 예를 들면, 무비 플레이어(300)는, 데이터베이스(320)를 참조하여, 지정된 플레이 리스트에 대응하는 클립 AV 스트림의 재생 위치를 파일 중의 바이트 위치에서 얻는다. 플레이 백 모듈(321)에서, 디코더 엔진(322)은, 이 재생 위치 정보에 기초하여, 클립 AV 스트림의 디코드를 제어한다.

무비 플레이어(300)는, 도 5에 도시된 바와 같이, 플레이 리스트의 재생 상황에 따라서 플레이(play), 스톱(stop) 및 포즈(pause)의 3상태를 갖는다. 플레이 상태는, 플레이 리스트의 재생을 행하고 있고, 시간이 경과되어 있는 상태를 가리킨다. 통상 재생 외, 2배속, 1/2배속 등의 변속 재생이나, 순방향 빨리 감기 및 역방향 빨리 감기도 플레이 상태에 포함된다. 포즈 상태는, 플레이 리스트의 재생을 행하고 있는 상태에서, 시간 축이 정지하고 있는 상태이다. 재생을 프레임 단위로 진행시키거나 되돌리거나 하는 소위 코마 전송 재생은, 포즈 상태와 플레이 상태를 반복하고 있는 상태이다. 스톱 상태는, 플레이 리스트를 재생하지 않고 있는 상태이다.

예를 들면, 무비 플레이어(300)의 상태는, 무비 플레이어(300) 내의 디코더 엔진(322)에서의 플레이, 포즈 및 스톱의 상태 천이에 수반하는 것으로, 디코더 엔진(322)의 상태 천이에 따라서 프로퍼티(323B)의 값이 갱신된다.

리쥼 인포메이션(324)은, 스톱 상태 직전의 상태를 기억한다. 예를 들면, 무비 플레이어(300)가 임의의 플레이 리스트를 디코드하여 플레이 상태로 되어 있을 때에, 상태가 스톱 상태로 천이하면, 스톱 상태의 직전의 상태를 기억한다. 또한, 리쥼 인포메이션(324)은, 하드웨어로서의 플레이어가 갖는 불휘발성 메모리에, 디스크의 타이틀마다 식별 가능하도록 복수를 기억시킬 수 있다. 예를 들면, 디스크는, 디스크의 타이틀마다 유니크한 식별 정보(타이틀 ID라고 함)를 갖고, 리쥼 인포메이션(324)을 이 식별 정보와 관련지어 기억한다. 이와 같이 함으로써, 리쥼 인포메이션(324)의 정보에 기초하여, 식별 정보가 대응하는 타이틀의 디스크가 다음으로 스톱 상태로부터 플레이 상태로 천이했을 때에, 그 디스크의 재생을, 이전 스톱 상태로 된 직전의 위치로부터 개시시킬 수 있다.

3.무비 플레이어의 이벤트 모델에 대하여

무비 플레이어(300)의 이벤트 모델에 대하여 설명한다. 무비 플레이어(300)는, 플레이 리스트를 재생하는 플레이 상태에서, 다양한 이벤트를 발생한다. 이 이벤트는, 이벤트 핸들러라고 하는, 스크립트로 기술된 처리 프로그램의 실행을 야기한다. 이벤트 핸들러는, 이벤트 발생에 의해 호출되는 메소드이다. 이 이벤트 발생에 의해 처리 프로그램의 실행을 개시하는 프로그램 실행 모델을, 이벤트 드리븐 모델이라고 한다. 이벤트 드리븐 모델에서는, 부정기한 이벤트가 발생하고, 이벤트 발생을 계기로 준비해 둔 프로그램이 실행된다. 이 일 실시 형태에서는, 스크립트 프로그램은, 이벤트 핸들러군에 의해 무비 플레이어 오브젝트(300)의 동작을 제어한다.

도 6은 본 발명의 일 실시 형태에 의한 무비 플레이어(300)의 이벤트 모델을 모식적으로 나타낸다. 도 6에서, 이벤트 핸들러 onEventA(), onEventB() 및, onEventC()는, 인터페이스로서, 각각의 이벤트 핸들러의 내용은, 스크립트로 기술되어 있다. 이벤트 핸들러의 내용은, 예를 들면 컨텐츠 제작자측에서 작성되어 실장된다. UMD 비디오 스크립트 규격에서는, 무비 플레이어 오브젝트(300)로부터 스크립트 프로그램에 통지되는 이벤트마다, 이벤트 핸들러가 준비된다. 도 6의 예에서는, 이벤트 A가 발생했을 때에 실행되는 처리 프로그램은, 이벤트 핸들러 onEventA()에 정해져 있다. 이벤트 B 및 이벤트 C에 대해서도 마찬가지로, 이벤트 B의 발생 시에는 대응하는 이벤트 핸들러 onEventB()가 실행되고, 이벤트 C의 발생 시에는 대응하는 이벤트 핸들러 onEventC()가 실행된다.

이벤트 발생에 따라서 호출되는 이벤트 핸들러는, 시스템측에서 선택되기 때문에, 컨텐츠 제작자측에서는, 어떤 이벤트가 발생했는지를 판단하는 처리를, 스크립트 프로그램 내에 기술해 둘 필요가 없다.

도 7은, 플레이 리스트의 재생 중에 발생하는 일례의 이벤트를 나타낸다. 플레이 리스트 PlayList의 선두에는, 챕터 마크 ChapterMark가 설정되어 있으므로, 플레이 리스트의 선두로부터의 재생 개시 시에는, 우선, 챕터 마크에 대응한 이벤트 Chapter가 발생한다. 또한, 챕터가 변할 때마다 이벤트 Chapter가 스크립트 레이어(302)에 통지되어, 대응하는 이벤트 핸들러 onChapter가 실행된다. 또한, 이벤트 마크 EventMark가 설정되어 있는 시각에 재생이 도달하면, 대응하는 마크 이벤트가 발생한다. 그리고, 플레이 리스트의 마지막까지 재생이 도달하면, 플레이 리스트의 마지막에서 재생이 일시 정지하고, 이벤트 PlayListEnd가 무비 플레이어(300)로부터 스크립트 레이어(302)에 통지된다. 스크립트 레이어(302)측에서는, 대응하는 이벤트 핸들러 onPlayListEnd() 내에서, 다른 플레이 리스트의 재생 개시가 지시된다. 이렇게 하여, 컨텐츠 제작자가 의도한 순서로, 일련의 플레이 리스트 재생이 계속되어 간다.

이와 같이, 플레이어의 동작 중에는 여러 가지 이벤트가 발생하는 것으로 하고, 이벤트 발생을 상위 프로그램에 전달함으로써, 상위 프로그램은 플레이어의 상태를 파악할 수 있게 된다. 상위 프로그램 쪽에서는, 각 이벤트 발생 통지 시에 실행되는 프로그램(이벤트 핸들러)을 준비해 둠으로써, 각종 이벤트 발생에 대처한다. 이벤트 및 이벤트 핸들러의 상세에 대해서는, 후술한다.

이벤트 핸들러가 컨텐츠 제작자에 의해 기술되어 있지 않은 경우에는, 규격으로 규정되어 있는 플레이어 편입의 동작(디폴트의 이벤트 핸들러)을 실행하거나, 혹은, 그 이벤트가 무시되어 아무것도 실행되지 않는다. 아무것도 처리를 행할 필요가 없을 때에는, 이벤트에 대응한 이벤트 핸들러를 기술하지 않도록 함으로써, 적극적으로 이벤트를 무시할 수 있다.

이벤트 모델로서는, 전술 외에, 임의의 이벤트에 대응하는 리스너를 오브젝트가 플레이어 오브젝트에 등록하고, 플레이어 오브젝트 내에서 발생한 이벤트가 등록된 이벤트이면, 플레이어 오브젝트로부터 그 이벤트를 등록한 오브젝트에 이벤트를 송신하고, 그 오브젝트에서 대응하는 메소드를 실행하도록 한 이벤트 리스너의 모델이나, 임의의 이벤트가 발생하여도 1개의 메소드를 호출하도록 한 단일 메소드의 모델 등이 생각된다.

이 일 실시 형태에 의한 이벤트 모델은, 이벤트 등록, 이벤트 등록 삭제와 같은 처리가 필요한 이벤트 리스너의 모델보다도 간단하다. 또한, 단일 메소드의 모델은, 어떤 이벤트가 발생했는지를 알고, 이벤트마다 준비되어 있는 처리 루틴을 절환한다고 하는 전 처리를, 그 메소드 중에 기술해 둘 필요가 있다. 메소드는 컨텐츠 제작자측이 실장하는 것이기 때문에, 모델로서는 간단해도, 컨텐츠 제작자측의 부담이 커진다. 또한, 큰 하나의 처리 프로그램(메소드)이 이벤트의 발생마다 불려지게 되고, 메모리의 영역을 많이 점유하여, 실행 속도도 늦어진다고 생각된다. 본 발명의 일 실시 형태에 의한, 이벤트마다 처리 프로그램(이벤트 핸들러)을 준비하는 모델에서는, 이러한 점에 대하여 유리하다고 할 수 있다.

4.무비 플레이어 오브젝트에 대해서

다음으로, 무비 플레이어 오브젝트(300)의 외부적인 사양에 대하여 설명한다. 일반적으로, ECMA 스크립트 언어사양에 따른 언어에 의해 정의된 오브젝트는, 프로퍼티와 메소드를 갖는다. 이 일 실시 형태에 의한 무비 플레이어 오브젝트(300)도.도 3 및 도 4를 이용하여 이미 설명한 바와 같이, 마찬가지로 하여 프로퍼티와 메소드를 갖는다. 프로퍼티는, 외부의 오브젝트로부터, 대상으로 되는 오브젝트명과 프로퍼티명을 지정함으로써, 직접적으로 판독 및 기입하는 것이 가능하다. 이에 한하지 않고, 프로퍼티값의 설정을 행하는 메소드 setXXX()(「XXX」는, 대상의 프로퍼티명)나, 프로퍼티값의 읽어내기를 행하는 메소드 getXXX()를 정의함으로써, 다른 오브젝트의 프로퍼티의 판독 및 기입을, 메소드로 행하는 것이 가능하게 된다.

도 8은, 무비 플레이어 오브젝트(300)가 갖는 일례의 프로퍼티를 일람하여 나타낸다. 이것은, 도 4에서의 프로퍼티(323)에 대응한다. 도 4에서의 리드 온리 파라미터(323A)에 속하는 프로퍼티는, 이하와 같다. 프로퍼티 scriptVersion은, UMD 비디오 스크립트의 버전을 나타낸다. 프로퍼티 languageCode는, UMD 비디오 플레이어에 설정된, 메뉴 표시 언어의 언어 코드를 나타낸다. 프로퍼티 audioLanguageCode는, UMD 비디오 플레이어에 설정된, 오디오 언어의 언어 코드를 나타낸다. 프로퍼티 subtitleLanguageCode는, UMD 비디오 플레이어에 설정된, 자막(서브타이틀) 언어의 언어 코드를 나타낸다.

디스크가 장전되었을 때에는, 이 리드 온리 파라미터(323A)에 설정된 프로퍼티 languageCode에 표시되는 언어 코드에 기초하여, 디스크로부터 읽어내는 스크립트 파일이 결정된다. 장전된 디스크에, 해당 언어에 대응하는 스크립트 파일이 없는 경우에는, 디폴트의 스크립트 파일이 읽어내어진다. 예를 들면, 복수의 스크립트 파일 중, 디스크 상에서 가장 선두측에 배치되는 파일이 디폴트의 스크립트 파일로서 읽어내어진다.

도 4에서의 플레이어 스테이터스(323B)에 속하는 프로퍼티는, 이하와 같다. 프로퍼티 playListNumber는, 현재 재생 중인 플레이 리스트의 번호를 나타낸다. 프로퍼티 chapterNumber는, 현재 재생 중인 챕터의 번호를 나타낸다. 프로퍼티 videoNumber는, 현재 재생 중인 비디오 스트림의 번호를 나타낸다. 프로퍼티 audioNumber는, 현재 재생 중인 오디오 스트림의 번호를 나타낸다. 프로퍼티 subtitleNumber는, 현재 재생 중인 자막 스트림의 번호를 나타낸다. 프로퍼티 playListTime은, 플레이 리스트 선두를 0으로 했을 때의 시각을 나타낸다. 프로퍼티 audioFlag는, 오디오 재생의 ON/OFF 및 듀얼 모노 LR의 지정을 나타낸다. 프로퍼티 subtitleFlag는, 자막 표시의 ON/OFF를 나타낸다.

또한, 듀얼 모노는, 스테레오 오디오의 좌우(L, R) 채널을, 상호 독립한 모노럴 오디오 채널로서 이용하는 모드이다.

이 플레이어 스테이터스(323B)에 속하는 각 프로퍼티는, 무비 플레이어(300)가 재생 또는 일시 정지 상태일 때에, 이들 정보가 존재한다. 정지 상태로 천이한 경우, 그 시점에서 플레이어 스테이터스(323B)에 속하는 각 프로퍼티는, 리쥼 인포메이션(324)으로서 백업된다. 이 때, 플레이어 스테이터스(323B)의 내용을 클리어하여도 된다.

도 9는, 무비 플레이어 오브젝트(300)가 갖는 일례의 메소드를 일람하여 나타낸다. 이것은, 도 3에서의 메소드(313)에 대응한다. 메소드 play()는, 비디오를 재생한다. 메소드 playChapter()는, 챕터를 지정하여 비디오를 재생한다. 메소드 stop

()은, 비디오의 재생을 정지한다. 메소드 pause()는, 비디오의 재생을 일시 정지한다. 메소드 playStep()은, 비디오를 코마 전송 재생한다. 메소드 changeStream()은, 비디오 스트림, 오디오 스트림 및/또는 자막 스트림을 변경한다. 메소드 getPlayerStatus()는, 무비 플레이어(300)에서의 재생, 정지, 일시 정지 등의 상태를 취득한다. 메소드 reset()은, 비디오의 재생을 정지하고, 리쥼 인포메이션(324)의 내용을 클리어한다.

UMD 비디오 규격에서는, 표시 화면 상의 일부분에 비디오를 표시할 수 있게 되어 있다. 이하의 4개의 메소드는, 이 경우의 비디오 표시에 관한 메소드이다. 메소드 setPos()는, 비디오의 표시 위치를 설정한다. 메소드 getPos()는, 비디오의 표시 위치를 취득한다. 메소드 setSize()는, 비디오의 표시 사이즈를 설정한다. 메소드 getSize()는, 비디오의 표시 사이즈를 취득한다.

또한, 실제로는, 무비 플레이어(300)와 네이티브 실장 플랫폼(301)은, 일체적으로 구성된다. 즉, 실제로 디스크가 장전되고, 이것을 재생하는 하드웨어로서의 UMD 플레이어와, UMD 플레이어를 제어하는 소프트웨어와의 관계에 대응지어, 어느 부분을 하드웨어에서 행하고, 어느 부분을 소프트웨어에서 행할지는, 실장 시의 구성에 의존한다. 예를 들면, UMD 플레이어를 퍼스널 컴퓨터 등으로 구성하는 경우에는, 디스크 드라이브 이외에는, 소프트웨어적으로 구성할 수 있다. 또한, 단체의 UMD 플레이어로서 구성하는 경우에는, 디스크 드라이브 이외에, 예를 들면 비디오 디코더나 오디오 디코더 등을 하드웨어적으로 구성할 수 있다. 따라서, 무비 플레이어(300)와 네이티브 실장 플랫폼(301) 사이에서 이루어지는 메소드나 커맨드, 이벤트는, 도 3에 일례가 도시된 바와 같은 명시적인 교환에 한정되지 않는다.

한편, 유저로부터의 키 입력에 대해서는, 도 3을 이용하여 이미 설명한 바와 같이, 유저 입력(310)을 네이티브 실장 플랫폼(301)이 우선, 수취한다. 즉, 네이티브 실장 플랫폼(301)은, 유저로부터의 키 입력을 유저 입력(310)으로서 수취하고, 유저 입력(310)이 무비 플레이어(300)에 대한 커맨드인지, 스크립트 레이어(302)의 스크립트 프로그램에 대한 이벤트인지를 판정하고, 판정 결과에 따라서, 제어 커맨드(311) 또는 키 이벤트(314)를 발생하고, 대응하는 상위 레이어(무비 플레이어(300) 또는 스크립트 레이어(302))에 통지한다.

도 10 및 도 11은, 유저 입력(310)에 의한 일례의 키 입력을 나타낸다. 또한, 도 10 및 도 11에 도시되는 「VK」로 시작되는 각 키는, 실장에 의존하지 않는 추상화한 가상적인 키이다. 도 10은, 무비 플레이어(300)의 조작에 관한 일례의 키 입력을 나타낸다. 키 VK_POWER는, 전원 키에 대응하는 기능을 제공한다. 키 VK_POWER_ON은, 전원 ON키에 대응하는 기능을 제공한다. 키 VK_POWER_OFF는, 전원 OFF키에 대응하는 기능을 제공한다. 키 VK_MENU는, 메뉴를 표시시키는 메뉴 키에 대응하는 기능을 제공한다. 키 VK_ENTER는, 「결정」을 지시하는 결정 키에 대응하는 기능을 제공한다. 키 VK_RETURN은, 처리의 스텝을 하나 되돌리는 되돌림 키에 대응하는 기능을 제공한다.

키 VK_PLAY는, 재생을 지시하는 재생 키에 대응하는 기능을 제공한다. 키 VK_STOP은, 재생의 정지를 지시하는 정지 키에 대응하는 기능을 제공한다. 키 VK_PAUSE는, 재생의 일시 정지를 지시하는 일시 정지 키에 대응하는 기능을 제공한다. 키 VK_FAST_FORWARD는, 빨리 감기 재생을 지시하는 빨리 감기 키에 대응하는 기능을 제공한다. 키 VK_FAST_REVERSE는, 빨리 되감기 재생을 지시하는 빨리 되감기 키에 대응하는 기능을 제공한다. 키 VK_SLOW_FORWARD는, 역방향의 슬로우 재생을 지시하는 슬로우(순방향) 키에 대응하는 기능을 제공한다. 키 VK_SLOW_REVERSE는, 역방향의 슬로우 재생을 지시하는 슬로우(역방향) 키에 대응하는 기능을 제공한다. 키 VK_STEP_FORWARD는, 순방향의 코마 전송 재생을 지시하는 코마 전송(순방향) 키에 대응하는 기능을 제공한다. 키 VK_STEP_REVERSE는, 역방향의 코마 전송 재생을 지시하는 코마 전송(역방향) 키에 대응하는 기능을 제공한다.

도 11은, 메뉴 조작에 관한 일례의 키 입력을 나타낸다. 키 VK_NEXT는, 「다음」을 의미하는 값을 입력하는 다음 지정 키에 대응하는 기능을 제공한다. 키 VK_PREVIOUS는, 「전」을 의미하는 값을 입력하는 전 지정 키에 대응하는 기능을 제공한다. 예를 들면, 키 VK_NEXT 및 키 VK_PREVIOUS를 이용하여, 전후의 챕터에의 이동을 지시할 수 있다.

키 VK_UP은, 「상」을 의미하는 값을 입력하는 상 방향 지정 키에 대응하는 기능을 제공한다. 키 VK_DOWN은, 「하」를 의미하는 값을 입력하는 하 방향 지정 키에 대응하는 기능을 제공한다. 키 VK_RIGHT는, 「우」를 의미하는 값을 입력하는 우 방향 지정 키에 대응하는 기능을 제공한다. 키 VK_LEFT는, 「좌」를 의미하는 값을 입력하는 좌 방향 지정 키에 대응하는 기능을 제공한다. 키 VK_UP_RIGHT는, 「우상」을 의미하는 값을 입력하는 우상 방향 지정 키에 대응하는 기능을 제공한다. 키 VK_UP_LEFT는, 「좌상」을 의미하는 값을 입력하는 좌상 방향 지정 키에 대응하는 기능을 제공한다. 키 VK_DOWN_RIGHT는, 「우하」를 의미하는 값을 입력하는 우하 방향 지정 키에 대응하는 기능을 제공한다. 키 VK_DOWN_LEFT는, 「좌하」을 의미하는 값을 입력하는 좌하 방향 지정 키에 대응하는 기능을 제공한다. 이들 방향 키를 이용함으로써, 예를 들면 화면 상의 커서 표시의 이동을 지시할 수 있다.

키 VK_ANGLE은, 멀티앵글의 비디오에 대한 앵글 절환을 지시하는 앵글 절환 키에 대응하는 기능을 제공한다. 키 VK_SUBTITLE은, 영어 자막, 일본어 자막, 자막 표시/비표시 등을 절환하는 자막 절환 키에 대응하는 기능을 제공한다. 키 VK_AUDIO는, 서라운드나 바이링걸 등 오디오 설정을 절환하는 오디오 절환에 대응하는 기능을 제공한다. 키 VK_VIDEO_ASPECT는, 비디오의 어스펙트비 절환을 지시하는 어스펙트 절환 키에 대응하는 기능을 제공한다. 키 VK_COLORED_KEY_1은, 착색 평션 키 1, 키 VK_COLORED_KEY_2는, 착색 평션 키 2, 키 VK_COLORED_KEY_3은, 착색 평션 키 3, 키 VK_COLORED_KEY_4는, 착색 평션 키 4, 키 VK_COLORED_KEY_5는, 착색 평션 키 5, 키 VK_COLORED_KEY_6은, 착색 평션 키 6에 각각 대응하는 기능을 제공한다.

전술한 도 10에 도시한 키 입력과 도 11에 도시한 키 입력에서는 역할이 서로 다르기 때문에, 통지처를 네이티브 실장 플랫폼(301)에서 분류할 필요가 있다. 전술한 바와 같이, 도 10에 도시되는 키 입력에 의해, 비디오, 오디오 및 자막의 재생에 관한지시가 이루어진다. 네이티브 실장 플랫폼(301)은, 유저 입력(310)으로서 도 10에 도시되는 키 입력을 수취하면, 수취한 키 입력을, 도 12A, 도 12B 및 도 12C에 도시하는 커맨드로 변환하여 무비 플레이어(300)에 통지한다.

한편, 도 11에 도시되는 키 입력은, GUI에 대한 유저 입력(310)이므로, 이 유저 입력은, 화면 구성이나 버튼을 배치하는 스크립트 레이어(302)에 통지되어 처리될 필요가 있다. 네이티브 실장 플랫폼(301)은, 유저 입력(310)으로서 도 11에 도시되는 키 입력을 수취하면, 도 3에서의 이벤트(314)로 변환하여 스크립트 레이어(302)에 통지한다. 도 13은, 이 키 입력에 대응하는 일례의 이벤트(314)를 나타낸다.

또한, 전술한 도 10 및 도 11에는, 키 VK_ANGLE, 키 VK_SUBTITLE, 키 VK_AUDIO와 같은, 스트림 절환에 관한 키 입력도 포함되어 있지만, 이들은, 스크립트 프로그램으로부터 무비 플레이어(300)에 대한 스트림 절환의 메소드와 마찬가지의 기능을 실현하기 위한 키 입력이다.

전술한 도 12A, 도 12B 및 도 12C의 커맨드에 대하여, 보다 상세하게 설명한다. 커맨드 uo_timeSearch(playListTime)는, 재생 중인 플레이 리스트의 지정 시각으로부터의 재생을 지시한다. 인수 playListTime은, 플레이 리스트의 선두를 0으로 했을 때의 시각을 나타낸다. 이 커맨드에서는, 플레이 리스트 번호의 지정은 할 수 없기 때문에, 인수 playListTime으로 표시되는 시각은, 현재 재생 중인 플레이 리스트의 범위 내에서의 지정 시각으로 된다. 커맨드 uo-Play()는, 예를 들면 1배속과 같은 소정의 재생 속도에서의 재생 개시를 지시한다. 개시 위치는, 리쥼 인포메이션(324)에 기초하여 결정된다. 리쥼 인포메이션(324)에 대응하는 정보가 없는 경우에는, 이 유저 조작은 무효로 된다. 이 커맨드는, 플레이 리스트 번호의 지정이 없는 메소드 play()를 실행하였을 때에 대응한다. 또한, 이 커맨드에서, 유저 조작에서는 플레이 리스트 번호를 지정할 수 없다.

커맨드 uo_playChapter(chapterNumber)는, 재생 중인 플레이 리스트의, 인수 chapterNumber로 지정된 챕터로부터의 재생 개시를 지시한다. 챕터의 지정이 없는 경우에는, 현재 재생 중인 챕터의 선두로부터의 재생 개시를 지시한다. 이것은, 챕터 번호의 지정이 없는 메소드 playChapter()에 대응한다. 커맨드 uo_playPrevChapter()는, 현재보다도 하나 전의 챕터로부터의 재생 개시를 지시한다. 커맨드 uo_playNextChapter()는, 현재의 다음의 챕터로부터의 재생 개시를 지시한다. 커맨드 uo_stop()은, 재생의 정지를 지시한다.

커맨드 uo_jumpToEnd()는, 플레이 리스트의 마지막에의 점프를 지시한다. 이 커맨드는, 무비 플레이어(300)에 대하여, 현재의 재생을 중지하여 이벤트 playListEnd를 발생시키도록 지시하는 유저 조작에 대응한다. 이 커맨드에 대응하여, 스크립트 레이어(302)에서는, 이벤트 핸들러 onPlayListEnd가 실행된다. 커맨드 uo_forwardScan(speed)은, 인수 speed로 지정된 재생 속도로의 순방향 재생을 지시한다. 커맨드 uo_backwardScan(speed)은, 인수 speed로 지정된 재생 속도로의 역방향 재생을 지시한다. 이들 커맨드 uo_forwardScan(speed) 및 커맨드 uo_backwardScan(speed)에서의 인수 speed는, UMD 비디오 플레이어의 실장에 의존한다.

커맨드 uo_playStep(forward)은, 순방향의 코마 전송 재생을 지시한다. 커맨드 uo_playStep(backward)은, 역방향의 코마 전송 재생을 지시한다. 커맨드 uo_pauseOn()는, 유저 조작에 기초하여 재생의 일시 정지를 지시한다. 커맨드 uo_pauseOff()는, 유저 조작에 기초하여 재생의 일시 정지 상태를 해제한다.

커맨드 uo_changeAudioChannel(value)은, 오디오의 채널 절환 또는 듀얼 모노 재생 시의 편 채널 절환을 지시한다. 이 커맨드의 실행 시에, 플래그 audioFlag의 값도 대응한 내용으로 변경한다. 커맨드 uo_setAudioEnabled(boolean)은, 오디오 스트림의 ON/OFF를 지정한다. 이 커맨드의 실행 시에, 플래그 audioFlag의 값도 대응한 내용으로 변경한다. 커맨드 uo_set SubtitleEnabled(boolean)은, 자막 스트림의 ON/OFF를 지정한다. 이 커맨드의 실행 시에, 플래그 subtitleFlag의 값도 대응한 내용으로 변경한다. 커맨드 uo_angleChange()는, 표시 앵글의 변경을 지시한다. 이 커맨드에 의한 유저 조작

이 무비 플레이어(300)에 전달되면, 무비 플레이어(300)는, 스크립트 레이어(302)에 대하여 이벤트 angleChange를 통지한다. 커맨드 uo_audioChange(audioStreamNumber)는, 재생하는 오디오 스트림의 변경을 지시한다. 커맨드 uo_subtitleChange(subtitleStreamNumber)는, 재생하는 자막 스트림의 변경을 지시한다.

전술한 도 13에 도시하는 이벤트 및 이벤트의 무비 플레이어(300)의 메소드와의 관계에 대하여, 보다 상세하게 설명한다. 이벤트 menu는, 메뉴로 점프한다. 이 이벤트는, 무비 플레이어(300)에 대해서가 아니라, 네이티브 실장 플랫폼(301)으로부터 스크립트 레이어(302)에 통지된다. 이 이벤트 menu가 스크립트 레이어(302)에 수취되면, 스크립트 레이어(302)는, 이벤트 핸들러 onMenu를 실행한다. 이벤트 exit는, 네이티브 실장 플랫폼(301)이 UMD 비디오 어플리케이션을 종료 시킬 때에, 네이티브 실장 플랫폼(301)으로부터 발생하는 이벤트이다. 이 이벤트 exit가 스크립트 레이어(302)에 수취되면, 스크립트 레이어(302)는, 이벤트 핸들러 onExit를 실행한다.

이벤트 up, 이벤트 down, 이벤트 left, 이벤트 right, 이벤트 focusIn, 이벤트 foeusOut, 이벤트 push 및 이벤트 cancel은, 화면에 표시되어 있는 GUI 부품인 버튼 화상에 포커스가 맞추어져 있는 경우에 발생하는 이벤트이다. 이 이벤트는, 무비 플레이어(300)에 대해서가 아니라, 네이티브 실장 플랫폼(301)으로부터 스크립트 레이어(302)에 통지된다. 또한, 버튼 화상에 포커스가 맞추어진 경우란, 예를 들면, 화면 상의 위치를 지시하기 위한 커서가 버튼 화상의 표시 좌표를 나타내고, 그 버튼 화상이 선택 가능하게 되어 있는 것 같은 상태이다. 이벤트 up, 이벤트 down, 이벤트 left 및 이벤트 right는, 버튼 화상에 대한 포커스가, 각각 상, 하, 좌 및 우의 버튼 화상으로 이동한 경우에 발생한다. 이벤트 focusIn은, 임의의 버튼 화상에 포커스가 맞추어진 경우에 발생하고, 이벤트 focusOut은, 포커스가 맞추어져 있는 버튼 화상으로부터 포커스가 벗어난 경우에 발생한다. 또한, 이벤트 push는, 포커스가 맞추어져 있는 버튼 화상에 대하여 누름 조작이 행해졌을 때에 발생한다. 이벤트 cancel은, 버튼 화상의 누름 조작에 대하여 캔슬 조작이 행해졌을 때에 발생한다.

이벤트 autoPlay 및 이벤트 continuePlay는, 스크립트 레이어(302)에서의 스크립트의 실행 개시를 지시하는 이벤트이다. 이벤트 autoPlay는, 디스크의 장전 시에 자동적으로 스크립트의 실행을 개시하도록 지시하는 이벤트이다. 이벤트 continuePlay는, 디스크 장전 시에, 예를 들면 리쥼 인포메이션(324)에 기초하여, 이전 중지된 시점으로부터의 스크립트의 실행 재개를 지시한다.

도 13에서 도시한 이벤트에 대해서는, 이벤트가 발생했을 때에 실행되는 프로그램이 존재한다. 이 이벤트에 대응한 프로그램을 이벤트 핸들러라고 칭한다. 이벤트와 이벤트 핸들러는, 예를 들면 이름으로 대응 관계를 지을 수 있다. 일례로서, 이벤트명의 선두에 「on」을 부가한 것이 이벤트 핸들러명으로 된다. 도 14 및 도 15는, 일례의 이벤트 핸들러를 나타낸다. 이벤트 핸들러의 내용을 컨텐츠 제작자가 기술함으로써, UMD 비디오 플레이어에 컨텐츠 제작자가 의도하는 다양한 동작을 실행시키는 것이 가능하게 된다.

도 14는, 무비 플레이어 오브젝트(300)가 갖는 일례의 이벤트의 일부와, 대응하는 이벤트 핸들러를 나타낸다. 이 도 14의 이벤트는, 전술한 도 3의 이벤트(312)에 대응하고, 무비 플레이어(300)로부터 스크립트 레이어(302)에 통지된다. 이벤트 핸들러는, 일종의 인터페이스이며, 그 내용은, 예를 들면 컨텐츠 제작자가 스크립트 언어를 이용하여 실장한다. 이벤트 핸들러를 이와 같이 구성함으로써, 이벤트 발생 시에, 컨텐츠 제작자가 의도하는 동작을 실현할 수 있다.

이벤트 mark 및 이벤트 핸들러 onMark()는, 이벤트 마크가 검출되었을 때에 실행된다. 이벤트 마크는, 예를 들면, 플레이 리스트 내에 매립되고, 플레이 리스트의 재생 중에 무비 플레이어(300)에 의해 검출된다. 무비 플레이어(300)에 의해 이벤트 마크가 검출되면, 무비 플레이어(300)로부터 스크립트 레이어(302)에 대하여 이벤트 mark가 통지된다. 스크립트 레이어(302)는, 이 이벤트 mark에 대응하는 이벤트 핸들러 onMark()를 실행한다. 마찬가지로 하여, 이벤트 palyListEnd 및 이벤트 핸들러 onPlayListEnd()는, 플레이 리스트가 종료했을 때에 실행된다. 이벤트 chapter 및 이벤트 핸들러 onChapter()는, 챕터 마크 검출 시에 실행된다. 챕터 마크는, 예를 들면, 플레이 리스트 내에 매립되고, 플레이 리스트의 재생 중에 무비 플레이어(300)에 의해 검출된다.

이벤트 angleChange 및 이벤트 핸들러 onAngleChange()는, 유저 조작에 의해 앵글 변경이 지시되었을 때에 실행된다. 예를 들면, 유저 조작에 따라서 키 입력 VK_ANGLE이 유저 입력(310)으로서 네이티브 실장 플랫폼(301)에 입력되면, 네이티브 실장 플랫폼(301)은, 그 유저 입력(310)을 커맨드 uo_angleChange()로 변환하여 무비 플레이어(300)에 건네준다. 무비 플레이어(300)는, 이 커맨드 uo_angleChange()에 따라서 이벤트 angleChange를 발생시켜, 스크립트 레이어(302)에 건네준다. 스크립트 레이어(302)는, 이 이벤트 angleChange에 대응한 이벤트 핸들러 onAngleChange()를 실행한다. 마찬가지로 하여, 이벤트 audioChange 및 이벤트 핸들러 onAudioChange()는, 유저 조작에 의해 오디오의 변경이 지시되었을 때에 실행된다. 이벤트 subtitleChange 및 이벤트 핸들러 onSubtitleChange()는, 유저 조작에 의해 자막 변경이 지시되었을 때에 실행된다.

도 15는, 시스템 오브젝트가 갖는 일례의 이벤트 핸들러의 일부를 나타낸다. 이 도 15에 도시되는 이벤트 핸들러는, 네이티브 실장 플랫폼(301)이 미리 갖고 있는 이벤트 핸들러이며, 네이티브 실장 플랫폼(301)으로부터 스크립트 레이어(302)에 통지된다.

이벤트 menu 및 이벤트 핸들러 onMenu()는, 메뉴로 점프한다. 이벤트 menu는, 예를 들면, 유저 조작 등으로 메뉴 키가 눌러졌을 때에, 네이티브 실장 플랫폼(301)으로부터 스크립트 레이어(302)에 통지되는 이벤트이다. 스크립트 레이어(302)는, 이 이벤트를 받아, 대응하는 이벤트 핸들러 onMenu()를 실행하고, 이벤트 핸들러 onMenu() 내에서 메뉴 화면을 구성하는 GUI 부품의 배치나 표시 등을 행한다. 이벤트 exit 및 이벤트 핸들러 onExit()는, 네이티브 실장 플랫폼(301)이 UMD 비디오 어플리케이션을 종료 시킬 때에, 네이티브 실장 플랫폼(301)으로부터 발생하는 이벤트 및 대응하는 이벤트 핸들러이다.

이벤트 exit는, 예를 들면, 유저 조작 등에 의해 UMD 비디오 플레이어의 동작의 종료가 지시되었을 때에, 네이티브 실장 플랫폼(301)으로부터 스크립트 레이어(302)에 통지된다. 스크립트 레이어(302)의 스크립트는, 통지된 이벤트 exit를 받아, 이벤트 핸들러 onExit() 내에서 종료 처리를 행할 수 있다. 이벤트 autoPlay 및 이벤트 핸들러 onAutoPlay(), 및, 이벤트 continuePlay 및 이벤트 핸들러 onContinuePlay()는, 각각 스크립트의 실행을 개시한다.

또한, 시스템 오브젝트의 이벤트 핸들러 이외에, 버튼에 관한 이벤트 핸들러가 있다. 이 버튼에 관한 이벤트 핸들러는, 본 발명과 관련성이 낮으므로, 설명을 생략한다.

도 16의 플로우차트를 이용하여, 유저 입력 이벤트를 계기로 하여, 준비된 프로그램이 실행되는 일례의 처리에 대하여, 개략적으로 설명한다. 도 16은, UMD 비디오 플레이어에서 디스크를 통상 재생 중에, 유저에 의해, 다음 챕터를 재생하는 것을 지시하기 위한 "next"키가 눌러졌을 때에, 이 키 입력에 대응하여, 다음의 챕터로 점프하여 재생을 개시함과 함께, 준비된 메시지를 화면 상에 표시하는 예이다.

예를 들면, UMD 비디오 플레이어에 의해 디스크를 통상 재생 중에, 유저가 UMD 비디오 플레이어의 리모트 컨트롤 커맨더를 이용하여 키 "next"를 누르면(스텝 S10), 네이티브 실장 플랫폼(301)에 대한 유저 입력(310)으로서, 키 VK_NEXT가 건네진다. 네이티브 실장 플랫폼(301)에서는, 이 유저 입력(310)에 대응하여 유저 커맨드 uo_playNextChapter()가 발생한다(스텝 S11). 이 유저 커맨드 uo_playNextChapter()는, 무비 플레이어(300)에 통지된다.

이 커맨드 uo_playNextChapter()를 수취한 무비 플레이어(300)는, 데이터베이스(320)를 검색하고, 플레이 리스트 정보로부터 현재 재생하고 있는 위치를 기준으로 하여, 다음 챕터 마크의 위치를 취득한다(스텝 S12). 스텝 S13에서, 다음 챕터 마크가 존재하는지의 여부가 판단되고, 만약, 존재하지 않는다고 판단된 경우, 챕터 점프를 행하지 않고, 현재의 재생이 계속된다.

한편, 스텝 S13에서, 다음 챕터 마크가 존재한다고 판단되면, 처리는 스텝 S14로 이행한다. 스텝 S14에서는, 무비 플레이어(300)는, 현재의 재생을 중지하고, 다음의 챕터 마크가 지시하는, 클립 AV 스트림 파일 내에서의 바이트 위치를, 데이터베이스(320)의 클립 인포메이션 파일의 특징점 정보로부터 취득한다. 그리고, 스텝 S15에서, 취득된 파일 내 바이트 위치에 액세스하고, 그 위치로부터 스트림의 읽어들이기를 개시하여 재생을 개시한다.

스텝 S16 이하는, 챕터가 절환된 것을 알리는 메시지를 화면 상에 표시하기 위한 일련의 수순이다. 챕터가 절환되어 챕터의 선두로부터의 재생이 개시되면, 챕터 이벤트가 발생한다(스텝 S16). 예를 들면, 챕터의 선두에 설치된 챕터 마크가 무비 플레이어(300)에 검출되어, 이벤트 chapter가 발생된다. 이 챕터 이벤트는, 무비 플레이어(300)로부터 스크립트 레이어(302)에 통지된다. 무비 플레이어(300)는, 이 이벤트의 통지 시에, 점프하는 챕터의 챕터 번호도 함께, 스크립트 레이어(302)에 대하여 통지한다. 스크립트 레이어(302)는, 통지된 이벤트에 대응하는 이벤트 핸들러, 예를 들면 이벤트 핸들러 onChapter()의 실행을 개시한다(스텝 S17).

이 예에서는, 이벤트 핸들러 내에는, 챕터가 절환되었을 때에 화면 상에 그 취지를 알리는 메시지를 표시하는 동작이 기술되어 있는 것으로 한다. 스크립트 레이어(302)의 스크립트는, 이 이벤트 핸들러를 실행하여, 이벤트 발생 시에 무비 플레이어(300)로부터 통지된 점프처의 챕터 번호를 취득하고(스텝 S18), 네이티브 실장 플랫폼(301)에 대하여, 예를 들면 취득한 챕터 번호의 챕터의 선두인 것 등, 소정의 메시지를 화면 상에 표시하는 지시를 낸다. 네이티브 실장 플랫폼(301)은 이 지시에 따라서, 화면 상에 메시지를 표시하고(스텝 S19), 이벤트 핸들러에 의한 처리가 종료된다(스텝 S20).

전술한 바와 같은 처리에 의해, 유저가 다음의 챕터의 재생 개시를 지시하는 키 "next"를 조작함으로써 챕터 점프가 행해지고, 점프처인 다음의 챕터의 재생 개시 시에 챕터의 선두인 것을 나타내는 메시지가 화면 상에 표시되게 된다.

이와 같이, 유저 입력 이벤트는, 무비 플레이어(300)의 상태를 변화시키고, 또한, 새로운 이벤트를 발생시키는 계기로도 되어, 새롭게 발생한 이벤트를 이용하여 다양한 처리를 행하게 할 수 있다.

도 17은, UMD 비디오 플레이어에 디스크가 로드되고 나서 이젝트될 때까지의 처리를 개략적으로 나타낸다. 또한, 도 17 중, 사선을 붙인 블록으로 기술된 처리는, 스크립트가 실행되어 있는 상태를 나타낸다.

우선, 유저에 의해 UMD 비디오 플레이어에 디스크가 장전되면, UMD 비디오 플레이어는, 소정의 동작에 의해 디스크를 로드하여, 재생 가능한 상태로 한다(스텝 S30). 디스크가 로드되면, 네이티브 실장 플랫폼(301)에 의해 리쥼 인포메이션(324)이 참조되고, 그 디스크에 대응하는 계속 재생 정보가 로드된다(스텝 S31).

다음으로, 상기 디스크에 대응하는 리쥼 인포메이션(324) 내가 참조되어, 계속 재생 정보가 존재하는지의 여부가 판단되고(스텝 S32), 존재하면, 네이티브 실장 플랫폼(301)으로부터 스크립트 레이어에 대하여 이벤트 continuePlay가 통지된다. 스크립트 레이어(302)는, 통지된 이벤트 continuePlay에 대응하는 이벤트 핸들러 onContinuePlay를 실행한다(스텝 S33). 스텝 S32에서, 상기 디스크에 대응하는 계속 재생 정보가 존재하지 않는다고 판단되면, 처리는 스텝 S34로 이행하고, 네이티브 실장 플랫폼(301)으로부터 스크립트 레이어(302)에 대하여 이벤트 autoPlay가 통지되고, 스크립트 레이어(302)는, 대응하는 이벤트 핸들러 onAutoPlay를 실행시킨다.

스텝 S35에서는, 이벤트 핸들러 onAutoPlay나 이벤트 핸들러 onContinuePlay의 기술 내용에 기초하여 디스크의 재생 동작 등이 행해지고, 디스크의 재생 동작에 수반하여 발생된 이벤트나, 그 이벤트에 대응하는 이벤트 핸들러가 실행된다.

여기에서, 네이티브 실장 플랫폼(301)으로부터 이벤트 exit가 발생되면, 스텝 S36에서, 스크립트 레이어(302)에서 대응하는 이벤트 핸들러 onExit가 실행되어, UMD 비디오 어플리케이션을 종료 시키기 위한 처리가 실행된다. 이벤트 exit는, 예를 들면 리모트 컨트롤 커맨더에 대한 소정의 조작에 따른 유저 입력(310)에 기초하여, 네이티브 실장 플랫폼(301)에서 발생된다.

이벤트 핸들러 onExit에 기초하는 스크립트 처리가 종료하면, 네이티브 실장 플랫폼(301)에 처리가 이행한다. 그리고, 스텝 S37에서, 무비 플레이어(300)에서, 재생 동작을 정지하는 처리가 실행된다. 이 때, 정지된 직전의 상태가 계속 재생 정보로서 리쥼 인포메이션(324)에 기억된다. 그리고, 디스크의 재생이 종료되고(스텝 S38), 동일한 디스크를 다시 재생하지 않을 때에는(스텝 S39), 스텝 S40에서, 네이티브 실장 플랫폼(301)에 의해 디스크가 이젝트되어, 일련의 처리가 종료된다. 또한, 동일한 디스크를 다시 재생할 때에는, 처리는 스텝 S31로 되돌린다.

도 18은, 스크립트 파일의 구성예를 나타낸다. 도 2를 이용하여 이미 설명한 바와 같이, 스크립트 파일은, 스크립트 레이어(302)를 구성하는 파일 "SCRIPT.DAT" 내의 파일로서 존재한다. 스크립트 파일은, 이벤트 핸들러군과 메인 처리부로 이루어진다. 이벤트 핸들러 군은, 1 또는 복수의 이벤트 핸들러가 배열된다. 이벤트의 발생이 스크립트 레이어(302)에 통지될 때마다, 통지된 이벤트에 대응한 이벤트 핸들러가 검색되어, 실행된다. 메인 처리부는, 예를 들면 각 이벤트 핸들러에 공통하여 이용되는 글로벌 변수 등의 정의가 기술되어, 통상적으로, 최초로 1회만 실행된다.

도 19는, 이벤트 핸들러 onAutoPlay()를 실행하는 일례의 수순을 나타낸다. UMD 비디오 플레이어에 디스크를 장전할 때에, 유저에 의해, 처음부터 재생을 행하도록, 무비 플레이어(300)에 대하여 재생 지시가 이루어진 경우(스텝 S50)에, 이 처리가 행해진다. 네이티브 실장 플랫폼(301)은, 스텝 S51에서, 스크립트 내에 이벤트 핸들러 onAutoPlay()가 존재하는지의 여부가 조사된다. 만약, 존재하면, 네이티브 실장 플랫폼(301)은, 이벤트 autoPlay를 스크립트 레이어(302)에 대하여 통지한다(스텝 S52). 이것을 받아, 스텝 S54에서, 스크립트 레이어(302)는, 이벤트 핸들러 onAutoPlay()를 실행한다. 이에 의해, 장전된 디스크가 자동적으로 재생 개시된다.

한편, 스텝 S51에서, 스크립트 내에 이벤트 핸들러 onAutoPlay()가 존재하지 않는다고 하면, 처리는 스텝 S53으로 이행하고, 네이티브 실장 플랫폼(301)은, 이벤트 exit를 스크립트 레이어(302)에 대하여 통지한다. 이 경우, 예를 들면, 메뉴 키 등을 조작하여, 네이티브 실장 플랫폼(301)에 실장되어 있는 메뉴 화면으로부터 재생 지시를 부여함으로써, 디스크의 재생을 개시할 수 있다. 스크립트 레이어(302)가 이벤트 핸들러 onExit()를 갖고 있는 경우, 이 이벤트 핸들러 onExit()가 실행된다.

도 20은, 이벤트 핸들러 onContinuePlay()를 실행하는 일례의 수순을 나타낸다. UMD 비디오 플레이어에 디스크를 장전할 때에, 유저에 의해, 계속 재생을 행하도록, 무비 플레이어(300)에 대하여 재생 지시가 이루어진 경우(스텝 S60)에, 이 처리가 행해진다. 네이티브 실장 플랫폼(301)은, 스텝 S61에서, 장전된 디스크에 대응하는 리쥼 인포메이션(324)이 존재하는지의 여부가 조사된다. 만약, 존재하지 않으면, 처리는 스텝 S62로 이행하고, 선두로부터의 재생으로 된다.

장전된 디스크에 대응하는 리쥼 인포메이션(324)이 존재하는 경우에는, 처리는 스텝 S63으로 이행하고, 스크립트 내에 이벤트 핸들러 onContinuePlay()가 존재하는지의 여부가 조사된다. 만약, 존재하면, 네이티브 실장 플랫폼(301)은, 이벤트 continuePlay를 스크립트 레이어(302)에 대하여 통지한다. 이것을 받아, 스크립트 레이어(302)는, 이벤트 핸들러 onContinuePlay()를 실행한다(스텝 S64). 이에 의해, 장전된 디스크가, 이벤트 핸들러 onContinuePlay()에 따라 재생이 재개된다.

한편, 스텝 S63에서, 스크립트 내에 이벤트 핸들러 onContinuePlay()가 존재하지 않는다고 하면, 처리는 스텝 S65로 이행하여, 디폴트의 이벤트 핸들러 onContinuePlay()가 실행된다. 디폴트의 이벤트 핸들러 onContinuePlay()는, 예를 들면, 리쥼 인포메이션(324)의 정보에 기초하여 전회의 재생 종료 위치로부터, 단순하게 재생을 개시한다.

또한, 이들, 이벤트 핸들러 onAutoPlay 및 이벤트 핸들러 onContinuePlay에 의한 유저 인터페이스는, 전술한 예에 한하지 않고, 다양한 방법이 생각된다. 예를 들면, 전술한 도 20에서는, 스텝 S60에서 유저에 의해 계속 재생이 지시되고 나서, 장전된 디스크에 대응하는 리쥼 인포메이션(324)이 존재하는지의 여부를 조사하고 있지만, 이것은, 순서를 반대로 하여, 리쥼 인포메이션(324)이 존재하는지의 여부를 먼저 조사하고, 존재하는 경우에, 계속 재생을 행할 것인지의 여부의 선택을 유저에게 재촉하여도 된다.

도 21은, 재생 종료 시의 일례의 처리를 나타낸다. 디스크의 재생 중에, 예를 들면 유저에 의해, 무비 플레이어(300)에 대하여 재생을 종료하는 지시가 이루어진 경우(스텝 S70)에, 이 처리가 행해진다. 재생 종료를 지시하는 유저 입력(310)이 네이티브 실장 플랫폼(301)에 대하여 입력되면, 네이티브 실장 플랫폼(301)은, 종료 처리를 개시한다(스텝 S71). 종료 처리는, 예를 들면 하기의 3개의 처리이다.

(1) 새로운 이벤트 발생의 억지

(2) 큐에 저장된 이벤트 핸들러의 파기

(3) 무비 플레이어(300)에 대한 제어 커맨드 uo_stop()의 발행

스텝 S71의 처리가 실행되어, 현재 실행되고 있는 이벤트 핸들러가 종료하면(스텝 S72), 다음의 스텝 S73에서, 네이티브 실장 플랫폼(301)으로부터 스크립트 레이어(302)에 대하여, 이벤트 exit가 통지된다. 스크립트 레이어(302)는, 이것을 받아, 스크립트 레이어(302)는, 이벤트 핸들러 onExit()를 실행한다(스텝 S74). 이벤트 핸들러 onExit()에 의해, 예를 들면, 재생 종료 시의 소정의 후처리나, 유저에 의한 설정 데이터를 기억하는 메소드 setUserData 등이 실행된다.

그리고, 다음 스텝 S75에서, 네이티브 실장 플랫폼(301)에 의해, 종료 처리가 이루어진다. 이 종료 처리에서는, 예를 들면, 불휘발성 메모리에 대한 계속 정보의 보존(즉, 재생 종료 직전의 상태의 리쥼 인포메이션(324)에 대한 백업)이나, 시스템 메뉴에의 천이 등이 행해진다.

이상과 같은 플레이어 모델에 의해, 비디오, 오디오 및 자막의 재생이 가능하게 된다. 또한, 컨텐츠 제작자가 미리 설정해 둔, 재생 중인 임의의 시각에 임의의 이벤트를 발생시켜서, 컨텐츠 제작자가 미리 준비해 둔 이벤트 핸들러를 실행하도록 하고 있기 때문에, 컨텐츠 제작자가 의도하는 동작을 실현할 수 있다. 또한, UMD 비디오 플레이어에 의한 디스크의 재생 중에 유저 조작이 있었던 경우는, 네이티브 실장 플랫폼(301)으로부터 무비 플레이어(300)에 대하여, 유저 조작에 따른 커맨드가 통지되어, 유저가 의도하는 그대로 플레이어의 상태를 변화시킬 수 있다. 나아가서는 또한, 유저 조작에 의한 유저 입력을 받은 네이티브 실장 플랫폼이, 스크립트 레이어(302)에 대하여 유저 입력에 대응하는 이벤트를 통지함으로써, 유저 조작에 따라서 컨텐츠 제작자가 준비한 동작을 실현하는 것이 가능하게 된다. 이와 같이 플레이어 모델을 구축함으로써, 비디오, 오디오 및 자막의 재생과, 인터랙티브한 조작을 유저에게 제공하는 것이 가능하게 된다.

5.스크립트 프로그램의 예

다음으로, 스크립트 레이어(302)의 스크립트 프로그램의 예에 대하여 설명한다. 우선, 도 22에 도시된 바와 같은 컨텐츠 재생의 흐름이, 컨텐츠 제작자에 의해 만들어져 있는 것으로 한다. 도 22에 도시되는 컨텐츠는, 표시되는 요소로서는, 플레이 리스트(400 및 401), 톱 메뉴(402), 및, 메시지(403)로 구성된다. 플레이 리스트(400)는, 디스크가 장전되면 자동적으로 표시되는 경고문 화면을 표시하기 위한 것이다. 플레이 리스트(401)는, 예를 들면 이 컨텐츠의 주된 목표인 영화의 본편이다. 톱 메뉴 화면(402)은, 플레이 리스트(401)의 재생을 지시할 수 있도록, 버튼 등의 GUI 부품이 배치된다. 또한, 메시지(403)는, 플레이 리스트(401)의 재생 중인 임의의 시각에 표시된다.

또한, 이 도 22의 구성에서는, 몇 개의 이벤트 핸들러가 준비되어 있다. 이벤트 핸들러 onAutoPlay()는, 디스크가 UMD 플레이어에 장전되면, 플레이 리스트(400)를 자동적으로 재생하고, 경고문을 표시시킨다. 이벤트 핸들러 onPlayListEnd()는, 플레이 리스트의 재생이 종료하면 호출되는 이벤트 핸들러로서, 이 도 22의 예에서는, 플레이 리스트(400)나 플레이 리스트(401)의 종료로 호출된다. 즉, 이벤트 핸들러 onPlayListEnd()는, 어느 플레이 리스트가 종료하였는지를 판정하고, 플레이 리스트(400)의 재생이 종료한 경우에는, 플레이 리스트(401)의 재생 개시를 지시한다. 또한, 플레이 리스트(401)의 재생이 종료한 경우에는, 톱 메뉴 화면(402)을 호출한다.

이벤트 핸들러 onMenu()는, 유저가 메뉴 키를 조작했을 때에 호출되고, 톱 메뉴(402)를 호출하여 화면에 표시한다. 이벤트 핸들러 onMark()는, 재생 중에 마크 Mark가 지시하는 시각에 도달했을 때에 실행된다. 이 도 22의 예에서는, 플레이 리스트(401)에 대하여 마크 Mark가 설정되어 있고, 플레이 리스트(401)의 재생이 마크 Mark가 지시하는 시각에 도달하면, 화면 상에 메시지(403)가 표시되도록 되어 있다.

즉, 도 22의 예에서는, UMD 비디오 플레이어에 디스크가 장전되면, 이벤트 핸들러 onAutoPlay가 호출되어 플레이 리스트(400)가 재생되고, 경고 화면이 표시된다. 플레이 리스트(400)의 재생 시간이 경과하여, 플레이 리스트(400)의 마지막에 도달하면, 이벤트 핸들러 onPlayListEnd가 호출되어, 플레이 리스트(400)가 마지막까지 재생된 것이 판정되고, 다음 플레이 리스트(401)가 재생된다. 여기에서, 플레이 리스트(401)의 재생 중에, 유저에 의해 메뉴 키가 조작되면, 이벤트 핸들러 onMenu가 호출되어, 톱 메뉴 화면(402)이 표시된다. 또한, 이벤트 핸들러 onMenu에 의해, 톱 메뉴 화면(402)에 대한 소정의 조작에 따라서, 플레이 리스트(401)의 선두로부터 재생이 개시된다. 또한, 플레이 리스트(401)의 재생 시각이 마크 Mark가 지시하는 시각에 도달하면, 이벤트 핸들러 onMark가 호출되어, 메시지(403)가 화면 상에 표시된다. 플레이 리스트(401)가 마지막까지 재생되면, 이벤트 핸들러 onPlayListEnd가 호출되어, 플레이 리스트(401)가 마지막까지 재생된 것이 판정되고, 톱 메뉴 화면(402)이 표시된다.

도 23은, 이 도 22에 도시한 바와 같은 동작을 실현하기 위한 일례의 스크립트 프로그램을 나타낸다. 전술한 바와 같이, 스크립트 프로그램은, 이벤트 핸들러를 배열할 수 있어, 이벤트의 발생에 따라 대응하는 이벤트 핸들러가 실행되도록 되어 있다. 스크립트 프로그램은, 후술하는 파일 "SCRIPT.DAT"에 저장된다.

무비 플레이어(300)에 대하여 플레이 리스트의 재생을 지시하는 메소드는, 「movieplayer.play()」이다. 괄호 내에는, 인수로서, 재생하는 플레이 리스트의 번호를 기술한다. 플레이 리스트의 재생이 종료하면, 이벤트 playListEnd가 발생한다. 이 이벤트 playListEnd가 발생하면, 스크립트로부터 이벤트 핸들러 movie player.onPlayListEnd()가 호출된다. 이 때, 스크립트에는, 이벤트 playListEnd와 함께, 오브젝트 event_info가 건네진다. 오브젝트 event_info에는, 어느 플레이 리스트가 종료했는지를 나타내는 플레이 리스트 번호 등이 저장된다. 스크립트에서는, 이 오브젝트 event_info의 내용에 의해, 다음의 동작을 바꿀 수 있다.

6.파일의 관리 구조에 대하여

다음으로, UMD 비디오 규격에 적용되는 파일의 관리 구조에 대하여, 도 24를 이용하여 설명한다. 파일은, 디렉토리 구조에 의해 계층적으로 관리되어, 디스크 상에 기록된다. 디스크의 파일 시스템은, ISO(International Organization for Standarization)-9660 혹은 UDF(Universal Disk Format) 등으로 규정된 파일 시스템을 적용할 수 있다.

루트 디렉토리 밑에, 파일 "TITLEID.DAT" 및, 디렉토리 "VIDEO"가 놓인다. 디렉토리 "VIDEO" 밑에는, 또한, 디렉토리 "RESOURCE", 디렉토리 "CLIP" 및 디렉토리 "STREAM", 및, 파일 "PLAYLIST.DAT"가 놓인다.

파일 "TITLEID.DAT"은, 타이틀(컨텐츠의 종류)마다 서로 다른 타이틀 식별자가 저장되는 파일이다. 1개의 디스크에 대하여, 1개의 파일 "TITLEID.DAT"를 갖는다.

디렉토리 "RESOURCE" 밑에는, 파일 "SCRIPT.DAT"가 놓인다. 이 파일 "SCRIPT.DAT"은, 전술한 바와 같이, 스크립트 레이어(302)를 구성하는 스크립트 프로그램이 저장된다. 디렉토리 "RESOURCE" 밑에는, 통상적으로, 1개의 파일 "SCRIPT.DAT"가 놓인다. 이에 한하지 않고, 디렉토리 "RESOURCE" 밑에, 복수의 파일 "SCRIPT.DAT"를 놓을 수도 있다. 이 때는, 예를 들면 파일명의 일부를 각각 변경하여, 상호 중복하지 않도록 한다. 복수의 파일 "SCRIPT.DAT"은, 예를 들면, 표시 언어가 상이한 복수의 메뉴 등을 준비할 때에, 언어마다 1개의 파일 "SCRIPT.DAT"가 이용된다. 이 경우에서도, 실제로 사용되는 파일 "SCRIPT.DAT"은, 1개로 된다.

디렉토리 "CLIP" 밑에는, 1 이상의 클립 인포메이션 파일이 놓인다. 클립 인포메이션 파일은, 파일명을, 델리미터인 피리오드의 앞이 「00001」등의 5문자 내지 몇 문자로 이루어지는 문자열(이 예에서는 숫자), 피리오드 뒤의 확장자가 「CLP」라고 한다. 확장자 「CLP」에 의해, 상기 파일이 클립 인포메이션 파일인 것을 식별할 수 있다.

디렉토리 "STREAM" 밑에는, 1 이상의 클립 AV 스트림 파일이 놓인다. 클립 AV 스트림 파일은, 파일명을, 델리미터인 피리오드의 앞이 「00001」 등의 5문자 내지 수문자로 이루어지는 문자열(이 예에서는 숫자), 피리오드의 뒤의 확장자가 「PS」라고 한다. 확장자 「PS」에 의해, 상기 파일이 클립 AV 스트림 파일인 것을 식별할 수 있다. 이 일 실시 형태에서는, 클립 AV 스트림 파일은, 비디오 스트림, 오디오 스트림 및 서브타이틀(자막) 스트림이 다중화되어, MPEG2(Moving Pictures Experts Group2)의 프로그램 스트림으로서, 전술한 확장자 「PS」로 식별되는 파일에 저장된다.

전술한 바와 같이, 클립 AV 스트림 파일은, 비디오 데이터 및 오디오 데이터를 압축 부호화 및 시분할 다중하여 얻어지는 파일로서, 이 파일을 읽어들이고, 디코드 처리를 행함으로써, 비디오 데이터 및 오디오 데이터가 얻어진다. 또한, 클립 인포메이션 파일은, 이 클립 AV 스트림 파일의 성질 등이 기술되는 파일로서, 클립 AV 스트림 파일과 대응한다. 이 일 실시 형태에서는, 클립 인포메이션 파일과 대응하는 클립 AV 스트림 파일로, 파일명에서의, 확장자의 앞의, 5문자 내지 수문자로 이루어지는 문자열을 일치시켜 둠으로써, 양자의 대응 관계를 용이하게 파악할 수 있다.

파일 "SCRIPT.DAT"은, 전술한 바와 같이, 스크립트 프로그램이 기술된 스크립트 파일로서, 이 일 실시 형태가 적용되는 디스크의 재생 형태를 인터랙티브한 것으로 하기 위하여 이용하는 프로그램이 저장되어 있다. 파일 "SCRIPT.DAT"은, 디스크에 저장되는 다른 파일에 앞서 읽어내어진다.

파일 "PLAYLIST.DAT"은, 클립 AV 스트림의 재생순을 지정하는 플레이 리스트가 기술된 플레이 리스트 파일이다. 도 25 내지 도 27을 이용하여, 파일 "PLAYLIST.DAT"의 내부 구조에 대하여 설명한다. 도 25는, 파일 "PLAYLIST.DAT"의 전체 구조를 나타내는 일례의 신택스를 나타낸다. 여기에서는, 신택스를 컴퓨터 장치 등의 프로그램의 기술 언어로서 이용되는 C언어의 기술법에 기초하여 나타낸다. 이것은, 다른 신택스를 도시하는 도면에서, 마찬가지이다.

필드 name_length는, 8비트의 데이터 길이를 갖고, 이 플레이 리스트 파일에 붙여진 명칭의 길이를 나타낸다. 필드 name_string은, 255바이트의 데이터 길이를 갖고, 이 플레이 리스트 파일에 붙여진 명칭을 나타낸다. 필드 name_string은, 그 선두로부터, 필드 name_length가 나타내는 바이트 길이까지가, 유효한 명칭으로서 사용된다. 예를 들면, 필드 name_length가 값 "10"을 갖는 경우에는, 필드 name_string의 선두로부터 10바이트 분이 유효한 명칭으로서 해석된다.

필드 number_of_PlayLists는, 16비트의 데이터 길이를 갖고, 계속하여 기술되는 블록 PlayList()의 개수를 나타낸다. 다음 행의 for 루프에 의해 필드 number_of_PlayLists에 나타내어지는 횟수분만큼, 해당 개수의 블록 PlayList()가 기술된다. 블록 PlayList()는, 플레이 리스트 그 자체이다.

블록 PlayList()의 일례의 내부 구조에 대하여 설명한다. 블록 PlayList()의 선두에는, 필드 PlayList_data_length가 배치된다. 필드 PlayList_data_length는, 32비트의 데이터 길이를 갖고, 그 필드 PlayList_data_length를 포함하는 블록 PlayList()의 데이터 길이를 나타낸다. 계속하여, 15비트의 데이터 길이를 갖는 필드 reserved_for_word_alignment와, 1비트의 데이터 길이를 갖는 플래그 capture_enable_flag_PlayList가 배치된다. 필드 reserved_for_word_alignment는, 데이터 길이가 1비트인 플래그 capture_enable_flag_PlayList와 조합하여, 블록 PlayList() 내에서의 배치를 16비트의 위치에 가지런히 하기 위하여 이용된다.

플래그 capture_enable_flag_PlayList는, 그 capture_enable_flag_PlayList를 포함하는 블록 PlayList()에 속하는 동화상의 2차 이용을 허가할지의 여부를 나타내는 플래그이다. 예를 들면, 이 플래그 capture_enable_flag_PlayList의 값이 "1"이면, 상기 PlayList()에 속하는 동화상 상의, 재생기 내에서의 2차 이용을 허가하는 것을 나타낸다.

또한, 전술에서는, 플래그 capture_enable_flag_PlayList를 1비트의 플래그로 했지만, 이것은 이 예에 한정되지 않는다. 예를 들면, 플래그 capture_enable_flag_PlayList를 복수 비트 구성으로서, 2차 이용의 단계적인 허가를 기술하도록 하여도 된다. 일례로서, 플래그 capture_enable_flag_PlayList를 2비트 구성으로 하고, 값이 "0"인 경우에는 2차 이용을 완전 금지로 하고, 값이 "1"인 경우에는 예를 들면 64화소×64라인 등, 소정의 해상도 이하로 압축 부호화한 경우만 2차 이용을 가능하게 한다. 또한, 값이 "2"이면, 제한 없이 2차 이용을 허가한다고 하는 이용이 생각된다. 이것에 한하지 않고, 2비트 구성 중 비트 0이 값 "1"인 경우에는 컨텐츠 재생 어플리케이션에서의 2차 사용을 허가하고, 비트 1이 값 "1"인 경우에는 동일 케이스 내의 다른 어플리케이션(예를 들면 벽지화상이나 스크린 세이버)에서의 2차 사용을 허가한다. 이 경우에는, 비트 0 및 비트 1의 값을 조합하여 이용할 수 있다.

필드 PlayList_name_length는, 8비트의 데이터 길이를 갖고, 이 블록 PlayList()에 붙여진 명칭의 길이를 나타낸다. 필드 PlayList_name_string은, 255비트의 데이터 길이를 갖고, 이 블록 PlayList()에 붙여진 명칭을 나타낸다. 필드 PlayList_name_string은, 그 선두로부터, 필드 PlayList_name_string이 나타내는 바이트길이까지가, 유효한 명칭으로서 사용된다.

필드 number_of_PlayItems는, 16비트의 데이터 길이를 갖고, 계속하여 기술되는 블록 PlayItem()의 개수를 나타낸다. 다음 행의 for 루프에 의해 필드 number_of_PlayItem2에 나타내어지는 횟수분만큼, 해당 개수의 블록 PlayItem()이 기술된다. 블록 PlayItem()은, 플레이 아이템 그 자체이다.

블록 PlayList() 내의 각 블록 PlayItem()에는, 식별 정보(ID)가 부여된다. 예를 들면, 블록 PlayList() 내의 최초로 기술되는 블록 PlayItem()은, 0번으로 되고, 이후, 블록 PlayItem()의 출현순으로, 1번, 2번, …으로 일련 번호가 붙여진다. 이 일련 번호가 각 블록 PlayItem()의 식별 정보로서 이용된다. 블록 PlayList()의 개수만큼 반복되는 for 루프의 인수 i를, 대응하는 블록 PlayItem()의 식별 정보로서 이용할 수 있다. 블록 PlayItem()의 다음에, 블록 PlayListMark()가 배치된다.

도 26을 이용하여, 블록 PlayItem()의 일례의 내부 구조에 대하여 설명한다. 블록 PlayItem()의 선두에는, 필드 length가 배치된다. 필드 length는, 16비트의 데이터 길이를 갖고, 상기 블록 PlayItem()의 길이를 나타낸다. 계속하여, 필드 Clip_Information_file_name_length가 배치된다. 필드 Clip_Information_file_name_length는, 16비트의 데이터 길이를 갖고, 이 블록 PlayItem()에 대응하는 클립 인포메이션 파일의 명칭의 길이를 나타낸다. 필드 Clip_Information_file_name은, 바이트 단위로 가변 길이의 데이터 길이를 갖고, 이 블록 PlayItem()에 대응하는 클립 인포메이션 파일의 명칭을 나타낸다. 필드 Clip_Information_file_name은, 그 선두로부터, 필드 Clip_Information_file_name_length가 나타내는 바이트 길이까지가, 유효한 명칭으로서 사용된다. 필드 Clip_Information_file_name에서 클립 인포메이션 파일이 지정되면, 전술한 파일명의 대응 관계에 의해, 그 클립 인포메이션 파일에 대응하는 클립 AV 스트림 파일을 특정할 수 있다.

필드 IN_time 및 필드 OUT_time은, 각각 32비트의 데이터 길이를 갖고, 블록 PlayItem() 내에서 필드 Clip_Information_file_name으로 지정한 클립 인포메이션 파일에 대응하는 클립 AV 스트림 파일의 재생 개시 위치 및 재생 종료 위치를 지정하는 시각 정보이다. 이들 필드 IN_time 및 필드 OUT_time의 정보를 이용함으로써, 클립 AV 스트림 파일의 선두 이외의 부분으로부터의 재생 개시를 지정할 수 있다. 마찬가지로, 클립 AV 스트림 파일의 후단 이외의 재생 종료를 지정할 수 있다.

도 27을 이용하여, 블록 PlayListMark()의 일례의 내부 구조에 대하여 설명한다. 블록 PlayListMark()의 선두에는, 필드 length가 배치된다. 필드 length는, 32비트의 데이터 길이를 갖고, 상기 블록 PlayListMark()의 길이를 나타낸다. 계속하여, 필드 number_of_PlayList_marks가 배치된다. 필드 number_of_PlayList_marks는, 16비트의 데이터 길이를 갖고, 계속되는 블록 Mark()의 개수를 나타낸다. 다음 행의 for 루프에 의해 필드 number_of_PlayList_marks에 나타내어지는 횟수분만큼, 해당 개수의 블록 Mark()가 기술된다.

블록 Mark()의 일례의 내부 구조에 대하여 설명한다. 블록 Mark()는, 선두에 필드 mark_type이 배치된다. 필드 mark_type은, 8비트의 데이터 길이를 갖고, 그 필드 mark_type을 포함하는 블록 Mark()의 종류를 나타낸다. 이 일 실시 형태에서는, 도 28에 일례가 도시된 바와 같이, 챕터 마크, 인덱스 마크 및 이벤트 마크의 3종류의 마크가 규정되어 있다. 챕터는, 플레이 리스트(블록 PlayList())를 분할하는 두출 단위이며, 인덱스는, 챕터를 또 분할하는 두출 단위이다. 챕터 마크 및 인덱스 마크는, 각각, 이들 챕터 위치 및 인덱스 위치를 시각 정보로 나타낸다. 이벤트 마크는, 마크 이벤트를 발생시키는 마크이다.

필드 mark_name_length는, 8비트의 데이터 길이를 갖고, 이 블록 Mark()에 붙여진 명칭의 길이를 나타낸다. 블록 Mark()의 최하 행에 배치되는 필드 mark_name_string은, 이 블록 Mark()에 붙여진 명칭을 나타낸다. 필드 mark_name_string은, 그 선두로부터 필드 mark_name_length가 나타내는 바이트 길이까지가, 유효한 명칭으로서 사용된다.

필드 ref_to_PlayItem_id, 필드 mark_time_stamp, 필드 entry_ES_stream_id 및 필드 entry_ES_private_stream_id의 4요소는, 블록 PlayList() 상에서 정의되는 블록 Mark()를, 클립 AV 스트림 파일과 대응짓는다. 즉, 필드 ref_to_ PlayItem_id는, 16비트의 데이터 길이를 갖고, 블록 PlayItem()의 식별 정보를 나타낸다. 이에 의해, 클립 인포메이션 파일과, 클립 AV 스트림 파일이 특정된다.

필드 mark_time_stamp는, 32비트의 데이터 길이를 갖고, 클립 AV 스트림 파일 내에서의 마크의 시간을 지정하기 위하여 이용된다. 도 29를 이용하여, 개략적으로 설명한다. 도 29에서, 플레이 리스트는, 번호 0, 1 및 2가 각각 지정된 3개의 플레이 아이템(PlayItem(#0), PlayItem(#1) 및 PlayItem(#2))으로 이루어지고, 플레이 리스트 상의 시각 t_0은, 번호 1의 플레이 아이템(PlayItem(#1))에 포함되는 것으로 한다. 또한, 번호 0, 1 및 2의 각 플레이 아이템은, 각각 대응하는 클립 인포메이션 파일을 통하여 클립 AV 스트림 파일의 프로그램 스트림(Program Sream) A, B 및 C에 각각 대응하고 있는 것으로 한다.

이러한 경우에, 플레이 리스트 상의 시각 t_0에 마크를 지정하는 경우, 필드 ref_to_PlayItem_id의 값을, 시각 t_0을 포함하는 플레이 아이템을 나타내는 "1"로 하고, 또한, 대응하는 클립 AV 스트림 파일 B 상에서 시각 t_0에 상당하는 시각을, 필드 mark_time_stamp에 기술한다.

도 27의 설명으로 되돌아가, 필드 mark_time_stamp에 계속하여 필드 entry_ES_stream_id 및 필드 entry_ES_private_stream_id가 배치된다. 필드 entry_ES_stream_id 및 필드 entry_ES_private_stream_id는, 각각 8비트의 데이터 길이를 갖고, 상기 블록 Mark()가 특정한 엘리멘터리 스트림에 관련지워져 있는 경우에, 그 엘리멘터리 스트림을 특정하기 위하여 이용된다. 필드 entry_ES_stream_id 및 필드 entry_ES_private_stream_id는, 각각 해당하는 엘리멘터리 스트림이 다중화 되어 있는 패킷(packet())의 스트림 ID(stream_id)와, 프라이빗 패킷 헤더(private_packet_header())의 프라이빗 스트림 ID(private_stream_id)를 나타낸다.

또한, 이들 패킷(packet())의 스트림 ID(stream_id), 프라이빗 패킷 헤더(private_packet_header())의 프라이빗 스트림 ID(private_stream_id)는, 예를 들면 MPEG2 시스템의 프로그램 스트림의 규정에 기초한다.

이들 필드 entry_ES_stream_id 및 필드 entry_ES_private_stream_id는, 예를 들면, 클립 AV 스트림 #0과 클립 AV 스트림 #1에서 상이한 챕터 구성인 경우 등에 이용된다. 해당하는 블록 Mark()가 특정한 엘리멘터리 스트림에 관련지워져 있지 않은 경우에는, 이들 2개의 필드의 값이 각각 "0"으로 된다.

다음으로, 도 30 내지 도 34를 이용하여, 클립 인포메이션 파일의 내부 구조에 대하여 설명한다. 클립 인포메이션 파일 "XXXXX.CLP"는, 전술한 바와 같이, 디렉토리 "STREAM" 밑에 놓여진, 대응하는 클립 AV 스트림 파일 "XXXXX.PS"의 성질 등을 기술한다.

도 30는, 클립 AV 스트림 파일 "XXXXX.CLP"의 전체 구조를 나타내는 일례의 신택스를 나타낸다. 클립 AV 스트림 파일 "XXXXX.CLP"는, 선두에, 필드 presentation_start_time 및 필드 presentation_end_time이 각각 배치된다. 필드 presentation_start_time 및 필드 presentation_end_time은, 각각 32비트의 데이터 길이를 갖고, 대응하는 클립 AV 스트림 파일의 선두와 후단의 시각을 나타낸다. 시각 정보는, MPEG2 시스템에서의 PTS(Presentation Time Stamp)를 이용할 수 있다. PTS는, 90kHz의 정밀도를 갖는다.

다음으로, 7비트의 데이터 길이를 갖는 필드 reserved_for_word_alignment와, 1비트의 데이터 길이를 갖는 플래그 capture_enable_flag_Clip이 배치된다. 필드 reserved_for_word_alignment는, 데이터 길이가 1비트인 플래그 capture_enable_flag_Clip과 조합하여, 파일 "XXXXX.CLP" 내에서의 배치를 16비트의 위치에 가지런히 하기 위하여 이용된다. 플래그 capture_enable_flag_Clip은, 상기 파일 "XXXXX.CLP"에 대응하는 클립 AV 스트림 파일에 포함되는 동화상의 2차 이용을 허가할지의 여부를 나타내는 플래그이다. 예를 들면, 이 플래그 capture_enable_flag_Clip의 값이 "1"이면, 상기 파일 "XXXXX.CLP"에 대응하는 클립 AV 스트림 파일의 동화상의, 재생기 내에서의 2차 이용을 허가하는 것을 나타낸다.

필드 number_of_streams는, 8비트의 데이터 길이를 갖고, 계속되는 블록 StreamInfo() 구조의 개수를 나타낸다. 필드 number_of_streams의 다음으로부터, for 루프에 의해 필드 number_of_streams로 나타내어지는 횟수분만큼, 블록 StreamInfo()가 기술된다. for 루프 후에는, 블록 EP_map()이 배치된다.

블록 StreamInfo()의 일례의 내부 구조에 대하여 설명한다. 블록 Stream Info()의 선두에는, 필드 length가 배치된다. 필드 length는, 16비트의 데이터 길이를 갖고, 상기 블록 StreamInfo()의 길이를 나타낸다. 계속하여, 각각 8비트의 데이터 길이를 갖는 필드 stream_id 및 필드 private_stream_id가 배치되고, 도 31에 일례가 도시된 바와 같이, 상기 블록 StreamInfo()를 엘리멘터리 스트림에 관련짓고 있다. 이 도 31의 예에서는, 상기 블록 StreamInfo()는, 필드 stream_id가 값 "0xE0" 내지 값 "0xEF"로 비디오 스트림에 관련지으며, 값 "0xBD"로 ATRAC(Adaptive Transform Acoustic Coding) 오디오 스트림, LPCM(Linear Pulse Code Modulation) 오디오 스트림 또는 자막 스트림과 관련짓는다. 또한, 상기 블록 StreamInfo()는, 필드 private_stream_id가 값 "0x00" 내지 값 "0x0F", 값 "0x10" 내지 값 "0x1F" 및 값 "0x80" 내지 값 "0x9F"이고, ATRAC 오디오 스트림, LPCM 오디오 스트림 및 자막 스트림에 각각 관련짓는다.

또한, 도 31에서의 값의 표기에서, 「0x」는, 후속하는 수치가 16진 표기인 것을 나타낸다. 이것은, 이하와 마찬가지의 표현에서, 공통이다.

여기에서, 블록 StreamInfo()는, 대별하여, 스트림 중에서 변화하지 않는 정보와 스트림 중에서 변화하는 정보의 2종류의 정보가 기술되어 있다. 스트림 중에서 변화하지 않는 정보는, 블록 StaticInfo()에 기술된다. 한편, 스트림 중에서 변화하는 정보는, 변화점을 시각 정보로 지정하고, 블록 DynamicInfo()에 기술된다.

블록 StreamInfo()에서, 블록 StaticInfo()의 뒤에 바이트 위치를 가지런히 하기 위한, 8비트의 데이터 길이를 갖는 필드 reserved_for_word_alignment가 배치되고, 그 다음으로, 필드 number_of_DynamicInfo가 배치된다. 필드 number_of_DynamicInfo는, 8비트의 데이터 길이를 갖고, 블록 StreamInfo() 내에 그 후에 기술되는 블록 DynamicInfo()의 개수가 나타내어진다. for 루프에 의해, 필드 number_of_DynamicInfo로 나타내어지는 횟수분만큼, 필드 pts_change_point 및 블록 DynamicInfo()가 기술된다.

필드 pts_change_point는, 32비트의 데이터 길이를 갖고, 대응하는 블록 DynamicInfo()의 정보가 유효해지는 시각을 PTS에 의해 나타낸다. 스트림마다 선두로 되는 시각도, 필드 pts_change_Point로 나타내어지고, 이것은, 파일 "XXXXX.CLP" 내에서 정의되는, 전술한 필드 presentation_start_time과 동등하게 된다.

도 32를 이용하여, 블록 StaticInfo()의 일례의 내부 구조에 대하여 설명한다. 블록 StaticInfo()는, 대응하는 엘리멘터리 스트림의 종류에 의해 내용이 서로 다르다. 대응하는 엘리멘터리 스트림의 종류는, 도 31을 이용하여 설명한, 필드 stream_id 및 필드 private_stream_id의 값에 기초하여 판단할 수 있다. 도 32에서는, 블록 StaticInfo()가 대응하는 엘리멘터리 스트림의 종류가 비디오 스트림, 오디오 스트림 및 서브타이틀(자막) 스트림 중 어느 하나인지를, if구문을 이용하여 각각 기술하고 있다. 이하, 블록 StaticInfo()에 대하여, 엘리멘터리 스트림마다 설명한다.

엘리멘터리 스트림이 비디오 스트림이었던 경우, 블록 StaticInfo()는, 각각 4비트의 데이터 길이를 갖는 필드 picture_size 및 필드 frame_rate, 1비트의 데이터 길이를 갖는 플래그 cc_flag로 이루어진다. 필드 picture_size 및 필드 frame_rate는, 상기 비디오 스트림의 화상의 사이즈 및 프레임 주파수를 각각 나타낸다. 플래그 cc_flag는, 상기 비디오 스트림이 클로즈드 캡션을 포함하는지의 여부를 나타낸다. 예를 들면, 플래그 cc_flag의 값이 "1"에서, 상기 비디오 스트림이 클로즈드 캡션을 포함한다. 필드 reserved_for_word_alignment는, 데이터 배치를 16비트로 가지런히 하기 위하여 이용된다.

엘리멘터리 스트림이 오디오 스트림이었던 경우, 블록 StaticInfo()는, 16비트의 데이터 길이를 갖는 필드 audio_language_code, 8비트의 데이터 길이를 갖는 필드 channel_configuration, 1비트의 데이터 길이를 갖는 플래그 lfe_exsistance 및 4비트의 데이터 길이를 갖는 필드 sampling_frequency로 이루어진다. 필드 audio_language_code는, 상기 오디오 스트림에 포함되어 있는 언어를 나타내는 코드를 나타낸다. 필드 channel_configuration은, 모노럴, 스테레오, 멀티채널 등, 오디오 데이터의 채널 속성을 나타낸다. 필드 lfe_existance는, 저역 강조 채널이 포함되어 있는지의 여부를 나타내고, 예를 들면 값이 "1"에서, 포함되어 있는 것을 나타낸다. 필드 sampling_frequency는, 오디오 데이터의 샘플링 주파수를 나타낸다. 필드 reserved_for_word_alignment는, 데이터 배치를 16비트로 가지런히 하기 위하여 이용된다.

엘리멘터리 스트림이 서브타이틀(자막) 스트림이었던 경우, 블록 StaticInfo()는, 16비트의 데이터 길이를 갖는 필드 subtitle_language_code 및 1비트의 데이터 길이를 갖는 플래그 configurable_flag로 이루어진다. 필드 subtitle_language_code는, 상기 자막 스트림에 포함되어 있는 언어를 나타내는 코드를 나타낸다. 플래그 configurable_flag는, 상기 자막 스트림을 행사할 때에, 문자의 크기나 위치의 변경을 허가할지의 여부를 나타내고, 예를 들면 값이 "1"에서, 허가하는 것을 나타낸다. 필드 reserved_for_word_alignment는, 데이터 배치를 16비트로 가지런히 하기 위하여 이용된다.

도 33를 이용하여, 블록 DynamicInfo()의 일례의 내부 구조에 대하여 설명한다. 블록 DynamicInfo()는, 선두에, 8비트의 데이터 길이를 갖는 필드 reserved_for_word_alignment가 배치된다. 계속되는 내용은, 대응하는 엘리멘터리 스트림의 종류에 따라 상이하다. 대응하는 엘리멘터리 스트림의 종류는, 도 31를 이용하여 설명한, 필드 stream_id 및 필드 private_stream_id의 값에 기초하여 판단할 수 있다. 도 33에서는, 블록 DynamicInfo()가 대응하는 엘리멘터리 스트림의 종류가 비디오 스트림, 오디오 스트림 및 서브타이틀(자막) 스트림 중 어느 것인지를, if구문을 이용하여 각각 기술하고 있다. 이하, 블록 DynamicInfo()에 대하여, 엘리멘터리 스트림마다 설명한다.

엘리멘터리 스트림이 비디오 스트림이었던 경우, 블록 DynamicInfo()는, 4비트의 데이터 길이를 갖는 필드 display_aspect_ratio로 이루어진다. 필드 display_aspect_ratio는, 비디오의 표시 출력 어스펙트비가 16:9인지 4:3인지를 나타낸다. 필드 reserved_for_word_alignment는, 데이터 배치를 16비트로 가지런히 하기 위하여 이용된다.

엘리멘터리 스트림이 오디오 스트림이었던 경우, 블록 DynamicInfo()는, 4비트의 데이터 길이를 갖는 필드 channel_assignment로 이루어진다. 필드 channel_assignment는, 상기 오디오 스트림이 2채널로 구성되어 있는 경우에, 출력이 스테레오인지 듀얼 모노인지를 나타낸다. 듀얼 모노는, 예를 들면 2개 국어의 음성을 재생 가능하게 할 때에 이용된다. 필드 reserved_for_word_alignment는, 데이터 배치를 16비트로 가지런히 하기 위하여 이용된다.

엘리멘터리 스트림이 자막 스트림이었던 경우, 블록 DynamicInfo()는, 데이터 배치를 16비트로 가지런히 하기 위하여 이용되는, 필드 reserved_for_word_alignment로 구성된다. 즉, 자막 스트림에 관해서는, 동적으로 변화하는 속성이 정의되어 있지 않다.

도 34를 이용하여, 블록 EP_map()의 일례의 내부 구조에 대하여 설명한다. 블록 EP_map()은, 엘리멘터리 스트림마다, 비트 스트림 내의 디코드 개시 가능 위치(엔트리 포인트)를, 시각 정보와 위치 정보를 이용하여 나타낸 것이다. 위치 정보는, 예를 들면 엘리멘터리 스트림이 기록되는 기록 매체에서의, 액세스의 최소 단위를 이용할 수 있다. 각 엘리멘터리 스트림은, 블록 EP_map()으로 나타내어진 위치로부터의 디코드 처리가 가능한 것으로 한다.

고정 레이트의 스트림에서는, 디코드 개시 가능 위치를 계산으로 구할 수 있으므로, 이 블록 EP_map()과 같은 정보는, 불필요하다. 한편, 가변 레이트의 스트림이나, MPEG계의 비디오의 압축 부호화 방식과 같이 액세스 유닛마다 데이터의 사이즈가 변하는 것 같은 스트림의 경우에는, 랜덤 액세스를 행하기 위해 중요한 정보로 된다.

블록 EP_map()은, 선두에, 배치를 16비트로 가지런히 하기 위하여, 8비트의 데이터 길이를 갖는 필드 reserve_for_word_alignment가 배치된다. 계속하여, 필드 number_of_stream_id_entries가 배치된다. 필드 number_of_stream_id_entries는, 8비트의 데이터 길이를 갖고, 이 블록 EP_map()에 기술되어 있는 엘리멘터리 스트림의 수를 나타낸다. 제1 for 루프에 의해, 필드 stream_id, 필드 private_stream_id 및 필드 number_of_EP_entries가, 필드 number_of_stream_id_entries로 나타내어지는 횟수분만큼, 기술된다. 또한, 제1 for 루프의 1회의 기술마다, 제2 for 루프에 의해, 필드 number_of_EP_entries로 나타내어지는 횟수분만큼, 필드 PTS_EP_start 및 필드 RPN_EP_start가 배치된다.

제1 for 루프 내에서, 최초로, 각각 8비트의 데이터 길이를 갖는 필드 stream_id 및 필드 private_stream_id가 배치되고, 도 31에 일례가 도시되도록 하여, 엘리멘터리 스트림을 특정하고 있다. 다음으로, 배치되는 필드 number_of_EP_entries는, 32비트의 데이터 길이를 갖고, 상기 엘리멘터리 스트림에 대하여 기술되어 있는 엔트리 포인트의 수를 나타낸다. 그 후, 제2 for 루프로, 필드 number_of_EP_entries가 나타내는 수만큼, 필드 PTS_EP_start 및 필드 RPN_EP_start가 각각 배치된다.

필드 PTS_EP_start 및 필드 RPN_EP_start는, 각각 32비트의 데이터 길이를 갖고, 엔트리 포인트 자체를 나타낸다. 필드 PTS_EP_start는, 엔트리 포인트의 클립 AV 스트림 파일 내에서의 시각을 PTS로 나타낸다. 한편, 필드 RPN_EP_start는, 엔트리 포인트의 클립 AV 스트림 파일 내에서의 위치를 예를 들면 2048 바이트 단위로 나타낸다.

이 일 실시 형태에서는, 디스크 상의 액세스 단위인 1섹터가 2048바이트로 된다. 그 때문에, 엔트리 포인트의 클립 AV 스트림 파일 내에서의 위치는, 필드 RPN_EP_start에 의해, 섹터 단위로 나타내어지게 된다.

여기에서, 비디오 스트림의 재생 개시 가능 위치의 직전에는, 반드시, 패킷 private_stream_2가 배치된다. 이 패킷 private_stream_2는, 비디오 스트림을 디코드하기 위해 이용 가능한 정보가 저장되는 패킷이다. 그 때문에, 비디오 스트림의 엔트리 포인트의 위치는, 상기 패킷 private_stream_2가 저장되는 팩 pack()의 위치로 된다.

블록 EP_map()은, 전술된 바와 같이 하여, 클립 AV 스트림 상의 시각과, 클립 AV 스트림 파일 내에서의 위치를 대응시키고 있다. 이에 의해, 클립 AV 스트림에의 액세스 포인트의 시각 정보(타임 스탬프)가 주어졌을 때에, 클립 AV 스트림 파일 중에서 데이터의 읽어내기를 개시해야 할 데이터 어드레스를 검색하는 것이 용이해지고, 디스크의 랜덤 액세스를 원활하게 행할 수 있다.

또한, 이 일 실시 형태에서는, 블록 EP_map()에서, 엘리멘터리 스트림마다의 시각 정보와 위치 정보의 조(제2 for 루프 내의 필드 PTS_EP_start와 필드 RPN_EP_start의 조)는, 필드 PTS_EP_start 및 RPN_EP_start의 양방에 대하여 올림차순(또는 내림차순)으로 미리 배열하여 등록하도록 하고 있다. 환언하면, 시각 정보와 위치 정보는, 미리 소정의 방향으로 재배열되어 있다. 이 때문에, 이대로의 데이터에 대하여 이분 검색을 실행하는 것이 가능하다.

또한, 본 발명의 일 실시 형태에서는, 비디오의 엘리멘터리 스트림은, MPEG2-Video의 규격에 기초하는 엘리멘터리 스트림으로서 설명하였지만, 이것은 이 예에 한정되지 않는다. 예를 들면 비디오의 엘리멘터리 스트림은, MPEG4-Visual이나, MPEG4-AVC에 의한 것이어도 된다. 또한, 오디오의 엘리멘터리 스트림은, ATRAC 오디오의 엘리멘터리 스트림으로서 설명했지만, 이것도 이 예에 한하지 않고, 예를 들면 MPEG1/2/4 오디오에도 적용 가능하다.

7.디스크 재생 장치에 대하여

다음으로, 본 발명의 일 실시 형태를 적용 가능한 디스크 재생 장치에 대하여 설명한다. 도 35는, 본 발명을 적용 가능한 디스크 재생 장치(100)의 일례의 구성을 개략적으로 나타낸다. 버스(111)에 대하여, CPU(Central Processing Unit)(112), 메모리(113), 드라이브 인터페이스(114), 입력 인터페이스(115), 비디오 디코더(116), 오디오 디코더(117), 비디오 입출력 인터페이스(118) 및 오디오 입출력 인터페이스(119)가 각각 접속된다. 이 디스크 재생 장치(100)의 각 부는, 버스(111)를 통하여 비디오 스트림, 오디오 스트림, 각종 커맨드나 데이터 등을 상호 교환할 수 있게 되어 있다.

드라이브 인터페이스(114)에는, 또한, 디스크드라이브(102)가 접속된다. 디스크드라이브(102)는, 드라이브 인터페이스(114)를 통하여 버스(111)와 데이터나 커맨드의 교환을 행한다.

CPU(112)는, ROM(Read Only Memory) 및 RAM(Random Access Memory)을 갖고(도시하지 않음), ROM에 미리 기억된 프로그램이나 데이터에 따라, 버스(111)를 통하여 이 디스크 재생 장치(100)의 각 부와 데이터나 커맨드의 교환을 행하고, 이 디스크 장치(100)의 전체를 제어한다. RAM은, CPU(112)의 워크 메모리로서 이용된다.

입력 인터페이스(115)는, 유저에 의해 실제로 입력 조작이 행해지는 입력 장치로부터의 입력 신호가 공급된다. 입력 장치는, 예를 들면, 적외선 신호 등으로 원격적으로 디스크 재생 장치(100)를 조작하는 리모트 컨트롤 커맨더나, 이 디스크 재생 장치(100)에 직접적으로 설치된 키 등이다. 입력 인터페이스(115)는, 이들 입력 장치로부터 공급된 입력 신호를, CPU(112)에 대한 제어 신호로 변환하여 출력한다.

디스크(101)는, 도 24 이후에서 설명한 바와 같은 포맷으로써, 플레이 리스트, 스크립트 프로그램, 클립 인포메이션 파일, 클립 AV 스트림 파일 등이 기록되어 있다. 디스크(101)가 디스크드라이브(102)에 장전되면, 자동 재생 또는 유저의 입력 조작에 따라 디스크(101)가 재생된다. 디스크(101)로부터 읽어내어진 스크립트 파일이나 플레이 리스트 파일, 클립 인포메이션 파일은, CPU(112)에 공급되어, 예를 들면 CPU(112)가 갖는 RAM에 기억된다. CPU(112)는, RAM에 기억된 이들 데이터나 스크립트 프로그램에 기초하여, 디스크(101)로부터 클립 AV 스트림 파일을 읽어낸다.

디스크(101)로부터 읽어내어진 클립 AV 스트림 파일은, 메모리(113)에 일단 저장된다. 비디오 디코더(116)는, CPU(112)의 명령에 기초하여, 메모리(113)에 저장된 클립 AV 스트림 파일의 비디오 스트림이나 자막 스트림을 디코드한다. 디코드된 비디오 데이터나 자막 데이터는, 예를 들면 CPU(112)에 의해 각각 확대, 축소 처리 등의 화상 처리를 실시함과 함께, 합성, 가산 처리를 실시하여, 1개의 비디오 데이터로 된다. 이들 화상 처리는, 이에 한하지 않고, 비디오 디코더(116)나 비디오 출력 인터페이스(118)에서 행할 수도 있다. 이 비디오 데이터는, 메모리(113)에 버퍼링되어, 비디오 출력 인터페이스(118)에 공급된다. 비디오 출력 인터페이스(118)는, 예를 들면, 공급된 비디오 데이터를 아날로그 비디오 신호로 변환하고, 비디오 출력 단자(120)에 도출한다.

마찬가지로, 오디오 디코더(117)는, CPU(112)의 명령에 기초하여, 메모리(113)에 저장된 클립 AV 스트림 파일의 오디오 스트림을 디코드한다. 디코드된 오디오 데이터는, 메모리(113)에 버퍼링되어, 오디오 출력 인터페이스(119)에 공급된다. 오디오 출력 인터페이스(119)는, 공급된 오디오 데이터를, 예를 들면 아날로그 오디오 신호로 변환하여 오디오 출력 단자(121)에 도출한다.

또한, 여기에서는, 도 35에 도시되는 각 부가 각각 독립된 하드웨어로 구성되어 있는 것과 같이 설명했지만, 이것은 이 예에 한정되지 않는다. 예를 들면, 비디오 디코더(116) 및/또는 오디오 디코더(117)는, CPU(112) 상에서 동작하는 소프트웨어에 의해 구성할 수 있다.

도 36A 및 도 36B는, 도 35에 도시한 디스크 재생 장치(100)에서의 동작을 보다 상세하게 설명하기 위한 기능 블록도이다. 디스크 재생 장치(100)는, 개략적으로는, 오퍼레이션 시스템(201)과, 비디오 컨텐츠 재생부(210)로 이루어진다. 비디오 컨텐츠 재생부(210)는, 실질적으로는, 오퍼레이션 시스템(201) 상에서 동작하는 소프트웨어 프로그램이다. 이에 한하지 않고, 비디오 컨텐츠 재생부(210)는, 소프트웨어와 하드웨어가 통합적에 동작하는 것으로 하여도 된다. 이하에서는, 비디오 컨텐츠 재생부(210)가 소프트웨어인 것으로서 설명한다. 또한, 도 36A 및 도 36B에서는, 디스크드라이브(102)는, 생략되어 있다.

오퍼레이션 시스템(201)은, 디스크 재생 장치(100)에 전원이 투입되면CPU(112)에서 최초로 기동하고, 각 부의 초기 설정 등 필요한 처리를 행하여, 어플리케이션 프로그램(여기서는 비디오 컨텐츠 재생부(210))을 ROM으로부터 호출한다. 오퍼레이션 시스템(201)은, 비디오 컨텐츠 재생부(210)의 동작 중에, 비디오 컨텐츠 재생부(210)에 대하여, 디스크(101)로부터의 파일의 읽어내기나 파일 시스템의 해석과 같은, 기본적인 서비스를 제공한다. 예를 들면, 오퍼레이션 시스템(201)은, 비디오 컨텐츠 재생부(210)로부터 전해진 파일 읽어내기 리퀘스트에 따라서, 드라이브 인터페이스(114)를 통하여 디스크 드라이브(102)를 제어하고, 디스크(101)에 기록되어 있는 데이터를 읽어낸다. 읽어내어진 데이터는, 오퍼레이션 시스템(201)의 제어에 의해, 비디오 컨텐츠 재생부(210)에 건네진다.

또한, 오퍼레이션 시스템(201)은, 멀티태스크 처리 기능을 구비하고, 복수의 소프트웨어 모듈을, 예를 들면 시분할 제어에 의해 외관 상 병렬적으로 제어할 수 있다. 즉, 도 36A 및 도 36B에 일례가 도시된, 비디오 컨텐츠 재생부(210)를 구성하는 각 모듈은, 오퍼레이션 시스템(201)의 멀티태스크 처리 기능에 의해, 모두, 병렬적인 동작이 가능하다.

이하, 비디오 컨텐츠 재생부(210)의 동작에 대하여, 보다 구체적으로 설명한다. 비디오 컨텐츠 재생부(210)는, 내부에 또한 몇 개의 모듈을 갖고 있어, 하기의 기능을 실현한다.

(1) 장전된 디스크(101)가 UMD 비디오의 규격에 준한 디스크(이하, UMD 비디오 디스크라고 함)인지의 여부를 판단한다.

(2) 장전된 디스크(101)가 UMD 비디오 디스크라고 판단한 경우, 디스크(101)로부터 스크립트 파일을 읽어내어, 스크립트 제어 모듈(211)에 건넨다.

(3) 장전된 디스크(101)가 UMD 비디오 디스크라고 판단한 경우, 나아가서는, 데이터베이스를 구성하는 파일(플레이 리스트 파일, 클립 인포메이션 파일 등)을 읽어내어, 플레이어 제어 모듈(212)에 건넨다.

이하, 비디오 컨텐츠 재생부(210)의 각 모듈의 동작에 대하여 설명한다.

스크립트 제어 모듈(211)은, 스크립트 파일 "SCRIPT.DAT"에 기술되어 있는 스크립트 프로그램을 해석하여 실행한다. 플레이어 모델의 설명에서 이미 설명한 바와 같이, 메뉴 화면 등의 화상의 작성 및 출력이나, 유저 입력에 따른 커서 이동, 메뉴 화면의 변경과 같은 GUI는, 스크립트 프로그램에 의해 그래픽스 처리 모듈(219)을 제어함으로써 실현한다. 또한, 스크립트 제어 모듈(211)은, 스크립트 프로그램의 실행에 의해, 플레이어 제어 모듈(212)의 제어 등이 가능하다.

플레이어 제어 모듈(212)은, 디스크(101)로부터 읽어내어진, 플레이 리스트 파일 "PLAYLIST.DAT"나, 클립 인포메이션 파일 "XXXXX.CLP"와 같은 파일에 저장된 데이터베이스 정보를 참조하여, 디스크(101)에 기록되어 있는 비디오 컨텐츠의 재생에 관한, 이하와 같은 제어를 행한다.

(1) 플레이 리스트나 클립 인포메이션과 같은 데이터베이스 정보를 해석한다.

(2) 컨텐츠 데이터 공급 모듈(213), 디코드 제어 모듈(214) 및 버퍼 제어 모듈(215)을 제어한다.

(3) 스크립트 제어 모듈(211) 또는 입력 인터페이스(115)로부터의 지시에 따라, 재생, 재생 정지, 재생 일시 정지와 같은 플레이어의 상태 천이 제어나, 스트림 절환 등의 재생 제어 처리를 행한다.

(4) 디코드 제어 모듈(214)로부터, 재생 중의 비디오 스트림에 대하여, 시각 정보를 취득하여, 시각 표시나 마크 이벤트의 생성 등을 행한다.

컨텐츠 데이터 공급 모듈(213)은, 플레이어 제어 모듈(212)의 지시에 따라, 디스크(101)로부터 클립 AV 스트림 파일과 같은 컨텐츠 데이터를 읽어내어, 버퍼 제어 모듈(215)에 건넨다. 버퍼 제어 모듈(215)은, 건네진 컨텐츠 데이터를 버퍼의 실체(215A)로서의 메모리(113)에 모아 넣는다. 컨텐츠 데이터 공급 모듈(213)은, 버퍼 제어 모듈(215)을 제어하고, 비디오 디코더 제어 모듈(216), 오디오 디코더 제어 모듈(217) 및 자막 디코더 제어 모듈(218)로부터의 요구에 따라, 메모리(113)에 모아 넣어진 컨텐츠 데이터를, 이들 모듈(216, 217 및 218)에 소정으로 공급한다. 또한, 컨텐츠 데이터 공급 모듈(213)은, 버퍼 제어 모듈(215)에 의해 모아 넣은 컨텐츠 데이터의 양을 소정으로 제어하도록, 디스크(101)로부터 컨텐츠 데이터의 읽어들이기를 행한다.

디코드 제어 모듈(214)은, 플레이어 제어 모듈(212)의 지시에 따라, 비디오 디코더 제어 모듈(216), 오디오 디코더 제어 모듈(217) 및 자막 디코더 제어 모듈(218)의 동작을 제어한다. 또한, 디코드 제어 모듈(214)은, 내부에 시계 기능을 갖고, 비디오 데이터와 오디오 데이터가 동기적으로 출력되도록, 각 디코더 제어 모듈(216, 217 및 218)의 동작을 제어한다.

버퍼 제어 모듈(215)은, 버퍼의 실체(215A)로서 메모리(113)의 일부를 배타적으로 이용한다. 또한, 버퍼 제어 모듈(215)은, 데이터 선두 포인터 및 데이터 기입 포인터를 기억한다. 버퍼 제어 모듈(215)은, 또한, 내부 모듈로서 비디오 읽어내기 기능, 오디오 읽어내기 기능 및 자막 읽어내기 기능을 갖는다. 비디오 읽어내기 기능의 내부에는, 비디오 읽어내기 포인터를 갖는다. 또한, 비디오 읽어내기 기능의 내부에는, 액세스 유닛 정보인 정보 au_information()을 축적하기 위한 레지스터를 구비한다. 오디오 읽어내기 기능의 내부에는, 오디오 읽어내기 포인터를 갖는다. 자막 읽어내기 기능의 내부에는, 자막 읽어내기 포인터와 자막 읽어내기 기능 플래그를 갖는다. 자막 읽어내기 기능 플래그는, 기입하는 값에 따라서 자막 읽어내기 기능의 유효/무효를 제어한다. 예를 들면, 자막 읽어내기 기능 플래그에 "1"을 기입하면, 자막 읽어내기 기능이 유효로 되고, "0"을 기입하면, 자막 읽어내기 기능이 무효로 된다.

버퍼 제어 모듈(215)의 내부 모듈인 비디오 읽어내기 기능, 오디오 읽어내기 기능 및 자막 읽어내기 기능은, 또한, 비디오 스트림, 오디오 스트림 및 자막 스트림이 다중화된 클립 AV 스트림으로부터, 각각의 스트림을 분리하는 디멀티 플렉서 기능을 갖는다. 본 발명의 일 실시 형태에서는, MPEG2 시스템의 프로그램 스트림의 형식에서 복수의 엘리멘터리 스트림이 시분할 다중되어, 클립 AV 스트림이 형성되어 있다. 따라서, 비디오 읽어내기 기능, 오디오 읽어내기 기능 및 자막 읽어내기 기능은, MPEG2 시스템의 프로그램 스트림에 대한 디멀티 플렉서 기능을 갖는다.

이 때문에, 비디오 읽어내기 기능은, 스트림 내에 소정으로 배치되는 필드 stream_id(도 31 참조)의 값을 판독하여, 보유한다. 마찬가지로, 오디오 읽어내기 기능 및 자막 읽어내기 기능은, 필드 stream_id 및 필드 private_stream_id(도 31 참조)의 값을 판독하여, 보유한다. 이들 필드 stream_id나 필드 private_stream_id의 값은, 공급된 비트 스트림을 해석할 때에 이용한다.

비디오 디코더 제어 모듈(216)은, 메모리(113)로부터 비디오 스트림의 단일의 비디오 액세스 유닛을 읽어내어 비디오 디코더(116)에 공급하도록, 버퍼 제어 모듈(215) 내의 비디오 읽어내기 기능에 대하여 지시를 낸다. 그리고, 비디오 디코더 제어 모듈(216)은, 비디오 디코더(116)를 제어하고, 비디오 디코더(116)에 공급된 비디오 스트림을 액세스 유닛 단위로 디코드한다. 비디오 스트림을 디코드하여 생성된 비디오 데이터는, 그래픽스 처리 모듈(219)에 공급된다.

마찬가지로, 오디오 디코더 제어 모듈(217)은, 메모리(113)로부터 오디오 스트림의 단일의 오디오 액세스 유닛을 읽어내어 오디오 디코더(117)에 공급하도록, 버퍼 제어 모듈(215) 내의 오디오 읽어내기 기능에 대하여 지시를 낸다. 또한, 이 일 실시 형태에서는, 오디오 스트림을 구성하는 액세스 유닛(오디오 프레임)은, 기지의 고정 길이로 한다. 그리고, 오디오 디코더 제어 모듈(217)은, 오디오 디코더(117)를 제어하여, 오디오 디코더(117)에 공급된 오디오 스트림을 액세스 유닛 단위로 디코드한다. 오디오 스트림을 디코드하여 생성된 오디오 데이터는, 오디오 출력 모듈(242)에 공급된다.

또한, 자막 디코더 제어 모듈(218)은, 메모리(113)로부터 자막 스트림의 단일의 자막 액세스 유닛을 읽어내어 자막 디코더 제어 모듈(218)에 공급하도록, 버퍼 제어 모듈(215) 내의 자막 읽어내기 기능에 대하여 지시를 낸다. 또한, 이 일 실시 형태에서는, 자막 스트림을 구성하는 자막 액세스 유닛은, 유닛의 선두에 그 유닛의 길이 정보가 저장되어 있다. 자막 디코더 제어 모듈(218)은, 자막 디코드 기능을 갖고, 공급된 자막 스트림을 디코드할 수 있다. 자막 디코더 제어 모듈(218)의 자막 디코드 기능에 의해 자막 스트림이 디코드된 자막의 화상 데이터는, 그래픽스 처리 모듈(219)에 공급된다.

그래픽스 처리 모듈(219)은, 전술한 바와 같이, 비디오 디코더 제어 모듈(216)의 제어에 기초하여 비디오 디코더(116)에서 디코드 된 비디오 데이터와, 자막 디코더 제어 모듈(218)에 의해 디코드 된 자막의 화상 데이터가 공급된다. 그래픽스 처리 모듈(219)은, 공급된 이들 비디오 데이터에 대하여 자막의 화상 데이터를 소정으로 가산하고, 출력하기 위한 비디오 신호를 생성한다. 그래픽스 처리 모듈(219)에서는, 또한, 스크립트 제어 모듈(211)이나 플레이어 제어 모듈(212)의 지시에 따라, 메뉴 화상이나 메시지 화상을 생성하고, 출력 비디오 신호에 대하여 합성(오버레이)한다.

예를 들면, 그래픽스 처리 모듈(219)은, 공급된 자막의 화상 데이터에 대하여, 스크립트 제어 모듈(211)로부터의 지시에 따라서 확대 처리나 축소 처리를 행하고, 비디오 데이터에 대하여 소정으로 가산한다.

또한, 그래픽스 처리 모듈(219)은, 미리 지정된 출력 비디오 디바이스의 어스펙트 레시오와, 디스크(101)로부터 재생된 컨텐츠 내에서 지정된 출력 어스펙트 레시오에 기초하여, 출력 신호의 어스펙트 변환을 행한다. 예를 들면, 출력 비디오 디바이스의 어스펙트 레시오가 16:9인 경우, 출력 어스펙트 레시오가 16:9이면, 비디오 데이터를 그대로 출력하고, 출력 어스펙트 레시오가 4:3이면, 출력되는 비디오 데이터를, 화상의 높이가 출력 비디오 디바이스의 화면높이에 일치하도록 스퀴즈(축소) 처리하고, 화상의 좌우에 흑 화상을 삽입하여 출력한다. 출력 비디오 디바이스가 4:3인 경우에는, 출력 어스펙트 레시오가 4:3이면, 비디오 데이터를 그대로 출력하고, 출력 어스펙트 레시오가 16:9이면, 출력되는 비디오 데이터를, 화상의 폭이 출력 비디오 디바이스의 화면폭에 일치하도록 스퀴즈 처리하고, 화상의 상하에 흑 화상을 삽입하여 출력한다.

그래픽스 처리 모듈(219)은, 또한, 플레이어 제어 모듈(212)로부터의 요구에 따라서, 현재 처리 중인 비디오 신호를 캡쳐하고, 플레이어 제어 모듈(212)에 되돌리는 것 같은 처리도 행한다.

비디오 출력 모듈(241)은, 메모리(113)의 일부를 배타적으로 점유하여 FIF0(First In First Out)의 버퍼로서 이용하여, 그래픽스 처리 모듈(219)에 의해 처리된 비디오 데이터를 이 버퍼에 일시적으로 모아 넣고, 소정의 타이밍에서 읽어내는 제어를 행한다. 버퍼로부터 읽어내어진 비디오 데이터는, 비디오 출력 인터페이스(118)로부터 출력된다.

오디오 출력 모듈(242)은, 메모리(113)의 일부를 배타적으로 점유하여 FIFO의 버퍼로서 이용하여, 오디오 디코더(119)로부터 출력된 오디오 데이터를 이 버퍼에 모아 넣고, 소정의 타이밍에서 읽어내는 제어를 행한다. 버퍼로부터 읽어내어진 오디오 데이터는, 오디오 출력 인터페이스(119)로부터 출력된다.

또한, 오디오 출력 모듈(242)은, 컨텐츠의 오디오 모드가 듀얼 모노(예를 들면 2개 국어)였던 경우, 미리 지정된 음성 출력 모드에 따라 오디오 데이터를 출력한다. 음성 출력 모드가 「주음성」으로 지정되어 있는 경우, 예를 들면 메모리(113)에서 좌채널의 오디오 데이터를 우채널에도 카피하여, 2채널의 출력을 양쪽 모두 좌채널의 오디오 데이터로서 출력한다. 음성 출력 모드가 「부음성」이었던 경우에는, 예를 들면 메모리(113)에서 우채널의 오디오 데이터를 좌채널에도 카피하고, 2채널의 출력을 양쪽 모두 우채널의 오디오 데이터로서 출력한다. 음성 출력 모드가 「주·부음성」인 경우나, 컨텐츠가 스테레오인 경우에는, 오디오 데이터를 그대로 출력한다.

이러한 음성 출력 모드의 설정은, 비디오 컨텐츠 재생부(210)가 생성하는 메뉴 화면 등에 의해, 유저가 대화적으로 행할 수 있도록 되어 있다.

불휘발성 메모리 제어 모듈(250)은, 플레이어 제어 모듈(212)로부터의 지시에 의해, 비디오 컨텐츠 재생부(210)가 종료해도 소거되지 않는 영역에의 데이터의 기입이나, 그 영역으로부터의 데이터의 읽어내기를 행한다. 타이틀 식별ID(Title_ID)를 키로 하여, 데이터 Saved_Player_Status 및 데이터 Saved_User_Data의 조를 복수건, 상기 영역에 기억하는 기능을 갖는다. 데이터 Saved_Player_Status로서, 플레이어 제어 모듈(212)이 갖는 데이터 Backup_Player_Status가 기억된다. 이 데이터 Backup_Player_Status는, 예를 들면 전술한 플레이어 스테이터스(323B)의, 플레이어 제어 모듈(212)이 종료하기 직전의 데이터에 대응하고, 데이터 Saved_Player_Status는, 리쥼 인포메이션(324)에 대응한다. 또한, 데이터 Saved_User_Data로서, 플레이어 제어 모듈(212)이 갖는 데이터 User-Data가 기억된다. 데이터 User_Data는, 유저에 의해 플레이어 제어 모듈(212)에 대하여 설정된 소정의 데이터이다.

예를 들면, 디스크 재생 장치(100)가 불휘발성의 메모리인 플래시 메모리 등을 갖는 경우, 불휘발성 메모리 제어 모듈(250)은, 이 플래시 메모리의 소정 영역에 이들 데이터 Saved_Player_Status 및 데이터 Saved_User_Data의 조를, 디스크(101)의 타이틀 ID와 관련지어 기억한다. 불휘발성 메모리 제어 모듈(250)이 데이터를 기억하는 기억 매체는, 플래시 메모리에 한하지 않고, 예를 들면 하드디스크 등이어도 된다.

8.유저 오퍼레이션의 제어에 대해서

다음으로, 본 발명의 일 실시 형태에 의한 유저 오퍼레이션 제한에 대하여 설명한다. 본 발명의 일 실시 형태에서는, 유저 오퍼레이션에 대한 제한의 조합을, 모드(유저 오퍼레이션 마스크 모드:UOP mask mode라고 함)로서 정의한다. 즉, 그 조작을 허가할지의 여부를 나타내는 플래그를 유저 오퍼레이션마다 설치하는 것은 아니고, 빈번하게 사용된다고 생각되는 유저 오퍼레이션의 세트를 미리 플레이어 측에서 준비해 두고, 컨텐츠 제작자측에서는, 준비된 모드를 선택함으로써, 유저 오퍼레이션에 대한 제한을 실현한다.

유저 오퍼레이션 마스크 모드의 정보는, 플레이 리스트의 신택스에서 필드 UOP_mask_mode로서 정의하고, 플레이 리스트마다 갖는 것으로 한다. 이 유저 오퍼레이션 마스크 모드 정보는, 플레이 리스트의 계층에서만 갖고, 복수의 계층에서 갖는 것은 하지 않는다.

이에 따르면, 유저 오퍼레이션에 대한 제한의 조합은, 유저 오퍼레이션 마스크 모드로서, 플레이어측에 실장되어 컨텐츠 제작자측에 제공된다. 그 때문에, 컨텐츠 제작자측에 의한 동작 검증의 부담이 경감된다.

또한, 컨텐츠 제작자가 유저 오퍼레이션의 제한을 행하고자 하는 경우, 미리 준비된 유저 오퍼레이션 마스크 모드를 선택하는 것만으로 되기 때문에, 유저 오퍼레이션을 보다 용이하게 제어할 수 있다. 그 때문에, 컨텐츠 제작자측의 제작 및 검증의 부담이 경감됨과 함께, 플레이어측의 실장 시의 검증의 부담도 경감된다.

이하, 본 발명의 일 실시 형태에서의 유저 오퍼레이션 제한에 대하여, 보다 상세하게 설명한다. 도 37는, 본 발명의 일 실시 형태에 의한 파일 "PLAYLIST.DAT"의 일례의 신택스를 나타낸다. 도 37에 일례가 도시된 바와 같이, 이 일 실시 형태에서는, 도 25를 이용하여 이미 설명한 UMD 비디오 규격에 의한 파일 "PLAYLIST.DAT"에 대하여, 필드 UOP_mask_mode가 추가되어 있다. 도 37의 예에서는, 필드 UOP_mask_mode는, 도 25의 파일 "PLAYLIST.DAT"에서의 필드 PlayList_data_length의 후의 필드 reserve_for_word_alignment와, 필드 capture_enable_flag_PlayList 사이에 추가되어 있다. 따라서, 필드 UOP_mask_mode는, 파일 "PLAY LIST.DAT"에 포함되는 플레이 리스트마다 기술된다.

또한, 이 필드 UOP_mask_mode의 위치는 일례로서, 이 예에 한정되는 것은 아니다.

도 4를 이용하여 이미 설명한 바와 같이, 무비 플레이어(300)는, 디스크(101)의 재생 개시 시에 파일 "PLAYLIST.DAT"를 읽어들이고, 이 디스크(101)의 재생 중에는, 읽어들인 플레이 리스트의 정보를 내부의 메모리에 보유하고 있다. 따라서, 필드 UOP_masu_mode의 정보도, 해당하는 플레이 리스트를 재생 중에는, 메모리에 보유되어 있다.

필드 UOP_mask_mode는, 4비트의 데이터 길이를 갖고, 이 파일 "PLAYLIST.DAT"에 포함되는 플레이 리스트마다 정의되는, 유저 오퍼레이션 마스크 모드를 나타낸다. 도 38는, 필드 UOP mask_mode가 나타내는 값의 의미를 예시한다. 값 「0x0」은, 상기 플레이 리스트의 유저 오퍼레이션 마스크 모드가, 모든 유저 오퍼레이션이 가능한 모드인 것을 나타낸다.

필드 UOP_mask_mode가 값 「0x1」로 되어 있을 때는, 상기 플레이 리스트에 대하여, 유저 오퍼레이션 마스크 모드 「1」이 설정되어 있는 것을 나타낸다. 유저 오퍼레이션 마스크 모드 「1」이 설정된 플레이 리스트는, 유저 오퍼레이션으로서는, 재생 정지(stop)만이 유효로 된다. 상기 플레이 리스트의 재생 중에 그 밖의 유저 오퍼레이션이 행해져도, 플레이어 측은, 무시한다.

또한, 유저 오퍼레이션 마스크 모드가 「1」로 설정되어 있는 플레이 리스트에 대하여, 상기 플레이 리스트 중의 임의의 시각으로부터의 재생을 개시하는, 소위 「스킵 재생」의 유저 오퍼레이션이 이루어졌을 때는, 상기 플레이 리스트의 선두로부터, 순방향의 소정 속 재생(예를 들면 1배속 재생)으로서 재생을 개시해야만 하도록, 정의한다. 즉, 다른 플레이 리스트를 재생 중에, 유저 오퍼레이션 마스크 모드가 「1」로 설정되어 있는 플레이 리스트에 대한 스킵 재생이 발생한 경우, 그 플레이 리스트의 선두로부터 순방향의 예를 들면 1배속이라고 하는 소정속도로의 재생이 행해진다.

이 유저 오퍼레이션 마스크 모드 「1」은, 예를 들면 영화 컨텐츠 등이 수록된 디스크(101)에서, 영화 컨텐츠에 앞서서 재생되는, 무단 복제나 무단 방송 등을 금지하는 메시지가 표시되는 경고 화면(FBI WARNING)을 재생하기 위한 플레이 리스트에 대하여 이용되는 것이 상정되어 있다.

필드 UOP_mask_mode가 값 「0x2」로 되어 있을 때는, 상기 플레이 리스트에 대하여, 유저 오퍼레이션 마스크 모드 「2」가 설정되어 있는 것을 나타낸다. 유저 오퍼레이션 마스크 모드 「2」가 설정된 플레이 리스트는, 유저 오퍼레이션으로서는, 그 플레이 리스트를 재생 중에, 유저 조작에 의해 그 플레이 리스트의 말미에 점프하는 것이 금지된다. 단, 재생의 정지는, 항상 허가된다. 또한, 순방향에의 고속 재생이나, 역방향에의 고속 재생은, 허가된다.

이 유저 오퍼레이션 마스크 모드 「2」는, 전술한 모드 「1」보다는 유저 오퍼레이션의 제한에 대한 강제력이 약하다. 이 유저 오퍼레이션 마스크 모드 「2」는, 예를 들면, 렌탈용의 컨텐츠가 수록된 디스크(101)의, 선두나 말미에 수록되어 있는 선전용 영상(트레일러)을 재생하는 플레이 리스트에 대하여 이용되는 것이 상정되어 있다.

또한, 필드 UOP_mask_mode에서, 값 「0x3」 내지 「0xF」는, 장래를 위한 예약 값이다.

다음으로, 전술한 필드 UOP_IDask_mode의 값을 이용하여 행해지는 유저 오퍼레이션의 제어에 대하여 설명한다. 도 39는, 무비 플레이어(300) 내에서 유저 오퍼레이션 제한 기능을 실현하기 위한 일례의 기능 블록도를 나타낸다. 무비 플레이어(300)는, 디스크(101)로부터 읽어들인 플레이 리스트의 속성 정보(500), 즉 필드 UOP_mask_mode가 나타내는 값에 기초하여 커맨드 필터 테이블(501)을 생성한다.

한편, 유저 오퍼레이션은, 네이티브 실장 플랫폼(301)에 대한 유저 입력(310)으로서 입력된다. 네이티브 실장 플랫폼(301)은, 입력된 유저 입력(310)을 제어 커맨드(311)로 변환하여, 무비 플레이어(300)에 공급한다. 이 제어 커맨드(311)는, 무비 플레이어(300) 내의 커맨드 필터(502)에 건네진다. 커맨드 필터(502)는, 커맨드 필터 테이블(501)을 참조하여, 건네진 제어 커맨드(311)를 플레이 백 모듈(321)에 건넬지의 여부를 판단한다. 필드 UOP_mask_mode에 의해 제한되는 유저 오퍼레이션은, 커맨드 필터 테이블(501)에서 필터링되어, 플레이 백 모듈(321)에 전해지지 않는 제어 커맨드(311)에 대응하는 유저 오퍼레이션이다.

도 40는, 커맨드 필터 테이블(501)의 일례의 작성 수순을 도시하는 플로우차트이다. 예를 들면 디스크 재생 장치(100)에서 디스크(101)가 로드되면(스텝 S80), 무비 플레이어(300)는, 디스크(101)로부터 플레이 리스트나 클립 인포메이션 파일을 읽어들인다. 그리고, 읽어들인 플레이 리스트의 속성 정보로부터, 필드 UOP_mask_mode를 읽어들인다(스텝 S81). 그리고, 읽어들인 필드 UOP_mask_mode에 나타내어지는 유저 오퍼레이션 마스크 모드에 대응한 커맨드 필터 테이블(501)을 작성한다(스텝 S82). 커맨드 필터 테이블(501)은, 플레이 리스트마다 작성된다.

도 41는, 유저 오퍼레이션 마스크 모드 「1」에 대응하는 일례의 커맨드 필터 테이블(501)을 나타낸다. 이 커맨드 필터 테이블(501)에서는, 상기 플레이 리스트의 선두 이외로부터의 재생 개시가 「금지」로 됨과 함께, 허가되는 제어 커맨드(311)는, 커맨드 uo_stop()(도 12A, 도 12B 및 도 12C 참조)만으로 된다.

도 42는, 유저 오퍼레이션 마스크 모드 「2」에 대응하는 일례의 커맨드 필터 테이블(501)을 나타낸다. 상기 플레이 리스트의 선두 이외로부터의 재생 개시가 「허가」됨과 함께, 도 12A, 도 12B 및 도 12C를 이용하여 설명한 각 제어 커맨드(311) 중, 커맨드 uo_jumpToEnd()만이 금지된다. 환언하면, 유저 오퍼레이션 마스크 모드「2」에서는, 커맨드 uo_jumpToEnd() 이외의 제어 커맨드(311)가 모두 허가된다.

도 41 및 도 42에서 설명한 바와 같은 커맨드 필터 테이블(501)은, 컨텐츠 제작자측에서 준비하는 것은 아니고, 무비 플레이어(300)의 내부에서 생성되는 것이다. 커맨드 필터 테이블(501)을 어떤 형식으로 플레이어 내부에 가질지는, 임의이며, 플레이어의 실장에 의존한다.

또한, 도 41 및 도 42에서는, 커맨드 필터 테이블(501)을 유저 오퍼레이션 마스크 모드 「1」 및 「2」에 대하여 각각 나타냈지만, 이것은 이 예에 한정되지 않는다. 예를 들면, 커맨드 필터 테이블(501)은, 유저 오퍼레이션 마스크 모드를 일람하여 통합하여 생성하여도 된다. 또한, if구문을 이용하여 기술할 수도 있다. if구문을 이용하는 경우에는, 커맨드 필터 테이블(501)의 기능을 스크립트 자체에 의해 실현하는 것이 가능하다.

도 43는, 커맨드 필터 테이블(501)을 이용하여 유저 오퍼레이션을 제한하는 일례의 처리를 도시하는 플로우차트이다. 또 이 플로우에 의한 처리가 개시되는데 앞서서, 디스크(101)가 플레이어에 로드되고, 로드 시에 읽어들인 파일 "PLAYLIST.DAT"에 기초하여, 커맨드 필터 테이블(501)이 생성되어 있는 것으로 한다.

스텝 S100에서, 플레이어에 대한 유저 조작이 발생하면, 이 유저 조작에 대응한 유저 입력(310)이 네이티브 실장 플랫폼(301)에 입력된다. 스텝 S101에서, 네이티브 실장 플랫폼(301)이 이 유저 입력(310)을 접수하면, 다음 스텝 S102에서, 네이티브 실장 플랫폼(301)은, 접수한 유저 입력(310)을 무비 플레이어(300)에 대한 제어 커맨드(311)로 변환하고, 무비 플레이어(300)에 통지한다.

무비 플레이어(300)는, 이 제어 커맨드(311)를 수취하면, 현재 재생 중인 플레이 리스트의 커맨드 필터 테이블(501)을 참조한다(스텝 S103). 그리고, 스텝 S104에서, 커맨드 필터 테이블(501)에 기초하여, 통지된 제어 커맨드(311)의 실행이 허가되어 있는지의 여부를 판단한다. 만약, 상기 제어 커맨드(311)의 실행이 허가되어 있지 않다고 판단되면, 처리는 스텝 S105로 이행하고, 무비 플레이어(300)는, 상기 제어 커맨드(311)에 의한 처리를 실행하지 않는다.

한편, 스텝 S104에서, 상기 제어 커맨드(311)의 실행이 허가되어 있다고 판단되면, 처리는 스텝 S106으로 이행한다. 스텝 S106에서는, 상기 제어 커맨드(311)가 현재 재생 중인 플레이 리스트 내에서 실행되는 것인지의 여부가 판단된다. 즉, 스텝 S106에서는, 예를 들면, 제어 커맨드(311)가 상기 플레이 리스트 내의 다른 챕터에 점프하는 챕터 점프나, 스트림 절환과 같은 동작을 지시하는, 현재 재생 중인 플레이 리스트 내에서 실행되는 것인지, 다른 플레이 리스트의 소정의 챕터로부터의 재생 개시를 지시하는 것 같은, 현재의 플레이 리스트 재생을 중단하여 새롭게 다른 플레이 리스트의 재생을 개시하는 것인지를 판단한다.

만약, 스텝 S106에서, 상기 제어 커맨드(311)가 현재 재생 중인 플레이 리스트 내에서 실행되는 것이라고 판단되면, 처리는 스텝 S107로 이행되고, 그 제어 커맨드(311)가 실행된다. 또한, 이 제어 커맨드(311)의 실행에 대하여, 이벤트 핸들러에 의해 제한을 부여할 수 있다. 즉, 유저 오퍼레이션에 대하여, 유저 오퍼레이션 마스크에 의한 필터링을 행한 후에, 다시 이벤트 핸들러에 의한 필터링을 행할 수 있다.

한편, 스텝 S106에서, 상기 제어 커맨드(311)가 현재 재생 중인 플레이 리스트에서 실행되는 것은 아니라고 판단되면, 처리는 스텝 S108로 이행된다. 스텝 S108에서는, 새롭게 재생이 개시되려고 하는 다른 플레이 리스트의 커맨드 필터 테이블(501)이 참조된다. 예를 들면, 전술한 스텝 S102에서 무비 플레이어(300)에 통지된 제어 커맨드(311)가, 현재 재생 중인 플레이 리스트로부터 다른 플레이 리스트에 점프하는 동작을 지시하는 커맨드인 것과 같은 경우, 점프처의 플레이 리스트의 커맨드 필터 테이블(501)이 참조된다.

처리는 스텝 S109로 이행되어, 새롭게 재생이 개시되려고 하는 다른 플레이 리스트의 커맨드 필터 테이블(501)에 기초하여, 그 다른 플레이 리스트에서, 선두로부터의 재생만이 허가되어 있는지의 여부가 판단된다. 만약, 선두로부터의 재생만이 허가되어 있다고 판단되면, 처리는 스텝 S110으로 이행된다. 그리고, 무비 플레이어(300)는, 상기 제어 커맨드(311)가 상기 다른 플레이 리스트의 선두 이외의 위치로부터의 재생을 지시하는 것이어도, 상기 다른 플레이 리스트의 선두로부터 재생 개시하도록, 플레이 백 모듈(321)에 대하여 지시한다.

한편, 스텝 S109에서, 상기 다른 플레이 리스트가 선두 이외의 위치로부터의 재생이 허가되어 있다고 판단되면, 처리는 스텝 S111로 이행된다. 그리고, 무비 플레이어(300)는, 상기 제어 커맨드(311)에 따라, 제어 커맨드(311)에 의해 지정된 시각이나 챕터로부터 상기 다른 플레이 리스트를 재생하도록, 플레이 백 모듈(321)에 대하여 지시한다.

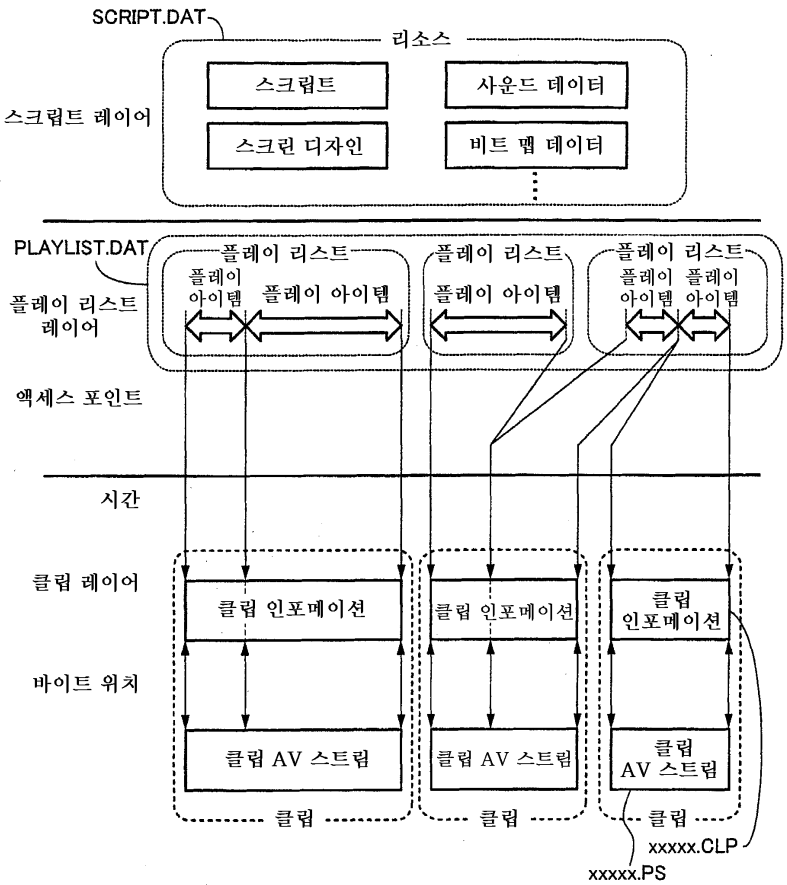
이상 설명한 바와 같이 하여, 본 발명의 일 실시 형태에 의한 유저 오퍼레이션의 제어가 실현된다.

도면

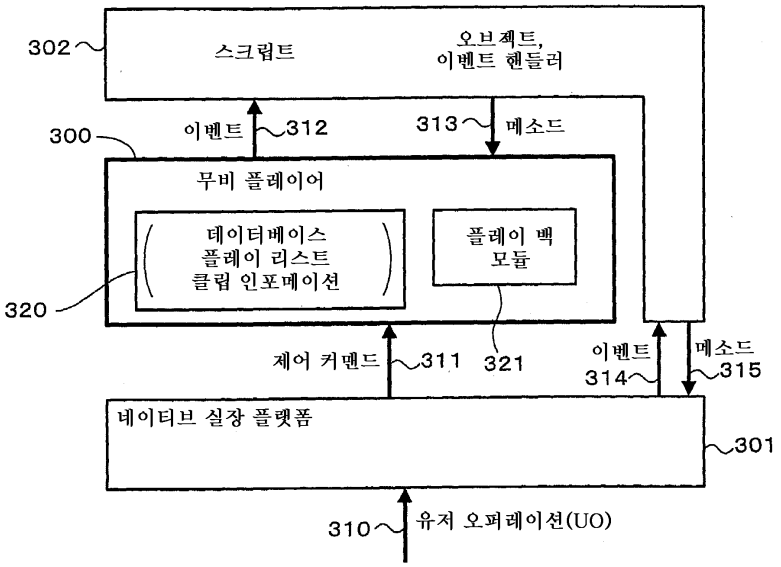
도면1

유저 조작	허가(0), 불허가(1)
타임 플레이, 타임 서치	1
챕터 플레이, 챕터 서치	0
타이틀 플레이	0
정지	0
리턴	0
전 프로그램, 다음 프로그램	1
빨리 감기	1
빨리 되감기	1
메뉴	0
리쥼	0
커서 이동	0
일시 정지	0
일시 정지 해제	0
스트림 절환	0
:	:

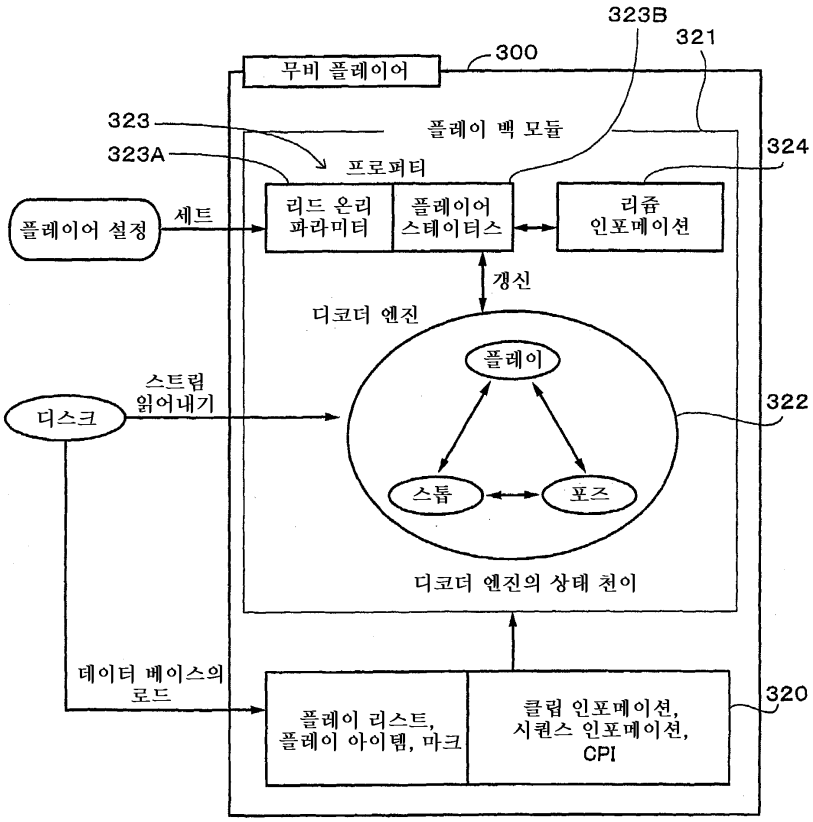
도면2



도면3


302
스크립트
오브젝트,
이벤트 핸들러
이벤트
312
313
메소드
300
무비 플레이어
데이터베이스
플레이 리스트
클립 인포메이션
플레이 백
모듈
320
제어 커맨드
311
321
이벤트
314
메소드
315
네이티브 실장 플랫폼
301
310
유저 오퍼레이션(UO)

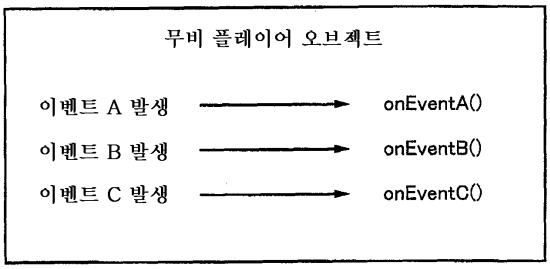
도면4


323B
321
무비 플레이어
300
323
플레이 백 모듈
323A
프로퍼티
324
플레이어 설정
세트
리드 온리
파라미터
플레이어
스테이터스
리쥼
인포메이션
갱신
디코더 엔진
플레이
스트림
읽어내기
디스크
322
스톱
포즈
디코더 엔진의 상태 천이
데이터 베이스의
로드
플레이 리스트,
플레이 아이템, 마크
클립 인포메이션,
시퀀스 인포메이션,
CPI
320

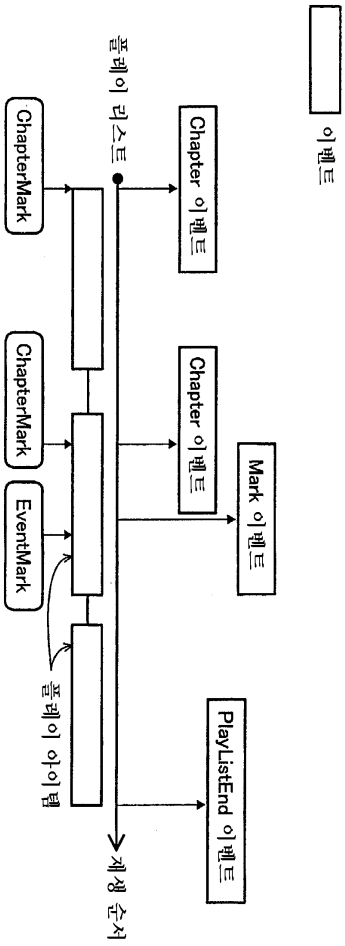
도면5

상태	설명
플레이	플레이 리스트의 재생을 행하고 있고, 시간이 경과되어 있는 상태를 가리킨다. 통상 재생 외, 변속 재생, 빨리 감기, 되감기도 포함한다. 코마 감기, 코마 되감기는 포즈와 플레이를 반복하고 있는 상태이다.
포즈	플레이 리스트의 재생을 행하고 있는 상태로, 시간 축이 정지하고 있는 상태
스톱	플레이 리스트를 재생하지 않고 있는 상태

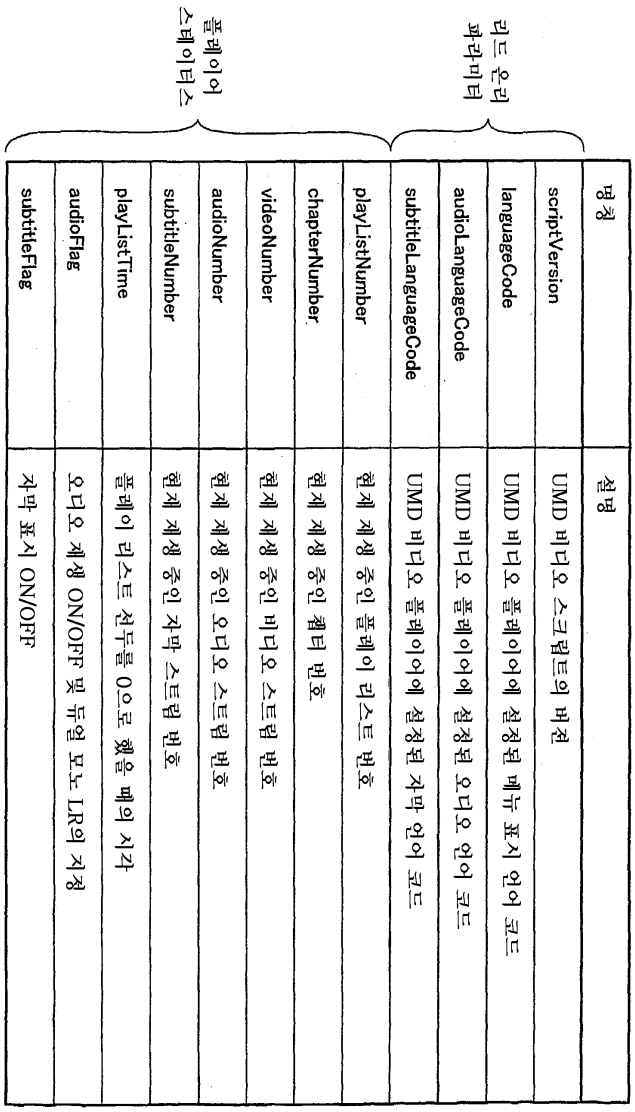
도면6



도면7



도면8


리드 온리 파라미터
플레이어 스테이터스
명칭
설명
scriptVersion
UMD 비디오 스크립트의 버전
languageCode
UMD 비디오 플레이어에 설정된 메뉴 표시 언어 코드
audioLanguageCode
UMD 비디오 플레이어에 설정된 오디오 언어 코드
subtitleLanguageCode
UMD 비디오 플레이어에 설정된 자막 언어 코드
playListNumber
현재 재생 중인 플레이 리스트 번호
chapterNumber
현재 재생 중인 챕터 번호
videoNumber
현재 재생 중인 비디오 스트림 번호
audioNumber
현재 재생 중인 오디오 스트림 번호
subtitleNumber
현재 재생 중인 자막 스트림 번호
playListTime
플레이 리스트 선두를 0으로 했을 때의 시각
audioFlag
오디오 재생 ON/OFF 및 듀얼 모노 LR의 지정
subtitleFlag
자막 표시 ON/OFF

도면9

명칭	설명
play()	재생
playChapter()	챕터 지정 재생
stop()	재생 정지
pause()	재생 일시 정지
playStep()	코마 감기
changeStream()	비디오, 오디오, 자막 스트림 변경
getPlayerStatus()	무비 플레이어의 재생, 정지, 일시 정지 등의 상태를 취득
reset()	재생을 정지하고, 리즘 인포메이션을 클리어
setPos()	비디오의 표시 위치의 설정
getPos()	비디오의 표시 위치의 취득
setSize()	비디오의 표시 사이즈의 설정
getSize()	비디오의 표시 사이즈의 취득

도면10

키 명칭	설명
VK_POWER	전원 키
VK_POWER_ON	전원 ON키
VK_POWER_OFF	전원 OFF키
VK_MENU	메뉴
VK_ENTER	결정
VK_RETURN	되돌림
VK_PLAY	재생
VK_STOP	정지
VK_PAUSE	일시정지
VK_FAST_FORWARD	빨리 감기
VK_FAST_REVERSE	빨리 되감기
VK_SLOW_FORWARD	슬로우(순방향)
VK_SLOW_REVERSE	슬로우(역방향)
VK_STEP_FORWARD	코마 전송(순방향)
VK_STEP_REVERSE	코마 전송(역방향)

도면11

키 명칭	설명
VK_NEXT	다음
VK_PREVIOUS	전
VK_UP	상
VK_DOWN	하
VK_RIGHT	우
VK_LEFT	좌
VK_UP_RIGHT	우상
VK_UP_LEFT	좌상
VK_DOWN_RIGHT	우하
VK_DOWN_LEFT	좌하
VK_ANGLE	앵글 절환
VK_SUBTITLE	자막 절환
VK_AUDIO	오디오 절환
VK_VIDEO_ASPECT	비디오의 어스펙트비 절환
VK_COLORED_KEY_1	착색 평션 키 1
VK_COLORED_KEY_2	착색 평션 키 2
VK_COLORED_KEY_3	착색 평션 키 3
VK_COLORED_KEY_4	착색 평션 키 4
VK_COLORED_KEY_5	착색 평션 키 5
VK_COLORED_KEY_6	착색 평션 키 6

도면12A

도 12A
도 12B
도 12C

유저 조작에 기인하는 제어 명령	설명
uo_timeSearch(playListTime)	재생중인 플레이 리스트의 지정 시각으로부터 재생한다. playListTime은 플레이 리스트 선두를 0으로 했을 때의 시각을 나타낸다. 플레이 리스트 번호는 지정할 수 없다. 그 때문에, 현재 재생 중인 플레이 리스트의 범위 내에서의 시각 지정으로 된다.
uo_play()	1x로 재생 개시한다. 개시 위치는 리쥼 인포메이션에 의해 결정된다. 리쥼 인포메이션이 없는 경우는, 이 유저 조작은 무효로 된다. playListNumber의 지정이 없는 play() 메소드를 실행했을 때에 대응. 유저 조작에서는, 플레이 리스트 번호를 지정할 수 없다.
uo_playChapter(chapterNumber)	재생 중인 플레이 리스트의 지정의 퀄터로부터 재생 개시한다. 퀄터의 지정이 없는 경우에는, 현재 재생 중인 퀄터의 선두로부터 재생 개시한다. chapter Number의 지정이 없는 play Chapter() 메소드에 대응

도면12B

uo_playPrevChapter()	전 챕터의 선두로부터 재생 개시한다.
uo_playNextChapter()	다음 챕터의 선두로부터 재생 개시한다.
uo_stop()	재생을 정지한다.
uo_jumpToEnd()	플레이 리스트의 마지막으로 점프한다. 무비 플레이어에 대해서, 현재의 재생을 중지하고, playListEnd 이벤트를 발생시키도록 지시하는 유저 조작. 스크립트에서는, onPlayListEnd 이벤트 핸들러가 실행된다.
uo_forwardScan(speed)	speed로 지정된 속도로 순방향 재생. speed는 UMD 비디오 플레이어 실장 의존
uo_backwardScan(speed)	speed로 지정된 속도로 역방향 재생. speed는 UMD 비디오 플레이어 실장 의존

도면12C

uo_playStep(forward)	순방향 코마 전송 재생
uo_playStep(backward)	역방향 코마 전송 재생
uo_pauseOn()	유저에 의한 일시 정지
uo_pauseOff()	일시 정지를 해제
uo_changeAudioChannel(value)	오디오의 채널수 절환 및 듀얼 모노 시의 편 채널 절환. audioFlag를 변경한다.
uo_setAudioEnabled(boolean)	오디오 스트림의 ON, OFF를 지정. audioFlag를 변경한다.
uo_setSubtitleEnabled(boolean)	자막 스트림의 ON, OFF를 지정. subtitleFlag를 변경한다.
uo_angleChange()	표시 앵글을 변경한다. 이 유저 조작이 무비 플레이어에 전달되면, 무비 플레이어는 스크립트에 angleChange 이벤트를 통지한다.
uo_audioChange(audioStream-Number)	재생하는 오디오를 변경한다.
uo_subtitleChange(subtitle-StreamNumber)	재생하는 자막을 변경한다.

도면13

이벤트	설명	무비 플레이어의 메소드와의 관계
menu	메뉴로 점프한다.	무비 플레이어가 아니라, 스크립트에 통지되는 이벤트. onMenu 이벤트 핸들러가 실행된다.
exit	네이티브 실장 플랫폼이 UMD 비디오 에플리케이션을 종료시킬 때에 네이티브 실장 플랫폼으로부터 발해지는 이벤트	onExit 이벤트 핸들러가 실행된다.
up,down,left,right focusIn, focusOut, push, cancel	화면에 포지되어 있는 버튼에 포커스가 맞추어져 있는 동안에 발생하는 이벤트	무비 플레이어가 아니라, 스크립트에 통지되는 이벤트
autoPlay, continuePlay	스크립트의 실행 개시를 지시하는 이벤트	

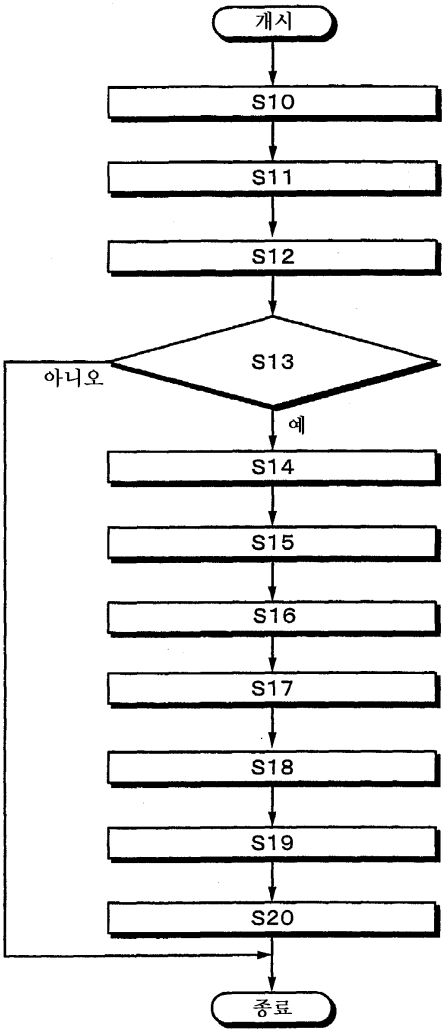
도면14

이벤트명	대응하는 이벤트 핸들러명	설명
mark	onMark()	이벤트 마크 검출 시에 실행된다.
playListEnd	onPlayListEnd()	플레이 리스트가 종료했을 때에 실행된다.
chapter	onChapter()	챕터 마크 검출 시에 실행된다.
angleChange	onAngleChange()	유저 조작의 앵글 변경이 지시되었을 때에 실행된다.
audioChange	onAudioChange()	유저 조작의 오디오 변경이 지시되었을 때에 실행된다.
subtitleChange	onSubtitleChange()	유저 조작의 자막 변경이 지시되었을 때에 실행된다.

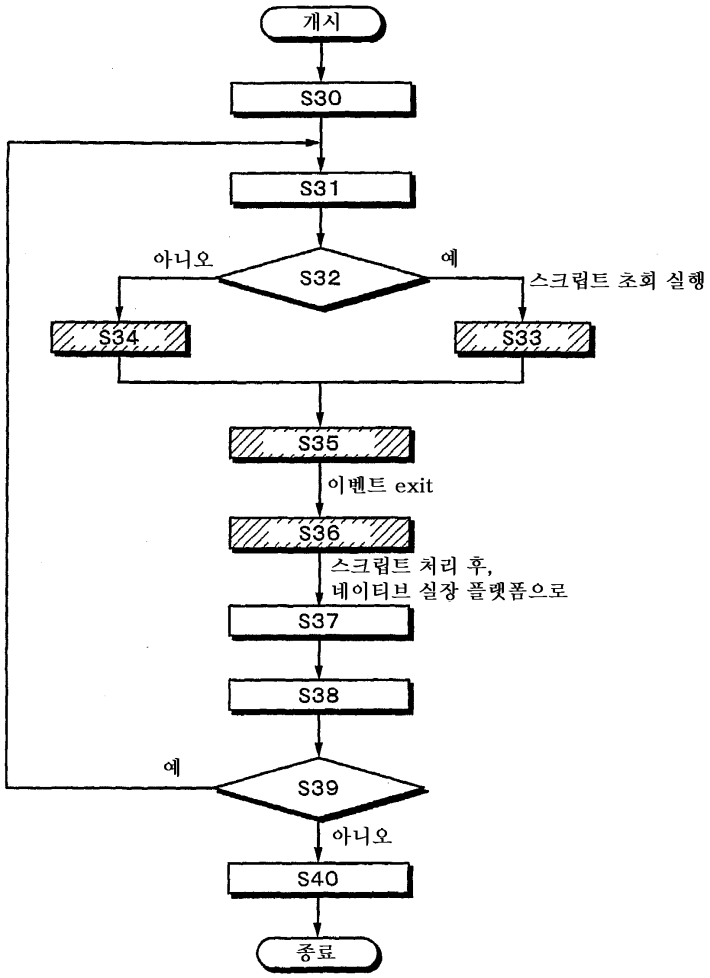
도면15

이벤트명	대응하는 이벤트 핸들러명	내용
menu	onMenu()	메뉴로 점프한다.
exit	onExit()	네이티브 실장 플랫폼이 UMD 비디오 애플리케이션을 종료시킬 때에 네이티브 실장 플랫폼으로부터 발해지는 이벤트
autoPlay, continuePlay	onAutoPlay(), onContinuePlay()	스크립트의 실행을 개시한다.

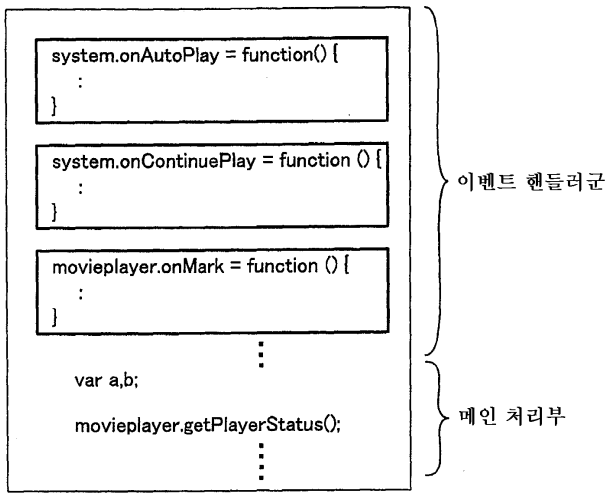
도면16



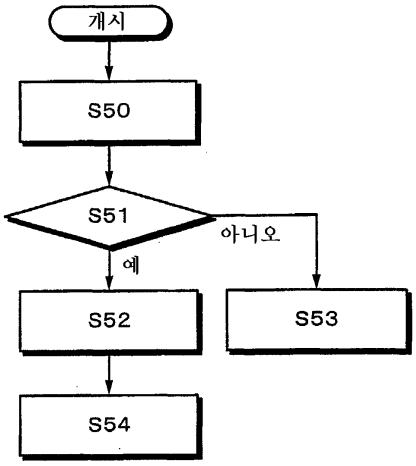
도면17


개시
S30
S31
아니오
S32
예
스크립트 초회 실행
S34
S33
S35
이벤트 exit
S36
스크립트 처리 후,
네이티브 실장 플랫폼으로
S37
S38
예
S39
아니오
S40
종료

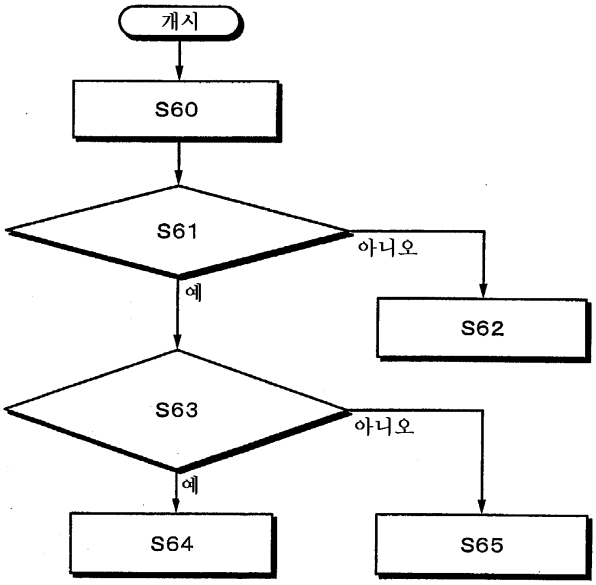
도면18


system.onAutoPlay = function() {
:
}
system.onContinuePlay = function () {
:
}
movieplayer.onMark = function () {
:
}
이벤트 핸들러군
var a,b;
movieplayer.getPlayerStatus();
메인 처리부

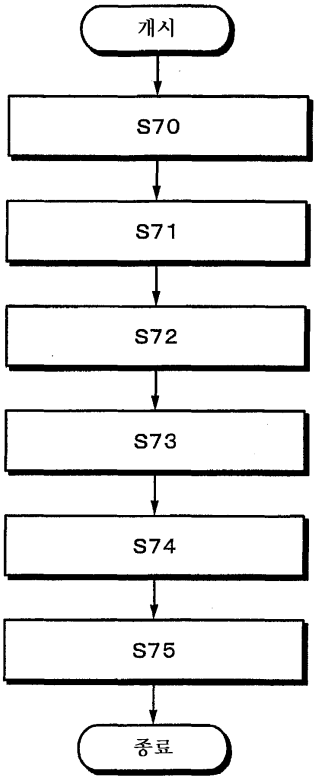
도면19



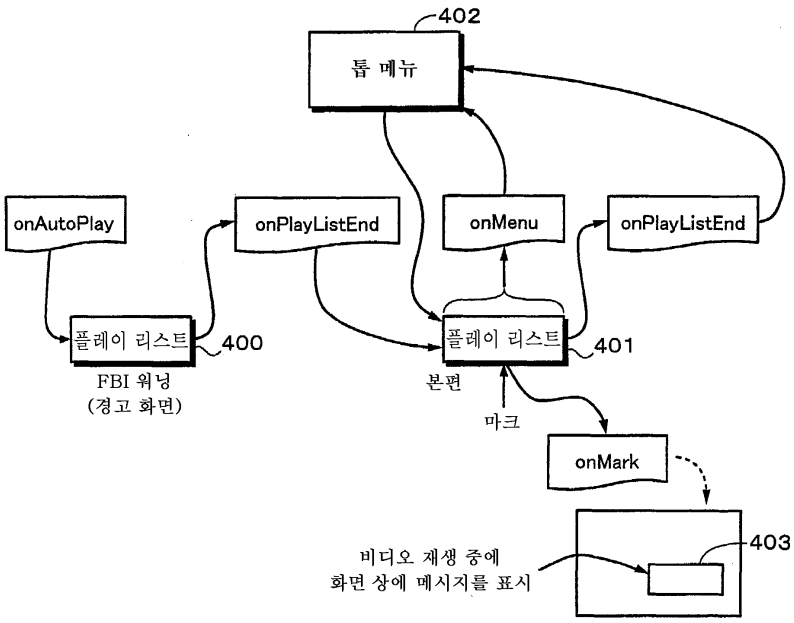
도면20



도면21


개시
S70
S71
S72
S73
S74
S75
종료

도면22


402
톱 메뉴
onAutoPlay
onPlayListEnd
onMenu
onPlayListEnd
플레이 리스트
400
FBI 워닝
(경고 화면)
플레이 리스트
401
본편
마크
onMark
403
비디오 재생 중에
화면 상에 메시지를 표시

도면23

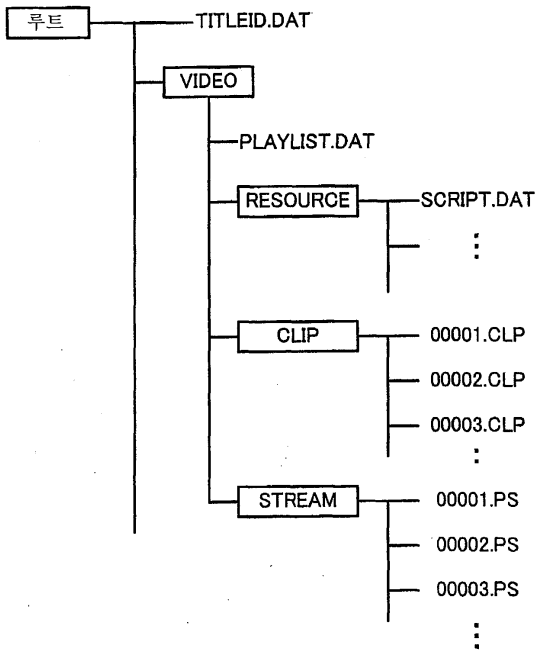
```
system.onAutoPlay = function(){
      //Play PlayList # 1 FBI warning.
      movieplayer.play(1);
}

movieplayersystem.onPlayListEnd = function(event_info){
      if(event_info.playListNumber == 1){
            // play feature film after FBI warning ends.
            movieplayer.play(2);
      }else{
            // transit to top menu after feature film ends.
            resource.pagetable["top_menu"].open();
      }
}

system.onMenu = function(){
      // transfer to top menu with display menu user
operation.
      resource.pagetable["top_menu"].open();
}

movieplayer.onMark = function(event_info){
    //display dialog when event mark encountered.
    if(event_info.mark_data == 1){
        resource.pagetable["dialog_window_1"].open();
    }
    ...
}
```

도면24



도면25

신택스	비트수	니모닉
"PLAYLIST.DAT" {		
name_length	8	uimsbf
name_string	8*255	bslbf
number_of_PlayLists	16	uimsbf
for(i=0; i<number_of_PlayLists; i++){		
PlayList(){ // A PlayList()		
PlayList_data_length	32	uimsbf
//속성 정보		
reserved_for_word_alignment	15	bslbf
capture_enable_flag_PlayList	1	bslbf
PlayList_name_length	8	uimsbf
PlayList_name_string	8*255	bslbf
//		
number_of_PlayItems	16	uimsbf
for (i=0; i<number_of_PlayItems; i++) {		
PlayItem()		
}		
PlayListMark()		
}		
}		
}		

도면26

신택스	비트수	니모닉
PlayItem() {		
length	16	uimsbf
Clip_Information_file_name_length	16	uimsbf
Clip_Information_file_name	8*Clip_Information_file_name_length	bslbf
IN_time	32	uimsbf
OUT_time	32	uimsbf
}		

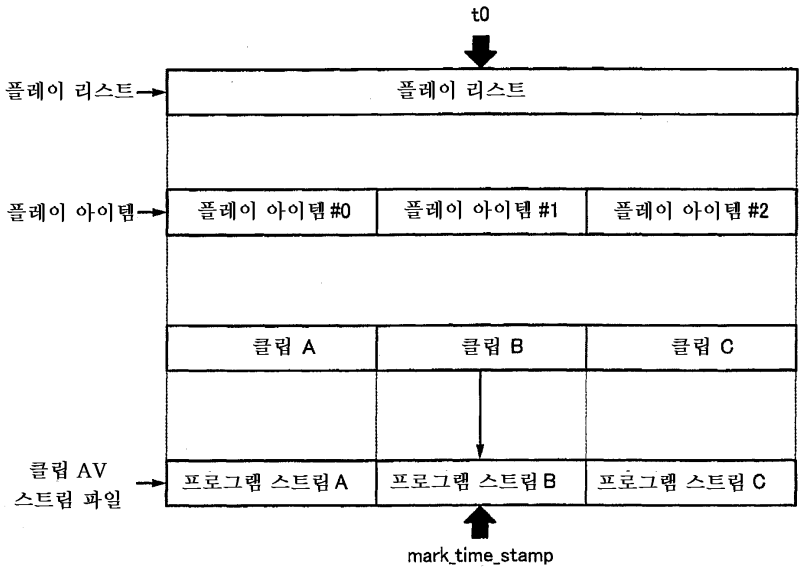
도면27

신택스	비트수	니모닉
PlayListMark() {		
length	32	uimsbf
number_of_PlayList_marks	16	uimsbf
for(i=0; i < number_of_PlayList_marks; i++) {		
Mark(){		
mark_type	8	uimsbf
mark_name_length	8	uimsbf
ref_to_PlayItem_id	16	uimsbf
mark_time_stamp	32	uimsbf
entry_ES_stream_id	8	uimsbf
entry_ES_private_stream_id	8	uimsbf
mark_data	32	bslbf
mark_name_string	8*24	bslbf
}		
}		
}		

도면28

mark_type	스트림 코딩
0	예약
1	챕터 마크
2	인덱스 마크
3	이벤트 마크
4-255	예약

도면29



도면30

신택스	비트수	니모닉
XXXXX.CLP{		
presentation_start_time	32	uimsbf
presentation_end_time	32	uimsbf
reserved_for_word_alignment	7	bslbf
capture_enable_flag_Clip	1	bslbf
number_of_streams	8	uimsbf
for (i = 0;i < number_of_streams;i++) {		
StreamInfo(){		
length	16	uimsbf
stream_id	8	uimsbf
private_stream_id	8	uimsbf
StaticInfo()		
reserved_for_word_alignment	8	bslbf
number_of_DynamicInfo	8	uimsbf
for (j = 0;j < number_of_DynamicInfo;j++) {		
pts_change_point	32	uimsbf
DynamicInfo()		
}		
}		
}		
EP_map()		
}		

도면31

엘리멘터리 스트림의 종류	stream_id	private_stream_id
비디오	0xE0~0xEF	(없음)
ATRAC 오디오	0xBD	0x00~0x0F
LPCM 오디오	0xBD	0x10~0x1F
자막	0xBD	0x80~0x9F

도면32

신택스	비트수	니모닉
StaticInfo() {		
if (stream == VIDEO) {		
reserved_for_word_alignment	16	bslbf
picture_size	4	uimsbf
frame_rate	4	uimsbf
reserved_for_word_alignment	7	bslbf
cc_flag	1	bslbf
} else if (stream == AUDIO) {		
audio_language_code	16	bslbf
channel_configuration	8	uimsbf
reserved_for_word_alignment	3	bslbf
lfe_existence	1	bslbf
sampling_frequency	4	*uimsbf*
} else if (stream == SUBTITLE) {		
subtitle_language_code	16	bslbf
reserved_for_word_alignment	15	bslbf
configurable_flag	1	uimsbf
}		
}		

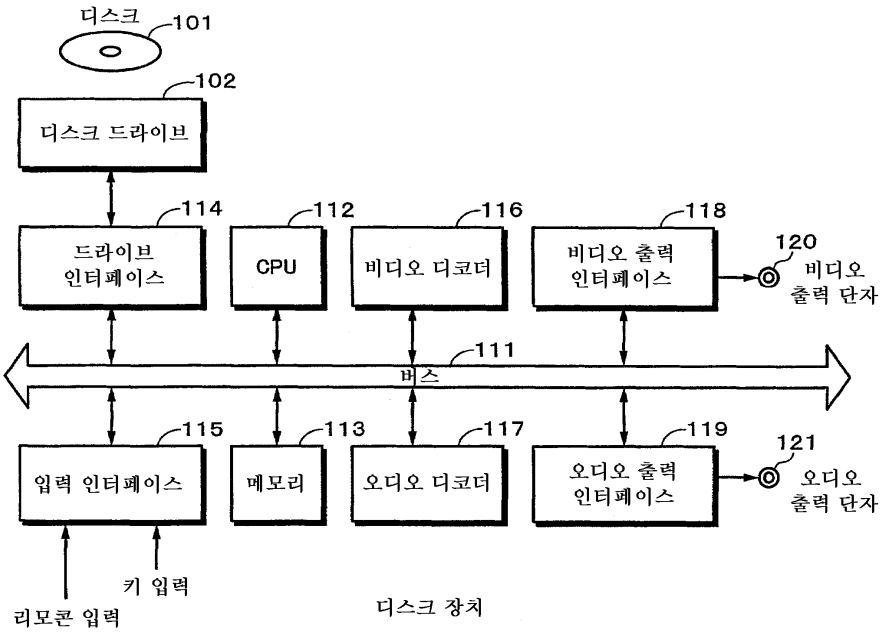
도면33

신택스	비트수	니모닉
DynamicInfo(i,j) {		
reserved_for_word_alignment	8	bslbf
if (stream == VIDEO){		
reserved_for_word_alignment	4	bslbf
display_aspect_ratio	4	uimsbf
} else if (stream == AUDIO) {		
reserved_for_word_alignment	4	bslbf
channel_assignment	4	uimsbf
} else if (stream == SUBTITLE) {		
reserved_for_word_alignment	8	bslbf
}		
}		

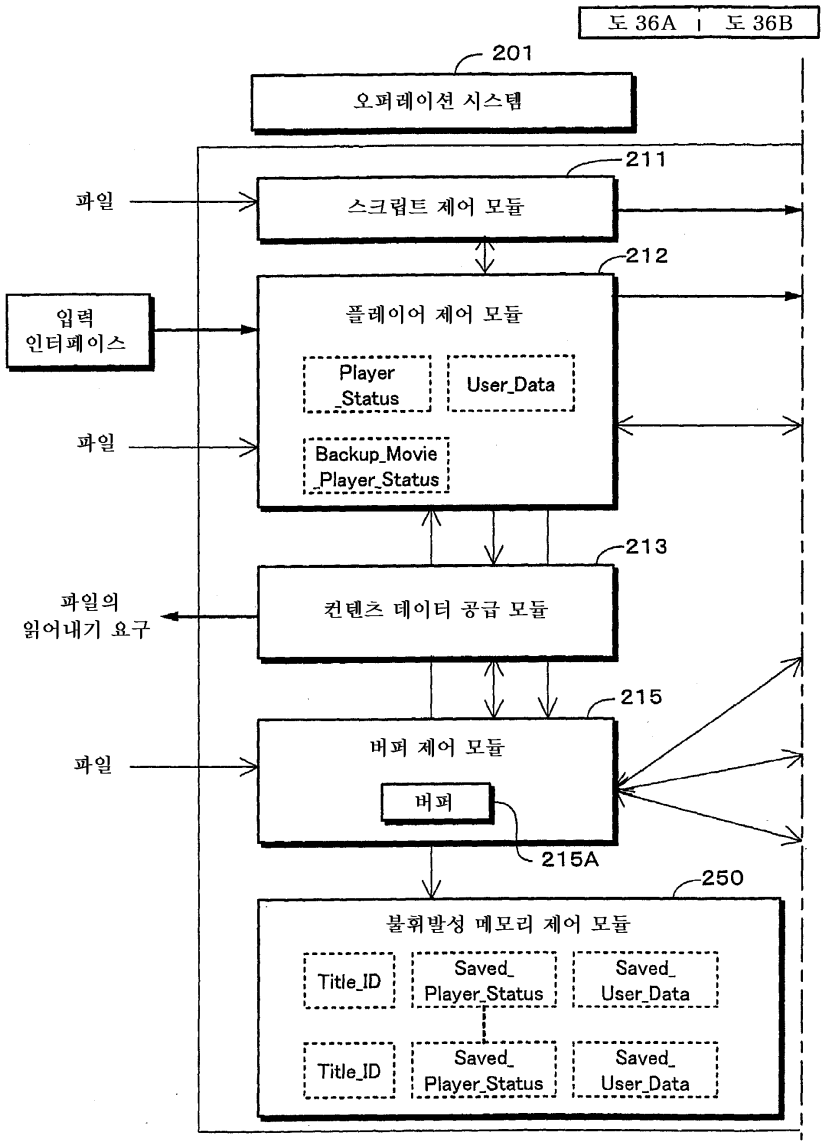
도면34

신택스	비트수	니모닉
EP_map(){		
reserved_for_word_alignment	8	bslbf
number_of_stream_id_entries	8	uimsbf
for (k=0; k<number_of_stream_id_entries; k++) {		
stream_id	8	bslbf
private_stream_id	8	bslbf
number_of_EP_entries	32	uimsbf
for (i=0; i<number_of_EP_entries; i++) {		
PTS_EP_start	32	uimsbf
RPN_EP_start	32	uimsbf
}		
}		
}		

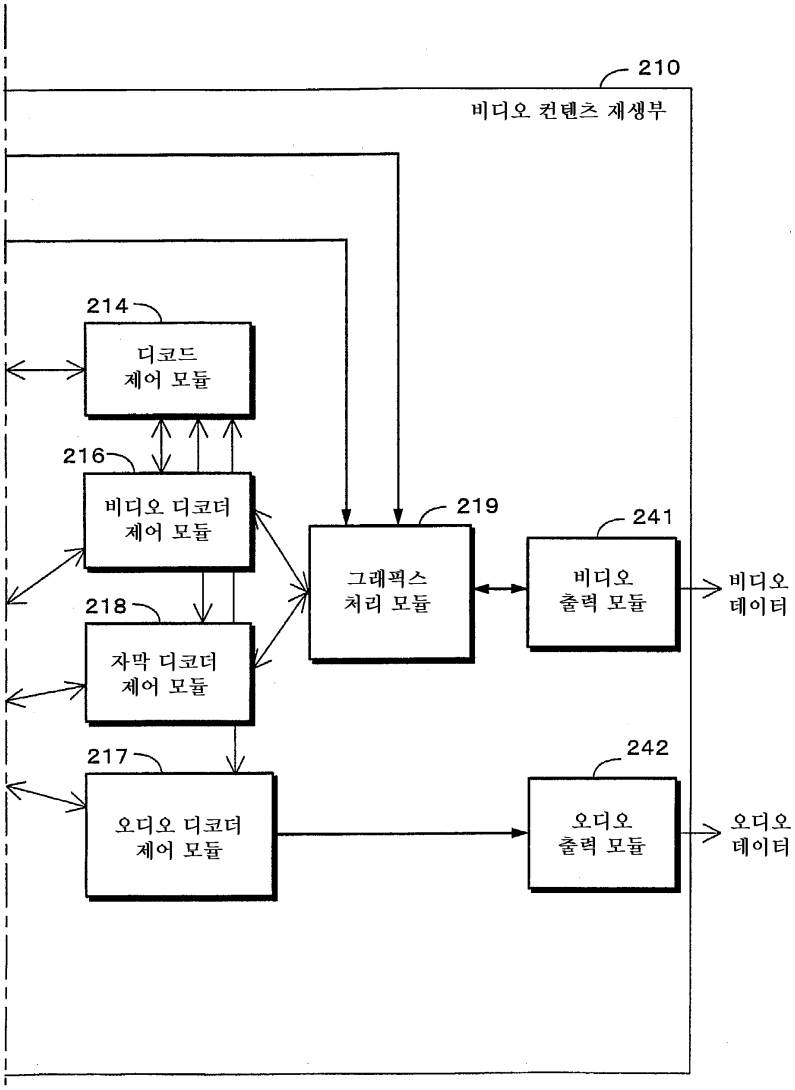
도면35



도면36A


도 36A | 도 36B
201
오퍼레이션 시스템
211
스크립트 제어 모듈
파일
212
플레이어 제어 모듈
입력
인터페이스
Player
Status
User_Data
파일
Backup_Movie
Player_Status
213
컨텐츠 데이터 공급 모듈
파일의
읽어내기 요구
215
버퍼 제어 모듈
파일
버퍼
215A
250
불휘발성 메모리 제어 모듈
Title_ID
Saved_
Player_Status
Saved_
User_Data
Title_ID
Saved_
Player_Status
Saved_
User_Data

도면36B


210
비디오 컨텐츠 재생부
214
디코드
제어 모듈
216
비디오 디코더
제어 모듈
218
자막 디코더
제어 모듈
217
오디오 디코더
제어 모듈
219
그래픽스
처리 모듈
241
비디오
출력 모듈
비디오
데이터
242
오디오
출력 모듈
오디오
데이터

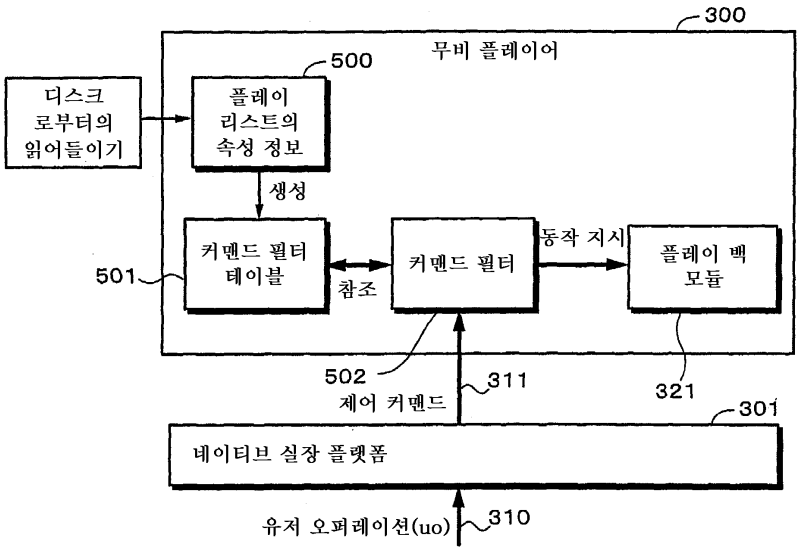
도면37

신택스	비트수	니모닉
"PLAYLIST.DAT" {		
name_length	8	uimsbf
name_string	8*255	bslbf
number_of_PlayLists	16	uimsbf
for(i=0; i<number_of_PlayLists; i++){		
PlayList(){ // A PlayList()		
PlayList_data_length	32	uimsbf
// 속성 정보		
reserved_for_word_alignment	11	bslbf
UOP_mask_mode	4	uimsbf
capture_enable_flag_PlayList	1	bslbf
PlayList_name_length	8	uimsbf
PlayList_name_string	8*255	bslbf
//		
number_of_PlayItems	16	uimsbf
for (i=0; i<number_of_PlayItems; i++) {		
PlayItem()		
}		
PlayListMark()		
}		
}		
}		

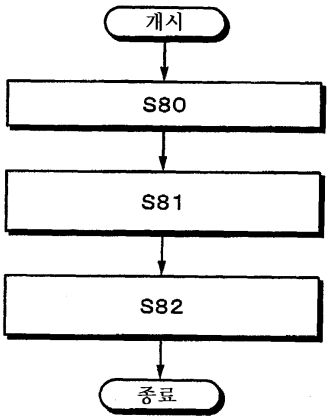
도면38

값	UOP_mask_mode
0x0	모든 유저 오퍼레이션을 허가
0x1	UOP 마스크 모드 1: 이 플레이 리스트를 재생 중에는, 유저 조작으로서는 스톱만이 유효하다. 그 외의 유저 조작은 행해져도, 플레이어는 무시한다. 이 플레이 리스트 내의 임의의 시각부터의 재생을 개시하는 유저 조작이 있었을 때에는, 플레이 리스트 선두로부터의 순방향 1배속 재생으로서 재생을 개시해야만 한다.
0x2	UOP 마스크 모드 2: 이 플레이 리스트를 재생 중에, 유저 조작에 의해 이 플레이 리스트의 재생을 중단하여 이 플레이 리스트의 마지막으로 점프하는 것은 금지된다. 단, 정지는 항상 허가된다. 빨리 감기나 빨리 되감기 등의 변속 재생 등의 유저 조작은 허가된다.
0x3-0xF	(예약)

도면39



도면40



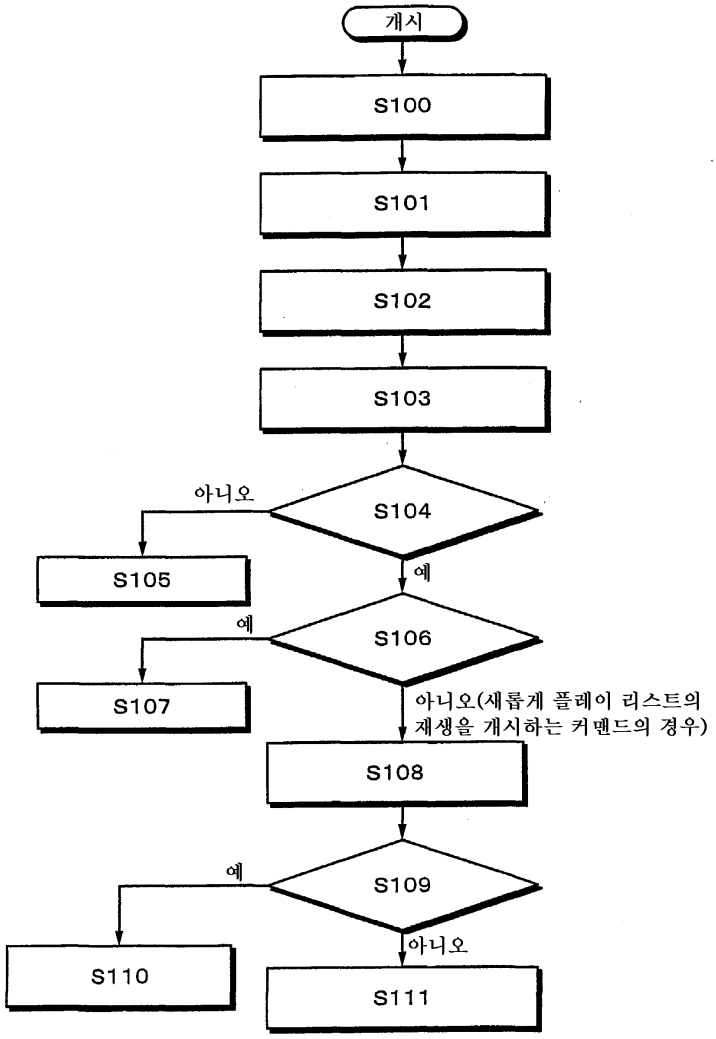
도면41

선두 이외로부터의 재생 개시	금지
허가되는 제어 커맨드	uo_stop()

도면42

선두 이외로부터의 재생 개시	허가
금지되는 제어 커맨드	uo_jumpToEnd()

도면43


개시
S100
S101
S102
S103
S104
아니오
S105
예
S106
예
S107
아니오(새롭게 플레이 리스트의
재생을 개시하는 커맨드의 경우)
S108
S109
예
S110
아니오
S111