

(19)대한민국특허청(KR)  
(12) 등록특허공보(B1)

(51) 。 Int. Cl. (45) 공고일자 2006년08월01일  
H04L 12/28 (2006.01) (11) 등록번호 10-0607217  
(24) 등록일자 2006년07월24일

(21) 출원번호	10-2000-7012297	(65) 공개번호	10-2001-0043328
(22) 출원일자	2000년11월04일	(43) 공개일자	2001년05월25일
번역문 제출일자	2000년11월04일		
(86) 국제출원번호	PCT/KR1999/000222	(87) 국제공개번호	WO 1999/57838
국제출원일자	1999년05월07일	국제공개일자	1999년11월11일

(81) 지정국      국내특허 : 알바니아, 아르메니아, 오스트리아, 오스트레일리아, 아제르바이잔, 보스니아 헤르체고비나, 바르바도스, 불가리아, 브라질, 벨라루스, 캐나다, 스위스, 리히텐슈타인, 중국, 쿠바, 체코, 독일, 덴마크, 에스토니아, 스페인, 핀란드, 영국, 그루지야, 헝가리, 이스라엘, 아이슬란드, 일본, 케냐, 키르기즈스탄, 북한, 대한민국, 카자흐스탄, 세인트루시아, 스리랑카, 리베이라, 레소토, 리투아니아, 룩셈부르크, 라트비아, 몰도바, 마다가스카르, 마케도니아공화국, 몽고, 말라위, 멕시코, 노르웨이, 뉴질랜드, 슬로베니아, 슬로바키아, 타지키스탄, 투르크멘, 터키, 트리니다드토바고, 우크라이나, 우간다, 우즈베키스탄, 베트남, 폴란드, 포르투갈, 루마니아, 러시아, 수단, 스웨덴, 싱가포르, 그라나다, 인도, 가나, 감비아, 크로아티아, 인도네시아, 시에라리온, 세르비아 앤 몬테네그로, 짐바브웨,

AP ARIPO특허 : 케냐, 레소토, 말라위, 수단, 스와질랜드, 우간다, 시에라리온, 가나, 감비아, 짐바브웨,

EA 유라시아특허 : 아르메니아, 아제르바이잔, 벨라루스, 키르기즈스탄, 카자흐스탄, 몰도바, 러시아, 타지키스탄, 투르크멘,

EP 유럽특허 : 오스트리아, 벨기에, 스위스, 리히텐슈타인, 독일, 덴마크, 스페인, 프랑스, 영국, 그리스, 아일랜드, 이탈리아, 룩셈부르크, 모나코, 네덜란드, 포르투갈, 스웨덴, 핀란드, 사이프러스,

OA OAPI특허 : 부르키나파소, 베닌, 중앙아프리카, 콩고, 코트디부아르, 카메룬, 가봉, 기니, 말리, 모리타니, 니제르, 세네갈, 차드, 토고, 기니 비사우,

(30) 우선권주장      60/084,578      1998년05월07일      미국(US)

(73) 특허권자      삼성전자주식회사  
경기도 수원시 영통구 매탄동 416

(72) 발명자      험플맨리차드  
미국캘리포니아94539프레몬트로우어빈트너스썬클343

왕동안  
미국캘리포니아95054산타클라라오크그로브드라이브#302440

(74) 대리인      리앤목특허법인

심사관 : 김대성

## (54) 네트워크내의 디바이스-디바이스간 명령 및 제어를 위한방법 및 시스템

### 요약

네트워크를 통하여 복수의 디바이스 중에서 명령 및 제어를 위한 방법 및 시스템은 제1 디바이스를 네트워크에 접속하는 단계; 제2 디바이스를 상기 네트워크에 접속하는 단계로서, 상기 제2 디바이스는 다른 네트워크 디바이스들에 의해 상기 제2 디바이스를 명령하고 제어하기 위하여 구조적인 포맷으로 어플리케이션 인터페이스 설명 데이터를 정하는 단계; 네트워크를 통하여 제1 디바이스에 어플리케이션 인터페이스 설명 데이터를 제공하는 단계; 및 제2 디바이스의 동작을 제어하기 위해 상기 어플리케이션 인터페이스 설명 데이터를 사용하여 네트워크를 통해 상기 제1 디바이스에서 상기 제2 디바이스로 제어 및 명령 데이터를 전송하는 단계를 포함한다.

### 대표도

도 7

### 명세서

#### 기술분야

본 발명은 네트워크 시스템에 관한 것으로, 보다 상세하게는 그것에 연결된 복수의 디바이스들을 구비한 홈 네트워크에 관한 것이다.

#### 배경기술

일반적으로 네트워크는 통신 링크와 통신 링크와 접속된 통신 능력을 가지는 다양한 디바이스들을 포함한다. 상기 디바이스들은 컴퓨터, 주변 장치, 라우터, 저장 장치, 그리고 프로세서와 통신 인터페이스를 구비한 제품을 포함한다. 네트워크의 일 실시 예로서 다양한 디바이스들이 상호 접속된 가정용 홈 네트워크를 들 수 있다. 보통의 가정은 퍼스널 컴퓨터 및 가정 내에서 전형적으로 발견되는 홈 디바이스들을 포함하는 여러가지 디바이스들을 포함할 수 있다. 용어로서의 "디바이스"는 전형적으로 로직컬 디바이스 또는 기능성 및 데이터를 교환하는 능력을 구비하는 다른 장치들을 포함하며, 모든 홈 디바이스들뿐만 아니라 범용 컴퓨터들을 또한 포함할 수 있다. 홈 디바이스는 보안 시스템, 극장 장비, TV, VCR, 스테레오 장비, 디지털 위성서비스(DSS:Digital Satellite Services)로도 알려진 직접 방송 위성 서비스(DBSS:Direct Broadcast Satellite Services), 살수 시스템, 조명 시스템, 전자 레인지, 식기 세척기, 오븐/스토브, 세탁기/건조기 및 자동차 내의 처리 시스템 등과 같은 전자 디바이스들을 포함한다.

일반적으로 홈 디바이스들은 집주인의 라이프 스타일과 생활수준을 향상시키는 일을 수행하기 위해 사용된다. 예를 들어, 식기 세척기는 더러운 식기들을 세척하는 일을 수행하고, 집주인이 손으로 식기를 세척해야 하는 것을 경감시켜 준다. VCR은 후에 집주인이 특정한 프로그램을 시청할 수 있도록 TV 프로그램을 녹화할 수 있다. 보안 시스템은 집주인의 귀중품을 보호하고, 원치 않는 침입에 의한 집주인의 공포를 감소시켜 줄 수 있다.

가정용 극장 장비와 같은 홈 디바이스들은 종종 다일의 공통 제어 유닛, 즉 원격 제어 장치를 사용하여 제어된다. 상기 단일의 공통 제어 유닛은 집주인이 단일의 인터페이스를 사용하여 여러가지 상이한 홈 디바이스들을 제어하고 명령할 수 있도록 한다. 따라서, 제조자들은 단일 인터페이스로부터 그들의 홈 디바이스들을 제어하고 명령하기 위한 제어 유닛을 발전시켜 왔다. 홈 디바이스들을 명령하고 제어하는 원격 제어 유닛의 사용과 관련된 단점은 각 홈 디바이스를 제어하고 명령하기 위한 정적인 명령 로직을 제공한다는 것이다. 따라서, 특정한 원격 제어 유닛은 필요한 제어 및 명령 로직을 포함하는 홈 디바이스들만을 제어하고 명령할 수 있다.

종래의 네트워크 시스템에서 사용자는 원격 제어 유닛 또는 디바이스 제어 패넬을 사용하여 명령들을 제공한다. 일단 사용자가 중지하면, 네트워크에서 자동 조작을 위한 명령을 제공하는 제어 유닛과 디바이스가 없게된다. 예를 들어, 홈 네트워크는 튜너 유닛 및 튜너가 없는 VCR을 포함할 수 있다. 사용자가 시간 지연 녹화를 위해 VCR에 프로그램 한 후에, 매카니

즘은 VCR을 위하여 이후에 자동적으로 튜너를 네트워크 상에 배치하고, VCR이 녹화할 수 있도록 비디오 정보를 제공하도록 튜너를 제어하는 것이 필요하다. 그와 같이 처음에 사용자가 제1 디바이스 세트를 제어하고 명령한 후에, 종래의 시스템은, 제2 디바이스 세트의 직접적인 사용자 제어 및 명령 없이 작업을 수행하기 위해 필요한 것으로서 제1 디바이스 세트가 네트워크에서 제2 디바이스 세트와 자동으로 통신하는 메카니즘을 제공하지 않는다.

따라서, 홈 네트워크에서 디바이스의 동적 중앙 제어 및 명령을 제공하는 방법과 시스템이 필요하다. 또한 사용자가 처음에 상호간에 통신하는 제1 디바이스 세트를 제어하고 명령하며, 제2 디바이스 세트에 대한 직접적인 사용자 제어 및 명령 없이 작업을 수행하기 위해 필요한 것으로서, 네트워크에서 제1 디바이스 세트가 제2 디바이스 세트와 자동으로 통신하는 능력을 제공하기 위한 방법 및 시스템이 필요하다. 또한 다양한 네트워크 디바이스들이 또 다른 다양한 네트워크 디바이스들을 자동으로 명령하고 제어하는 능력을 제공하기 위한 방법 및 시스템이 필요하다.

## 발명의 상세한 설명

본 발명은 상기 필요성들을 만족시킨다. 일 실시예에서 본 발명은 제1 디바이스를 네트워크에 접속하는 단계; 제2 디바이스를 상기 네트워크에 접속하는 단계로서, 상기 제2 디바이스는 다른 네트워크 디바이스들에 의해 상기 제2 디바이스를 명령하고 제어하기 위하여 구조적인 포맷으로 어플리케이션 인터페이스 설명 데이터를 정하는 단계; 네트워크를 통하여 제1 디바이스에 어플리케이션 인터페이스 설명 데이터를 제공하는 단계; 및 제2 디바이스의 동작을 제어하기 위해 상기 어플리케이션 인터페이스 설명 데이터를 사용하여 네트워크를 통해 상기 제1 디바이스에서 상기 제2 디바이스로 제어 및 명령 데이터를 전송하는 단계에 의해 네트워크를 통하여 다수의 디바이스들 사이의 명령과 제어를 위한 방법 및 시스템을 제공한다.

적어도 상기 어플리케이션 인터페이스 설명 데이터의 한 부분은 사용을 위해 제1 디바이스에 전송될 수 있거나, 또는 상기 제1 디바이스는 네트워크를 통하여 다른 디바이스들에 어플리케이션 인터페이스 설명 데이터를 쿼리할 수 있다. 더욱이, 어플리케이션 인터페이스 설명 데이터는 제1 디바이스가 액세스하기 위해 데이터 베이스에 저장될 수 있다. 상기 어플리케이션 인터페이스 설명 데이터는 상기 제1 홈 디바이스가 상기 제2 홈 디바이스의 동작을 제어하기 위해 원격 진행 호출 정보를 포함할 수 있다. 게다가 상기 어플리케이션 인터페이스 설명 데이터는 상기 제2 디바이스의 성능을 확인하기 위한 성능 데이터를 포함할 수 있다. 상기 어플리케이션 인터페이스 설명 데이터를 위한 구조적인 포맷은 XML 포맷을 포함할 수 있다.

바람직 하기로는, 다수의 디바이스들은 네트워크에 접속되어 있고, 각 디바이스는 네트워크에 접속된 하나 또는 그 이상의 다른 디바이스들에 의해 상기 구조적인 포맷에서 디바이스의 명령 및 제어를 위한 어플리케이션 인터페이스 설명 데이터를 포함한다. 이와 같은 경우에, 다수의 디바이스의 둘 또는 그 이상의 어플리케이션 인터페이스 설명 데이터는, 디바이스들의 동작을 제어하기 위해 둘 또는 그 이상의 디바이스 각각에 상응하는 어플리케이션 인터페이스 설명 데이터를 사용하여 네트워크를 통해 상기 제1 디바이스로부터 상기 둘 또는 그 이상의 디바이스들에 제어 및 명령 데이터를 전송할 수 있도록 하기 위해서, 네트워크를 통하여 제1 디바이스에 제공될 수 있다.

또 다른 관점에서, 본 발명은 사용자 인터페이스 데이터의 디스플레이가 가능하도록 하는 클라이언트 디바이스를 홈 네트워크에 접속하는 단계; 적어도 제1 홈 디바이스의 사용자 명령 및 제어를 위한 사용자 인터페이스를 정의하는 선택된 포맷에서 사용자 인터페이스 데이터를 포함하는 상기 제1 홈 디바이스를 상기 홈 네트워크에 접속하는 단계; 구조적인 포맷에서 제2 홈 디바이스의 디바이스 명령과 제어를 위한 어플리케이션 인터페이스 설명 데이터를 포함하는 상기 제2 홈 디바이스를 네트워크에 접속하는 단계; 클라이언트 디바이스에서 상기 제1 홈 디바이스의 사용자 인터페이스 데이터를 수신하는 단계; 상기 클라이언트 디바이스 상에서 상기 제1 홈 디바이스의 사용자 인터페이스 데이터에 의해 정의된 사용자 인터페이스를 디스플레이하는 단계; 상기 클라이언트 디바이스 상에 디스플레이된 사용자 인터페이스와 함께 상호 작용하는 사용자에게 응답하여 사용자로부터 사용자 입력을 받아들이는 단계; 및 서비스를 수행하기 위해 상기 어플리케이션 인터페이스 설명 데이터를 사용하여 상기 제1 및 제2 홈 디바이스가 상호간에 통신하도록 하는 사용자 입력에 기반하여 상기 클라이언트 디바이스로부터 상기 제1 홈 디바이스로 제어 및 명령 데이터를 전송하는 단계에 의해, 홈 네트워크를 통하여 서비스를 수행하기 위한 방법 및 장치를 제공한다.

그와 같이 하여, 상기 사용자는 상기 클라이언트 디바이스 상에서 디스플레이된 사용자 인터페이스로부터 제1 및 제2 홈 디바이스를 선택할 수 있다. 더욱이, 제1 홈 디바이스는 클라이언트 디바이스를 통하여 제1 홈 디바이스로의 사용자 입력에 기반을 둔 어플리케이션 인터페이스 설명 데이터를 사용하여 제2 홈 디바이스에 제어 및 명령 정보를 보냄으로써 제2 홈 디바이스를 제어할 수 있다. 상기 어플리케이션 인터페이스 설명 데이터는 상기 제2 홈 디바이스를 위한 성능 데이터를

포함할 수 있고, 상기 제1 홈 디바이스는 상기 제2 홈 디바이스의 어플리케이션 인터페이스 설명 데이터의 범위 내에서 성능 데이터를 쿼리 할 수 있고, 클라이언트 디바이스 상에 디스플레이된 제1 홈 디바이스의 사용자 인터페이스를 통하여 사용자에게 의해 제2 홈 디바이스의 명령과 제어를 허락하는 제1 디바이스 내의 사용자 인터페이스를 갱신할 수 있다.

둘 또는 그 이상의 홈 디바이스들은 네트워크에 접속될 수 있고, 각 홈 디바이스는 네트워크에 접속된 하나 또는 그 이상의 다른 홈 디바이스들에 의해 구조적인 포맷에서 홈 디바이스의 명령 및 제어를 위해 어플리케이션 인터페이스 설명 데이터를 저장할 수 있다. 둘 또는 그 이상의 각 디바이스들 내의 어플리케이션 인터페이스 설명 데이터는 각각의 홈 디바이스를 위한 성능 데이터를 포함할 수 있다. 이와 같은 경우에 상기 제1 홈 디바이스는 둘 또는 그 이상의 홈 디바이스들의 어플리케이션 인터페이스 데이터 범위 내에서 성능 데이터를 쿼리 할 수 있고, 클라이언트 디바이스 상에 디스플레이된 제1 홈 디바이스의 사용자 인터페이스를 통하여 사용자에게 의해 둘 또는 그 이상의 홈 디바이스들의 명령 및 제어를 허용하기 위해 제1 홈 디바이스 내의 사용자 인터페이스 데이터를 갱신할 수 있다.

이러한 본 발명의 특징, 관점 및 이점은 다음의 상세한 설명, 부가된 청구항, 및 첨부된 도면에 의해 더 잘 이해가 될 것이다.

### 도면의 간단한 설명

도 1은 본 발명의 제1 관점에 따른 네트워크의 구현을 위한 블록도이다.

도 2는 디바이스 제어 및 통신 개요의 제1 실시 예에 따른 도 1의 블록도이다.

도 3은 다수의 클라이언트와 서버 디바이스를 포함하는 본 발명에 따른 홈 네트워크 시스템에 대한 제1 실시 예를 보이는 블록도이다.

도 4는 도 3의 클라이언트 디바이스 및 서버 디바이스 구현의 제1 실시 예를 보이는 블록도이다.

도 5는 클라이언트 디바이스 구현의 제1 실시 예를 보이는 도면이다.

도 6은 서버 디바이스 구현의 제1 실시 예를 보이는 도면이다.

도 7은 상호간에 통신 및 제어를 가능하게 하는 네트워크 서버 디바이스의 제2 실시 예를 보이는 블록도이다.

도 8은 네트워크에서 소스 서버 디바이스, 싱크 서버 디바이스 및 클라이언트 디바이스를 포함하는 오디오/비디오 모델의 구조에 대한 제1 실시 예를 보이는 블록도이다.

도 9는 오디오/비디오 모델의 다른 실시 예를 보이는 도면이다.

도 10은 네트워크 디바이스를 위한 성능 데이터 테이블의 실시 예를 보이는 도면이다.

도 11은 네트워크 디바이스를 위한 특성 데이터 테이블의 실시 예를 보이는 도면이다.

도 12는 네트워크 디바이스들 사이에서 명령 메시지를 발생하는 빌딩 블록에 대한 제1 실시 예의 구성을 보이는 도면이다.

도 13은 명령 메시지를 발생하는 도 12에 대한 빌딩 블록의 다른 실시 예의 구성을 보이는 도면이다.

도 14는 네트워크 클라이언트 및 서버 디바이스들 사이에서 상호작용의 제3 실시 예를 보이는 도면이다.

도 15는 네트워크 디바이스 인터페이스의 API 확장을 정의하는 제1 실시예의 블록도이다.

도 16은 다른 서버 디바이스의 인터페이스 설명 문서를 액세스하는 서버 디바이스 어플리케이션을 위한 제1 실시예의 구조를 보이는 도면이다.

도 17은 제어기 서버 디바이스와 제어된 서버 디바이스 사이의 내부-디바이스 제어 구조의 다른 실시 예를 보이는 도면이다.

도 18은 네트워크 디바이스 사이의 API 레벨에 통신 스택에서 웹 표준 공통 미들웨어 계층을 제공하는 XML 프로토콜의 구현을 보이는 도면이다.

도 19는 서버 디바이스간 명령 제어 구조의 또 다른 구현을 보이는 도면이다.

도 20은 디바이스 인터페이스 라이브러리와 홈 디바이스를 위한 가전제품 데이터베이스 사이의 관계를 보이는 도면이다.

도 21은 디바이스 인터페이스 정의에 대한 계층적 형태의 구현을 보이는 도면이다.

도 22는 도 21의 디바이스 인터페이스 정의에서 계층들의 제1 일 실시 예를 보이는 도면이다.

도 23은 전송기 및 수신기 디바이스 사이에서 공통 전송 및 해석 절차를 보이는 도면이다.

도 24는 본 발명의 제1 관점에 따른 번역 서비스를 제공하기 위한 패킷 타입과 포맷의 부분적인 리스트 테이블의 제1 실시 예를 보인 도면이다.

### 실시예

일 관점에서, 본 발명은 홈 네트워크와 같은 네트워크에서 디바이스간 통신을 제공한다. 홈 디바이스들이 보다 지능적으로 되고 정보를 공유할 수 있게 되면서, 디바이스간 통신은 사용자가 상기 디바이스들의 정보 공유 성능을 이용할 수 있도록 네트워크 내의 디바이스들을 상호 접속할 수 있도록 허용한다. 그와 같은 것으로서, 디바이스간 통신은 사용자에게 네트워크에 연결된 디바이스들을 충분하고 유연하게 사용하는 능력을 부여함에 있어서 결정적인 역할을 수행한다.

도 1을 참조하면, 본 발명의 일 실시예에서, 네트워크(10)는 통신 링크(16)를 통하여 상호 연결된 적어도 하나의 클라이언트 디바이스(12) 및 적어도 하나의 서버 디바이스(14)를 포함한다. 통신 링크(16)는 다양하게 연결된 홈 디바이스들 사이에서 데이터를 전송하고 수신하기 위해 물리적 계층(매체)을 제공하는 1394 시리얼 버스를 포함할 수 있다. 상기 1394 시리얼 버스는 시간-다중화 오디오/비디오(A/V) 스트림 및 표준 IP(Internet Protocol) 통신을 양 쪽 모두 지원한다. 실시예에서, 홈 네트워크는 홈 네트워크를 위한 통신 계층으로서 IP 네트워크 계층을 사용한다. 그러나 다른 통신 프로토콜들이 홈 네트워크를 위해 통신을 제공하기 위해 사용될 수 있다.

각 클라이언트 디바이스(12)는 네트워크(10) 내의 하나 또는 그 이상의 서버 디바이스(14)와 통신할 수 있다. 더욱이, 각 서버 디바이스(14)는 네트워크(10)에서 하나 또는 그 이상의 다른 서버 디바이스들(14), 그리고 하나 또는 그 이상의 클라이언트 디바이스들(12)과 통신할 수 있다. 각 클라이언트 디바이스(12)는 사용자 입력을 수신하기 위한 마우스와 키보드와 같은 입력 장치를 사용자가 네트워크에 연결된 디바이스들과 상호 작용하도록 제어 사용자 인터페이스를 제공하는 디스플레이를 포함하는 사용자 통신 인터페이스를 포함할 수 있다. 사용자 인터페이스는 사용자에게 정보를 제공하기 위한 GUI(Graphical User Interface) 디스플레이 18을 포함할 수 있다. 도 2를 참조하면, 여기에 정의된 것과 같이, 각 서버 디바이스(14)는 사용자에게 제어 사용자 인터페이스를 제외한, 서비스를 제공하고, 각 클라이언트 디바이스(12)는 네트워크(10)와 상호 작용하는 사용자를 위하여 제어 사용자 인터페이스를 제공한다. 그와 같이 하여 클라이언트 디바이스들(12)만이 사용자와 직접적으로 상호 작용을 하고, 서버 디바이스들(14)은 클라이언트 디바이스(12) 및 서버 디바이스(14)와만 상호 작용을 한다. 예를 들어, 서비스는 MPEG 소싱/싱킹(Sourcing/Sinking) 및 디스플레이 서비스를 포함할 수 있다.

도 3은 다수의 클라이언트 디바이스(12) 및 다수의 서버 디바이스(14)를 포함하는 홈 네트워크(10)의 예를 보이는 블록도이다. 각 서버 디바이스(14)는 네트워크에서 사용자에게 서비스를 제공하기 위한 자원으로서 하드웨어를 포함할 수 있다. 더욱이 각 서버 디바이스(14)는 서버 하드웨어를 제어하기 위한 서버 또는 제어 프로그램(20)을 저장할 수 있고, 도 4에 도시된 바와 같이 제어 프로그램(20)과 함께 사용자 인터페이스를 위한 그래픽 제어 객체(GCO:Graphical Control Object) 사용자 인터페이스 설명서(22)를 포함할 수 있다.

제어 클라이언트 디바이스 12와 제어된 서버 디바이스(12) 사이의 제어를 위하여, 클라이언트 디바이스 12는, 예를 들어 네트워크를 통하여 서버 디바이스(14)로부터 클라이언트 디바이스(12)에 GCO(22)를 전송함으로써, 서버 디바이스(14)의 GCO(22)를 액세스한다. 그 다음에 클라이언트 디바이스(12)는 네트워크를 통하여 클라이언트 디바이스(12)로부터 서버 디바이스(14)의 제어 프로그램(20)과 통신을 수행하는 사용자를 위한 제어 사용자 인터페이스 GUI(18)을 창출하기 위하여 전송된 GCO(22)를 사용한다. 사용자는 적어도 클라이언트 디바이스(12)로부터 서버 디바이스(14)의 제어 프로그램(20)에 명령과 제어를 제공한다.



서버 디바이스 자체에서 각 서버 디바이스(14)의 GCO(22) 저장은 몇몇 서버 디바이스(14)와 함께 네트워크에서 클라이언트 디바이스(12)의 프로세싱과 저장 요구를 감소시킬 수 있다. 더욱이, 서버 디바이스(14)에서 GCO(22)를 저장하는 것은 각 서버 디바이스(14)가 사용자에게 그 자신의 GUI 모양과 느낌을 제공하도록 허락하고, 클라이언트 디바이스(12)에서 수정없이 GCO(22)의 수정 또는 갱신하는 것을 허용한다.

실시 예에서 클라이언트 디바이스(12)와 서버 디바이스(14) 사이의 명령 및 제어를 제공하는 도 4를 참조하면, 상기 클라이언트 디바이스(12)는 클라이언트 디바이스(12)에 저장되거나 원하는 서버 디바이스(14)로부터 네트워크를 통하여 클라이언트 디바이스(12)에 전송된 GCO(22)를 사용하여 GUI(18)를 디스플레이 하기 위한 렌더러(Renderer)(24)를 포함할 수 있다. 예를 들어, 초기의 디바이스 선택 상태에서, 상기 클라이언트 서버(12)는 네트워크를 통하여 적어도 한 개의 서버 디바이스(14)의 GCO(22)를 패치(Fetch) 할 수 있고, 상기 렌더러(24)는 상기 서버 디바이스(14)를 제어하기 위해 GCO(22)를 사용하여 GUI(18)를 디스플레이 한다. 바람직하게는, 상기 GUI(18)는 서버 디바이스(14)에서 커스텀화 되고, 서버 디바이스(14)를 제어하기 위해 만들어진 내장 명령 세트를 포함할 수 있다.

또한, 다양한 서버 디바이스 14의 GUI(18)는 예를 들어: (1) GUI(18)를 디스플레이 하는 클라이언트 디바이스 렌더러(24)를 위한 공통 GCO 모델 타입, (2) 다양한 서버 디바이스(14)로부터 클라이언트 디바이스(12)에 GCO(22)를 전송하기 위한 공통 통신 프로토콜, (3) 제어되는 특정 서버 디바이스(14)의 대응하는 서버 디바이스(14)의 제어 프로그램에서 만들어진 지식을 요구하지 않는 클라이언트 디바이스(12)로부터 대응되는 서버 디바이스(14)의 제어 프로그램(20)에 GUI 상호 작용을 위한 공통 통신 프로토콜과 같은 공동체를 포함할 수 있다.

여전히 도 4를 참조하면, 서버 디바이스(14)는 서비스를 제공하기 위해 서버 하드웨어를 제어하는 하나 또는 그 이상의 제어 프로그램(20)을 포함할 수 있다. 상기 서버 디바이스(14)의 GCO(22)로부터 GUI 인터페이스(18)는 제어 프로그램(20)에 인터페이스를 제공한다. 상기 서버 디바이스(14)는 요청된 서비스를 제공함에 있어 서버 디바이스(14) 및 서버 디바이스 하드웨어의 제어 상태를 지적하는 제어 상태 데이터(26)을 포함할 수 있다.

예를 들어, 제어 상태 데이터(26)는, VCR 서버 디바이스에서 녹화 동작을 수행하기 위해 설정되는 타이머 설정과 같이, 서버 디바이스(14)를 위해 GUI(18)에서 제어 정보의 상태를 포함할 수 있다. 제어 상태 데이터(26)는 제어된 서버 디바이스(14)에 저장되고, 서버 디바이스(14)의 사용자 제어를 위해, 제어 클라이언트 디바이스(12)에서 서버 디바이스(14)의 GUI(18)를 통하여 사용자에게 디스플레이 된다. 바람직하게는, 상기 서버 디바이스(14)의 GUI(18)를 디스플레이하기 위한 제어 클라이언트 디바이스(12)는 제어된 서버 디바이스(14)를 위한 제어 상태 데이터(26)의 지식을 유지하지 않는다.

각 서버 디바이스(14)는 하나 또는 그 이상의 클라이언트 디바이스들(12)에 의해 제어될 수 있다. 그러한 것으로서, 상기 서버 디바이스(14)에 저장된 제어 상태 데이터(26)은 각 제어 클라이언트 디바이스(12)에서 서버 디바이스(14)의 GUI(18)내의 정보 상태를 포함한다. 예를 들어, 사용자가 제1 클라이언트 디바이스(12)를 사용하여 서버 디바이스(14)를 제어할 때, 사용자 제어가 완성되면, 제1 클라이언트 디바이스(12)에서 서버 디바이스(14)의 GUI(18) 내의 정보는 서버 디바이스(14)에 의해 서버 디바이스(14)의 제어 상태 데이터(26)에 저장된다.

다른 한편으로, 사용자가 제1 클라이언트 디바이스(12)에서 서버 디바이스(14)의 GUI(18)와 상호 작용하는 동안, 상기 서버 디바이스(14)의 제어 상태 데이터(26)는 제1 클라이언트 디바이스(12)에서 서버 디바이스(14)의 GUI(18) 내의 정보로 갱신되고, 사용자 제어가 완성되면, 상기 제어 상태 데이터(26)은 서버 디바이스(14)에서 유지된다. 사용자가 제2 클라이언트 디바이스(12)를 사용하여 서버 디바이스(14)를 제어할 때, 상기 제어 상태 데이터(26)은 뒤따르는 제어를 위해 제2 클라이언트 디바이스(12)에서 서버 디바이스(14)의 GUI(18)를 통하여 사용자에게 이용할 수 있도록 만들어진다. 사용자는 또한 이후에 서버 디바이스(14)를 제어하기 위해 제1 클라이언트 디바이스(12)를 사용할 수 있고, 여기서 상기 제어 상태 데이터(26)는 뒤따르는 제어를 위해 제1 클라이언트 디바이스(12)에서 서버 디바이스(14)의 GUI(18)를 통하여 사용자에게 이용할 수 있도록 만들어진다. 상기 서버 디바이스(14)는 또한 클록(28)을 포함하거나, 이후에 기술된 것처럼 사용자로부터 입력된 시간 또는 클록에 기반을 둔 시간 지연 작용을 허락하기 위하여 현재 시간을 유지할 수 있다.

클라이언트 디바이스(12)와 서버 디바이스(14)는 이태면 DTV와 같이 하나의 유닛으로 물리적으로 함께 묶여질 수 있다. 그러한 경우에, 상기 서버 디바이스(14)는 서버 하드웨어를 제어하기 위한 제어 프로그램(20)을 포함하고, 상기 클라이언트 디바이스(12)는 적어도 서버 하드웨어의 제어 및 명령을 위해 제어 프로그램(20)에 제어 사용자 인터페이스를 제공한다. 도 5는 클라이언트 디바이스(12)의 예를 보이는 도면으로, (1) GUI 디스플레이를 위한 PDA(RemoteC), (2) GUI를 디스플레이하고, 오디오 및/또는 비디오 프로그램 스트림 목적지 서버로 구성된 싱크 서버를 포함하는 DTV(STB), (3) GUI를 디스플레이하고, 다중의 서비스를 제공하기 위해 적어도 하나의 서버 디바이스를 포함하는 PC를 포함할 수 있다. DTV 또는 PC 클라이언트 디바이스내의 하드웨어 혹은 실행 가능한 것들은 다른 클라이언트 디바이스에 의해서도 제어될

수 있다. 도 6은 서버 디바이스(14)의 예를 보이는 도면으로, (1) 소스 서버 디바이스로서의 DVDP SmartCard, (2) 싱크 서버 디바이스로서의 오디오 증폭기, (3) 소스 또는 싱크 서버 디바이스로서의 DVCR, (4) 원격 서버 디바이스를 관리하기 위한 관리 서버를 포함한다. 상기 관리 서버는 예를 들어 DSB-STB, 케이블 TV-STB, 또는 ATSC-STB를 포함할 수 있다. 이와 같은 디바이스들은 STB 내부 작업의 로컬 제어 또는 관리를 위한 관리 서버를 포함한다. 더욱이, 외부 네트워크를 통하여 액세스된 외부 서버들은 예를 들어, Video-on-Demand, Enhanced-TV 및 인터넷 상거래와 같은 서비스를 위해 로컬 클라이언트 디바이스에 의해 사용될 수 있다.

도 7을 참조하면, 두 개의 서버 디바이스(14) 사이의 통신 및 제어는 이들 사이의 명령어 및 제어 데이터를 통신하는 서버 디바이스(14)의 제어 프로그램(20)에 의해 수행된다. 서버 디바이스(14)는 네트워크를 통하여 하나 또는 그 이상의 다른 서버 디바이스(14)를 제어할 수 있다. 그리고, 서버 디바이스(14)는 하나 또는 그 이상의 서버 디바이스(14), 그리고 하나 또는 그 이상의 클라이언트 디바이스(12)에 의해 제어될 수 있다. 더욱이, 사용자는 서버 디바이스(14)의 제1 세트를 제어 및 명령하기 위해 클라이언트 디바이스(12)를 사용할 수 있고, 상기 서버 디바이스(14)의 제1 세트는, 사용자에게 서비스를 수행하기 위해 필수적인 것으로서, 사용자를 혼란하게 하지 않고 서버 디바이스 14의 제2 세트를 자동으로 명령하고 제어할 수 있다.

예를 들어, 자동 시간-지연 동작을 수행하기 위해, 사용자는 서버 디바이스(14)의 제1 세트를 제어하는 클라이언트 디바이스(12)에 "로그 온(Log On)" 하고, 원하는 서비스를 지정할 수 있다. 그리고 나서 사용자는 상기 클라이언트 디바이스(12)로부터 "로그 오프(Log Off)"한다. 서버 디바이스(14)의 제1 세트는 이들 자신들 사이에서 통신 및 제어를 수행하고, 그 이후에, 제1 세트에서 하나 또는 그 이상의 서버 디바이스가 사용자의 개입 없이 원하는 서비스를 총괄적으로 제공하기 위해 필수적인 것으로서 서버 디바이스(14)의 제2 세트를 자동적으로 제어한다.

도 7은 두 개의 서버 디바이스(14)가 서로 통신하고 제어하는 실시 예를 보이는 도면이다. 각 서버 디바이스(14)는 위에서 기술된 제어 프로그램(20), 클럭(28), 및 상술한 제어 상태 데이터(26)를 포함한다. 각 서버 디바이스(14)는 또한 서버 디바이스(14)가 클라이언트 디바이스(12)에 의해 직접 제어되도록 하는 GCO(22)를 포함할 수 있다. 그러나, GCO(22)는 클라이언트 디바이스(12)에 의해 직접 제어되지 않으며, 단지 다른 서버 디바이스(14)와 통신하는 서버 디바이스(14)에는 포함될 필요가 없다. 각 서버 디바이스(14)는 또한 명령어(CL:Command Language) 인터페이스(30) 및 명령어 라이브러리를 포함한다. 상기 명령어 라이브러리는 상기 서버 디바이스(14)가 그것의 서비스를 제공하기 위해 정보를 송신 및 수신하는데 사용하는 명령들을 포함한다. 그러나, 도 4 및 위에서 기술된 바와 같이 명령어는 사용자 제어를 위해서 반드시 필요한 것은 아니다.

도 8은 네트워크에서 소스 서버 디바이스(14), 싱크 서버 디바이스(14), 그리고 클라이언트 디바이스(12)를 포함하는 오디오/비디오(A/V) 모델의 예를 보여준다. 상기 소스 서버 디바이스(14)는 소스 서버 디바이스(14)의 데이터 스트림 소스 하드웨어(32)를 제어하기 위한 제어 프로그램(20)을 포함하고, 상기 싱크 서버 디바이스(14)는 싱크 서버 디바이스(14)의 데이터 스트림 싱크 하드웨어(34)를 제어하기 위한 제어 프로그램(20)을 포함한다. 예시적인 동작에서, 사용자는 데이터 스트림 소스 하드웨어(32)를 가동시키기 위해 소스 서버 디바이스(14)를 제어하고, 데이터 스트림 싱크 하드웨어(34)를 가동시키기 위해 싱크 서버 디바이스(14)를 제어하는 클라이언트 디바이스(12)를 사용한다. 데이터 스트림 소스 하드웨어(32)로부터 데이터 스트림 싱크 하드웨어(34)로의 데이터 전송을 초기화함에 있어서, 사용자는 상기 클라이언트 디바이스(12)를 포기할 수 있다. 다른 한편으로는, 사용자는 장래의 초기 데이터 전송을 프로그램 할 수 있고, 클라이언트 디바이스(12)를 포기할 수 있다. 이후로 소스 서버 디바이스(14)의 데이터 스트림 소스 하드웨어(32)와 싱크 서버 디바이스(14)의 데이터 스트림 싱크 하드웨어(34)는 사용자에게 의해 프로그램된 시간에서 자동으로 데이터 전송을 초기화한다.

예를 들어, 상기 데이터 스트림 소스 하드웨어 32는 직접 위성 방송(DBS:Direct Broadcast Satellite)와 같은 튜너-엑세스 디바이스를 포함할 수 있다. DBS는 케이블 텔레비전을 대신하는 다중-채널 대안이고, 위성으로부터 직접적으로 작은 위성 접시(지름이 3피트 18인치)에 케이블 같은 텔레비전 프로그래밍을 제공한다. DBS와 함께, 몇몇의 표준 아날로그 텔레비전 신호는, 단일 위성 트랜스폰더로 디지털적으로 압축되며, 공중의 지정된 장소에 위치한 위성 접시도 200개 이상의 채널이 수신할 수 있도록 허용한다. 상기 데이터 스트림 싱크 하드웨어(34)는 재생 시에 압축된 디지털 비디오 신호를 디코딩 할 수 있는 명령 및 디지털 비디오 카세트 레코더(DVCR)를 포함할 수 있다. 사용자는 DVCR을 위한 "시간-지연 녹화" 이벤트 데이터와 튜너-엑세스 디바이스를 위한 "시간-지연 선택 프로그램" 이벤트 데이터를 포함하는 명령 및 제어 데이터를 제공한다. 시간 지연 후에, 상기 튜너-엑세스 디바이스는 원하는 프로그램을 선택하고, 사용자로부터 추가적인 제어 조작이 없이도 프로그램 데이터를 수신 및 녹화하는 DVCR에 소스 프로그램 데이터를 제공한다.

도 9는 적어도 네트워크에서 소스 서버 디바이스(14) SERVER1, 싱크 서버 디바이스(14) SERVER2와 클라이언트 디바이스(12)를 포함하는 또 다른 A/V 모델을 보이는 도면이다. 상기 클라이언트 디바이스(12)는 서버 디바이스(14) SERVER1 및 SERVER2와 다른 서버 디바이스(14) 예를 들어 SERVER3 및 SERVER4(미도시)를 선택하고 제어하기 위해 사용자를

위한 선택 정보를 디스플레이 하는 사용자 인터페이스가 있는 세션 관리자(36)를 포함한다. 선택 정보는 서버 디바이스(14) SERVER1, SERVER2, SERVER3 및 SERVER4를 각각 선택하게 하는 사용자가 세션 관리자(36)에서 Serv1, Serv2, Serv3 및 Serv4에 할당된 아이콘 심볼들을 포함한다. 상기 소스 서버 디바이스(14) SERVER1은 DVCR을 포함할 수 있고, 상기 싱크 서버 디바이스(14) SERVER2는 1/2 DTV를 포함할 수 있다.

동작의 일 예에서, 서버 디바이스(14) SERVER1과 SERVER2의 선택에 있어서, 상기 클라이언트 디바이스(12)는 각 서버 디바이스(14)의 GCO(22)를 클라이언트 디바이스에 전송하고 각 서버 디바이스(14) SERVER1 및 SERVER2에 대응되는 GUI(18)를 디스플레이 한다. 사용자는 서비스를 위해 대응되는 서버 디바이스(14)에 명령과 제어를 제공하는 각 서버 디바이스(14)의 GUI(18)와 상호 작용할 수 있다. 각 서버 디바이스(14)는 혼자 또는 다른 서버 디바이스(14)와 조합하여 서비스를 제공할 수 있다. 더욱이, 상기 세션 관리자(36)는, 서비스를 수행하기 위해 대응되는 서버 디바이스(14)가 필요로 하는 것으로서, 상기 클라이언트 디바이스(12)에서 상기 서버 디바이스(14)의 GUI(18) 사이에 제어 상태 데이터(26)을 전송한다. 사용자 명령과 제어 정보를 기반으로 하여, 둘 이상의 서버 디바이스(14)들은 사용자 요구 서비스를 제공하기 위해 그들 사이에서 명령 및 제어 정보를 통신할 수 있다.

상기 세션 관리자(36)는 네트워크(10)에서 다양한 서버 디바이스(14)들에 의해 제공되는 유용한 홈 네트워크 서비스를 액세스하고 디스플레이 하는 기능을 가진 소프트웨어 에이전트를 포함할 수 있다. 상기 소프트웨어 에이전트는 부가적으로 네트워크(10)에서 다양한 서버 디바이스(14)의 성능을 정합시킬 수 있고, 호환 성능을 가지는 상기 서버 디바이스(14)만을 위한 선택 정보를 디스플레이 한다. 더욱이, 세션 관리자(36)는, 사용자가 서버 디바이스(14)에 의미 있는 명령 및 제어 정보를 제공하도록 돕기 위해 한 개의 서버 디바이스(14)의 GUI(18)에서 만들어진 선택과, 다른 서버 디바이스(18)의 GUI(18)에서 만들어진 선택을 정합시킬 수 있다.

동작의 또 다른 예에서, 세션 관리자 36은 네트워크를 검색하고, 네트워크에 접속된 서버 디바이스(14)를 발견하는 소프트웨어 에이전트를 실행한다. 상기 소프트웨어 에이전트는 또한 서버 디바이스(14)의 성능을 결정하기 위하여 각 서버 디바이스(14)에 저장된 성능 데이터를 액세스하고, 사용자에게 그 성능에 대한 정보를 제공한다. 그리고 나서 상기 세션 관리자(36)는 도 9에 도시된 바와 같이 서버 디바이스 SERVER1, SERVER2, SERVER3 및 SERVER4를 위한 선택 아이콘 Serv1, Serv2, Serv3 및 Serv4를 디스플레이 한다.

상기 세션 관리자(36)는 사용자가 4개의 모든 아이콘 중에서 선택할 수 있도록 하기 위해 초기에 모든 선택 아이콘 Serv1, Serv2, Serv3 및 Serv4를 인에이블 시킨다. 사용자가 Serv1 선택 아이콘 상에서 클릭하여 서버 디바이스 SERVER1을 선택한 후에, 상기 세션 관리자(36)는 서버 디바이스 SERVER3 및 SERVER4가 서버 디바이스 SERVER1과 성능 상 호환될 수 없다고 결정한다. 그러한 것으로서, 상기 세션 관리자(36)는 서버 디바이스 SERVER3 및 SERVER4를 위해 선택 아이콘 Serv3 및 Serv4를 각각 디스에이블 시킨다. 그리고 나서 사용자는 서버 디바이스 SERVER2를 명령 및 제어하기 위해 아이콘 Serv2 위에서 클릭할 수 있다.

사용자가 선택된 서버 디바이스(14)의 GUI(18)와 상호 작용하는 것과 같이, 사용자에게 의해 각 GUI(18)로 입력된 제어 및 명령 정보는 사용자에게 의해 뒤따른 서버 디바이스 선택에 영향을 미치는 부가적인 성능 정보를 제공한다. 예를 들어 만약에 VCR 서버 디바이스(14)가 선택되면, 다른 서버 디바이스(14)를 위한 선택 아이콘을 인에이블 또는 디스에이블 시킴에 있어서 세션 관리자(36)에 의한 뒤따른 조작이 재생 또는 녹화하는 사용자 결정에 의해 영향을 받는다.

네트워크에서 각 서버 디바이스(14)는 도 9에 도시된 서버 디바이스에 관계된 예를 통하여 위에서 논의된 바와 같이 하나 또는 그 이상의 서비스 성능을 가진다. 각 서비스 성능은 정보의 소싱(Sourcing) 또는 싱킹(Sinking)을 포함한다. 예를 들어, TV는 비디오 및 오디오 스트림을 수신하는 싱킹 성능을 가지고, VCR은 비디오 및 오디오 신호를 소스(전송) 및 싱크(수신)할 수 있고, PC는 비디오, 오디오 그리고 데이터를 전송 및 수신할 수 있다. 각 소싱 성능은 상보적이고 호환적인 싱킹 성능을 가진다. 이와 유사하게, 각 싱킹 성능도 상보적이고 호환적인 소싱 성능을 가진다. 예를 들어 하나의 디바이스에서 비디오 출력 성능은 또 다른 디바이스의 비디오 입력 성능에 의해 보완이 된다.

각 디바이스(14)는 네트워크 상에서 여러가지의 다른 서비스를 위해 소스 또는 싱크될 수 있기 때문에, 도 10의 예로 도시된 바와 같이 각 디바이스(14)는 성능 데이터 테이블(성능 테이블 1)을 저장한다. 테이블 1의 제1 열은 디바이스(14)의 서비스 성능을 확인하고, 제2 열은 제1 열에 대응되는 서비스를 위해 상기 디바이스(14)가 소스인지 싱크인지 여부를 확인한다. 성능 데이터 테이블 1을 사용하여 새로운 서비스는 더 오래된 디바이스와 호환성을 유지하는 동안 실행될 수 있다. 예를 들어, 만약 더 오래된 서비스와 호환되는 새로운 서비스가 개발된다면, 새로운 서비스 및 오래된 서비스 둘 다, 실행 디바이스는 오래된 서비스를 사용하는 오래된 디바이스들과 호환성을 유지함에 따라, 상기 새로운 서비스를 실행하는 디바이스를 위한 성능 데이터 테이블 1에 기입될 수 있다.



제1 실시 예에서, 디바이스 관리자는 소스 및 싱크 서비스 디바이스의 매칭 또는 비교를 수행한다. 예를 들어, 상기 디바이스 관리자는 호환성 매칭과 함께 다양한 디바이스(14) 및 지정된 디바이스(14)의 성능 및 특성을 비교하는 소프트웨어 에이전트로써 실행될 수 있다. 예를 들어, 서비스가 네트워크에 걸쳐 제1 디바이스(14)로부터 제2 디바이스(12)까지 미디어 스트림인 경우, 상기 디바이스 관리자는 사용자가 제1 디바이스(14)의 성능과 호환되는 제2 디바이스(14)를 분별력 있게 선택하도록 돕기 위해 제1 및 제2 디바이스(14)의 성능을 비교한다. 다음은 서버 디바이스(14)의 실시 예를 위한 서비스 성능 리스트의 예를 나타낸다.

Stream\_format\_video\_dv

Stream\_format\_video\_mpeg2tp

Stream\_format\_video\_dsstpt

Stream\_format\_video\_mpeg2pes

Stream\_format\_video\_mpeg210901-tp

도 11에 도시된 예에 의해, 각 디바이스(14)는 디바이스의 관련된 특성을 포함하는 특성 데이터 테이블(특성 테이블 2)을 더 저장한다. 이름과 값은 테이블 2에서 각 특성을 정의한다. 문자 길이가 테이블 2에 도시되어 있기는 하지만, 그것들은 요청되지 않는다. 상기 특성 데이터는 네트워크(10) 상에서 정보 처리 상호 운용을 용이하게 하고 디바이스 정보를 저장하는 다른 디바이스(14)에 유용하다. 예를 들어, 아래에 기술된 바와 같이 디바이스 페이지는 디바이스 이름을 저장하는 특성 테이블 2를 사용한다. 다른 필드는 필요할때 특성 데이터 테이블 2에 추가될 수 있다.

위에서 기술된 사용자-클라이언트 디바이스 제어 모델에서, 특성 데이터는 클라이언트 디바이스(12)에서 서버 디바이스(14)의 GUI 페이지 상에 디스플레이 될 수 있다. 다르게는, 상기 특성 데이터를 디스플레이 하기 위해 제2 레벨 디바이스 정보 홈페이지가 사용될 수 있다. 더욱이, 텍스트 또는 XML(Extensible Markup Language) 파일의 형태로의 특성 데이터가 소프트웨어 에이전트에 의해 액세스될 수 있다. 디바이스-디바이스 제어 모델을 위해, 제어된 디바이스를 위한 특성 데이터는 디바이스 인터페이스 및 어플리케이션 인터페이스에 저장된다.

특성 테이블 2에서 상기 디바이스 위치 특성 필드는 각 디바이스(14)를 위한 위치 또는 그룹을 저장하기 위해 사용된다. 디바이스 타입 특성 필드는 VCR, DVD, DTV, 캠코더, PC, 보안 시스템 등과 같은 특정한 디바이스(14)를 위한 디바이스 타입을 기입한다. 상기 디바이스 타입 특성 필드는, 만약 디바이스 자체가 디바이스 아이콘을 공급하지 않는다면, 디바이스 페이지에서 디바이스를 표현하는 디폴트 디바이스 아이콘을 선택하기 위해 사용된다. 특성 테이블 2는 디폴트 소스 및 디폴트 싱크 특성 필드를 위한 다중의 기입을 포함할 수 있다. 그와 같은 각각의 기입은 디바이스(14)에 의해 처리되는 각 데이터 타입을 위한 다른 디폴트 소스 또는 싱크 디바이스(14)를 표현한다.

바람직하게, 성능 및 특성 데이터는 계층 언어를 사용하여 구조화된 데이터로 패키징 된다. 이것은 GCO 전송 및 서버 디바이스-서버 디바이스 제어와 같은 다른 목적으로 사용된 성능과 특성을 회복시키는 공통 방법을 제공한다. 예를 들어, 특성 데이터는 다음의 구조화된 데이터 포맷을 포함할 수 있다.

<DEVICEATTRIBUTES>

<ATTRIBUTE name=DeviceManufacturer value="Samsung Inc.">

<ATTRIBUTE name=ManufacturerURL value=www.samsung.com>

<ATTRIBUTE name=ManufacturerIcon value="logo.gif">

<ATTRIBUTE name=DeviceName value="Samsung DSS">

<ATTRIBUTE name=DeviceModel value="SCH1900">

<ATTRIBUTE name=DeviceType value=DDS>

<ATTRIBUTE name=DeviceLocation value="Livingroom">

<ATTRIBUTE name=DeviceIcon value="device.gif">

<ATTRIBUTE name=DeviceAddress value=105.144.30.17>

</DEVICEATTRIBUTES>

일 예로써, 성능 데이터는 다음의 구조화된 포맷을 포함할 수 있다.

<DEVICECAPABILITIES>

<CAPABILITY type=MPEG2 value=Source>

<CAPABILITY type=MPEG2 value=Sink>

<CAPABILITY type=MPEG3 value=Source>

<CAPABILITY type=MPEG3 value=Sink>

</DEVICECAPABILITIES>

어플리케이션 인터페이스 언어는 다른 서버 디바이스(13)가, 서버 디바이스-서버 디바이스를 포함하여, 디바이스-디바이스 제어를 수행하도록 사용된다. 상기 어플리케이션 인터페이스 언어는 명령어를 포함하고, 아래에 기재된 바와 같이 XML을 사용하여 기술될 수 있다. 하나의 서버 디바이스(14)의 제어 프로그램(20)은, GUI(18)를 사용하지 않거나 또는 사용자의 개입없이, 네트워크를 통하여 또 다른 서버 디바이스(14)의 제어 프로그램(20)을 원격으로 제어한다. 디바이스-디바이스의 제어의 예는 자동 조작이다. 사용자는 초기에 원하는 서비스를 위해 클라이언트 디바이스(12)를 통한 제어를 제공하고, 이어서 둘 이상의 서버 디바이스(14)들이 서비스를 제공하기 위해 사용자의 상호작용 없이도 서로간에 자동으로 통신하고 제어한다.

도 12 및 도 13을 참조하면, 바람직하게 표준 어플리케이션 인터페이스 언어가 다양한 서버 디바이스(14)내의 다양한 제어 프로그램(20) 사이에서 정보처리 상호 운용을 허락하도록 사용된다. 일 실시 예에서, 상기 표준 어플리케이션 인터페이스 언어는 다음과 같은 구성 블록인 (1) 서비스 기능 데이터 베이스에서와 같은 서비스의 기능적 규격 블록(40), (2)메시지 요소가 구성되는 블록(42), (3)산업 표준 포맷 블록(44), (4) 메시지 압축 블록 46 및 구조화된 메시지 데이터를 출력하는 메시지 문자열 구성 블록(48)들을 포함한다.

도 12는 명령 메시지 발생 기능을 수행하는 구성 블록의 구성 예를 보이는 도면이다. 각 메시지 아이템은 서비스의 기능적 규격으로 구성되고, 메시지 아이템에 대해 산업 표준 압축 형태 (Hex) 라벨을 선택함으로써 표준화된다. 메시지 아이템과 같은 그룹은 완전한 명령 문자열을 만들기 위해 조합된다. CAL 및 AV/C와 같이 존재하는 명령어는 도 12에 도시된 바와 같이 동작한다. 그러나, 그러한 명령어 메카니즘은 물리적인 인터페이스 상에서 물리적 장치에 대하여 2진 코드 또는 16진 코드 메시지 및 시스템 작동을 규정하고 하드웨어 규정에 근거한다. 따라서, 그러한 명령어는, 제어기 디바이스(14)에서 하나의 제어 프로그램(20)은 네트워크 상의 제어된 디바이스(14)내의 또 다른 제어 프로그램(20)을 지정하고 제어하는 어플리케이션 레벨 즉, 소프트웨어 레벨에서, 제어 시스템 규격이 이름, 주소, 디바이스 성능 발견, 통신 언어 그리고 명령 메시지를 포함하는 제어 메카니즘에 기반을 둔 네트워크 계층에 대해 덜 바람직할 수 있다. 상기 제어 메카니즘은 컴퓨터와 같은 다중도, 다중 어플리케이션 디바이스뿐만 아니라 DVCR과 같은 장치를 포함하는 디지털 장치와 같은 디바이스에 대해 더욱 적합하다.

도 13은 명령 메시지 발생 기능을 수행하는 도 12의 구성 블록의 바람직한 예를 보이는 도면이다. 도 13에서, 산업 표준화 포맷(44)와 메시지 압축(46)의 위치들이 도 12와 다르다. 다수의 텍스트 표준화 양식은 완전한 메시지를 만드는 기능적 규격 서비스(40)으로부터 선택된다. 나중에 상기 메시지는 프로토콜 스택의 하부 계층에 의해 압축될 수 있다. 도 13은 가전 제품(CE)에 대해 서비스 또는 디바이스 명령 및 제어를 수행하는 방법을 나타낸다. 메시지 구성은 XML 표준화 선택스에 의해 정의될 수 있고, 메시지 압축은 HTTP와 같은 또 다른 프로토콜 계층에 의해 수행될 수 있다. 명령 인터페이스 언어는

하위 하드웨어 레벨보다는 제어 프로그램(20) 인터페이스 레벨에서 사용된다. 그러한 것으로서, 네트워크 프로토콜 스택은 상기 언어내의 명령에 의해 관리되고, 각 제어기 디바이스(14)와 제어된 디바이스(14)는 그들 사이의 메시지 전송을 위해 네트워크의 통합된 구성으로써 보여질 수 있다.

도 14를 설명하면, 클라이언트 디바이스(12)와 서버 디바이스(14) 사이의 세 가지 상호 작용의 세 가지 실시 예들을 보여준다. 제1 실시 예 "A"에서, 사용자는 원격 서비스 어플리케이션 "S"와 통신한다. 상기 사용자는 사용자 인터페이스로서 클라이언트 디바이스에서, 브라우저를 사용하고, 여기서 상기 브라우저는 서비스 어플리케이션 "S"에서 서비스 프로그램 20을 제어하고 HTML(Hyper Text Markup Language) 또는 XML 포맷으로 응답을 수신한다. XML 기반의 비동기 명령 메시지 포스팅을 받아들이는 브라우저를 구비한 두 번째 서버가 포함된다. 예를들어, DVCR에 있어서 두 번째 서버(14)는 "VCR FAILED : TAPE BROKE"와 같은 명령 메시지를 받아들이 수 있다. 브라우저를 포함하는 소프트웨어 에이전트는 사용자에 의한 나중의 주의와 DVCR의 제어를 위해 브라우저의 GUI내에 사용자를 위한 명령 메시지를 디스플레이 되도록 이용된다. 바람직하기로는, XML 기반의 클라이언트 디바이스(12)는 서버 디바이스 대 서버 디바이스 명령과 제어를 위해 다른 곳에서 초기화된 명령에 응답하는 HTTP1.1 서버 성능을 포함한다.

제2 실시 예 "B"에서, 사용자는 소프트웨어 클라이언트 제어 프로그램(50)에 의해 대체된다. 상기 소프트웨어 클라이언트 제어 프로그램(50)은 XML 기반의 서비스 어플리케이션 "S"로의 명령 포스팅(Postings)을 생성하고, 회신되는 XML 명령 포스팅을 수신한다. 그리고, 제3 실시 예 "C"에서, 상기 소프트웨어 클라이언트 제어 프로그램(50)은, 명령과 응답이 두 제어 프로그램(20) 사이에서 교환되는 제어 프로그램(20)과 같은 어플리케이션에 의해 대체된다. 이 점에 있어서, 실시 예 "B"는 무용(Null) 서비스를 가지는 실시 예 "C"의 특별한 경우이다.

XML에 기반을 둔 어플리케이션 인터페이스 언어는 월드 와이드 웹(World wide web) 혹은 인터넷에 의해 인에이블 되는 디바이스 또는 서비스를 위해 제1 서버 디바이스(14)와 제2 서버 디바이스(14)(디바이스-대-디바이스 또는 서비스-대-서비스)사이를 제어하기 위해 사용된다. 어플리케이션 인터페이스 언어는 웹 표준, 미들웨어 계층에 기반을 둔다. 일 실시 예에서, 디바이스-디바이스 제어는 네트워크(10)에서 또 다른 서버 디바이스(14)로부터 하나의 서버 디바이스(14)에서 제어 프로그램(20) 또는 어플리케이션을 원격으로 제어하는 것을 포함한다. 그러한 것으로서, 제어 프로그램(20)에 대한 인터페이스(API)는 API 확장을 사용하는 네트워크를 통하여 이용가능하게 된다. 바람직하기로는, 상기 API 확장은 전체의 정보처리 상호 운용을 제공하기 위하여 XML을 기반으로 한 인터페이스와 같은 표준 포맷을 사용한다.

이제, 도 15를 설명하면, 그것은 네트워크 내에서 통신하는 서비스 A로써 디자인된 제1 어플리케이션 A와 B로써 디자인된 제2 어플리케이션 B를 위한 API 확장의 블록도가 보여진다. 예를 들어 서비스 A는 네트워크에서 제1 서버 디바이스 A를 위한 제어 프로그램일 수 있고, 서비스 B는 네트워크에서 제2 서버 디바이스 B를 위한 제어 프로그램일 수 있다. 서버 디바이스 B는 서버 디바이스 A에 명령을 전송한다. 예를 들어, 제1 및 제2 서비스 디바이스 A 및 B는 CE 디바이스를 포함할 수 있다.

서비스 A를 위한 API 확장을 참조하면, 제1 최상위 블록(52)는 CE 디바이스를 설명하는 영어 단어를 사용하여 CE 대상 및 방법의 종합적인 정의 또는 데이터 베이스를 제공한다. 상기 종합적인 정의 또는 데이터 베이스는 C, XML 또는 대상 및 그들 각자의 방법들을 표현할 수 있는 다른 포맷으로도 존재할 수 있다. XML을 사용하는 종합적인 정의 또는 데이터 베이스는 XCE 정의로 불리운다. 제2 블록(54)는 인터페이스 데이터 타입 정의 INTERFACE.DTD로 디자인된 모든 디바이스(14)를 위해 XML 형식으로 API를 표현하는 포맷을 제공한다.

수단(Tool) A로 디자인된 소프트웨어 에이전트는 서비스 A를 위한 XCE 정의의 서브세트를 사용하고, XML 형태 문서를 생성하는 서비스 A를 위한 인터페이스 데이터 타입 INTERFACE.DTD, INTERFACE-A.XML를 사용한다. 상기 문서 INTERFACE-A.XML은 서비스 A를 위한 문서 타입 정의 INTERFACE.DTD에 따라 서비스 A에 의해 지원되는 대상 및 방법을 설명한다. 다른 데이터 타입 정의는 또한 INTERFACE-A.XML 문서를 작성하기 위해 사용될 수 있다.

소프트웨어 수단 A는 또한 네트워크 인터페이스 상에서, 서비스 B로부터 예를 들어 C에서 프로그램된, 서비스 A까지 XML 메시지로부터 변환된 룩-업 테이블(56) 및 실행할 수 있는 2진수로 컴파일된 룩-업 테이블(56)을 생성한다. 바람직하기로는, 룩-업 테이블(56)은 컴파일 시에 작성되고, 여기서 런-타임 동안, 서비스 B로부터 입력되는 XML 형태의 메소드(Method) 메시지들(명령들)이 서비스 A를 위해 컴파일된 어플리케이션 C 코드에 의해 작성된 API 형식으로 변환된다. 상기 룩-업 테이블(56)은 XML 대상 방법 요청 서비스로 A를 위한 장치 고유의 요청으로 변환하는 런-타임 번역을 제공한다. 룩-업 테이블(56)은 서비스 A를 위해 서버 디바이스 A 상에서 국부적인 실행을 위한 디바이스의 제어 프로그램(20)과 함께 컴파일된다.

INTERFACE-A.XML은 수신된 메시지에서 에러가 발생했을 때 정당성 체크를 위해 서비스 A에 의해 사용될 수 있다. INTERFACE-A.XML은 또한 서비스 A와 통신하기 전에 서비스 A를 위한 메시지 포맷을 결정하는 서비스 B와 같은 외부 어플리케이션에 의해 사용될 수 있다. 더욱이, 만약에 서비스 B로부터 서비스 A로의 메시지가 에러를 발생시키면, 서비스 B는 에러를 진단하기 위해 INTERFACE-A.XML 문서를 액세스할 수 있다.

서비스 B에 대한 API 확장을 참조하면, 상기 제1 블록(58)은 위의 서비스 A를 위해 XCE 정의와 같은 CE 객체의 종합적인 정의 또는 데이터 베이스를 제공한다. 다음 블록(60)은 서비스 A를 위한 API와 같은 원격의 API 서비스 또는 디바이스에 XML 형태 메소드(명령) 호출을 위한 언어 정의를 제공한다. 상기 언어 정의는 네트워크 상에서 객체와의 상호 작용을 설명하는 문서 타입 정의 방법 요청 CALL.DTD 이다.

수단 B로 디자인된 소프트웨어 에이전트는 명령을 서비스 B를 위해 컴파일된 C 프로그램 코드를 XML 형식의 메소드 요청으로 변환하는 룩업테이블을 작성하기 위해 적어도 서비스 B를 위한 XCE 규정 내의 객체 및 메소드의 서브 세트를 이용한다. 그와 같은 것으로서, 예를들어, 룩-업 테이블(62)는 서비스 B(예를 들어, "재생(Play)")에 의해 야기된 방법과 서비스 A에서 네트워크 인터페이스를 가로질러 방법 호출을 수행하는 XML 문서 또는 메시지 사이에서 변환을 제공한다. 소프트웨어 수단 B에 의해 사용되는 XCE 정의의 서브세트는 네트워크의 사용 확장 및 본질에 의존한다. 예를 들어, 상기 서브세트는 홈 네트워크 상에서 모든 가능한 서비스의 포괄적 혹은 제한된 사용을 제공하기 위해 선택될 수 있다.

따라서, 상기 API 확장은 XML을 사용하는 네트워크 상에서 다양한 디바이스 사이에서 통신을 제공한다. 위의 예에서, 서비스 B를 위한 제어 프로그램(20)은 API에서 방법 호출을 생성하고, API 호출은 내부-디바이스 통신을 위한 웹/인터넷 표준 XML에 부합하도록 XML 형태로 변환된다. XML 메소드 호출(메시지)은 네트워크를 통하여 서비스 A에 전송되고, 서비스 A는 네트워크 인터페이스로부터의 XML 방법 호출을 서비스 A를 위한 프로그램 코드 API 정의로 재변환한다. 상기 변환 및 재변환은 다른 디바이스 사이에서 이진 호환성을 다르게 요구하는 프로그램 코드 API를 가지는 네트워크내의 다양한 디바이스들을 위한 웹/인터넷 호환성을 제공한다. 도 15에서의 블록도를 사용하는 XML 인터페이스 블록의 실시 예가 아래에 도시된다.

-----

interface.dtd

rules for describing an object interface in xml

-----

<!ELEMENT parameter #PCDATA>

<!ATTLIST parameter valueCDATA #REQUIRED>

<!ELEMENT method (#PCDATA, (parameter)+)>

<!ELEMENT object (#PCDATA)

<!ELEMENT call (object, (method)+)>

-----

interface.h

example object interface in c

-----

/\* object \*/

```
typedef struct stream {
```

```
int id;
```

```
};
```

```
/* method */
```

```
void StreamPlay (int id, int speed);
```

```
void StreamStop (int id);
```

```
-----
```

```
interface.xml
```

```
the same object in xml using ruels of interface.dtd
```

```
-----
```

```
<object>Stream
```

```
<method>Play
```

```
<parameter type="int">id</parameter>
```

```
<parameter type="int">speed</parameter>
```

```
</method>
```

```
<method>Stop
```

```
<parameter type="int">id</parameter>
```

```
</method>
```

```
<object>
```

```
-----
```

```
call.dtd
```

```
rules for describing a c function call in xml
```

```
-----
```

```
<!ELEMENT parameter #PCDATA>
```

```
<!ATTLIST parameter TypeCDATA #REQUIRED>
```

```
<!ELEMENT method (#PCDATA, (parameter)+ )>
```

```
<!ELEMENT method (#PCDATA, (method)+ )>
```

controller.c

example controller command in c

```
StreamPlay (0x1ae, 500);
```

call.xml

the same command in xml using call.dtd

```
<!-- example to play a stream -->
```

&lt;call&gt;

&lt;object&gt;stream&lt;/object&gt;

&lt;method&gt;Play&lt;/method&gt;

```
<parameter value="500">speed</parameter>
```

더욱이, 위에서는 상술된 이용 가능한 서비스의 설명 문서인 INTERFACE.XML의 설명 문서를 작성하도록 사용된 인터페이스 정의 INTERFACE.DTD 및 CALL.DTD의 예를 보여준다. 상기 CALL.DTD 정의는 XML RPC(Remote Procedure Call) 또는 XMLRPC 메시지와 같은 방법 호출 또는 함수 호출 메시지를 생성하기 위한 규정 세트를 포함한다. 상기 CALL.DTD 정의는 제어기 서비스(14)의 출력 인터페이스를 설명한다. 홈 네트워크에서, 예를 들어 INTERFACE.XML은 홈 네트워크 상에서 이용 가능한 서비스를 표현한다. 이용 가능한 서비스는 CE 영역에서 전체 서비스의 서브세트이다.

OTR(One-Touch-Record) 개요에서, 사용자는 위성 STB와 같은 튜너-액세스-디바이스의 제어를 한다. 사용자는 프로그램 리스트를 나타내는 그래픽적인 사용자 인터페이스 표현과 같은 EPG(Electronic Program Guide)를 사용하여 튜닝을 제어한다. OTR 녹화는 사용자가 시간 지연된 녹화를 위해 VCR을 프로그램하기 위해 VCR 그래픽 사용자 인터페이스를 액세스하지 않아도 녹화를 수행하도록 EPG로부터 장래 프로그램의 선택을 포함하는 서비스를 제공한다. OTR은 VCR



의 제어를 자동화한다. 아래는 OPR.XML에서 예시적인 동작 제어 리스트이다 : (1) StreamOpen=위성 STB로부터 네트워크로 출력된 선택된 프로그램 스트림을 재생; OTR을 위해 이 제어는 STB 디바이스에서 국부적이다; (2) StorageOpen= 저장 서비스 열기; 및 (3) StorageRecord= 네트워크를 통해 VCR에 녹화 명령 전송.

-----  
call.dtd

rules for describing a c function call in xml

-----  
<!ELEMENT parameter #PCDATA>

<!ATTLIST parameter valueCDATA #REQUIRED>

<!ELEMENT method (#PCDATA, (parameter)+ )>

<!ELEMENT object (#PCDATA)>

<!ELEMENT call (object,method)>

-----  
interface.dtd

example for describing an object interface in xml

-----  
<!ELEMENT parameter #PCDATA>

<!ATTLIST parameter valueCDATA #REQUIRED>

<!ELEMENT method (#PCDATA, (parameter)+ )>

<!ELEMENT object (#PCDATA, method+ )>

-----  
interface.xml

this document describes various CE services offered -a subset of the whole CE space.

-----  
<?xml version="1.0"?>

<!DOCTYPE interface SYSTEM "interface.dtd">

<object>Stream

<method>Open

```

<parameter type="int">id</parameter>

<parameter type="int">channel</parameter>

</method>

<method>Close

<parameter type="int">id</parameter>

</method>

</object>

<object>Control

<method>Set

<parameter type="int">id</parameter>

<parameter type="int">level</parameter>

</method>

</object>

<object>Storage

<method>Open

<parameter type="int">id</parameter>

<parameter type="int">channel</parameter>

</method>

<method>Record

<parameter type="int">id</parameter>

</method>

<method>Play

<parameter type="int">id</parameter>

<parameter type="int">speed</parameter>

</method>

<method>Stop

<parameter type="int">id</parameter>

```

```

</method>

<method>Close

<parameter type="int">id</parameter>

</method>

</object>

<object>Display

<method>Open

<parameter type="int">id</parameter>

<parameter type="int">channel</parameter>

</method>

<method>Render

<parameter type="int">id</parameter>

</method>

<method>Blank

<parameter type="int">id</parameter>

</method>

<method>Control

<parameter type="int">id</parameter>

<parameter type="int">cid</parameter>

<parameter type="int">level</parameter>

</method>

<method>Close

<parameter type="int">id</parameter>

</method>

</object>

<!-- -----
otr.xml

```

an xml representation of one touch record

c representation:

StreamOpen (100,2);/\* play a stream (pushed by satellite feed \*/

StorageOpen (24,2);/\* open a storage service \*/

StorageRecord (24);/\* record the stream \*/

-----  
<?xml version="1.0"?>

<!DOCTYPE interface SYSTEM "call.dtd">

<call>

<object>stream</object>

<method>open</method>

<parameter value="100">id</parameter>

<parameter value="2">channel</parameter>

</call>

<call>

<object>storage</object>

<method>open</method>

<parameter value="100">id</parameter>

<parameter value="2">channel</parameter>

</call>

<call>

<object>storage</object>

<method>Record</method>

<parameter value="100">id</parameter>

</call>

도 15에 관계하여 위에서 논의된 것과 같이, 제1 디바이스 B는 A의 디바이스 성능과 API 인터페이스의 세부 사항들을 조사하기 위해 제2 디바이스 A의 INTERFACE.XML 문서를 액세스하고, 제2 디바이스 A의 지지된 기능 및 명령 항목을 결정할 수 있다. 특별하게, 상기 제1 디바이스 B는 제1 디바이스 B와 제2 디바이스 A에 의해 지원된 오버랩핑(Overlapping)

, 그리고 따라서 이용 가능한 메소드를 결정할 수 있다. 도 16은 어플리케이션 B를 포함하는 제1 서버 디바이스 B가 어플리케이션 A를 포함하는 제2 서버 디바이스 A의 INTERFACE-A.XML 문서를 액세스하는 예를 보여준다. 상기 제1 서버 디바이스 B는 제2 서버 디바이스 A에서 INTERFACE-A.XML 문서와 비교하기 위한 INTERFACE-B.XML를 포함한다

하나의 시나리오에서, 상기 제1 서버 디바이스 B는 네트워크에서 제2 서버 디바이스 A를 제어하고자 한다. 상기 제2 디바이스 A의 INTERFACE-A.XML 문서는 제2 서버 디바이스 A로부터 제1 서버 디바이스 B에 전송되고, 제2 서버 디바이스 A의 성능 및 API 인터페이스 방법을 질문하는 어플리케이션 B에 의해 사용된다. 이것은 제1 서버 디바이스 B가 XML 원격 절차 호출 XMLRPC를 사용하여 제2 서버 디바이스 A를 제어하는 것을 허락한다. 다른 시나리오에서, 상기 제1 서버 디바이스 B는 적어도 한번 제2 서버 디바이스 A와 통신을 시도하고 통신 확인이 실패한 후에, 위의 단계를 수행한다. 또 다른 시나리오에, 상기 제1 서버 디바이스 B는 제1 서버 디바이스 B에 INTERFACE-A.XML 문서를 전송하지 않고 원격으로 제2 서버 디바이스 A내의 INTERFACE-A.XML에게 쿼리한다.

INTERFACE-A.XML 문서의 항목을 조사함에 있어서, 상기 제1 서버 디바이스 B는 위에 기술된 것처럼, XML 포맷에서 제2 서버 디바이스 A에 전송하기 위한 명령을 창출할 수 있다. 일반적으로 상기 제1 서버 디바이스 B는, 위에서 기술된 것처럼 제1 및 제2 서버 디바이스 B 및 A에 의해 사용된 XCE 정의의 서브셋으로 오버랩 하는 INTERFACE-A.XML의 적어도 일부 항목을 해석할 수 있다. 만약 상기 제1 서버 디바이스 B가 INTERFACE-A.XML 문서의 일부 항목의 해석이 불가능하다면, 제1 서버 디바이스 B는 그 부분을 무시하거나, 더 아래에 기술된 것처럼 번역에 의해 그 부분을 설명하도록 도와주는 어플리케이션을 패치(Fetch) 할 수 있다.

도 17을 참조하면, 제어기 서버 디바이스(14) 및 제어된 서버 디바이스(14) 사이의 디바이스-디바이스 또는 내부 디바이스 제어하는 다른 예가 도시된다. 상기 제어기 디바이스(14)는 제어기 어플리케이션 E를 포함하고 제어된 디바이스(14)는 실행 가능한 어플리케이션 C를 포함한다. 제어된 디바이스(14)는 INTERFACE-A.XML, 어플리케이션 C의 어플리케이션 인터페이스 설명을 더 포함한다. 어플리케이션 E는 제어된 디바이스(14)의 성능 및 제어된 서버 디바이스(14)의 API 인터페이스 방법을 쿼리하기 위하여 제어된 디바이스(14) 내의 어플리케이션 인터페이스 설명 A를 액세스한다. 다음에 어플리케이션 E는 제어된 디바이스(14)의 하드웨어 또는 서비스 D를 제어하기 위해 XML 원격 절차 호출을 사용하여 어플리케이션 C를 명령하고 제어한다. 스케줄러(Scheduler) 디바이스는 VCR의 시간-지연-녹화 제어기와 같이 날짜가 되면 구동하는 제어기 디바이스 14의 한 경우일 수 있다.

제1 예에서, 상기 어플리케이션 E는 네트워크를 통한 원격 조회에 의해 어플리케이션 인터페이스 설명 A를 액세스한다. 제2 예에서, 상기 어플리케이션 E는 제어된 디바이스(14)에서 제어기 디바이스(14)로 어플리케이션 인터페이스 설명 A의 복사본(Copy)를 전송함으로써 어플리케이션 인터페이스 설명 A를 액세스한다. 다음에 어플리케이션 E는 국부적으로 인터페이스 설명 A에 쿼리한다. 제3 예에서, 어플리케이션 인터페이스 설명 A는 인터페이스 설명을 위해 라이브러리 공간을 제공하는 라이브러리 디바이스(64)에 전송되고, 어플리케이션 E는 라이브러리에서 인터페이스 설명 A를 원격으로 쿼리한다. 상기 라이브러리 디바이스(64)는 직접적인 제어 동작 및 응답이 사용가능한 관련된 어플리케이션의 주소(URL)를 저장한다.

도 18을 참조하면, XML 프로토콜은 네트워크에서 다양한 디바이스(14)의 제어 프로그램(20) 사이의 API 레벨에서 통신 스택(66)에 있어서 웹 표준 공통 미들웨어 계층을 제공한다. 각 디바이스(14)에서 통신 스택의 최 상부 어플리케이션은 네트워크를 통하여 통신 메시지를 송수신하고, 디바이스를 위한 디바이스 하드웨어 또는 서비스 소프트웨어를 국부적으로 제어하는 디바이스 스택으로 소프트웨어 계층들과 통신한다.

XML 층 OUT(68)로 디자인된 제1 XML 층 API는 메시지를 송신하기 위해 사용되고, XML 층 IN(70)으로 디자인된 제2 XML 층 API는 메시지를 수신하기 위해 사용된다. XCE 정의와 메소드 호출의 XML 정의는 즉, 위에서 기술된 문서 타입 정의 CALL.DTD는, XML 층 OUT(68)을 생성하도록 사용된다. 더욱이 XCE 정의 및 메소드 호출을 위한 XML 정의는 즉, 위에서 기술된 문서 타입 정의 INTERFACE.DTD는, XML 층 IN(70)을 생성하도록 사용된다. 예를 들어 제어기 어플리케이션은 XML 층 OUT(68)을 사용하고, 제어된 어플리케이션은 XML 층 IN(70)을 사용한다.

도 19를 참조하면, 서버 디바이스-서버 디바이스 명령 및 제어 구조의 또 다른 실시 예를 보여준다. XML-기반 제어 구조는 웹과 인터넷에 의해 구동되는 디바이스들 또는 서비스들을 위한 디바이스-디바이스(서비스-대-서비스)를 위해 사용된다. 제1 디바이스 A는 XML 명령 메시지를 사용하는 네트워크를 통하여 제2 디바이스 B에서 제어 프로그램(20)을 원격으로 제어할 수 있다. 각 디바이스에 대한 인터페이스는 상기 디바이스 내에서 어플리케이션에 대한 인터페이스를 포함하고 XML 포맷으로 서술된다. 더 아래에 기술된 것처럼, 상기 인터페이스들은 네트워크를 통하여 다른 디바이스에 의한 정정 및 설명을 위해 미들웨어 계층 상에서 확장되고 사용가능하게 될 수 있다.

각각의 서버 디바이스 A 및 B는 네트워크를 통하여 다른 서버 디바이스를 제어하고, 네트워크를 통하여 다른 서버 디바이스에 의해 제어되기 위한 하드웨어와 소프트웨어를 포함한다. 도 19에서, 홈 네트워크 디바이스 A는 제어기 디바이스 또는 모듈이고, 홈 네트워크 디바이스 B는 제어된 디바이스 또는 모듈이다. 각각의 디바이스 A 및 B는 인터페이스 문서 INTERFACE.XML 및 문서 타입 정의 INTERFACE.DTD로 구성된 국부적인 디바이스 XML 인터페이스(72)를 포함한다. INTERFACE.XML 문서는 대응되는 디바이스 14에 의해 지원되는 객체, 메소드 및 파라미터의 설명을 포함한다. 상기 INTERFACE.DTD 문서는, 위에서 기술된 것처럼, 디바이스의 XML 인터페이스에서 구체적인 정당성 체크를 위해 사용될 수 있다.

각각의 디바이스 A 및 B는 또한 XML 인터페이스 및 XMLRPC 명령과 같은 XML 메시지를 파싱하고 확인하기 위한 프로그램 코드로 구성된 XML 파서(Parser)(74)를 포함한다. 상기 XML 파서(74)는 도 18과 관계하여 위에 기술된 상기 XML 계층 IN(70)과 유사하다. 더욱이, 각각의 디바이스 A 및 B는 XMLRPC의 메소드명과 파라미터들을 재생하기 위해 XMLRPC 메시지에서 송출하는 메소드명과 파라미터들을 인코딩하기 위한 XMLRPC 인코더 및 입력되는 XMLRPC 메시지를 디코딩하기 위한 XMLRPC 디코더를 포함한다. XMLRPC 코덱(76)은 디바이스 XML 인터페이스(72) 및 디바이스-디바이스 제어 구조로부터 독립되고, 그것으로써 디바이스-디바이스 제어 구조의 다른 특징을 변화시키지 않고 다른 XMLRPC 포맷의 사용을 허용한다.

프로그램 코드로 구성된 인터페이스 패처(Fetcher)는 또 다른 디바이스 또는 홈 네트워크 인터페이스 라이브러리(80)로부터 직접적으로 또 다른 디바이스의 디바이스 인터페이스를 패치하기 위해 각각의 디바이스 A 및 B에 의해 사용된다. 디바이스(14)가 제어기 디바이스이면, 제어기 디바이스(14)내의 제어기 어플리케이션 프로그램 코드(82)는, XML 파서(74), 인터페이스 패처(78) 및 XMLRPC 코덱(76)과 같은 제어기 디바이스(14)내의 소프트웨어 및 하드웨어를 통제함으로써, 네트워크를 통하여 다른 디바이스(14)의 명령 및 제어에 영향을 준다. 디바이스(14)가 제어된 디바이스이면, 제어된 디바이스(14)내의 제어된 어플리케이션 프로그램 코드(84)는 다른 디바이스(14)에 의해 제어될 디바이스(14)를 위해 디바이스(14) 내의 소프트웨어 및 하드웨어를 통제한다. 각각의 디바이스 A 및 B내의 홈 네트워크 디바이스 웹 서버(86)는 네트워크 상의 디바이스들 사이의 통신을 관리한다. 각 디바이스 A 및 B 내의 고유 록업 테이블(88)에 대한 XML은 XMLRPC 메시지(예를 들어 방법 이름, 구성요소 이름 및 타입)를 디바이스의 고유 인터페이스(예를 들어 고유 방법 이름, 구성요소 및 타입)로 변환하는 제어된 어플리케이션(84)에 의해 사용된다. 상기 테이블(88)은, XML 메시지 내에서 메소드 및 파라미터의 이름과 디바이스의 고유 인터페이스가 같은 경우에는 사용되지 않는다.

각각의 디바이스 A 및 B는 하나 이상의 취급자(Handler)(90)를 더 포함하고, 각 취급자(90)은 제어된 어플리케이션(84) 내부로부터 특정 디바이스 기능의 고유 구현으로의 포인터를 포함한다. 대부분의 디바이스에서 디바이스 기능의 고유 구현은 실행시간에서의 이진 코드를 포함한다. 이진 코드는 예를 들어 C 및 자바를 포함하여, 컴파일 시간에 고 수준의 언어로부터 생성될 수 있다. 그러한 것으로, 가전제품 제조자는 존재하는 취급자 및 기능 구현에 영향을 미치지 않고 새로운 기능을 위한 더 나은 취급자(90)를 추가할 수 있다. 각각의 디바이스 A 및 B에서 하드웨어 서비스(92)는 디바이스 기능의 고유 구현을 포함한다. 각각의 디바이스 A 및 B내의 또한 디바이스 기능의 고유 구현으로 API를 구비하는 고유 인터페이스(94)를 포함한다.

더욱이, 홈 네트워크 대상 요청 브로커(HNORB:Home Network Object Request Broker)(79) 및 인터페이스 라이브러리(IL:Interface Library)와 같은 네트워크 대상 요청 브로커는 홈 네트워크(10)를 위한 미들웨어 계층(98)을 제공한다. 도 19에 도시된 바와 같이, 상기 미들웨어 계층(98)은 제3 디바이스(96) 또는 분리된 제어 허브에 위치될 수 있다. 상기 HNORB(79)는 네트워크 10에 접속된 다른 디바이스(14)의 존재를 발견하는 하나의 디바이스(14)에 의해 사용되기 위한 소프트웨어 에이전트를 포함한다. 상기 HNORB 소프트웨어 에이전트는 디바이스 이름을 명칭의 계층적인 트리구조로 조직하고, 디바이스 인터페이스들을 상기 검색 가능한 인터페이스 라이브러리로 조직하며, 그리고 인터페이스 정보를 요청하는 디바이스에게 디바이스 인터페이스를 제공한다.

HNORB(79) 및 IL(80)으로 구성된 미들웨어 계층은 지정된 국부적인 홈 네트워크(10)의 외부로부터 액세스될 수 있는 선택된 홈 디바이스와 같이 인터넷으로 직접 연결될 수 있다. 하나의 국부적인 홈 네트워크 내의 미들웨어 계층(98)은 두 개의 홈 네트워크(10)를 구비하는 통합된 네트워크를 제공하기 위하여 인터넷을 통하여 다른 국부적인 홈 네트워크에서 미들웨어 계층(98)에 접속될 수 있다. 이러한 경우에, 적당한 스트림 부호화를 가지는 인증된 사용자는 비디오를 재생하고 TV로 그 것을 보기 위해 사용자의 제2 가정내의 TV로부터 사용자의 제1 가정내의 DVD 체인저(Changer)를 액세스할 수 있다.

상기 인터페이스 라이브러리(80)를 사용하기 위해서, 적어도 하나의 HNORB&IL은 국부적인 홈 네트워크(10) 상에서 동작 중이어야 한다. 하나 이상의 HNORB&IL이 또한 이용될 수 있다. 예를 들어, 케이블 모뎀, 몇몇 DTV, 그리고 중앙 가정



허브는 모두 그들 자신의 HNORB&IL 소프트웨어 에이전트를 가질 수 있다. HNORB&IL의 위치를 정하기 위해 디바이스 (14)는 국부적인 홈 네트워크를 통하여 방송 메시지를 전송한다. 디바이스(14)에서 응답한 제1 HNORB&IL은 디바이스 (14)에 의해 사용된다. 일단 HNORB&IL이 위치되면, 디바이스(14) 및 HNORB&IL은 등록, 인터페이스 요청 및 패치를 위한 지점간 TCP(Transmission Control Protocol) 또는 UDP(User Datagram Protocol)과 디바이스 특업 서비스를 확인할 수 있다. 만약 UDP 프로토콜을 이용할 수 없으면, TCP 프로토콜이 IEEE 1394와 같은 고 대역폭 접속을 위해 사용될 수 있다. HTTP-기반의 XMLRPC는 HNORB&IL 통신을 위한 디바이스를 위해 사용될 수 있다. 예를 들어, 디바이스 14는 하나 이상의 구성요소로서 디바이스 인터페이스를 패스하기 위해 HNORB의 "레지스터" 메소드를 원격으로 호출할 수 있고, 또는 XMLRPC 호출은 XMLRPC 응답 또는 되돌아오는 값으로써 IL로부터 부분적 또는 전체적인 디바이스 인터페이스를 재생할 수 있다.

앞서 진술한 바와 같이, 하나 이상의 HNORB&IL은 동시에 국부적인 홈 네트워크(10)에서 동작할 수 있고, 거기서 각 HNORB&IL은 사용가능한 디바이스의 서브셋을 인식할 수 있으며, 하나의 HNORB&IL은 그것이 발견할 수 없는 디바이스 14를 위치시키기 위해 다른 HNORB&IL과 통신할 수 있다. 하나의 국부적인 홈 네트워크(10) 상의 다중 HNORB&IL은, UDP 및 TCP와 같은, 방송 메시지를 사용함으로써 자동으로 상호간에 위치할 수 있다. 이러한 경우에, 다중 HNORB&IL은 다수의 인터페이스 라이브러리(80)가 분산 인터페이스 라이브러리를 구성하는 동안 분산 대상 요청 브로커를 구성한다. 고장 허용 오차를 제공하기 위해, 만약 하나의 HNORB&IL가 갑자기 종료해야 한다면, 이 HNORB&IL와 함께 등록된 모든 디바이스들은 통지를 받게 되고, 상기 디바이스들은 다른 유용한 HNORB&IL와 함께 자동으로 등록할 수 있다.

각 디바이스 인터페이스는 조합되고, 일관되고, 단일한 논리적인 이름을 가지고 있다. 다른 디바이스들은, 상기 디바이스의 위치 또는 실제 네트워크 주소가 변환된 후 조차도, 디바이스를 인식하고 액세스하는 상기 일관되고, 단일한 논리적인 이름을 사용할 수 있다. 논리적인 이름의 맵핑(Mapping)과 실제 디바이스 주소의 맵핑은 HNORB에서 명명 서비스를 위한 소프트웨어 에이전트에 의해 취급된다. 바람직하기로는, 표준화된 명명 방법이 사용된다. 보다 바람직하기로는, 디바이스 이름을 계층적 트리 구조로 조작하기 위해 계층적인 명명 구조가 사용된다. 이 계층적인 구조는 파일 시스템에서와 유사하게 "/"를 사용하여 표현될 수 있다. 그 구조는, 가정/MPEG2/TV와 같은 다른 서비스 타입에 의해; 또는 가정/거실/VCR과 같은 다른 위치에 의한 것과 같이, 다른 방법으로 생성될 수 있다. 여러가지 명명 트리는 실행과 능률을 위해 공존할 수 있다.

도 19에서의 제어기 서버 디바이스 A 및 제어기 서버 디바이스 B 사이의 명령 및 제어 예에서, 상기 미들웨어 계층(98)은 제3 디바이스(96)에 있거나, 또는 분리된 중심 허브에 있을 수 있다. 어두운 블록은 도 19에서 묘사된, 특수한 명령 및 제어 프로세서를 위해 사용된 디바이스 구성요소를 보여준다. 동작 시나리오에서, 디바이스 A 및 B가 네트워크를 통하여 사용가능하게 되고 액세스 가능하게 된 후에, 각 디바이스는 중앙 HNORB 및 IL 미들웨어 층(98)에게 그 자체 및 그 것의 XML 인터페이스를 기입/제출한다. 만약 중앙 HNORB 및 IL 미들웨어 계층이 사용가능하지 않다면, 각 디바이스는 그 자체를 알리기 위해 국부적인 홈 네트워크를 통하여 메시지를 방송한다.

상기 디바이스 A의 제어기 어플리케이션(82)은 제어된 디바이스 B의 전체 또는 부분의 디바이스 인터페이스에 쿼리를 시도한다. 만약 인터페이스 라이브러리(80)가 유용하지 않다면, 제어기 디바이스 A는 네트워크를 통하여 디바이스 B에 요청을 처음에 전송하고, 그리고 나서 디바이스 B로부터 디바이스 B의 XML 인터페이스를 수신함으로써, 제어기 디바이스 B로부터 직접적으로 제어된 디바이스 B의 디바이스 인터페이스를 요청하고 패치할 수 있다. 그러나, 만약 인터페이스 라이브러리(80)가 사용 가능하다면, 제어기 디바이스 A는 인터페이스 라이브러리(80)로부터 제어된 디바이스 B의 전체 또는 부분의 디바이스 인터페이스를 요청할 수 있다. HNORB의 소프트웨어 에이전트는 인터페이스 라이브러리(80)로부터 상기 디바이스 B의 XML 디바이스 인터페이스를 획득하고, 제어기 디바이스 A에 그것을 회송한다.

일단 제어기 디바이스 A가 제어된 디바이스 B의 XML 디바이스 인터페이스를 수신하면, 디바이스 A의 제어기 어플리케이션은 디바이스 B의 디바이스 인터페이스를 분석(Parse)하고 번역하기 위해 디바이스 A의 XML 파서 74를 사용한다. 그리고 나서 디바이스 A의 XMLRPC 코덱 76은 파싱 결과를 사용하여 바람직한 XMLRPC 명령 메시지를 생성한다. 상기 XMLRPC 명령 메시지는 네트워크를 통하여 제어된 디바이스 B에 전송된다. 상기 XMLRPC 명령 메시지를 수신하면, 디바이스 B의 제어된 어플리케이션 84는 수신된 XML 명령 메시지를 분석하고 번역하기 위해 디바이스 B의 XML 파서 74를 이용한다. 그리고 나서 디바이스 B의 XMLRPC 코덱 76은, 요청된 서비스를 수행하기 위해 디바이스 B 기능을 위한 메소드명 및 파라미터들을 포함하는 명령 메시지에서 방법 호출 정보를 획득하기 위해 파싱 결과를 디코딩한다.

디바이스 B의 제어된 어플리케이션(84)은 다음에 디바이스 B의 고유 인터페이스를 통하여 디바이스 B의 고유 기능 실행을 액세스하고 내보내기 위해 디바이스 B에서 고유 특업 테이블(88) 및 취급자(90)에 대해 XML을 사용한다. 만약 함수가 어떤 응답 또는 되돌아오는 값을 생성하면, 상기 응답 또는 되돌아오는 값은 XML 또는 XMLRPC 메시지로 인코딩 되고,

제어기 디바이스 A로 전송된다. 더욱이, 미들웨어 계층 HNORB 및 IL은 제어된 디바이스 B에 대한 참조를 제어된 디바이스 A에 제공하고, 여기서 디바이스 A는 국부적인 디바이스 A 고유 기능에 대한 호출과 마찬가지로 디바이스 B 고유 기능에 대한 원격 호출을 발생한다.

바람직하기로는, 모든 디바이스가 네트워크를 통하여 RPC 호출을 설명하고 디코드 할 수 있도록, 표준 XMLRPC 포맷이 사용된다. 제어된 디바이스(14)의 디바이스 인터페이스가 제어기 디바이스(14)에 의해 쿼리되고 조사될 수 있기 때문에, 바람직하게는 능력을 향상시키기 위해 충분한 디바이스 인터페이스 정보를 가지는 단순화된 XMLRPC 포맷이 사용된다. 다음 예는 OTR(One Touch Record) 및 TDR(Time Delayed Record) 동작을 위해 XMLRPC를 호출하는 두 개의 가능한 포맷을 보여준다.

#### 예 I :

XML RPC 호출, 상세 태그 및 인터페이스 정보를 포함하는 포맷예:

##### 1. OTR 호출 예:

```
<?xml version="1.0"?>

<call>

<object>DVCR1.record</object>

<method>timeDelayedRecord</method>

<parameter>

<name>channel</name>

<value><int>4</int></value>

</parameter>

<parameter>

<name>recordTime</name>

<value><time>2:10:30</time></value>

</parameter>

</parameters>

</call>
```

##### 2. TDR 호출 예:

```
<?xml version="1.0"?>

<call>

<object>DVCR1.record</object>

<method>timeDelayedRecord</method>
```

```

<parameter>

<name>channel</name>

<value><channelName>NBC</channelName></value>

</parameter>

<parameter>

<name>startTime</name>

<value><datetime.iso8601>19990401T19:05:35</datetime.iso8601></value>

</parameter>

<parameter>

<name>recordTime</name>

<value><time>2:00:00</time></value>

</parameter>

</parameters>

</call>

```

## 예 II:

XML RPC 호출, 감소된 태그 및 인터페이스 정보를 지닌 포맷 예:

### 1. OTR 호출 예:

```

<?xml version="1.0"?>

<call>

<object>DVCR1.record</object>

<method>timeDelayedRecord</method>

<parameter value="4">channel</parameter>

<parameter value="2:10:30">recordTime</parameter>

</call>

```

### 2. TDR 호출 예:

```

<?xml version="1.0"?>

```

```
<call>

<object>DVCR1.record</object>

<method>oneTouchRecord</method>

<parameter value="NBC">channel</parameter>

<parameter value="19990401T19:05:35">startTime</parameter>

<parameter value="2:00:00">recordTime</parameter>

</call>
```

도 20을 참조하면, 홈 디바이스(14)를 위한 디바이스 인터페이스는 표준화된 용어를 사용하는 산업 표준 구조화된 데이터 베이스(100)을 기반으로 한다. 새로운 인터페이스를 위한 인터페이스 데이터 및 용어는 데이터 베이스(100)에 추가될 수 있다. CE 대상을 설명하기 위해 영어 단어를 사용하는 포괄적인 정의 혹은 데이터베이스, 메소드 그리고 파라미터들은 CE 데이터베이스(102)에 규정된다. 포괄적인 정의 또는 데이터베이스는 대상 및 그들 각각의 방법 및 구성요소를 나타내는 C, XML 또는 다른 포맷이될 수 있다. 표준화된 XML 용어를 사용하는 포괄적인 정의 또는 데이터베이스는 XCE 정의 또는 데이터 베이스(104)라고 불려진다.

제어기 및 제어된 어플리케이션(82), (84)는 XCE 데이터 베이스(104)에 기반을 둔 XML의 표준 인터페이스 서브 세트를 사용하여 프로그램 된다. 각 디바이스 인터페이스는 XML 형태로 상기 어플리케이션(82), (84)와 함께 저장된다. 비록 상기 XCE 데이터 베이스(104)가 XML에 있을 필요가 없다 하더라도, 도 15와 관련하여 위에 설명된 것처럼, 본 발명의 실시예에서 컴파일 시에 작성된 상기 서브 세트 인터페이스는 XML 내에 있다.

도 20에서, 임베디드 장치 14를 위해, '제조회사' 정보로서 디자인된 정보는 제조사 장치 14에 내장된 것이고, '홈 네트워크'로서 디자인된 정보는 네트워크에서 장치의 동작적 실행시간 특징의 일부이다. N 디바이스를 위한 1...N로서 디자인된 디바이스 XML 인터페이스(72)는, 표준화된 XCE 데이터 베이스(104)에서 데이터의 부분이다. 홈 네트워크 인터페이스 라이브러리(HNIL:Home Network Interface Library)(106)은 홈 네트워크에 접속된 사용가능한 디바이스(14)의 디바이스 인터페이스 집합을 제공한다. HNIL(106)은 전체 XCE 데이터 베이스(104)의 서브세트이다.

도 16에서, 디바이스 인터페이스는 디바이스 A를 위한 인터페이스의 내용을 조사하는 디바이스 B내의 어플리케이션 B를 위해 디바이스 A로부터 디바이스 B로 전송된다. 위에서 세부적으로, 디바이스 인터페이스는 디바이스에 의해 유지되는 객체, 메소드 파라미터들의 설명을 포함하고, 예를 들어 디바이스 A에 대한 INTERFACE-A.XML로서 언급된다. 디바이스 XML 인터페이스(72)는 XML 포맷의 디바이스 인터페이스이다. XCE 데이터 베이스(104)의 내용은 디바이스 인터페이스를 제공하는 서비스 기반의 구조이다.

도 20을 참조하면, 상기 XCE 데이터 베이스(104)는, 또한 CE 디바이스 14를 표현하는 XML을 사용하기 위해 표준화된 규칙 세트를 제공하는 CE 디바이스를 위한 표준화된 XCE 인터페이스 문서 타입 정의(DTD:Document Type Definition)을 제공한다. 상기 DTD 또는 그것의 서브세트는 정당성 체크를 위해 사용될 수 있다. 제조 회사 수단(108)으로서 디자인된 소프트웨어 에이전트는 규정된 CE 디바이스를 위한 표준화된 XCE 정의의 서브세트를 필터링하고 사용하며, 예를 들어 INTERFACE.XML 및 INTERFACE.DTD와 같은 CE 디바이스의 XML 디바이스 인터페이스(72)를 생성하는 표준화된 XCE 인터페이스 DTD를 사용한다. 상기 문서 INTERFACE.XML은 표준화된 XCE 인터페이스 DTD에 따라 특수한 디바이스에 의해 지원되는 객체, 메소드 및 파라미터들의 설명을 포함한다. 상기 문서 INTERFACE.DTD는 표준화된 XCE 인터페이스 DTD의 서브세트이고, 디바이스의 XML 인터페이스를 위한 정당성 체크를 위해 사용될 수 있다. 다른 문서 타입 정의로 또한 INTERFACE.XML 문서를 생성하기 위해 사용될 수 있다.

상기 XML 인터페이스 문서 및 상기 DTD 문서를 포함하는 CE 디바이스의 XML 인터페이스(72)는 홈 네트워크 인터페이스 라이브러리(106)와 같은 일반적으로 액세스가능한 라이브러리에 저장된다. 소프트웨어 에이전트(110)은 네트워크 상의 모든 액세스가능한 디바이스(14)의 디바이스 인터페이스(72)를 수집하고, 디바이스 이름/주소 정보와 함께 검색 가능

한 구조화된 인터페이스 라이브러리(106)에 그들을 배치한다. 상기 인터페이스 라이브러리(106)은 XCE 데이터 베이스(104)의 서브 세트이고, 인터페이스 라이브러리(106)을 만드는 과정은 부분 또는 모든 XCE 데이터 베이스(104)를 재구성하는 것과 유사하다. 상기 인터페이스 라이브러리(106)은 홈 네트워크 내의 모든 디바이스의 디바이스 인터페이스의 집합 또는 가장 최근에 사용된 디바이스 인터페이스(72)가 저장되는 홈 네트워크에서 모든 디바이스(14)의 유용성에 의존하는 캐시(Cache)로서 기능을 할 수 있다. 디바이스(14)가, DVD 재생기에서 디스크를 바꾸는 것과 같은 이벤트에 기인하여 그 디바이스 인터페이스(72)를 갱신한 경우에는, 디바이스 인터페이스(72)의 일부가 이벤트 서비스를 기반으로 하여 갱신된다.

도 21을 참조하면, 바람직하기로는, 각 디바이스(14)의 디바이스 인터페이스 정의(72)는 계층적인 형태를 가진다. 왜냐하면 이것은 홈 디바이스에 있어서, 디바이스 인터페이스 정의(72)가 길어질 수 있기 때문이다. 전형적으로, 시간 지연된 녹화를 위한 단일의 기능과 같은 하나 또는 소수의 기능은 동시에 액세스되고, 따라서 상기 디바이스 인터페이스(72)의 유일한 하계 작은 부분이 사용된다. 전체 디바이스 인터페이스(72)를 작성 하기 보다, 디바이스 인터페이스(72)의 부분만을 작성하는 것이 더 효과적이다. 계층적인 디바이스 XML 인터페이스를 사용함으로써, 제어기 디바이스(14)는, 제어기 디바이스(14) 또는 HNORB 및 IL 미들웨어 계층(98)로부터 XML 디바이스 인터페이스를 위한 요구함에 있어서 요구되는 기능 카테고리 또는 기능을 특정함으로써 제어된 디바이스(14)의 부분적인 디바이스 인터페이스(72)를 요청할 수 있다. 후자의 경우에, HNORB 및 IL 미들웨어 계층(98)은 디바이스 인터페이스(72)의 요구된 부분을 회송한다.

도 21을 참조하면, 상기 계층적인 디바이스 인터페이스 구조는 (1) 현재 유용한 디바이스를 기입한 각 홈 네트워크의 XML 인터페이스를 위한 제1 계층(112), (2) 기능 카테고리를 기입한 각 디바이스의 일반적인 XML 인터페이스를 위한 제2 계층(114), (3) 디바이스를 위한 각 기능 카테고리의 XML 인터페이스를 규정하기 위한 제3 계층(116), (4) 기능 카테고리내의 각 기능의 XML 인터페이스를 규격화하기 위한 제4 계층(118)을 포함하는 4개의 층을 포함한다. 홈 네트워크 내부에서, 단지 세 개의 하위 층, (114), (116) 및 (118)이 사용되고, 홈 네트워크 외부에서 제1 층(112)가 사용된다.

도 22는 상기 층들 (112), (114), (116), (118) 및 대응된 인터페이스 예를 보여준다. 각 계층내의 인터페이스는(가능하다면) 양방향 링크를 제공하는 X 링크(Link) 또는 X 포인터(Pointer)와 같은 링크들을 통하여 상위 또는 하위 층에 링크된다. X 링크는 (1) XML 문서내의 링크를 그것으로 인식되는 것을 허용하는 X 링크 성분, (2) 링크들이 XML 문서의 정확한 서브 파트로 어드레싱하도록 허용하는 X 포인터 성분인 두 파트를 가지는 하이퍼링크 기능성의 패키지를 포함한다. 그러한 것으로서, X 링크는 어떻게 링크가 XML 문서에 삽입되는가를 관리하는데, 여기서 상기 링크는 GIF 파일과 같은 데이터를 가리킬 수 있다. 더욱이 X 포인터는, 어디든지부터(예를들어 HTML 파일로부터), XML 문서에 링크할 때 URL로 갈 수 있는 조각 식별자(Fragment Identifier)를 결정한다.

본 발명에 따른 다른 서버 디바이스(14)를 제어하는 서버 디바이스(14)를 위한 전형적인 명령 및 제어 모델에서, 제1 디바이스(14)는 제2 인터페이스 계층(114)에서 제2 디바이스(14)의 디바이스 인터페이스를 쿼리하도록 시도한다. 기능 카테고리(FC:Function Category)를 선택한 후에, 상기 제1 디바이스(14)는, 녹화 카테고리나 같은 제2 디바이스(14)에서 특정한 기능 카테고리의 인터페이스 계층(116)을 쿼리한다. 더욱이, 상기 제1 디바이스(14)는 상기 기능들에 대한 호출을 만들기 위해 OTR 또는 TDR과 같이 특정한 기능을 인터페이스 계층(118)에 쿼리한다. 계층적 또는 트리 구조는 더 효과적인 인터페이스 기능을 찾도록 만들고, 네트워크 대역폭을 절감한다. 인터페이스 파일 구조 및 층의 예가 아래와 같을 수 있다.

제1 층 112- HN1.xml

제2 층 114- VCR1.xml

제3 층 116- VCR1\_RecordCategory.xml

제4 층 118- VCR1\_RecordCategory\_OTR.xml

유사하게, 홈 네트워크 인터페이스 라이브러리(106)은 바람직하게 계층적이고, 디바이스의 다른 서비스 타입 또는 방들(Rooms)과 같은 다른 위치에 의해 다양한 방법으로 구조화될 수 있다. 상기 계층적인 구조는 다른 홈 네트워크 또는 인터넷에 대한 국부적인 홈 네트워크(10)의 인터페이스이다.

XML 신택스(Syntax)로 구현될 수 있는 예시적인 계층적인 디바이스 인터페이스 정의(72)가 아래에 도시된다.

consumer (document\_file, doc)

```

+ ----document_file<server_home.dtd, server_auto.dtd>

+ ----doc (services_home, server_auto, server_samsung_web_site, avc_commands, cal_commands,,)

+ ----services_home (xml_utility, client, server_av, lighting, comms, hvac, utility, security, appliances,
convenience,,)

+ ----xml_utility (download_DTD_file,,)

+ ----client (acknowledge, attention, error, post_message, sound, stop_schedule, stop_all,,)

+ ----sound(alarm, ring, buzz,,)

+ ----server_av (controls_gen, source, sink)

+ ----controls_gen (ping, process_infor, setup,,)

+ ----process_info (s/w_id, h/w_id)

+ ----h/w_id (ser_no, manuf, model, class,,)

+ ----s/w_id (ser_no, exe_name, version,,)

+ ----setup (clock,,)

+ ----clock (hours, minutes, seconds)

+ ----source (service_id, media, rate, protocol, stream_format, controls_av,,)

+ ----sink (service_id, media, rate, protocol, stream_format, controls_av,,)

+ ----service_id (url,,)

+ ----media (tpt_stream, ram, disk, tape,,)

+ ----disk (name, number,,)

+ ----rate<value>

+ ----protocol (61883/1394, UDP/IP/Ethernet,,)

+ ----61883/1394 (isoch_ch_no)

+ ----stream_format (video, audio,,)

+ ----video (dv, mpeg2tpt, dsstpt, mpeg2pes, mpeg10801-tpt,)

+ ----audio (mpeg3, ac-3, midi,,)

+ ----controls_av (flow_control, tune, timer_record, ui_control,,)

+ ----timer_record (tune, flow_control)

```



```

+ ----flow control (play, stop, goto, record,,)

+ ----play(time_params)

+ ----record(time_params)

+ ----time_params (now, start, duration, end,,)

+ ----tune (send_epg, channel,,)

+ ----channel (number, id, time_params,,)

+ ----ui_control (display, acoustic)

+ ----display (brightness, contrast, color/tint, horiz_size, vert_size,,)

+ ----acoustic (volumn, base, treble, balance, fade,)

+ ----lighting (sensors, lights, send_epg)

+ ----sensors (living_room, sky,,)

+ ----lights (rooms_up, rooms_down, yard,,)

+ ----rooms_up (bed1, bed2, bed3, bed4,,)

+ ----bed1 (lamp, dimmer,,)

+ ---dimmer<value>

+ ----rooms_down (family, kitchen, living, dining, soho, garage,,)

+ -----yard (front, back)

+ ----comms (homehub, intercom, telco,)

+ ----homehub (send_device_list, send_configuration, sene_snmp_mib,,)

+ ----intercom ()

+ ----telco ()

+ ----hvac (controls_gen, controls_havc,,)

+ ----controls_hvac (a/c, heat, temp, humidity,)

+ ----temp (low, high, hysteresis,,)

+ ----utility (meters, energy_mgmt,,)

+ ----meters (water, gas, electric,,)

+ ----water <value>, gas<value>, electric<value>

```

```
+ ----security (sensors, send_epg, alarm,,)

+ ----sensors (peripheral, motion,,)

+ ----peripheral (rooms_up, rooms_down,,)

+ ----motion (room_down, yard,,)

+ ----appliances (microwave, range, oven, fridge, freezer, coffee, toaster, washer, dryer, water_heater,,)

+ ----microwave (send_epg, controls,,)

+ ----fridge (temp,)

+ ----water_heater (temp,)

+ ----convenience (window, curtain_open, door/gate, pool/spa, bath, fountain, lift, jacuzzi,,)

+ ----curtain_open<value>

+ ----server_auto (message, server_auto_ford_explorer_98,,)

+ ----server_auto_ford_explorer_98 (mileage, maintenance,,)

+ ----mileage <data>

+ ----maintenance <data>

+ ----server_samsung_web_site (message, service, help,,,)

+ ----avc_commands<,,,command_string,,,>

+ ----service_id (url,,)

+ cal_commands<,,,command_string,,,>

+ ----service_id (url,,)
```

상기 계층적인 디바이스 인터페이스 정의 72는 다음과 같은 분야를 포함할 수 있다.

'document file'이름으로, 이는 XCE 데이터베이스(104) 또는 XCE 데이터베이스(104)의 버전 부분의 정당성 및 정정의 검증을 위해 XML 파서에 의해 사용될 수 있는 문서 타입의 규정(DTD 파일의 이름을 제공한다. 상기 XCE 구조의 상이한 부분을 위한 여러가지 DTD 파일들이 있을 수 있고, 여기서 상기 DTD는 RPC.CALL을 위한 문서 타입 정의 및 통신을 위한 INTERFACE.DTD와 다르다.

'doc'이름으로, 이는 호환성, 특성, 통신 및 제어 인터페이스의 영역의 최상위 레벨 이름을 제공한다.

'Services\_home'으로, 이는 가정 자동화, 가전 제품들, 효율성 등을 위한 영역을 제공한다.

'Server\_auto'로, 이는 차고 안에 있는 자동차에 대해, 한 대 또는 그 이상의 자동차 타입에 대한 유용한 메시지 인터페이스를 보여준다. 예를 들어, 'server\_auto\_ford\_explorer\_98'은 특정 자동차에 대한 인터페이스이다. 이것은 자동차의 마일리지(Mileage) 및 보수 인터페이스를 액세스 할 수 있도록 허용하고, 직접 체크하는 자동차 제조회사 또는 차고가 직접 체크 또는 원격 진단하는 원격 액세스로 사용될 수 있다.

'server\_samsung\_web\_site'으로, 이는 가정 외부의 제조 회사 웹사이트와 통신을 제공하고, 메시지, 서비스, 도움 등을 위한 인터페이스를 포함한다.

'AVC\_commands' 및 'CAL\_commands'로, 이는 예를 들어, AV/C 및 CAL 언어를 해석할 수 있는 물려받은 디바이스를 제공한다. 이 구조 부분은 상기 언어에서 명령을 확인하며, 거기서 명령은 XML에서 부가되거나 수행된다. 그러한 것으로서, 상기 항목은 XCE(Web) 객체가 아니고, 프로토콜 변환기 어플리케이션은 원래의 CAL 또는 AV/C 어플리케이션 소프트웨어로의 인터페이스로 사용될 수 있다.

위의 설명에서, 'Services\_home'은 A/V 가전제품을 포함하는 메인 구조를 제공한다. 상기 구조의 한 부분은 특정한 예로 비디오 서비스 싱크 및 스트림 목적지(예를 들어, DVCR) 제어 인터페이스에 대하여 상세하게 확장된다. 전형적인 홈 네트워크에서 제어 인터페이스는 다음을 포함할 수 있다.

'xml\_utility'로, 이는 갱신된 DTD 파일, 인터페이스 파일, 프로그램 파일 등을 다운로드 하는 것과 같이 지원 유틸리티(Utility) 네트워크 기능을 지원하기 위한 세부 사항을 제공한다.

'client'로, 이는 웹 브라우저를 포함하는 클라이언트 디바이스 12의 인터페이스 세부 사항을 설명한다. 예를 들어, 'acknowledgment'는 발송된 메시지 또는 명령으로 제어기의 인증 수락을 표시한다.

'server\_av'로, 이는 STB, DVCR, DTV, DVD, AUDIO 등을 포함하는 유용한 모든 오디오 및 비디오 서비스에 대하여 제어 및 용량 인터페이스를 제공한다.

'lighting'으로, 이는 가정 자동화 조명 제어기에 인터페이스를 제공하며, 센서들 및 조명 등을 포함한다.

'comms'로, 이는 는 전형적으로 사용 목적 또는 디바이스의 셋-업 또는 구성요소의 원격 관리 또는 구성을 복구하기 위해 통신 디바이스에 제어 인터페이스를 제공한다.

'hvac'로, 이는 HVAC 시스템의 원격제어를 위한 인터페이스를 제공하고, 예를 들어 그날의 피크(Peak) 부하 기간 동안에 가정의 HVAC 시스템을 끄기 위해 공익 설비 회사(Utility company)에 의해 가정 외부로부터 상기 시스템의 제어하기 위해 사용될 수 있다. 더욱이, 상기 인터페이스는, 자동 온도 조절 장치 제어 보다 더 복잡한 제어 메카니즘을 제공하는 제어기에 기반을 둔 디바이스를 위한 장치에 의해, 가정 내부로부터 HVAC 시스템을 제어하기 위해 사용될 수 있다.

'utility'로, 이는 예를 들어, 가정을 위한 계량기(Utility Meter)를 독취하기 위한 인터페이스를 제공한다.

'security'로, 이는 보안 센서 및 알람 세팅(Alarm setting)을 위한 인터페이스를 제공한다. 그와 같은 것으로서, 홈 네트워크 디바이스 상에서 인터페이스 사용, 어플리케이션 진행은 그 디바이스의 감시 및 제어를 위한 가정 주변의 센서 및 감지기를 액세스할 수 있다.

'appliances'로, 이는 예를 들어, 원격 제어 또는 온도 설정 감지 또는 제어기 디바이스로부터 다른 제어 및 구성요소를 포함하는 주방, 유용 및 일반적인 가정 장치를 위한 인터페이스를 제공한다. 일 시나리오에서, 전자 레인지는 음식 아이템 패키지 상의 바코드 정보를 스캔할 수 있고, 주어진 전자레인지 시스템 타입에 대해 음식의 조리 시간을 알아내기 위해 제조업체 데이터베이스를 액세스 할 수 있다. 디바이스-디바이스 명령 및 제어를 사용하는 장치의 통합은 주방 또는 거실에서 전화를 받았을 때 자동으로 식기 세척기를 정지시키고 텔레비전을 뮤팅(Muting)시키는 것과 같은 서비스를 제공하기 위해 많은 제어 시나리오들을 제공한다.

'convenience'로, 이는 예를 들어 커튼, 창문, 블라인드 또는 세탁 제어기에 대한 인터페이스와 같이 편의 서비스를 제공하기 위해 장치에 대한 인터페이스를 제공한다.

위의 설명에서, 'server\_av'는 A/V 서비스를 제공하는 A/V 장치에 대하여 제어 인터페이스를 위한 구조의 한 부분이고, 'controls-gen', 'source', 'sink' 성능으로 다시 나누어진다.

'controls-gen'은 디바이스 제조자 배포 및 디바이스 존재의 핑(Ping) 테스트와 같은 일반적인 유틸리티 인터페이싱을 위한 인터페이스를 제공한다. 더욱이, 소프트웨어 및 하드웨어 확인 및 버전 정보와 같은 제조 특성이 포함될 수 있다. 상기 인터페이스를 제공하는 디바이스는 어떤 제어 동작도 초래하지 않고 상기 소프트웨어에 대한 이름 또는 확인을 제공하는 데이터를 돌려보낸다. 또한 날짜 시계의 시간을 세트하는 인터페이스가 포함될 수 있다.

'sink'로, 이는 미디어 스트림 서비스 디바이스를 위한 인터페이스를 제공한다. 그 구조는 VCR과 같은 특별한 디바이스 이름이기보다는 제공된 서비스(예를 들어, 비디오 스트림 녹화 및 리플레이)에 기반을 둔 구성이다. 예를 들어, 튜너와 DVD 플레이어는 둘 다 네트워크에 대해 비디오 프로그램 포맷을 가지는 비디오 프로그램 스트림 소스이고, 시작 및 정지와 같이 제어될 수 있다. 특정한 디바이스 제어에서의 차이점은 구조화 정의의 하부 계층에 의해 어드레스된다.

'source'는 'sink' 인터페이스와 유사한 인터페이스를 제공한다.

상술한 내용을 참조하면, 'service\_id' 또는 'application\_interface\_id'는 이름, 주소 또는 웹 주소, 또는 하나 이상 디바이스의 URL 위치를 포함한다. XCE 데이터베이스(104)는 인터페이스에 대해 합의된 전체를 구성하기 때문에, 통상 DHCP (Dynamic Host Configuration Protocol) 소프트웨어 에이전트는 각 디바이스에 어드레스 및 디폴트 이름을 할당하는 것을 실행하고, 상기 어드레스 및 디폴트 이름은 서비스 또는 디바이스의 인터페이스에 추가된다. 그리고 나서, 소프트웨어 에이전트(110)는 '부분적인 XCE 네트워크(Network partial XCE)'를 생성하기 위해 홈 네트워크에 국부적으로 접속된 모든 디바이스로부터 서브 세트 또는 '부분적인 XCE 디바이스(Device partial XCE)'를 포함하는 디바이스 인터페이스(72)를 모은다. 부가적인 관련 외부 인터페이스들이 외부 제어를 위한 구조에 추가될 수 있다. 예를 들어, 'service\_id'는 네트워크에 접속된 디바이스의 디바이스 인터페이스에 따라 소프트웨어 에이전트로부터 가입을 포함하는 수신된 구조 또는 네트워크 인터페이스 라이브러리(106)내의 이름/주소일 수 있다. 그후에, 사용자는 데이터베이스에서 서비스를 검색할 수 있고, 상기 이름/주소를 사용하여 라이브러리의 특정한 데이터 부분을 포함하는 어플리케이션을 액세스할 수 있다. 그러한 것으로서, 네트워크는 상기 이름/주소 정보에 의해 식별된 다중의 동일한 서비스들을 포함할 수 있다.

'media'로, 이는 예를 들어, 튜너로부터의 전송 스트림, PC DRAM으로부터 RAM, DC 또는 DVD를 위한 디스크 및 테이프를 포함하는 미디어의 타입을 위한 인터페이스를 제공한다. 상기 미디어는 명명되고 확인될 수 있고, 제어기 디바이스는 네트워크 상에서 제공되는 일반적인 미디어를 확인하는 XCE 데이터베이스를 검색할 수 있다. DVD 디스크와 같은 새로운 미디어가 네트워크 상에서 제공될 때, 디스크 상에서 프로그램 자료를 확인하는 디바이스 인터페이스(72)의 일부가 알맞게 변환된다. 그러한 것으로서, 전체 디바이스 인터페이스(72)가 전송될 필요는 없고, 단지 적합한 부분이 XCE 데이터 베이스로 전송된다. 어텐션(Attention) 신호의 수신으로, 라이브러리 소프트웨어 에이전트(110)은 새로운 갱신을 패치할 수 있고, 적합한 장소에 그것을 위치시킬 수 있다. 디스크 미디어의 부가는 상기 네트워크에 서비스를 추가하거나 또는, 상기 네트워크에 다른 장치를 접속하는 것과 유사하다.

'rate'로, 이는 예를 들어, 6Mbps/Sec 또는 19.2Mbps/Sec와 같이 디바이스 인터페이스를 위한 데이터 스트림 레이트(Rate)에 대한 값을 제공한다.

'protocol'로, 이는 상기 데이터 스트림을 위해 사용된 프로토콜을 확인한다. 만약 예를 들어, 61883/1394 또는 U에/IP와 같이, 하나 이상의 프로토콜이 제공되면, 바람직한 프로토콜이 선택될 수 있다.

'stream\_format'으로, 이는 디지털 스트림 오디오 및 비디오 분할을 위한 패킷 포맷 및/또는 압축 표준을 제공한다. 만약, 하나 이상의 포맷이 제공된다면, 바람직한 포맷이 인터페이스 메시지를 통하여 선택될 수 있다. 제어기 어플리케이션 82는 거기에 호환적인 것이 있는지 없는지를 결정하기 위해 사용 가능한 포맷을 조사할 수 있다.

'controls\_av'로, 이는 A/V 미디어 장치를 위한 메인 제어 인터페이스를 제공한다.

'Flow\_control'로, 이는 특정한 디바이스를 위한 방법으로서 PLAY, STOP, GOTO, RECORD 등과 같은 데이터 스트림 제어를 제공한다. 그 방법은 예를 들어, PC 소프트웨어를 제외하고, 임베디드(Embedded) 장치에 대해 변하지 않는다. tkdrl 제어는 지연된 동작을 위해 시간 구성요소를 포함할 수 있다.

'Tuning'으로, 이는 튜닝 제어를 위한 인터페이스를 제공한다. 제어기 디바이스 14는 제어된 디바이스 14의 인터페이스로 위에서 기술된 전자 프로그램 가이드(EPG:ElectronicProgram Guide) 데이터 구조를 회신 요청을 보낼 수 있다.

'UI control'로, 이는 디스플레이에 대한 밝기 및 콘트라스트, 오디오에 대한 볼륨 및 베이스와 같은 수정을 제어하기 위해 제어된 어플리케이션 84에 제어 인터페이스를 제공한다.

'Timer\_record'로, 이는 지연된 시간 기록을 구현하기 위해 제어기 어플리케이션 82에 대해 셋-업 데이터를 위한 인터페이스를 제공한다. 직접적인 채널 동조 정보 및 플로우 제어(time\_aparams) 정보가 사용될 수 있다.

위의 설명은 클라이언트 디바이스(12)에 동일하게 적용될 수 있다. CE 영역을 위한 다른 신택스 XCE 정의 또는 데이터베이스가 사용될 수 있다. 대안적인 신택스 XCE 데이터 베이스는 예를 들어 가정 자동화, 장치 및 자동차를 포함하는 모든 서비스 설명을 포함한다. 서비스 객체가 제어를 위해 적응성 및 파라미터들을 제공하는 경우, 제어 메쏘드는 원하는 대로 상기 객체를 제어하기 위해 사용된다. 이진수 또는 16진수 데이터 열을 포함하는 AV/C 및 CAL 명령어에서 명령 예가 아래에 도시된다.

```
consumer (document_file, doc)

+ ----document_file<server_home.dtd, server_auto.dtd>

+ ----doc (avec_commands, cal_commands, services_home, server_auto,)

server_auto,

server_samsung_web_site, server_auto_ford_explorer_98,,)

+ ----avc_commands<...command_string...>

+ ----call_commands<...command_string...>

+ ----services_home (client, av, lighting, comms, hvac, utility, security, appliance, convenience,,)

+ ----xml_utility (download_DTD_files,,)

+ ----client (acknowledge, attetion, error, post_schedule, soundstop_schedule, stop_all,,)

+ ----sound (alarm, ring, buzz,,)

+ ----server_av (source, sink)

+ ----soruce (service_id, media, rate, protocol, stream_format, controls_gen, controls_av,,)

+ ----sink (service_id, media, rate, protocol, stream_format, controls,,)

+ ----service_id (url,,)

+ ----media (tpt_stream, ram, disk, tape,,)

+ ----disk (name, number,,)

+ ----rate<valu>

+ ----protocol (61883/1394, UDP/IP/Ethernet,,)

+ ----61883/1394 (isoch_ch no)

+ ----stream format (video, audio,,)
```

```

+ ----video (dv, mpeg2tpt, dsstpt, mpeg2pes, mepeg1080i-tpt,)
+ ----audio (mpeg3, ac-3, midi,,)
+ ----control_gen (ping, process_info, setup,,)
+ ----controls_av (flow_control, tune, timer_record, ui_control,,)
+ ----process_info (s/w_id, h/w_id)
+ ----h/w_id (ser_no, manuf, model, class,,)
+ ----s/w_id (ser_no, exe_name, version,,)
+ ----setup (clock,,)
+ ----clock (hours, minutes, seconds)
+ ----time_record (tune, flow_control)
+ ----flow_control (play, stop, goto, record,,)
+ ----play (time_params)
+ ----record (time_params)
+ ----tune (send_epg, channel,,)
+ ----channel (number, id, time_params,,)
+ ----time_params (now, start, duration, end,,)
+ ----ui_control (display, acoustic)
+ ----display (brightness, contrast, color/tint, horiz_size, vert_size,,)
+ ----acoustic (volume, bass, treble, balance, fade,)
+ ----lighting (screen, light, send_epg)
+ ----sensors (living_room, sky,,)
+ ----lights (rooms_up, rooms_down, yard,,)
+ ----rooms_up (bed1, bed2, bed3, bed4,,)
+ ----rooms_down (family, kitchen, living, dining, soho, garage,,)
+ ----yard (front, back)
+ ----bed1 (lamp, dimmer,,)
+ ----dimmer<value>

```



```

+ ----comms (netman, intercom, telco,)

+ ----netman (send_device_list, send_configuration, send_snmp_mib,,)

+ ----intercom ()

+ ----telco ()

+ ----hvac (controls_gen, controls_hvac,,)

+ ----controls_hvac (a/c, heat, temp, humidity,)

+ ----temp (low, high, hysteresis,,)

+ ----utility (meters, energy_mgmt,,)

+ ----meters (water, gas, electric,,)

+ ----water<value>, gas<value>, electric<value>

+ ----security (sensors, send_epg, alram,,)

+ ----sensors (peripheral, motion,,)

+ ----peripheral (rooms_up, rooms_down,,)

+ ----motion (rooms_up, yard,,)

+ ----appliances (microwave, range, over, fridge, freezer, coffee, toaster, washer, dryer, water-heater,,)

+ ----microwave (send_epg, controls,,)

+ ----fridge (temp,,)

+ ----water_heater (temp)

+ ----convenience (window, curtain_open, door/gate, pool/spa, bath, fountain, lift,,)

+ ----curtain_open<value>

+ ----server_auto (message, mileage, maintenance,,)

+ ----mileage<data>

+ ----maintenance<data>

```

다른 관점에서, 본 발명은 네트워크에서 디바이스-디바이스 명령 및 제어를 위해 존재하는 명령어 구현의 사용을 제공한다. 디바이스는 내부 객체 및 존재하는 전송 메카니즘에 따라, 실행 시에 이진수 열을 생성하는 API들을 포함할 수 있다. 이러한 경우, 네트워크에서 한 개의 디바이스 14로부터 또 다른 디바이스 14에 XML 원격 절차 호출(XML RPC : XML Remote Procedure Calls)를 제공하기 위해서, 존재하는 어플리케이션 인터페이스 구현은 XML 서비스 API에 대한 호출로 대체된다. 그러한 것으로서, 본래의 구현은 XML 서비스 API를 위한 래퍼(Wrapper)와 동등하다. 도 18은 또한 XCE/XML 서비스 API에서 래퍼를 대체된 인터페이스 구현으로, 점선 내에서의 CAL 또는 AV/C와 같은 다른 명령어를 사용하여 생성된 어플리케이션을 보여준다. CAL 명령어로부터 XML RPC 포맷을 변환시키는 예가 아래에 도시되어 있다.

-----

existing implementation:

```
void DeviceCALCommand (int command) {
.
.
/*
crate CAL formatted byte string to represent this object/method and output to the wire
*/
CreateCALFormattedByteString(command); */different for every protocol*/
sendCALByteString(); */different for every protocol*/
*/
}
```

-----

wrapping the XML Service API call:

```
void DeviceCALCommand (int command) {
{
/*
replace CAL implementation with calls to the XML ServiceAPI
*/
CreateXMLMessage (command); */always the same*/
send XMLMessage () ; */always the same*/
}
}
```

또 다른 관점에서, 도 23을 참조하면, 본 발명은 네트워크내의 다른 디바이스 사이의 디바이스간 통신을 위한 표준 명령 프로토콜 및 제어 언어 번역을 제공한다. 정보를 공유하는 다른 디바이스를 위하여, 상기 정보는 요청 디바이스가 해석할 수 있는 포맷이어야 한다. 그리고, 하나의 디바이스(120)가 다른 디바이스(22)를 제어하기 위하여, 상기 두 개의 디바이스는 서로의 명령을 해석하기 위해 공통의 언어를 사용해야한다. 본 발명은 데이터를 위한 공통 식별 포맷 및 명령 프로토콜을 제공한다.

일 실시 예에서, 공통 표시 또는 데이터 및 명령 프로토콜 패키지 하는 방법이 제공되고, 그것에 따라 수신 디바이스(122)는 전송된 데이터의 본래 포맷을 결정할 수 있다. 만약 수신 디바이스가 본래의 포맷을 해석할 수 있다면, 그때에 데이터를 직접적으로 수신할 수 있다. 상기 그렇지 않으면, 수신 디바이스(122)는 번역 디바이스(124) 또는 요청 디바이스(122)가 해석할 수 있는 바람직한 포맷으로 데이터를 변형하는 어플리케이션을 요청할 수 있다. 상기 번역 디바이스(124) 또는 어플리케이션은 원래의 데이터의 본래 포맷을 결정하고, 상기 데이터를 상기 요구되는 포맷으로 번역하며, 요청 디바이스 112에 번역된 데이터를 송신한다.

다음에 상기 요청 디바이스(122)는 마치 상기 데이터가 송신 디바이스(120)에 의해 요청 디바이스의 본래 언어 포맷으로 원래 제공되는 것처럼 데이터를 처리한다. 상기 요청 디바이스(122)는 요청 디바이스의 본래 포맷으로 송신 디바이스(120)로 회신을 또한 보낼 수 있거나 번역 디바이스(124) 또는 송신 디바이스(120)의 본래 포맷으로 번역하는 어플리케이션을 통해 프록시에 의해 응답을 전송할 수 있다. 상기 번역 방법은 명령 프로토콜, 데이터 파일 및 오디오/비디오 스트림을 포함하는 정보를 위해 사용될 수 있다.

위에서 기술된 공통 포맷을 사용하지 않는 디바이스들을 위하여, 본 발명은 유연하지 않은(Non-compliant) 디바이스로/부터의 명령 프로토콜을 포함하는 데이터의 번역을 제공한다. 예를 들어, 유연하지 않은 디바이스(120)가 유연한 디바이스(122)에 데이터를 송신할 때, 상기 유연한 디바이스(122)는 상기 데이터의 본래 포맷의 결정에 기반을 결정하여 데이터를 번역할 수 있다. 예를 들어 상기 유연한 디바이스(122)는 데이터 내부의 특정한 비트 패턴을 위한 데이터를 조사할 수 있다. 유연한 디바이스가 알려진 유연하지 않은 디바이스에 데이터를 전송할 때, 상기 유연한 디바이스는 유연하지 않은 디바이스의 본래 포맷의 결정에 기반을 둔 전송 전에 상기 데이터를 번역할 수 있다.

예시적인 구현은 IP 및 HTTP 프로토콜을 지원하는 홈 네트워크를 위한 것일 수 있다. 상기 홈 네트워크는 바람직한 기능을 위한 어플리케이션 및 다양한 타입의 서비스를 획득하기 위해 인터넷에 접속될 수 있다. 그러한 것으로서, 공통 포맷 메소드는 인터넷 및 홈 네트워크 상의 동작에 대한 인터넷 프로토콜 및 절차와 호환성을 갖도록 만들어 질 수 있다.

공통 데이터 포맷을 제공하는 하나의 예는 홈 네트워크를 통하여 전송을 위한 데이터 패키지를 생성하는 XML을 사용하고 있다. 상기 데이터는 명령 프로토콜, 오디오 또는 비디오 스트림, 그래픽 또는 어플리케이션을 포함할 수 있다. 상기 데이터는 XML 형태로 데이터의 본래 포맷 및 패키지의 항목을 확인하는 표준 헤더로 래핑된다. 상기 헤더는 데이터 타입과 XML 코드의 데이터 부분을 유일하게 확인하는 것을 허용하고, 그것에 의하여 상기 데이터는 필요에 따라 번역될 수 있고, 수신 후에 적당한 어플리케이션에 제공될 수 있다.

웹 표준 하에서, 상기 확인 작업은 파일 전송의 타입 및 항목을 확인하기 위해 파일 이름 확장을 사용하는 브라우저에 의해 수행된다. 다음에 브라우저는 그 파일을 처리하는 적당한 플러그-인 모듈을 이동시킨다. 홈 네트워크에서, XML은 IP 상의 모든 홈 네트워크 전송을 위에서 기술된 공통 확인 방법을 가지고 제공하는 데이터 전송을 확인하기 위해 사용된다.

한편으로, 홈 네트워크를 통한 모든 데이터 전송의 항목을 유일하게 확인하는 소프트웨어 계층이 홈 네트워크 프로토콜 스택에서 제공될 수 있다. 상기 소프트웨어 층은 XML 대신에 사용될 수 있다. 본 발명의 공통 포맷 및 확인 원리는 XML을 사용하는 실시 예 또는 확인 방법으로서 상기 소프트웨어 계층에 동일하게 적용한다.

도 23에서 데이터 패키지 전송을 수신하면, 수신 디바이스(122)는 그안의 데이터의 포맷을 결정하는 데이터 패키지의 XML 확인 헤더를 조사한다. 만약 상기 데이터가 디바이스(122)에 의해 인식할 수 있는 포맷이라면, XML 확인 헤더 정보는 버려지고, 상기 디바이스는 데이터를 직접 처리한다. 그렇지 않으면, 상기 디바이스(122)는 수신된 XML 패키지를 XML 번역 요청 패키지로 변환하고, 번역 서버 디바이스 124로 요청 패키지 및 상기 데이터를 송신한다.

상기 번역 서버 디바이스(124)는 상기 데이터를 번역하고, 상기 번역된 데이터를 XML 번역 응답 패키지로 변환한다. 다음에 상기 번역 서버(124)는 요청 디바이스(122)로 응답 패키지를 회송한다. 번역 에러인 경우에, 상기 번역 서버(124)는 요청 디바이스(122)에 번역 응답 에러 상태를 제공할 수 있다. 번역된 데이터를 수신하면, 상기 요청 디바이스(122)는 번역된 데이터를 응답 패키지에서 처리한다.

XML데이터 패키지 또는 패킷의 예는 다음과 같을 수 있다:

```
<IDENTITY type=format=AV/C>...packet data...</IDENTITY>
```

번역 요청 패키지 또는 패킷의 예는 다음과 같이 있을 수 있다:

<TRANSLATION REQUEST type=Command format=CAL>

<IDENTITY type=Command format=AV/C>...packet data...</IDENTITY>

</TRANSLATION REQUEST>

번역 요청 패키지 또는 패킷 예는 다음과 같을 수 있다:

<TRANSLATION RESPONSE type=Command format=CAL>...packet data...</TRANSLATION RESPONSE>

전송 응답 에러 상태 패키지 또는 패킷의 예는 다음과 같을 수 있다:

<TRANSLATION RESPONSE type=Command format=CAL>...packet data...<ERROR condition=Unrecognized command>Translation could not be performed</ERROR></TRANSLATION RESPONSE>

더욱이, 도 24에서 테이블 3은 패키지 또는 패킷 타입 및 포맷의 부분적인 리스트를 포함한다.

번역 서비스를 제공하기 위해, 번역 서버(124)는 DHCP 서버의 그것과 유사한 방법으로 네트워크 구성 동안에 네트워크에서 확인된다. 상기 번역 서버(124)는 네트워크가 구성된 후에 일정한 기간 동안 네트워크내의 모든 디바이스에 그것의 IP 주소를 방송한다. 번역 서비스와 호환할 수 있는 모든 디바이스(120), (122)는 네트워크를 부팅 하는 동안에 네트워크를 통하여 방송하는 번역 서버(124)의 IP 주소를 저장한다.

한편으로, 요청 디바이스 122는 홈 네트워크를 통하여 번역 요청을 방송할 수 있다. 네트워크에서 번역 요청을 수신하는 모든 번역 서버(124)는 요청 디바이스(122)에 번역 응답을 송신함으로써 번역 요청에 응답할 수 있다. 다음에 요청 디바이스(122)는 응답 번역 서버들 중에서 하나의 번역 서버(124)를 선택한다. 일 예를 들어, 요청 디바이스(122)는 번역 요청에 응답하는 첫번째 번역 서버(124)를 선택한다. 또 다른 예에서 번역 서버(124)들은 번역 요청을 충족시키는 번역 서버(124)의 선택을 위해 그들 자신 사이에서 그리고/또는 요청 디바이스(122)와 교섭할 수 있다.

본 발명의 또 다른 실시 예에서, 다중의 번역 서버(124)가 모든 번역 요청을 실행하기 위해 사용된다. 예를 들어, 단일의 번역 서버(124)가 모든 요청을 번역하는 성능을 가지고 있지 않을 수도 있다. 이러한 경우, 각 번역 서버의 주소 및 각 번역 서버가 제공할 수 있는 번역 서비스의 타입을 확인할 필요가 있다. 각 디바이스(120), (122)는 모든 번역 서버 IP 주소 리스트와 각 번역 서버(124)가 제공하는 번역 서비스 타입에 상응하는 리스트, 그리고 선택적으로 연관된 번역 어플리케이션을 저장할 수 있다.

효율을 위하여, 만약 송신 디바이스(120)가 송신 디바이스(120)의 그것과 다른 본래 포맷을 사용하는 것으로 알려진 수신 디바이스(122)에 데이터를 전송하기를 바란다면, 상기 송신 디바이스(120)은 번역 서버(124)를 통한 프록시에 의해 수신 디바이스(122)에 데이터를 전송할 수 있다. 상기 송신 디바이스(120)은 번역 서버(124)에 전송 요청 명령과 유사하게 번역 서버(124)에 명령을 전송하고, 번역된 데이터에 대한 목적지로서 수신 디바이스(122)의 주소를 포함한다.

수신 디바이스(122)가 데이터 스트림의 번역을 요구하는 경우에는, 송신 디바이스(120)은 직접 번역 서버(124)로 데이터 스트림을 라우팅할 수 있고, 상기 번역 서버(124)는 위에서 기술된 것처럼 수신 디바이스(122)에 번역된 데이터를 차례로 전송한다. 다른 한편으로는, 상기 송신 디바이스(120)은 상기 수신 디바이스(122)에 데이터 스트림을 전송할 수 있고, 다음에 수신 디바이스(122)는 번역 및 상기 수신 디바이스(120)로 번역된 데이터의 회송을 위해 번역 서버(124)에 데이터 스트림을 라우팅한다.

설명 속에서, 제어 메카니즘은 분배되고, 협조적인 하이퍼미디어 정보 시스템을 위한 어플리케이션-레벨 프로토콜을 제공하는 HTTP 1.1에 기반되어 있다. HTTP는 많은 태스크들을 위한 폭넓은 사용에서 일반적이고, 상태가 없고, 객체기반의 프로토콜이다. HTTP의 특징은 전송된 데이터와 독립적으로 시스템이 구성되는 것을 허용하는 데이터 표현의 분류 및 교섭이다. 바람직하기로는, 홈 네트워크 상에서 디바이스 및 어플리케이션에 의해 사용된 네트워크 프로토콜은 IP(Internet Protocol)이다. 그러나 다른 프로토콜 또한 사용될 수 있다.

본 발명이 그것으로부터 바람직한 버전과 관련하여 상당히 자세하게 하더라도, 다른 버전도 가능하다. 따라서, 첨부된 청구범위는 여기에 포함된 바람직한 버전의 설명에 의해 한정되어서는 안된다.

## (57) 청구의 범위

### 청구항 1.

네트워크를 거쳐 다수의 디바이스들 사이에서 명령하고 제어하기 위한 방법에 있어서,

- (a) 네트워크에 제1 디바이스를 접속하는 단계;
- (b) 제2 디바이스를 상기 네트워크에 접속하는 단계로써, 상기 제2 디바이스는 네트워크에 접속된 적어도 하나의 다른 디바이스들에 의해 상기 제2 디바이스를 명령하고 제어하기 위해 구조적인 포맷으로 어플리케이션 인터페이스 설명 데이터를 저장하는 단계;
- (c) 상기 네트워크를 통하여 상기 제1 디바이스에 상기 어플리케이션 인터페이스 설명 데이터를 제공하는 단계; 및
- (d) 상기 제2 디바이스의 동작을 제어하기 위해 상기 어플리케이션 인터페이스 설명 데이터를 사용하여 네트워크를 통해 상기 제1 디바이스에서 상기 제2 디바이스로 제어 및 명령 데이터를 전송하는 단계를 포함하는 방법.

### 청구항 2.

제 1항에 있어서, 상기 (c)단계는 네트워크를 거쳐 상기 어플리케이션 인터페이스 설명 데이터의 위치를 지정하는 단계, 및 네트워크를 거쳐 상기 제1 디바이스에 상기 어플리케이션 인터페이스 설명 데이터를 제공하는 단계를 더 포함하는 것을 특징으로 하는 방법.

### 청구항 3.

제 1항에 있어서,

- (i) (b) 단계는 둘 이상의 디바이스들을 상기 네트워크에 접속하는 단계로서, 각 디바이스는 상기 구조화된 포맷에서 상기 네트워크에 접속된 하나 또는 그 이상의 다른 디바이스에 의한 상기 디바이스의 명령 및 제어를 위해 어플리케이션 인터페이스 설명 데이터를 상기 구조화된 포맷에 저장하는 단계;
- (ii) (c) 단계는 상기 네트워크를 거쳐 상기 제1 디바이스로 상기 복수의 디바이스의 어플리케이션 인터페이스 설명 데이터를 제공하는 단계; 및
- (iii) (d)단계는 상기 복수의 디바이스의 동작을 제어하는 상기 복수의 디바이스 각각에 대응되는 상기 어플리케이션 인터페이스 설명 데이터를 사용하여 상기 네트워크를 통해 제1 디바이스로부터 상기 복수의 디바이스들에게 제어 및 명령 데이터를 전송하는 단계를 포함하는 것을 특징으로 하는 방법.

### 청구항 4.

제 3항에 있어서, 상기 (ii)단계는 네트워크를 거쳐 상기 어플리케이션 인터페이스 설명 데이터의 위치를 지정하는 단계, 및 상기 네트워크를 거쳐 상기 제1 디바이스에 상기 어플리케이션 인터페이스 설명 데이터의 제공하는 단계를 더 포함하는 것을 특징으로 하는 방법.

### 청구항 5.

제 3항에 있어서, 상기 (ii)단계는 적어도 상기 제1 디바이스에 상기 복수의 디바이스의 어플리케이션 인터페이스 데이터를 제공하는 단계를 더 포함하고, 상기 (iii) 단계는 상기 복수의 디바이스 등 적어도 하나의 동작을 제어하는 상기 복수의 디바이스 각각에 대응되는 상기 어플리케이션 인터페이스 설명 데이터를 사용하여 적어도 제1 디바이스로부터 상기 네트워크에 접속된 상기 복수의 디바이스에 제어 및 명령 데이터를 전송하는 것을 포함하는 것을 특징으로 하는 방법.

#### 청구항 6.

제 1항에 있어서, 상기 (c) 단계는 상기 네트워크를 거쳐 상기 제1 홈 디바이스에 상기 어플리케이션 인터페이스 설명 데이터의 적어도 일부를 전송하는 단계를 포함하는 것을 특징으로 하는 방법.

#### 청구항 7.

제 1항에 있어서, 상기 (c) 단계는 상기 네트워크를 거쳐 상기 제2 디바이스 내의 상기 어플리케이션 인터페이스 설명 데이터를 쿼리하는 상기 제1 디바이스를 포함하는 것을 특징으로 하는 방법.

#### 청구항 8.

제 1항에 있어서, 상기 (c) 단계는 상기 네트워크에 접속된 데이터베이스 디바이스로부터 상기 어플리케이션 인터페이스 설명 데이터를 쿼리하는 상기 제1 디바이스를 포함하는 것을 특징으로 하는 방법.

#### 청구항 9.

제 1항에 있어서, 상기 어플리케이션 인터페이스 설명 데이터는 상기 제2 홈 디바이스의 동작을 제어하는 상기 제1 홈 디바이스를 위한 원격 절차 호출 정보를 포함하는 것을 특징으로 하는 방법.

#### 청구항 10.

제 1항에 있어서, 상기 어플리케이션 인터페이스 설명 데이터는 상기 제2 디바이스의 성능을 확인하기 위한 성능 데이터를 포함하는 것을 특징으로 하는 방법.

#### 청구항 11.

제 1항에 있어서, 상기 디바이스들은 사용자 인터페이스를 디스플레이 할 수 없는 것을 특징으로 하는 방법.

#### 청구항 12.

제 1항에 있어서, 상기 구조화된 포맷은 XML 포맷을 포함하는 것을 특징으로 하는 방법.

#### 청구항 13.

다바이스들을 명령하고 제어하기 위한 네트워크 시스템에 있어서,

(a) 서로 통신하는 디바이스들에 의해 사용될 수 있는 통신 매체를 제공하는 물리 계층;

(b) 구조화된 포맷에서 적어도 하나의 다른 디바이스에 의해 제어된 디바이스를 명령 및 제어하기 위해 구조화된 포맷에 어플리케이션 인터페이스 설명 데이터를 포함하는 적어도 하나의 제어된 디바이스; 및

(c) 상기 어플리케이션 인터페이스 설명 데이터를 획득하기 위한 제어 어플리케이션 수단을 포함하고, 상기 제어된 디바이스의 동작을 제어하기 위해 상기 어플리케이션 인터페이스 설명 데이터를 사용하여 상기 제어된 디바이스에 제어 및 명령 데이터를 전송하는 적어도 하나의 제어기 디바이스를 포함하는 네트워크 시스템.

#### 청구항 14.

제 13항에 있어서, 복수의 제어된 디바이스들로서, 각각의 제어된 디바이스는 상기 구조화된 포맷에서 적어도 상기 제어기 디바이스에 의해 각각 제어되는 디바이스의 명령 및 제어를 위한 어플리케이션 인터페이스 설명 데이터를 저장하는, 복수의 제어된 디바이스를 더 구비하여, 상기 어플리케이션 제어수단은 하나 또는 그 이상의 상기 제어된 디바이스의 동작을 제어하기 위해 상기 어플리케이션 인터페이스 설명 데이터를 사용하여 하나 또는 그 이상의 상기 제어된 디바이스에 제어 및 명령 데이터를 전송하기 위한 하나 또는 그 이상의 상기 제어된 디바이스의 상기 어플리케이션 인터페이스 설명 데이터를 선택적으로 획득하는 것을 특징으로 하는 네트워크 시스템.

#### 청구항 15.

제 13항에 있어서, 상기 어플리케이션 제어 수단은 제어된 디바이스에 상기 제어 및 명령 데이터를 생성하고 전송하기 위해 상기 제어기 디바이스에 상기 어플리케이션 인터페이스 설명 데이터의 적어도 일부를 전송함으로써 상기 어플리케이션 인터페이스 설명 데이터를 획득하는 것을 특징으로 하는 네트워크 시스템.

#### 청구항 16.

제 13항에 있어서, 상기 어플리케이션 제어 수단은 제어된 디바이스에서 상기 어플리케이션 인터페이스 설명 데이터를 쿼리함으로써 상기 어플리케이션 인터페이스 설명 데이터를 획득하는 것을 특징으로 하는 네트워크 시스템.

#### 청구항 17.

제 13항에 있어서, 상기 어플리케이션 제어 수단은 데이터베이스 디바이스로부터 상기 어플리케이션 인터페이스 설명 데이터를 쿼리함으로써 상기 어플리케이션 인터페이스 설명 데이터를 획득하는 것을 특징으로 하는 네트워크 시스템.

#### 청구항 18.

제 13항에 있어서, 상기 어플리케이션 인터페이스 설명 데이터는 제어된 디바이스의 동작을 제어하는 제어기 디바이스를 위해 원격 절차 호출 정보를 포함하는 것을 특징으로 하는 네트워크 시스템.

#### 청구항 19.

제 13항에 있어서, 상기 어플리케이션 인터페이스 설명 데이터는 제어된 디바이스의 성능을 확인하기 위한 성능 데이터를 포함하는 것을 특징으로 하는 네트워크 시스템.

#### 청구항 20.

제 13항에 있어서, 상기 제어기 및 제어된 디바이스는 사용자 인터페이스를 표시할 수 없을 특징으로 하는 네트워크 시스템.

## 청구항 21.

제 13항에 있어서, 상기 구조화된 포맷은 XML 포맷을 포함하는 것을 특징으로 하는 네트워크 시스템.

## 청구항 22.

제 13항에 있어서, 복수의 제어기 디바이스들을 더 포함하고 각 제어 디바이스는 상기 하나 또는 그 이상의 제어된 디바이스의 동작을 제어하기 위해 상기 어플리케이션 인터페이스 설명 데이터를 사용하여 상기 하나 또는 그 이상의 제어된 디바이스에 명령 및 제어 정보를 전송하기 위한 하나 또는 그 이상의 제어기 디바이스의 어플리케이션 인터페이스 설명 데이터를 획득하는 어플리케이션 제어 수단을 포함하는 것을 특징으로 하는 네트워크 시스템.

## 청구항 23.

홈 네트워크를 거쳐 서비스를 수행하는 방법에 있어서,

(a) 클라이언트 디바이스를 상기 홈 네트워크에 접속하는 단계로서, 상기 클라이언트 디바이스는 사용자 인터페이스 데이터를 디스플레이 할 수 있는 단계;

(b) 네트워크를 통하여 상기 클라이언트 디바이스에 의해 제어가 가능한 상기 홈 네트워크에 제1 홈 디바이스를 접속하는 단계;

(c) 제2 홈 디바이스를 상기 홈 네트워크에 접속하는 단계로서, 네트워크에 접속된 하나 또는 그 이상의 홈 디바이스에 의해 제2 홈 디바이스의 디바이스 명령 및 제어를 위한 구조화된 포맷으로 어플리케이션 인터페이스 설명 데이터를 저장하는 단계; 및

(f) 상기 클라이언트 디바이스로부터 상기 서비스를 수행하는 상기 어플리케이션 인터페이스 설명 데이터를 사용하여 상기 제1 및 제2 디바이스가 서로 통신하게 하는 상기 사용자 입력에 기반을 둔 상기 제1 홈 디바이스에 제어 및 명령 데이터를 전송하는 단계를 포함하는 홈 네트워크를 거쳐 서비스를 수행하는 방법.

## 청구항 24.

제 23항에 있어서, 상기 (e) 단계는 상기 클라이언트 디바이스 상에 디스플레이되는 상기 사용자 인터페이스로부터 상기 제2 홈 네트워크를 선택하기 위해 사용자 입력을 수용하는 단계를 더 포함하는 것을 특징으로 하는 방법.

## 청구항 25.

제 24항에 있어서, 상기 제1 홈 디바이스는 클라이언트 디바이스를 거쳐 상기 제1 홈 디바이스로의 상기 사용자 입력에 기반을 둔 상기 어플리케이션 인터페이스 설명 데이터를 사용하여 상기 제2 홈 디바이스에 제어 및 명령 정보를 전송함으로써 상기 제2 홈 디바이스를 제어하는 단계를 더 포함하는 것을 특징으로 하는 방법.

## 청구항 26.



제 25항에 있어서, 네트워크를 거쳐 제1 홈 디바이스에 상기 어플리케이션 인터페이스 설명 데이터를 제공하는 단계를 더 포함하는 것을 특징으로 하는 방법.

## 청구항 27.

제 23항에 있어서, 상기 어플리케이션 인터페이스 설명 데이터는 상기 제2 홈 디바이스를 위한 성능 데이터를 포함하고, (i) 상기 제2 홈 디바이스의 상기 어플리케이션 인터페이스 설명 데이터 내부에서 호환 데이터를 쿼리하는 단계, 및 (ii) 상기 클라이언트 디바이스 상에 디스플레이된 상기 제1 홈 디바이스의 사용자 인터페이스를 통하여 사용자에게 의해 상기 제2 홈 디바이스의 명령 및 제어를 허용하기 위해 성능 데이터를 사용하여 상기 제1 홈 디바이스의 상기 사용자 인터페이스 데이터를 갱신하는 단계를 더 포함하는 것을 특징으로 하는 방법.

## 청구항 28.

제 23항에 있어서, 둘 또는 그 이상의 홈 디바이스를 상기 네트워크에 접속하는 단계로서, 각 홈 디바이스는 구조화된 포맷에서 네트워크에 접속된 하나 또는 그 이상의 다른 홈 디바이스에 의해 상기 홈 디바이스의 명령 및 제어를 위해 어플리케이션 인터페이스 설명 데이터를 저장하는 둘 또는 그 이상의 홈 디바이스를 네트워크에 접속하는 단계를 더 포함하는 것을 특징으로 하는 방법.

## 청구항 29.

제 28항에 있어서, 상기 둘 또는 그 이상의 홈 디바이스의 각각에서 상기 어플리케이션 인터페이스 설명 데이터는 각 홈 디바이스를 위한 성능 데이터를 포함하고, (i) 상기 둘 또는 그 이상의 홈 디바이스의 상기 어플리케이션 인터페이스 설명 데이터 내부에서 호환 데이터를 쿼리하는 단계, 및 (ii) 상기 클라이언트 디바이스 상에 디스플레이된 상기 제1 홈 디바이스의 사용자 인터페이스를 통하여 사용자에게 의해 상기 둘 또는 그 이상의 홈 디바이스의 명령 및 제어를 허용하기 위해 성능 데이터를 사용하여 제1 홈 디바이스의 상기 사용자 인터페이스 데이터를 갱신하는 단계를 더 포함하는 것을 특징으로 하는 방법.

## 청구항 30.

제 28항에 있어서, 네트워크를 거쳐 상기 제1 홈 디바이스에 복수의 상기 둘 또는 그 이상의 홈 디바이스의 상기 어플리케이션 인터페이스 설명을 제공하는 단계를 더 포함하는 것을 특징으로 하는 방법.

## 청구항 31.

제 30항에 있어서, 상기 복수의 홈 디바이스들의 동작을 제어하는 상기 복수의 홈 디바이스들 각각에 상응하는 상기 어플리케이션 인터페이스 설명 데이터를 사용하여 네트워크를 거쳐 상기 제1 홈 디바이스로부터 상기 복수의 홈 디바이스에 제어 및 명령 데이터를 전송하는 단계를 더 포함하는 것을 특징으로 하는 방법.

## 청구항 32.

제 30항에 있어서, 네트워크를 거쳐 상기 어플리케이션 인터페이스 설명의 위치를 지정하고, 데이터를 네트워크를 거쳐 상기 제1 홈 디바이스에 상기 어플리케이션 인터페이스 설명 데이터를 제공하는 단계를 더 포함하는 것을 특징으로 하는 방법.

### 청구항 33.

제 30항에 있어서, 상기 복수의 홈 디바이스들 중의 다른 홈 디바이스에게 상기 복수의 홈 디바이스들 중 하나의 홈 디바이스의 상기 어플리케이션 인터페이스 설명 데이터를 제공하는 단계를 더 포함하는 것을 특징으로 하는 방법.

### 청구항 34.

제 33항에 있어서, 상기 다른 홈 디바이스의 동작을 제어하는 상기 또 다른 홈 디바이스에 상응하는 상기 어플리케이션 인터페이스 설명을 사용하여 상기 네트워크를 거쳐 상기 하나의 홈 디바이스로부터 상기 또 다른 홈 디바이스에 제어 및 명령 데이터를 전송 하는 단계를 더 포함하는 것을 특징으로 하는 방법.

### 청구항 35.

제 23항에 있어서, 상기 어플리케이션 인터페이스 설명은 상기 제2 홈 디바이스의 동작을 제어하는 상기 제1 홈 디바이스를 위한 원격 절차 호출 정보를 포함하는 것을 특징으로 하는 방법.

### 청구항 36.

제 35항에 있어서, 상기 어플리케이션 인터페이스 설명은 상기 제2 디바이스의 성능을 확인하기 위한 성능 데이터를 포함하는 것을 특징으로 하는 방법.

### 청구항 37.

제 23항에 있어서, 상기 선택된 포맷은 HTML 포맷을 포함하는 것을 특징으로 하는 방법.

### 청구항 38.

제 23항에 있어서, 상기 구조화된 포맷은 XML 포맷을 포함하는 것을 특징으로 하는 방법.

### 청구항 39.

디바이스들을 명령하고 제어하기 위한 네트워크 시스템에 있어서,

- (a) 서로 통신하는 디바이스들에 의해 사용될 수 있는 통신 매체를 제공하는 물리 층;
- (b) 사용자에게 의해 적어도 제1 디바이스의 사용자 명령 및 제어를 위한 사용자 인터페이스를 정의하는 선택된 포맷내의 사용자 인터페이스 데이터를 저장하는 제1 서버 디바이스;
- (c) 하나 또는 그 이상의 디바이스에 의해 상기 제2 서버 디바이스의 디바이스 명령 및 제어를 위한 구조화된 포맷에서 어플리케이션 인터페이스 설명 데이터를 저장하는 제2 서버 디바이스; 및
- (d) 사용자 인터페이스 데이터의 디스플레이를 가능하게 하고, 사용자로부터 입력을 수용하기 위해 상기 클라이언트 디바이스 상에서 상기 제1 서버 디바이스의 상기 사용자 인터페이스를 디스플레이하기 위한 사용자 인터페이스 제어기를 포함하고, 사용자에게 의해 요청된 서비스를 수행하기 위해 상기 어플리케이션 인터페이스 설명 데이터를 사용하여 상기 제1 및 제2 서버 디바이스가 서로에게 통신하도록 하기 위해, 사용자 입력에 기반을 둔 제1 서버 디바이스에 제어 및 명령 데이터를 전송하는, 클라이언트 디바이스를 포함하는 디바이스들을 명령하고 제어하기 위한 네트워크 시스템.

#### 청구항 40.

제 39항에 있어서, 상기 사용자 인터페이스 제어기는 상기 클라이언트 디바이스 상에 디스플레이된 사용자 인터페이스로부터 상기 제2 서버 디바이스 선택을 위한 사용자 입력을 수용하는 것을 특징으로 하는 네트워크 시스템.

#### 청구항 41.

제 40항에 있어서, 상기 제1 서버 디바이스는 상기 클라이언트 디바이스를 거쳐 상기 제1 서버 디바이스로의 사용자 입력에 기반을 둔 상기 어플리케이션 인터페이스 설명 데이터를 사용하여 상기 제2 서버 디바이스에 제어 및 명령 정보를 전송함으로써 제2 서버 디바이스를 제어하기 위한 어플리케이션 제어 수단을 포함하는 것을 특징으로 하는 네트워크 시스템.

#### 청구항 42.

제 41항에 있어서, 상기 어플리케이션 제어 수단은 상기 제1 서버 디바이스로부터 상기 어플리케이션 인터페이스 설명 데이터를 획득하는 것을 특징으로 하는 네트워크 시스템.

#### 청구항 43.

제 40항에 있어서, 상기 어플리케이션 제어 수단은 데이터베이스로부터 상기 어플리케이션 인터페이스 설명 데이터를 획득하는 것을 특징으로 하는 네트워크 시스템.

#### 청구항 44.

제 39항에 있어서, 상기 어플리케이션 설명 데이터는 상기 제2 서버 디바이스를 위한 성능 데이터를 포함하고, 상기 어플리케이션 제어 수단은 상기 어플리케이션 인터페이스 설명으로부터 상기 성능 데이터를 획득하고, 상기 클라이언트 디바이스 상에 디스플레이된 상기 제1 서버 디바이스의 사용자 인터페이스를 거쳐 사용자에게 의한 상기 제2 서버 디바이스의 명령 및 제어를 허용하는 성능 데이터를 사용하여 상기 제1 홈 디바이스 내에서 상기 사용자 인터페이스 데이터를 갱신하는 것을 특징으로 하는 네트워크 시스템.

#### 청구항 45.

제 39항에 있어서, 상기 구조화된 포맷에서 하나 또는 그 이상의 디바이스에 의해 상기 둘 또는 그 이상의 서버 디바이스의 명령 및 제어를 위해 어플리케이션 인터페이스 설명 데이터를 각각 저장하는 둘 또는 그 이상의 서버 디바이스를 더 포함하는 것을 특징으로 하는 네트워크 시스템.

#### 청구항 46.

제 45항에 있어서, 상기 둘 또는 그 이상의 서버 디바이스 각각에서 상기 어플리케이션 인터페이스 설명 데이터는 각각의 서버 디바이스를 위한 성능 데이터를 포함하고, 상기 어플리케이션 제어 수단은 상기 둘 또는 그 이상의 서버 디바이스들의 상기 어플리케이션 인터페이스 데이터로부터 상기 성능 데이터를 획득하고, 상기 클라이언트 디바이스 상에 디스플레이된 상기 제1 서버 디바이스의 사용자 인터페이스를 거쳐 사용자에게 의한 상기 둘 또는 그 이상의 서버 디바이스들의 명령 및 제어를 허용하기 위해 상기 성능 데이터를 사용하여 상기 제1 디바이스 내에서 상기 사용자 인터페이스를 업데이트하는 것을 특징으로 하는 네트워크 시스템.

#### 청구항 47.

제 45항에 있어서, 상기 어플리케이션 제어 수단은 상기 둘 또는 그 이상의 서버 디바이스의 동작을 제어하기 위해 상기 둘 또는 그 이상의 서버 디바이스 각각에 상응하는 상기 어플리케이션 인터페이스 설명 데이터를 사용하여 상기 둘 또는 그 이상의 서버 디바이스들에게 제어 및 명령 데이터를 전송하는 것을 특징으로 하는 네트워크 시스템.

#### 청구항 48.

제 39항에 있어서, 상기 어플리케이션 인터페이스 설명 데이터는 상기 제1 서버 디바이스가 상기 제2 서버 디바이스의 동작을 제어하기 위해 원격 절차 호출 정보를 포함하는 것을 특징으로 하는 네트워크 시스템.

#### 청구항 49.

제 39항에 있어서, 상기 선택된 포맷은 HTML 포맷을 포함하는 것을 특징으로 하는 네트워크 시스템.

#### 청구항 50.

제 39항에 있어서, 상기 구조화된 포맷은 XML 포맷을 포함하는 것을 특징으로 하는 네트워크 시스템.

#### 청구항 51.

제 23항에 있어서, 상기 (b)단계에서, 상기 제1 홈 디바이스는 네트워크를 통하여 사용자에게 의해 적어도 제1 홈 디바이스의 사용자 명령 및 제어를 위한 사용자 인터페이스를 규정하는 선택된 포맷으로 사용자 인터페이스를 저장하는 것을 특징으로 하는 홈 네트워크를 거쳐 서비스를 수행하는 방법.

#### 청구항 52.

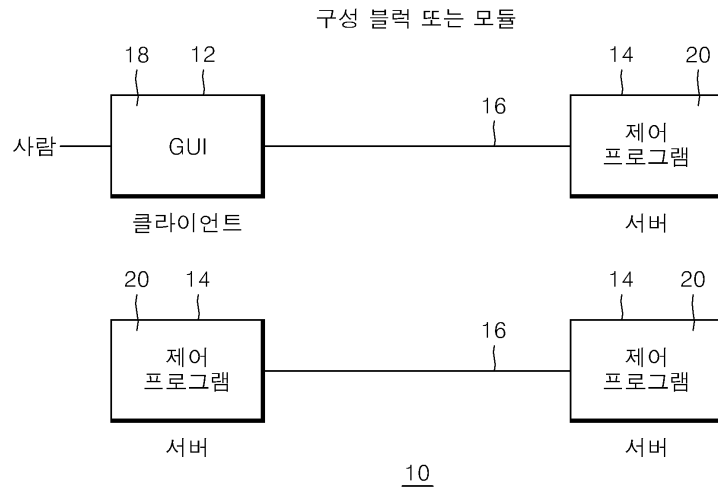
제 23항에 있어서,

(d) 상기 홈 네트워크를 거쳐 상기 클라이언트 디바이스에서 상기 제1 홈 디바이스의 상기 사용자 인터페이스 데이터를 수신하는 단계; 및

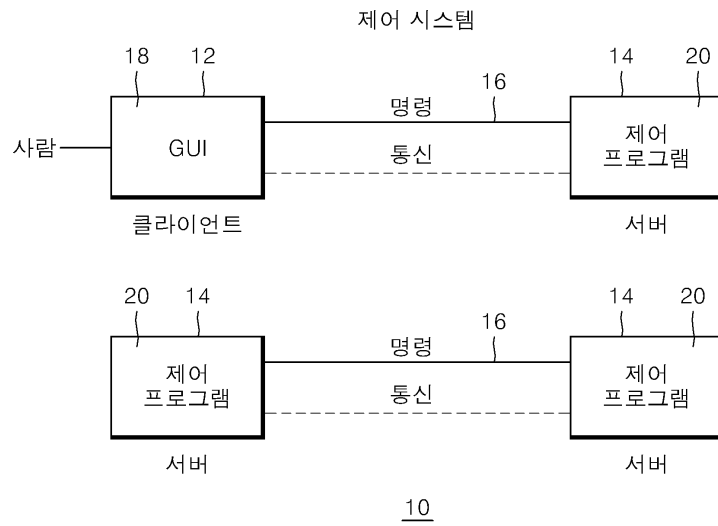
(e) 상기 클라이언트 디바이스 상에 디스플레이 된 상기 사용자 인터페이스와 상호작용 하는 사용자에게 응답하여 상기 사용자로부터 사용자 입력을 수용하는 단계를 더 포함하는 것을 특징으로 하는 홈 네트워크를 거쳐 서비스를 수행하는 방법.

도면

도면1

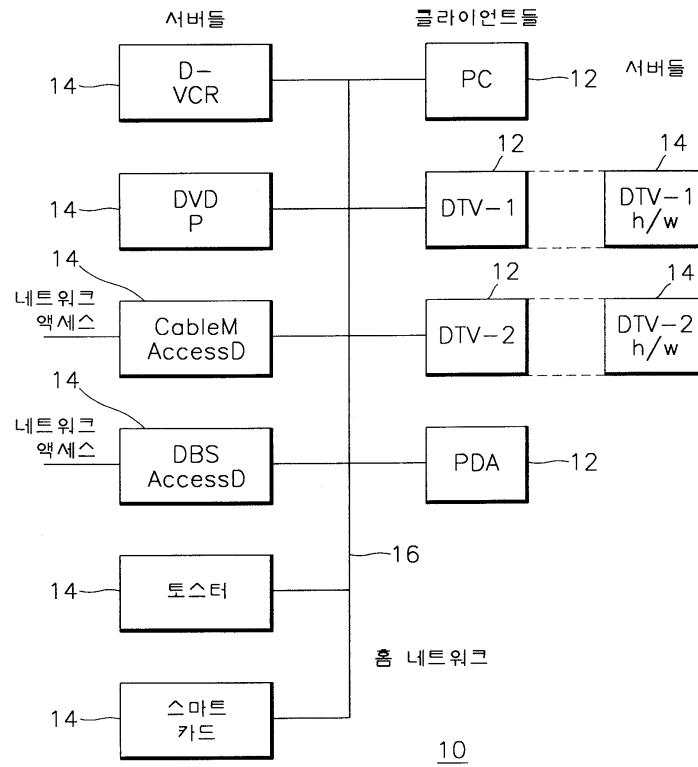


도면2



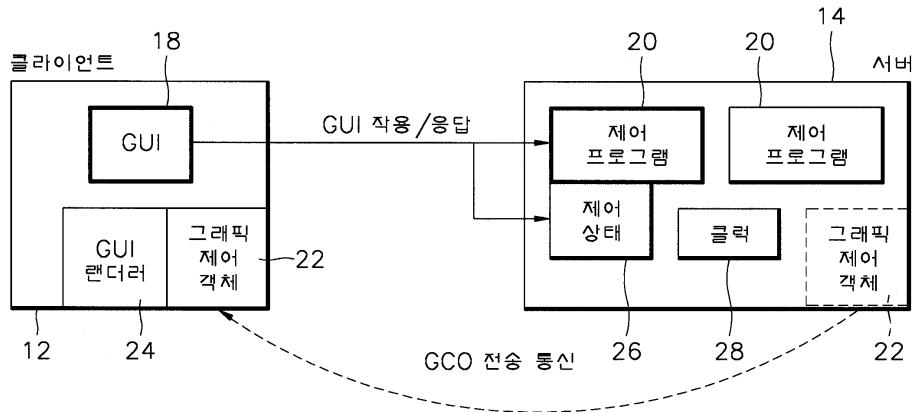
도면3

다수의 클라이언트 다수의 서버 홈 네트워크 제어 모델



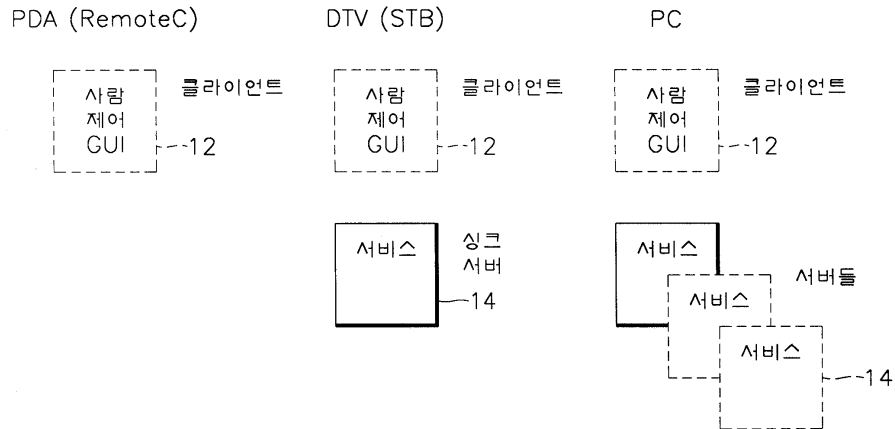
도면4

모듈 구성 GUI 제어



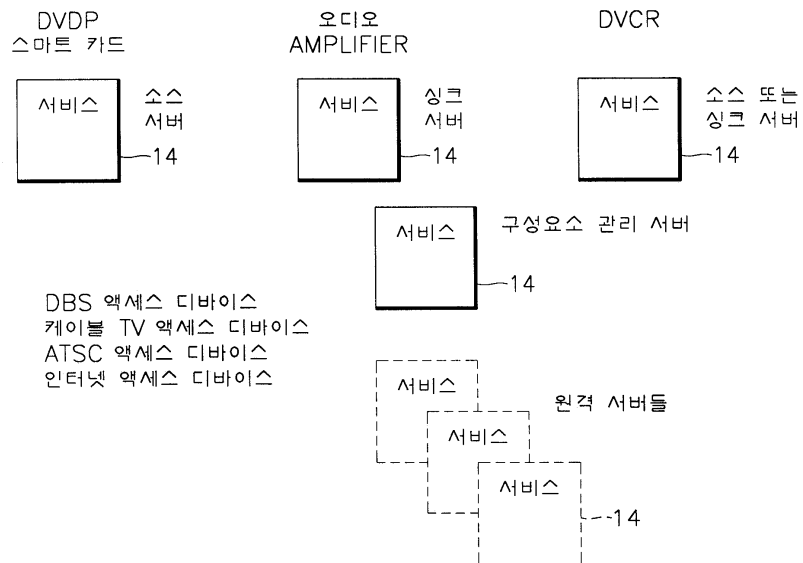
도면5

클라이언트들의 구성 요소



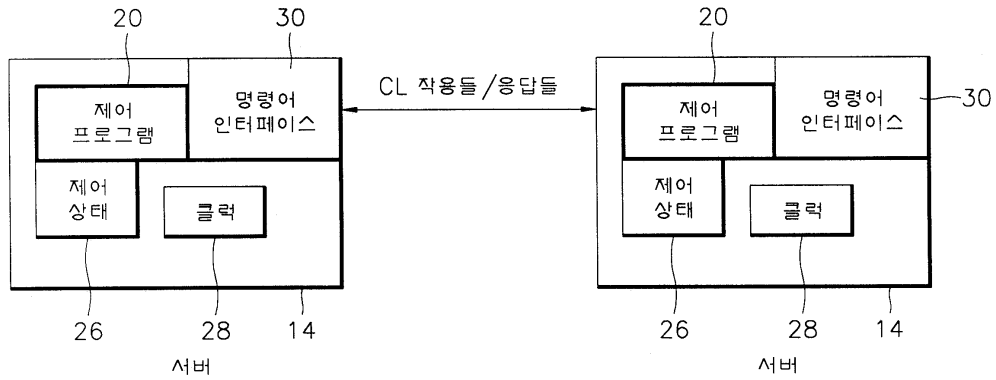
도면6

서버의 구성 요소 (Boxes)



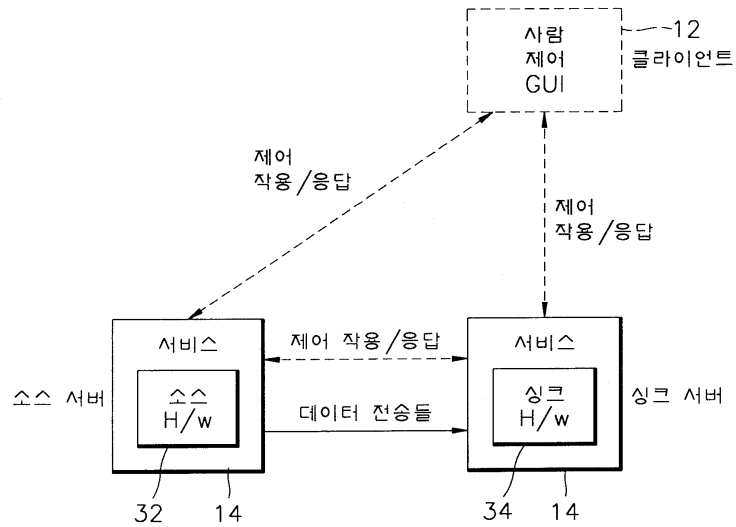
도면7

서버/서버 CL을 위한 모듈 구성들



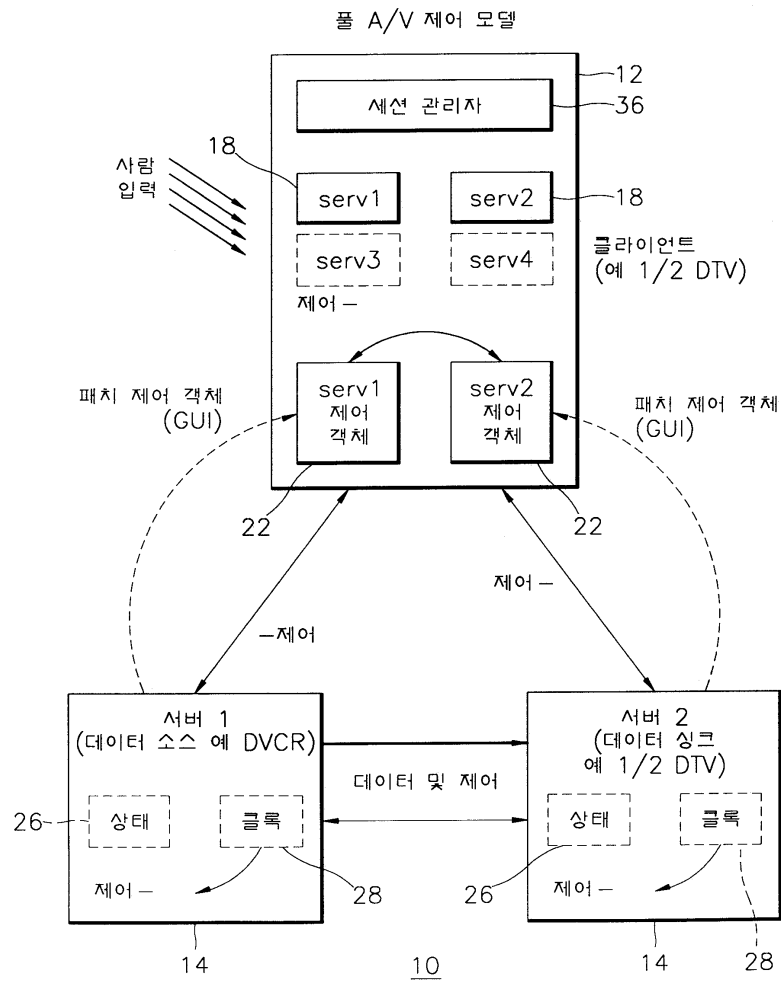
도면8

기본 A/V, 클라이언트/서버/서버, 모델





도면9



도면10

table 1 : 성능 테이블

Service	Sink/Source
Service A	Sink
Service B	Source
Service C	Source
Service D	Sink

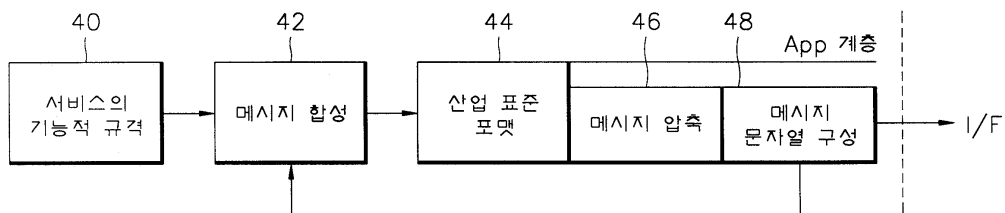
도면11

Tovle 2 : 특성 테이블

이름	길이	값
DeviceManufacturer	20 chars	Device manufacturer's name
ManufacturerURL	60 chars	device manufacturer's home page URL
manufacturerIcon	20 chars	Name of Device manufacturer's Icon
DeviceName	30 chars	Device name
DeviceModel	20 chars	Device model number
DeviceType	20 chars	device category
DeviceLocation	30 chars	device group or location
DefaultSource	15 chars	Data type, Default source device (IP address)
DefaultSink	15 chars	Data type, Default sink device (IP address)
DeviceIcon	20 chars	Name of device icon

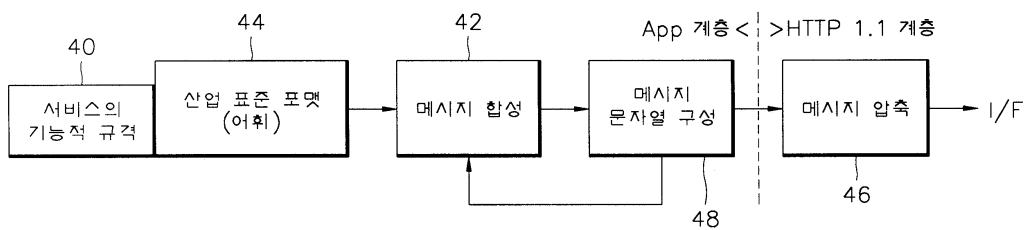
도면12

CAL 및 AV/C에 의해 사용된 메시지 구성

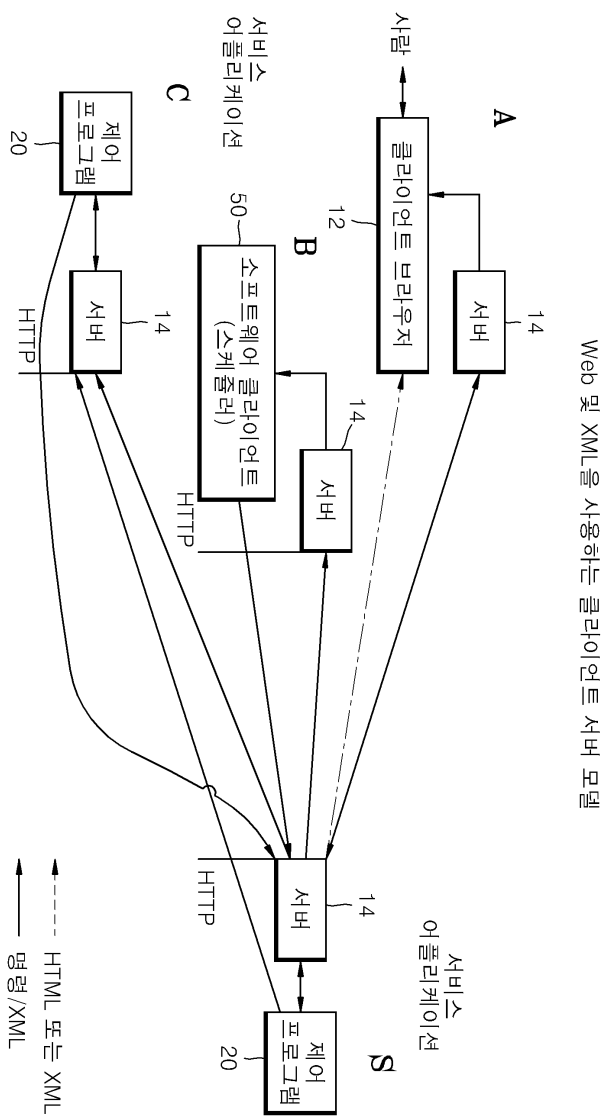


도면13

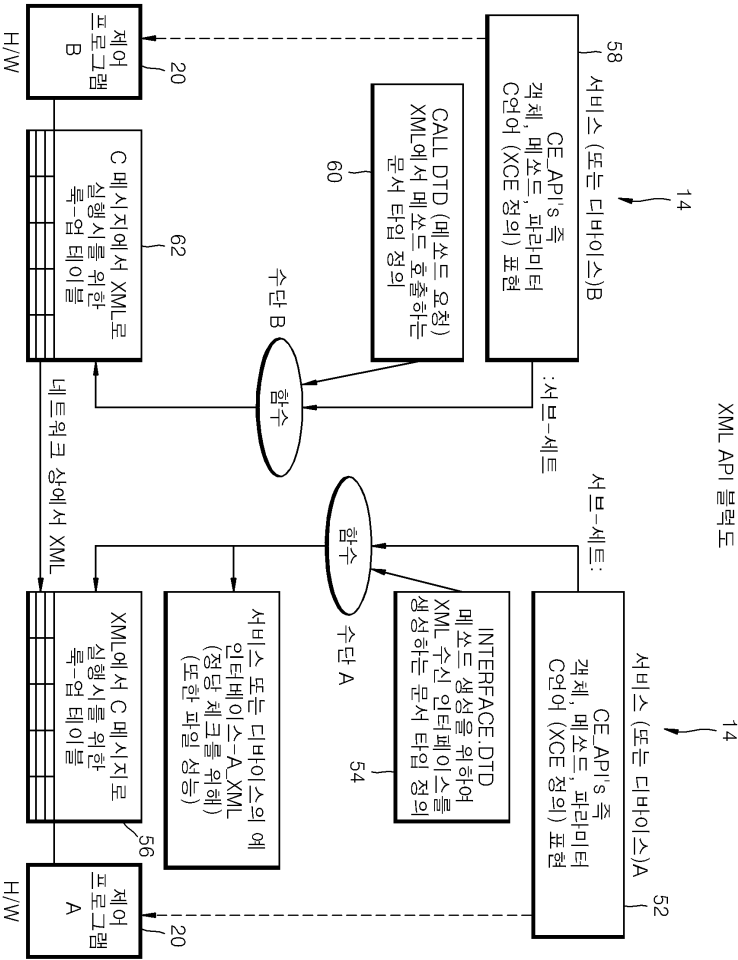
새로운 XCE/XML의 메시지 구성



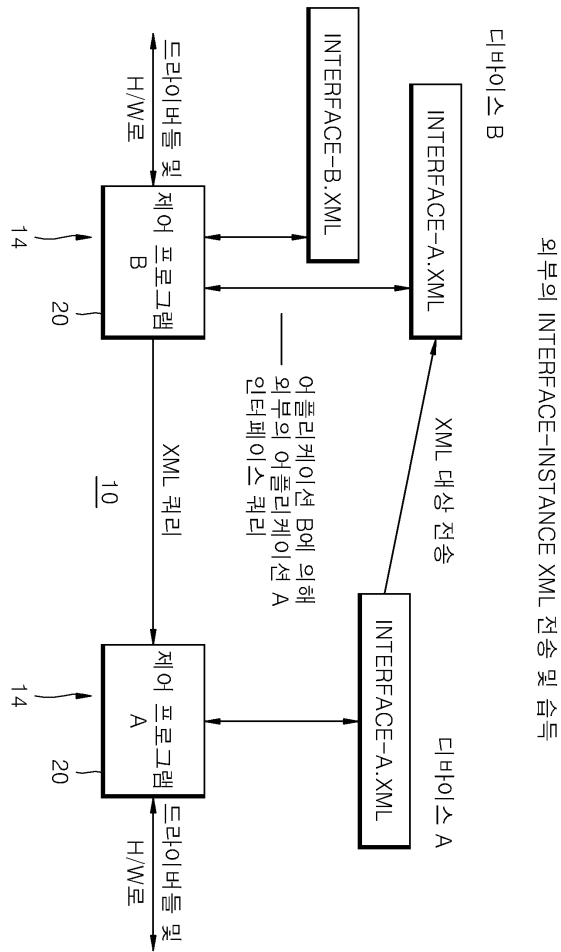
도면14



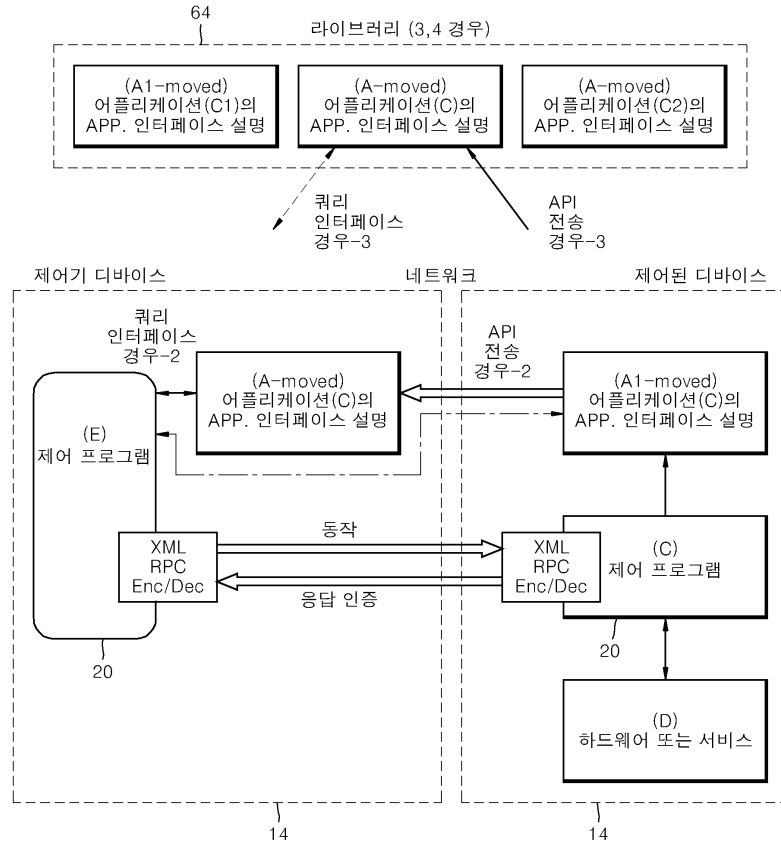
도면15



도면16

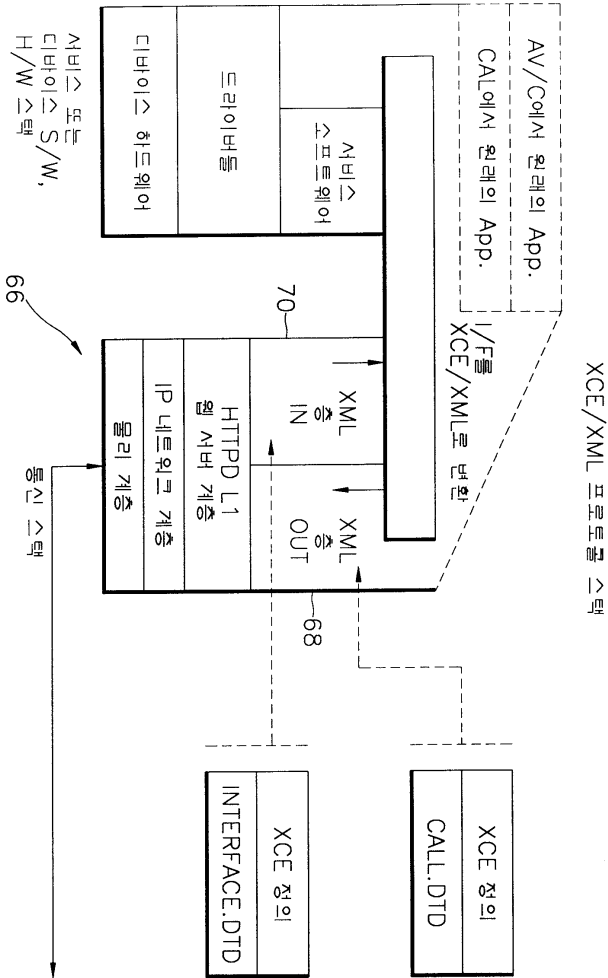


도면17

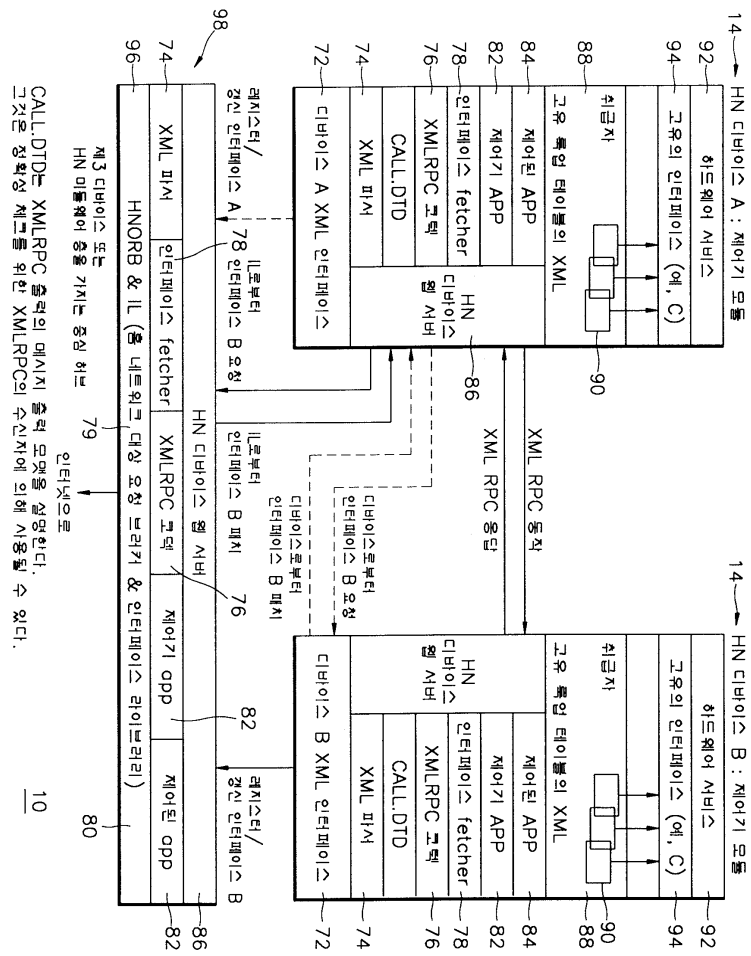


- 경우1. (E)에 의해 원격으로 쿼리하기 위한 XML 어플리케이션-C 인터페이스 (A)
- 경우2. (E)에 의해 국부적으로 쿼리하기 위한 XML 어플리케이션-C 인터페이스 (A)를 제어기로 이동
- 경우3. (E)에 의해 원격으로 쿼리하기 위한 XML 어플리케이션-C 인터페이스 (A)를 제3 디바이스-모든 인터페이스를 위한 집합장소 (라이브러리)로 이동  
라이브러리는 직접 제어 동작 및 응답이 유용하도록 관련된 어플리케이션의 주소(URL) 가지고 있어야 한다
- 경우4. 경우 3과 같지만 '간접적인' 제어 동작(및 응답)이 라이브러리 디바이스에서 또한 명령된다  
이경우에 라이브러리는 '송달자'와 함께 만들어진다

도면18

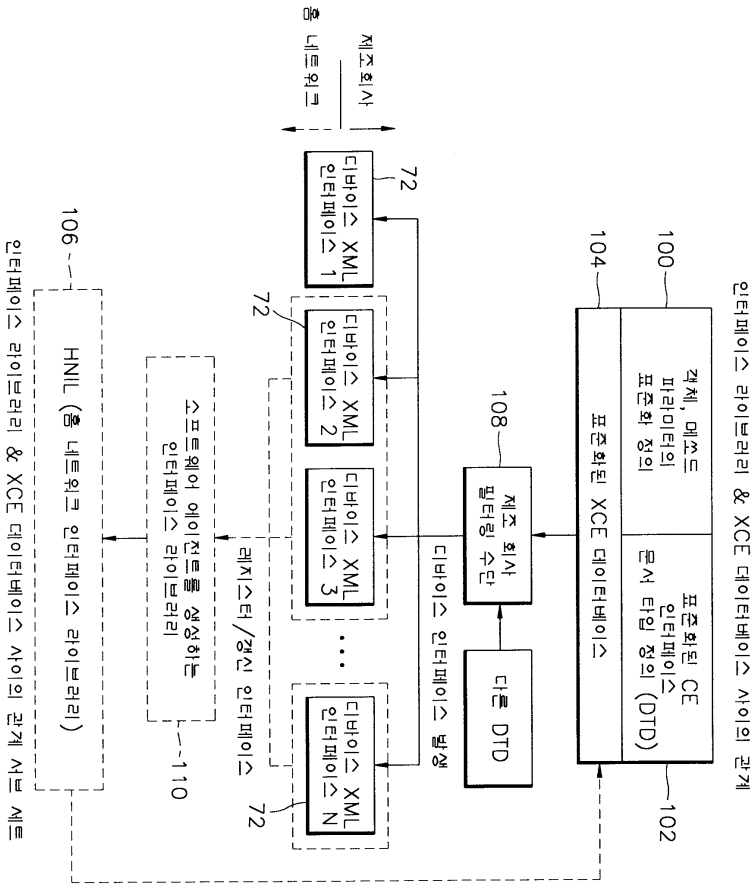


도면19

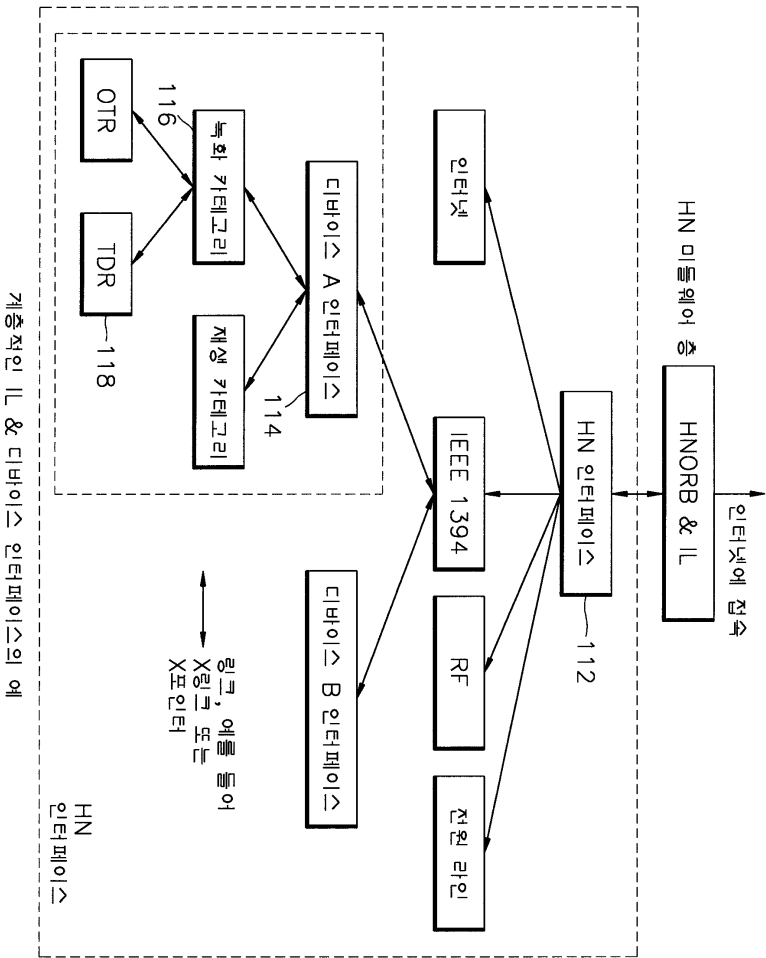




도면20



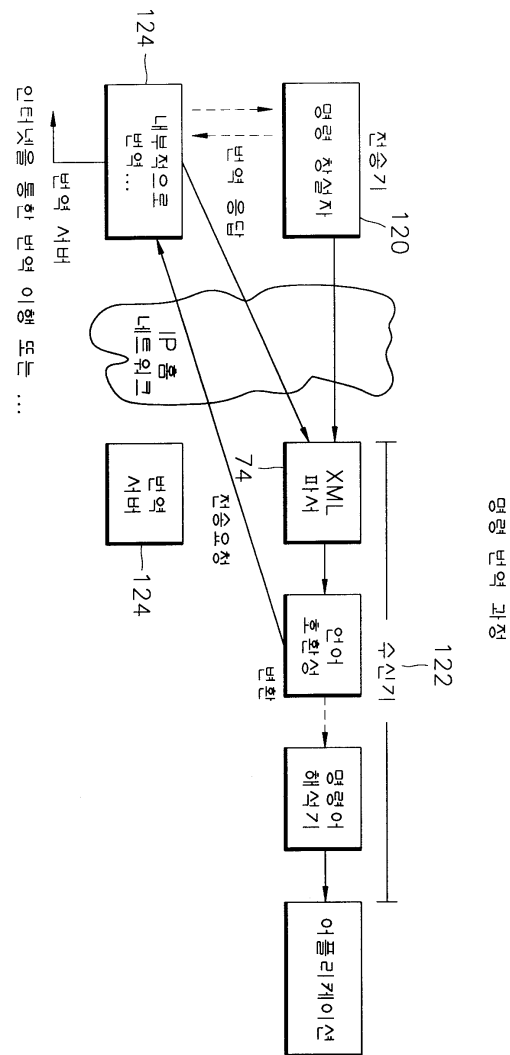
도면21



도면22

계층적인 인터페이스 구조, 4계층			
인터페이스 계층		인터페이스 예	
112 —	1	각 HN을 위한 XML 인터페이스, 현재 유용한 디바이스 리스트	My HN (VCR, TV ...)
114 —	2	각 디바이스를 위한 일반적인 인터페이스, 기능 카테고리 리스트	일반적인 VCR 인터페이스 (녹화, 재생...)
116 —	3	디바이스에 대한 각 기능 카테고리를 위한 XML 인터페이스 규칙	VCR. 녹화 카테고리
118 —	4	기능 카테고리에서 각 기능을 위한 XML 인터페이스 규칙	VCR. 녹화. tdr 기능

도면23



도면24

테이블 3 : 포맷과 타입 예

타입	포맷
명령	CAL, AV/C, X-10
영상	jpeg, gif, bitmap, tiff
언어	English, French, etc
비디오_스트림	mpeg2
비디오_클립	avi, quickime, mpeg
텍스트	html, plain
오디오	wav, aiff
어플리케이션	maword, pdf, postscript, gzip