



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2008-0098004
(43) 공개일자 2008년11월06일

(51) Int. Cl.

G06F 15/16 (2006.01) G06F 9/44 (2006.01)

(21) 출원번호 10-2008-7018118

(22) 출원일자 2008년07월23일

심사청구일자 없음

번역문제출일자 2008년07월23일

(86) 국제출원번호 PCT/US2007/000154

국제출원일자 2007년01월03일

(87) 국제공개번호 WO 2007/089385

국제공개일자 2007년08월09일

(30) 우선권주장

11/275,806 2006년01월30일 미국(US)

(71) 출원인

마이크로소프트 코포레이션

미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
마이크로소프트 웨이

(72) 발명자

그리프, 에사이아스, 이.

미국 98052-6399 워싱턴주 레드몬드 원 마이크로
소프트 웨이

(74) 대리인

양영준, 백만기

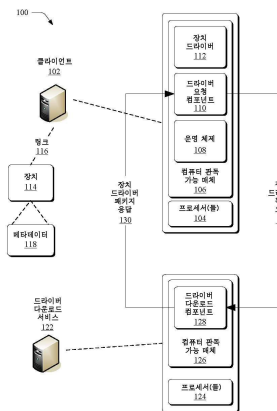
전체 청구항 수 : 총 20 항

(54) 장치 드라이버에 대한 다운로드 서비스를 제공하기 위한시스템, 방법 및 스키마

(57) 요약

장치 드라이버에 대한 다운로드 서비스를 제공하기 위한 시스템 및 방법은 클라이언트 및 드라이버 다운로드 서비스를 포함한다. 클라이언트는 드라이버 다운로드 서비스로부터 장치에 대한 드라이버가 설치될 수 있는 위치를 요청한다. 드라이버 다운로드 서비스는 장치 드라이버가 설치될 수 있는 위치로 응답한다. 그리고, 클라이언트는 각각의 위치로부터 장치 드라이버를 요청한다.

대표도 - 도1



특허청구의 범위

청구항 1

장치에, 상기 장치에 대한 장치 드라이버를 페치(fetch)하는 적어도 하나의 종점(endpoint)에 대한 제1 요청을 전송하고, 적어도 상기 장치 드라이버에 대한 제2 요청을 상기 종점으로 전송하도록 구성되는 드라이버 요청 컴포넌트를 포함하는 적어도 하나의 클라이언트; 및

상기 제2 요청을 수신하고, 상기 제2 요청에 대한 응답을 제공하기 위한 적어도 하나의 드라이버 다운로드 서비스

를 포함하는 시스템.

청구항 2

제1항에 있어서,

상기 드라이버 요청 컴포넌트는, 상기 드라이버의 위치가 요청되는 장치를 나타내는 장치 식별자(ID), 상기 드라이버가 요청되는 운영 체제를 나타내는 운영 체제 ID, 및 상기 드라이버가 요청되는 프로세서 아키텍처(architecture) 유형을 나타내는 아키텍처 ID 중 적어도 하나를 상기 제2 요청에 포함시키도록 구성되는 시스템.

청구항 3

제1항에 있어서,

상기 드라이버 요청 컴포넌트는 상기 드라이버에 의해 지원되는 인간이 판독 가능한 언어를 나타내는 요청 언어 파라미터(requested language parameter)를 상기 제2 요청에 포함시키도록 구성되는 시스템.

청구항 4

제1항에 있어서,

상기 드라이버 요청 컴포넌트는, 상기 장치 드라이버를 포함하는 적어도 하나의 파일을 상기 응답으로부터 추출하고 상기 장치 드라이버를 상기 클라이언트에 설치하도록 구성되는 시스템.

청구항 5

제1항에 있어서,

상기 드라이버 요청 컴포넌트는, 상기 장치 드라이버를 포함하는 위치를 상기 응답으로부터 추출하고, 상기 장치 드라이버를 포함하는 적어도 하나의 파일을 상기 위치로부터 액세스하고, 상기 장치 드라이버를 상기 클라이언트에 설치하도록 구성되는 시스템.

청구항 6

제1항에 있어서,

상기 장치를 더 포함하며,

상기 장치는 상기 장치 드라이버가 요청될 수 있는 적어도 하나의 종점을 특징하는 메타데이터를 포함하고,

상기 장치는 상기 제1 요청에 응답하여 상기 클라이언트에게 상기 메타데이터를 제공하도록 구성되는 시스템.

청구항 7

제1항에 있어서,

상기 드라이버 다운로드 서비스는, 상기 제2 요청을 수신하고 상기 제2 요청에 대한 응답을 제공하도록 구성되는 드라이버 다운로드 컴포넌트를 포함하는 시스템.

청구항 8

제1항에 있어서,

클라이언트 서비스는 상기 장치로부터 획득되는 메타데이터를 참조하도록 구성되며,

상기 메타데이터는 상기 장치 드라이버가 요청될 수 있는 적어도 하나의 종점을 특정하는 시스템.

청구항 9

제1항에 있어서,

상기 클라이언트는 상기 장치로부터 획득되는 메타데이터를 참조하도록 구성되며,

상기 메타데이터는 상기 드라이버가 요청될 수 있는 종점으로서,

장치 종점;

상기 클라이언트와 연관된 관리 종점(administrative endpoint); 및

상기 장치의 제조업자와 연관된 제조업자 종점

중 적어도 하나의 종점을 특정하는 시스템.

청구항 10

제1항에 있어서,

상기 드라이버 다운로드 서비스는 상기 장치 드라이버에 대한 적어도 하나의 파일을 포함하는 응답을 전송하도록 구성되는 시스템.

청구항 11

제1항에 있어서,

상기 드라이버 다운로드 서비스는 상기 장치 드라이버가 설치될 수 있는 적어도 하나의 위치를 포함하는 응답을 전송하도록 구성되는 시스템.

청구항 12

제1항에 있어서,

상기 드라이버 다운로드 서비스는 상기 장치 드라이버에 관련된 설치 정보를 포함하는 응답을 전송하도록 구성되는 시스템.

청구항 13

컴퓨터 기반 시스템에 의해 적어도 부분적으로 실행 가능한 방법으로서,

장치로부터 상기 장치에 대한 장치 드라이버가 요청될 수 있는 적어도 하나의 종점을 특정하는 메타데이터를 요청하는 단계; 및

적어도 상기 메타데이터에서 특정된 종점으로부터 상기 장치 드라이버를 요청하는 단계를 포함하는 방법.

청구항 14

제13항에 있어서,

상기 드라이버에 대한 장치 식별자, 운영 체제 및 프로세서 아키텍처 중 적어도 하나를 특정하는 단계를 더 포함하는 방법.

청구항 15

제13항에 있어서,

상기 장치 드라이버에 대한 요청 언어(requested language)를 특정하는 단계를 더 포함하는 방법.

청구항 16

제13항의 방법을 적어도 부분적으로 구현하는 스키마(schema).

청구항 17

컴퓨터 기반 시스템에 의해 적어도 부분적으로 실행 가능한 방법으로서,
장치에 대한 장치 드라이버 요청을 수신하는 단계; 및
상기 요청에 대한 응답을 전송하는 단계
를 포함하는 방법.

청구항 18

제17항에 있어서,
상기 방법은 상기 장치에 의해 제공되는 메타데이터에서 특정된 적어도 하나의 종점에 의해 행해지며,
상기 메타데이터는 상기 드라이버가 요청될 수 있는 위치로서,
상기 장치;
클라이언트와 연관된 관리 종점; 및
상기 장치의 제조업자와 연관된 제조업자 종점
의 위치들 중 적어도 하나의 위치를 특정하는 방법.

청구항 19

제17항의 방법을 적어도 부분적으로 구현하는 스키마.

청구항 20

제17항에 있어서,
상기 응답을 전송하는 단계는,
상기 장치 드라이버가 이용 불가능함을 나타내는 메시지;
상기 장치 드라이버에 대한 적어도 하나의 파일; 또는
상기 장치 드라이버에 대한 적어도 하나의 파일이 폐치될 수 있는 위치
중 하나를 포함하는 응답을 전송하는 단계를 포함하는 방법.

명세서

배경 기술

- <1> 현대 컴퓨팅 시스템은, 주변 장치가 컴퓨팅 시스템에 직접 접속되어 있든 아니면 근거리 통신망 또는 원거리 통신망을 통해 컴퓨터 시스템에 접속 가능하게 되어 있든 간에, 이러한 주변 장치를 자주 사용하곤 한다. 통상적으로, 이러한 주변 장치는 컴퓨팅 시스템이 그 주변 장치를 전체적으로 이용할 수 있게 하는 소프트웨어 패키지인 관련 장치 드라이버(associated device driver)를 포함한다.
- <2> 일반적으로, 주변 장치의 제조업자는 예를 들면 CD(compact disk), 플로피 또는 기타 기계 판독 가능 매체에 장치 드라이버를 제공하고, 이러한 매체를 주변 장치와 함께 패키지에 포함시킨다. 컴퓨팅 시스템에 새로운 주변 장치를 설치할 때, 사용자는 통상적으로 컴퓨팅 시스템에 매체를 로드시켜, 운영 체제 또는 기타 유틸리티가 매체로부터 장치 드라이버를 판독하고 장치 드라이버를 설치할 수 있게 한다.
- <3> 상기 접근 방식은 일부 환경에서는 효과를 발휘하지만, 몇 가지 단점이 있다. 새로운 주변 장치를 설치할 때, 또는 하나의 기계로부터 다른 기계로 주변 장치를 이동할 때, 사용자에게는 드라이버를 로드해야 한다는 부담이 있다. 그러나, 드라이버를 포함하는 매체가 시간이 경과됨에 따라 손실되거나 손상될 수 있어서, 이 때문에 드

라이버를 액세스할 수 없게 되기도 한다. 또한, 주변 장치의 제조업자는 통상적으로 장치가 제조될 때 매체를 로드한다. 그러나, 장치 드라이버는 주변 장치의 예상 수명 기간 동안 빈번하게 갱신되어, 원래 공급된 장치 드라이버를 쓸모없게 할 수 있다.

<4> 상기한 문제점을 다루기 위한 한가지 접근 방식은, 제조업자가 예를 들면, 인터넷을 통해 액세스 가능한 웹사이트 상에 갱신된 장치 드라이버를 제공하는 것이다. 숙련되거나 정통한 사용자에게 대해서는 효과를 발휘하지만, 많은 사용자에게 있어서 제조업자 웹사이트로부터 장치 드라이버를 찾고, 액세스하고, 로드하는 프로세스는 여전히 위압적이고 에러를 발생시키기 쉬울 수 있다. 예를 들면, 각각의 제조업자가 자신의 웹사이트를 상이하게 조직하고, 각각의 웹사이트는 장치 드라이버를 찾고, 다운로드하고, 설치하기 위한 상이한 인터페이스를 제공할 수 있다. 상이한 제조업자들로부터 다양한 장치 드라이버의 위치를 찾고 이를 설치해야만 하는 사용자는 좌절감과 혼란감을 느끼게 될 수 있다. 또한, 이러한 접근 방식은 사용자가 인터넷 접속성을 갖는 경우에만 적합하고, 만약 사용자가 장치 드라이버를 포함하는 원본 매체를 분실했고, 또한 인터넷 접속성도 갖지 않는 경우에는 사용자가 곤경에 빠질 수 있다.

<5> 다른 접근 방식에서는, 운영 체제 또는 기타 시스템 수준 유틸리티의 공급업자가 갱신 서비스를 제공할 수 있다. 이러한 갱신 서비스는 장치 드라이버의 기존 저장소(store)로부터 적절한 장치 드라이버를 자동으로 검색하고 그 위치를 찾을 수 있다. 넓게 보면, 이러한 서비스는 드라이버를 찾고 있는 장치의 유형을 나타내는 고유한 식별자를 수신함으로써, 또한 입력된 고유한 식별자와 매칭되는 엔트리에 대한 장치 드라이버의 저장소를 검색함으로써 동작된다. 장치 드라이버의 저장소는 운영 체제에 의해 국부적으로 유지될 수도 있고, 아니면 인터넷을 통해 액세스 가능할 수도 있다. 그러나, 어느 경우에서든, 장치 드라이버의 위치는 미리 알려져 있고, 유일한 의문 사항은 그 위치가 적절한 드라이버를 포함하고 있는지 여부이다.

<6> <발명의 개요>

<7> 장치 드라이버에 대한 다운로드 서비스를 제공하기 위한 시스템, 방법 및/또는 기술("툴")은 클라이언트 및 드라이버 다운로드 서비스를 포함한다. 클라이언트는 장치에 대한 드라이버가 페치(fetch)될 수 있는 위치를 드라이버 다운로드 서비스로부터 요청한다. 드라이버 다운로드 서비스는 장치 드라이버가 페치될 수 있는 위치로 응답한다.

<8> 본 개요는 실시예에서 이하 상술되는 개념의 선택을 간략화된 형태로 도입하기 위하여 제공된다. 본 명세서는 청구 대상의 핵심적이거나 본질적인 특징을 식별하기 위한 것이 아니고, 청구 대상의 범주를 정하는 데 도움을 주려고 사용되는 것도 아니다.

실시예

<16> 개요

<17> 이하의 문헌은 여러 가지 기술 및 프로세스가 가능한 툴을 설명한다. 이하의 논의는 툴이 장치 드라이버에 대한 다운로드 서비스를 제공하는 예시적인 방법을 설명한다. 또한, 본 논의는 툴에 의해 실행되는 다른 기술을 설명한다.

<18> 제한이 아닌 단순한 편의를 위해, 본 문헌은 섹션들로 조직되고, 이 섹션은 대응하는 제목에 의해 도입된다. 우선, 운영 환경이 도 1과 관련하여 설명되어 있다. 도 1은 장치 드라이버에 대한 다운로드 서비스를 제공하는 것에 관한 운영 환경을 도시하고, 또한 예시적인 데이터 흐름을 제공한다.

<19> 다음, 데이터 구조 및 스키마가 도 2 내지 도 4와 관련하여 설명된다. 도 2 내지 도 4는 장치 메타데이터, 장치 드라이버 위치에 대한 요청, 및 이러한 요청에 대한 응답의 예들을 도시한다.

<20> 마지막으로, 프로세스 흐름 및 프로토콜이 도 5 및 도 6과 관련하여 설명된다. 도 5 및 도 6은 장치 드라이버가 페치될 수 있는 위치를 요청하고 이러한 요청에 응답하기 위한 예시적인 프로세스 흐름 및 프로토콜을 도시한다.

<21> 운영 환경

<22> 도 1은 장치 드라이버에 대한 다운로드 서비스를 제공하기에 적절한 운영 환경(100)을 도시한다. 운영 환경(100)은 하나 이상의 클라이언트(102)를 포함할 수 있다. 도 1은 설명의 편의상 하나의 대표적인 클라이언트(102)만을 도시하지만, 운영 환경(100)의 가능한 구현을 제한하는 것은 아니다. 일반적으로, 운영 환경(100)은 임의의 수의 클라이언트(102)를 포함할 수 있다. 클라이언트(102)는 네트워크 또는 다른 서버, 데스크탑 컴퓨터

터, 랩탑 또는 노트북 컴퓨터, 이동 전화, PDA(personal digital assistant), 핸드헬드 컴퓨터 등과 같은 컴퓨팅 장치를 포함할 수 있다.

- <23> 클라이언트(102)는 하나 이상의 프로세서(들)(104) 및 컴퓨터 판독 가능 매체(106)를 포함할 수 있다. 컴퓨터 판독 가능 매체(106)는, 프로세서(104)에 의해 실행될 때, 여기서 설명되는 임의의 툴을 실행하는 명령어를 포함할 수 있다. 프로세서(104)는 컴퓨터 판독 가능 매체(106)에 내장되어 있거나 이에 인코딩되어 있는 명령어에 액세스하고/하거나 이를 실행하도록 구성될 수 있다. 또한, 프로세서(104)는 소정의 구조를 갖는 것으로 분류되거나 특징될 수 있다.
- <24> 또한, 컴퓨터 판독 가능 매체(106)는 운영 체제(108)를 포함할 수 있으며, 이 운영 체제(108)는 임의의 상업적으로 이용 가능한 운영 체제의 형태를 취할 수 있다. 운영 체제(108)의 적절한 예는, 미국 워싱턴주 레드몬드(Redmond)에 위치한 Microsoft Corporation사로부터 이용 가능한 운영 체제의 WINDOWS® 제품군 중 임의의 것을 포함할 수 있지만, 이제 제한되지 않는다. 운영 체제(108)의 다른 예는, LINUX® 운영 체제 중 임의의 것, 또는 미국 캘리포니아주 쿠퍼티노(Cupertino)에 위치한 Apple Computer, Inc.사로부터 이용 가능한 임의의 운영 체제를 포함할 수 있다.
- <25> 또한, 컴퓨터 판독 가능 매체(106)는 드라이버 요청 컴포넌트(110)를 포함할 수 있다. 드라이버 요청 컴포넌트(110)는 모듈, 프로그램, 또는 클라이언트(102) 외부의 하나 이상의 엔터티와 직접적으로 또는 간접적으로 상호작용할 수 있는 기타 엔터티로서 구현될 수 있다. 드라이버 요청 컴포넌트(110)의 예시적인 기능 및 성능은 툴에 대한 설명과 관련하여 이하 상술된다. 개략적으로, 드라이버 요청 컴포넌트(110)는 클라이언트(102)가 하나 이상의 장치(114)에 대응하는 하나 이상의 장치 드라이버(112)를 요청하고 이를 획득할 수 있게 한다.
- <26> 클라이언트(102)는 대응하는 링크(116)에 의해 하나 이상의 장치(114)에 접속될 수 있다. 도 1은 설명의 편의상 하나의 대표적인 장치(114)만을 도시하고 있지만, 운영 환경(100)의 가능한 구현들을 제한하지 않는다. 장치(114)는 클라이언트(102)에 직접 연결될 수 있다. 이러한 경우, 링크(116)는 예를 들면, USB 케이블 및 관련 포트들을 포함할 수 있다.
- <27> 다른 경우, 장치(114)는 네트워크에 의해 클라이언트(102)에 연결될 수 있다. 이 네트워크는 명백한 설명을 위해 도 1에는 도시되어 있지 않지만, 임의의 적절한 형태를 취할 수 있으며, 개괄적으로 링크(116)에 의해 나타나 있다. 예를 들면, 네트워크는 LAN(local area network), 인터넷과 같은 WAN(wide area network) 또는 이들의 임의의 조합일 수 있다. 이러한 경우, 클라이언트(102)는 이 네트워크를 통해 장치(114)에 액세스할 수 있다. 반대로, 장치(114)가 네트워크를 통해 복수의 클라이언트(102) 간에 공유될 수 있다.
- <28> 장치(114)는 클라이언트(102)의 외부 또는 주변에 있는 임의의 장치일 수 있다. 장치(114)의 예에는 프린터, MFP(multi-function peripheral), 스캐너, 카메라, 마이크 등이 포함될 수 있지만, 이에 제한되지 않는다.
- <29> 장치(114)의 예들은 대응하는 장치 드라이버(112)들과 연관되어 있다. 장치(114)가 클라이언트(102)에 접속되거나 또는 네트워크를 통해 클라이언트가 이용할 수 있게 되는 경우, 클라이언트(102)는 여기에 기재된 툴을 사용하여 적절한 장치 드라이버(112)를 획득할 수 있다. 예를 들면, 운영 체제(108)는 장치(114)를 감지하고, 또한 장치(114)에 대한 장치 드라이버(112)를 포함하고 있지 않다고 판단할 수 있다. 다르게는, 운영 체제(108)는 장치(114)에 대한 장치 드라이버(112)를 포함하지만, 장치 드라이버(112)가 예전의 것(outdated)이라고 판단할 수 있다.
- <30> 여기에 기재된 바와 같은 툴의 동작을 지원하기 위하여, 장치(114)는, 클라이언트(102)가 장치 드라이버(112)에 대한 새로운 파일 또는 갱신된 파일을 획득할 수 있는 장소를 특정하는 메타데이터(118)를 포함할 수 있다. 이러한 메타데이터(118)의 예는 이하 도 3에 도시되고 설명되어 있다.
- <31> 장치 드라이버(112)를 획득하든 아니면 이를 갱신하든 간에, 어느 경우에서든, 클라이언트(102) 상의 컴포넌트는 적절한 장치 드라이버(112)에 대한 위치를 획득하기 위한 요청을 제출할 수 있다. 이 요청의 일부로서, 클라이언트(102)는 장치(102)로부터 메타데이터(118)를 요청하고 수신할 수 있다. 메타데이터(118)는 클라이언트(102)가 장치 드라이버(112)를 요청할 수 있는 위치 또는 종점(endpoint)을 특정한다. 이 정보가 부여되면, 클라이언트(102)는 장치 드라이버(112)에 대한 적절한 요청을 메타데이터(118)에서 특정된 하나 이상의 종점으로 지향시킬 수 있다. 도 1은 장치 드라이버 요청(120)과 같은 요청의 예를 도시한다. 예를 들면, 드라이버 요청 컴포넌트(110)는 장치 드라이버 획득 요청(get device driver request)(120)을 제출할 수 있다.
- <32> 장치 드라이버 획득 요청(120)은 드라이버 다운로드 서비스(122)에 전송될 수 있다. 드라이버 다운로드 서비스(122)의 인스턴스는 메타데이터(118)에 의해 특정된 각각의 종점에서 제공된다. 하나의 가능한 구현에 불과한

일 구현에서는, 드라이버 다운로드 서비스(122)가 장치(114)에 의해 호스트될 수 있다. 다른 가능한 구현에서는, 기업 환경에 배치되는 서버와 같이, 드라이버 다운로드 서비스(122)가 복수의 클라이언트(102)를 서비스하는 집중형 엔터티에 의해 호스트될 수 있다. 또한, 드라이버 다운로드 서비스(122)가 장치(114)의 제조업자에 의해 호스트되고, 인터넷과 같은 WAN을 통해 액세스될 수 있다.

<33> 어느 경우에서든, 드라이버 다운로드 서비스(122)를 호스트하는 엔터티는 컴퓨팅 장치를 포함할 수 있고, 이 컴퓨팅 장치는 프로세서(124) 및 컴퓨터 판독 가능 매체(126)를 포함할 수 있다. 컴퓨터 판독 가능 매체(126)는 여기에 기재된 툴을 사용하여 장치 드라이버 획득 요청(120)을 수신하고 서비스하는 드라이버 다운로드 컴포넌트(128)를 포함할 수 있다. 드라이버 다운로드 컴포넌트(128)는 장치 드라이버 획득 요청(120)에 대해 장치 드라이버 패키지 응답(130)을 제공한다. 예를 들면, 드라이버 요청 컴포넌트(110)는 장치 드라이버 패키지 응답(130)을 수신하고 처리할 수 있다. 그리고, 장치 드라이버(112)는 예를 들면, 운영 체제(108) 또는 드라이버 요청 컴포넌트(110)에 의해 클라이언트(102)에 설치될 수 있다.

<34> 본 명세서에서는 "장치 드라이버 획득 요청", "드라이버 요청 컴포넌트", "장치 드라이버 패키지 응답" 및 "드라이버 다운로드 컴포넌트"란 용어를 단지 편의상 사용할 뿐, 이에 제한되지 않음을 이해할 것이다. 또한, 운영 환경(100)의 구현은 상이한 명칭 하에서도 유사한 기능을 제공할 수 있음을 이해할 것이다.

<35> 데이터 구조 및 스키마

<36> 도 1에서는 운영 환경(100)에 대해 설명하였고, 본 논의는 이제 운영 환경(100)의 다양한 컴포넌트에 의해 사용될 수 있는 다양한 데이터 구조 및 스키마의 설명으로 들어간다. 본 설명은 메타데이터(118)에 대한 보다 상세한 논의로 시작한다. 메타데이터(118)의 예는 이제 도 2와 관련하여 설명된다.

<37> 도 2는 드라이버 다운로드 컴포넌트(128)가 장치 드라이버(112)를 제공할 수 있는 소정의 장치(114)에 관한 메타데이터(118)를 도시한다. 클라이언트(102)가 이용할 수 있는 다양한 장치들(114) 중 적어도 일부는 대응하는 메타데이터(118)의 인스턴스를 저장한다. 소정의 장치(114)에 있어서, 메타데이터(118)는 메타데이터(118)와 관련된 장치(114)를 식별하는 장치 ID 필드(202)를 포함할 수 있다.

<38> 메타데이터(118)의 필드(204, 206 및 208)는 장치 드라이버(112)가 획득될 수 있는 다양한 위치 또는 종점에 대응한다. 필드(204, 206 및 208) 중 적어도 하나의 필드는 메타데이터(118)의 소정의 인스턴스에 대해 포플레이트(populate)된다.

<39> 하나의 가능한 구현에 불과한 일 구현에서, 장치 드라이버(112)는 하나 이상의 클라이언트(102)를 관리하는 관리자 또는 다른 사람에 의해 소정의 위치에 집중될 수 있다. 이러한 구현에서, 필드(204)는 장치 드라이버(112)가 배치될 수 있는 드라이버 다운로드 컴포넌트(128)에 대응하는 관리 종점(administrative endpoint)에 대한 참조를 포함할 수 있다. 관리 종점은 예를 들면, 법인(corporate) 또는 다른 인트라넷, 또는 인터넷을 통해 액세스될 수 있다. 특정한 관리 종점 참조는 책임있는 관리자에 의해 소정의 기업체에 규정될 수 있다.

<40> 다른 가능한 구현에서, 장치 드라이버(112)는 예를 들면, 장치(114)의 제조업자에 의해 제공되는 웹사이트로부터 이용 가능하다. 이러한 구현에서, 필드(206)는 제조업자 또는 다른 IHV(independent hardware vendor)에 의해 호스트되는 웹 서비스 또는 웹사이트와 같은 제조업자 종점에 대한 참조를 포함할 수 있다. 장치 드라이버(112)는 제조업자 종점에서 호스트되는 드라이버 다운로드 컴포넌트(128)로부터 이용 가능하며, 이는 예를 들면 인터넷을 통해 액세스 가능하다.

<41> 다른 가능한 구현에서, 장치 드라이버(112)는 장치(114) 그 자체에서 호스트되는 드라이버 다운로드 컴포넌트(128)로부터 이용 가능하다. 이러한 구현에서, 필드(308)는, 장치(114)에서 호스트되는 드라이버 다운로드 컴포넌트(128)로부터 장치 드라이버(112)가 획득될 수 있는 장치 종점에 대한 참조를 포함한다. 클라이언트(102)가 네트워크 접속성을 갖지 않는 경우, 클라이언트(102)는 장치(114)로부터 장치 드라이버(112)를 획득하도록 지향될 수 있다.

<42> 메타데이터(118)의 상술한 필드(202-208)를 도시하고 설명함에 있어서, 메타데이터(118) 또는 그와 등가의 구조물의 구현이 도 2에 도시된 필드 이외의 필드를 포함할 수 있음을 이해할 것이다. 도 2에 도시된 예가 제한되지 않음을 도 2에 도시된 생략 부호에 의해 나타내었다.

<43> 단, 예를 들면 장치(114)는, 예를 들면 드라이버 요청 컴포넌트(110), 또는 보다 일반적으로 클라이언트(102)로부터 요청이 있을 때, 메타데이터(118)를 제공할 수 있다. 이 요청은 "Get Device Metadata" 요청 등으로서 레이블될 수 있다. 클라이언트(102) 및/또는 드라이버 요청 컴포넌트(110)는 도 1에 도시된 하나 이상의 장치 드

라이버 획득 요청(120) 중 적어도 일부를 과플레이트하기 위하여 메타데이터(118)를 사용할 수 있다.

- <44> 소정의 장치(114)에 대한 메타데이터(118)에 과플레이트되는 중점(예를 들면, 204, 206 및/또는 208)에 있어서, 드라이버 요청 컴포넌트(110)는 장치 드라이버(112)가 성공적으로 획득될 때까지, 각각의 장치 드라이버 획득 요청(120)을 이들 중점으로 과플레이트하고 전송할 수 있다. 예를 들면, 드라이버 요청 컴포넌트(110)는 위 요청(120)의 필드를 메타데이터(118)로부터의 장치 ID 필드(202)로 과플레이트할 수 있다.
- <45> 소정의 중점에서, 드라이버 다운로드 컴포넌트(126)는 어느 장치(114)가 드라이버(112)를 찾는지를 판단하기 위하여 메타데이터(118)의 장치 ID 필드(202)를 참조할 수 있다. 그리고, 드라이버 다운로드 컴포넌트(126)는 이 장치(114)에 대한 장치 드라이버(112)를 검색할 수 있다. 예를 들면, 복수의 장치 드라이버(112)는 드라이버 다운로드 컴포넌트(128)에 의한 참조를 위하여 데이터 저장소에 집합될(collected) 수 있다. 데이터 저장소는 입력된 장치 드라이버 획득 요청(120)의 장치 ID 필드와 일치하는 장치 드라이버(112)의 위치를 찾기 위하여 검색될 수 있다.
- <46> 소정의 장치(114)에 대한 메타데이터(118)는 처음에 장치(114)의 제조업자에 의해 과플레이트될 수 있다. 메타데이터(118)는 예를 들면, 장치(114)의 펌웨어에 저장될 수 있다. 일부의 경우, 장치(114)의 제조업자에 의해 저장된 대로의 메타데이터(118)는 예를 들면, 복수의 클라이언트(102)를 관리하는 시스템 관리자에 의해 교체(replace)되거나 무효로 될(overridden) 수 있다.
- <47> 본 설명에서는 메시지에 대한 표준 아웃라인을 규정하기 위하여 이하의 구문(syntax)을 사용한다. 구문은 XML 인스턴스로서 나타나고, 이탤릭체의 값은 값 대신에 데이터 유형을 나타낸다. 카디널리티(cardinality)를 나타내기 위하여 구성 요소 및 어트리뷰트에 이하와 같이 문자들이 추가된다.
- <48> ● "?"(0 또는 1)
- <49> ● "*" (0 이상)
- <50> ● "+" (1 이상)
- <51> ● 문자 "|"는 선택 사항들(alternatives) 사이에서의 선택을 나타내는 데 사용된다.
- <52> ● 문자 "[" 및 "]"는 포함된 아이템이 카디널리티 또는 선택에 관한 그룹으로 다뤄지는 것을 나타내는 데 사용된다.
- <53> 생략 부호(즉, "...")는 다른 자 요소(child) 또는 어트리뷰트 콘텐츠를 허용하는 확장 가능성을 나타낸다. 그 표시된 확장 지점에 추가적인 자 요소 및/또는 어트리뷰트가 추가될 수 있다.
- <54> 메타데이터(118)의 예시적인 XML 기반 구현은,
- ```
<dds:DriverDownloadLocations>
 <dds:DeviceIdentifier>xs:anyURI</dds:DeviceIdentifier>
 [<dds:AdminastrativeEPR>
 endpoint-reference
 </dds:AdminastrativeEPR>]?
 [<dds:ManufacturerEPR>
 endpoint-reference
 </dds:ManufacturerEPR>]
 [<dds:DeviceEPR>
 endpoint-reference
 </dds:DeviceEPR>]?
 ...
</dds:DriverDownloadLocations >
```
- <55>
- <56>
- <57> 와 같이 나타난다.
- <58> 다양한 장치(114)에 이용가능한 메타데이터(118)의 예에 대해 설명하였고, 본 설명은 이제 도 3에 자세히 도시된 장치 드라이버 획득 요청(120)으로 들어간다.

- <59> 도 3은 도 1에 도시된 바와 같은, 장치 드라이버 획득 요청(120)의 예시적인 내용을 도시한다. 예를 들면, 장치 드라이버 획득 요청(120)은 클라이언트(102)로부터 메타데이터(118)에서 특정된 각각의 종점에 제공된 드라이버 다운로드 서비스(122)로 전달되는 메시지로써 구현될 수 있다. 장치 드라이버 획득 요청(120)은 장치 식별자(ID) 필드(302)를 포함할 수 있고, 이는 드라이버(112)를 찾는 장치(114)를 식별하고, 장치(114)에 대한 정확한 장치 드라이버(112)를 획득하는 데 사용된다.
- <60> 운영 체제 ID 필드(304)는 클라이언트(102)의 운영 체제(108)를 식별한다. 상이한 클라이언트(102) 상에서 실행 중일 수 있는 상이한 운영 체제(108)에는 상이한 장치 드라이버(112)가 제공될 수 있음을 이해할 것이다.
- <61> 아키텍처(architecture) ID 필드(306)는 클라이언트(102) 상의 프로세서(104)의 아키텍처를 식별한다. 클라이언트(102) 상에서 실행 중일 수 있는 상이한 운영 체제(108) 및 상이한 아키텍처에는 상이한 장치 드라이버(112)가 제공될 수 있음을 이해할 것이다. 이러한 아키텍처의 예는 x86, x64, 아이테니엄(Itanium) 등을 포함할 수 있지만, 이에 제한되지 않는다.
- <62> 몇몇 경우에서, 장치 드라이버(112)는 특정 언어에 국지화될(localized) 수 있다. 이러한 경우, 장치 드라이버 획득 요청(118)은 요청 언어(requested language) 필드(308)를 파플레이트할 수 있다. 요청 언어 필드(308)는 장치 드라이버(112)가 어느 언어를 지원해야 하는지를 나타낼 수 있다. 예를 들면, 장치 드라이버(112)는 인간이 판독 가능한 언어로 대화 상자에 프롬프트, 라벨 또는 다른 텍스트를 제공할 수 있다. 따라서, 요청 언어 필드(308)는 영어, 스페인어, 불어, 독어, 또는 기타 인간이 판독 가능한 임의의 언어를 나타낼 수 있다.
- <63> 위에서 제안된 바와 같이, 장치 드라이버(112)의 일부 예는 국지화되지 않는다. 따라서, 요청 언어 필드(308)는, 장치 드라이버 획득 요청(120)이 비국지적인 장치 드라이버(112)를 요청하고 있을 때는 파플레이트되지 않는 선택적인 필드(optional field)로서 보일 수 있다. 요청 언어 필드(308)의 선택적인 속성은 도 3의 블록(308)의 점선으로 된 아웃라인으로 나타내었다. 또한, 요청 언어 필드(308)를 선택적인 것으로 특징짓는 것에 있어서, 여기의 설명은 여기에 도시된 다른 구성 요소가 모든 구현에서 본질적이거나 강제적인 것임을 말하는 것으로 해석되지 않음을 주의한다.
- <64> 장치 드라이버 획득 요청(120)의 상기 필드(302-306)를 도시하고 설명함에 있어서, 장치 드라이버 획득 요청(120) 또는 그와 등가의 요청의 구현은 도 3에 도시된 필드 이외의 필드를 포함할 수 있음을 이해할 것이다. 도 3에 도시된 예가 제한되지 않음을 도 3에 도시된 생략 부호에 의해 나타내었다.
- <65> 장치 드라이버 획득 요청(120)의 예시적인 XML 기반 구현은,
- ```
<dds:GetDeviceDriver>
  <dds:DeviceIdentifier>xs:anyURI</dds:DeviceIdentifier>
  <dds:OSIdentifier>xs:anyURI</dds:OSIdentifier>
  <dds:RequestedArchitecture>xs:string</dds:RequestedArchitecture>
  <dds:RequestedLanguage>xs:string</dds:RequestedLanguage>
  ...
</dds:GetDeviceDriver>
```
- <66>
- <67> 와 같이 나타난다.
- <68> 도 3과 관련하여 장치 드라이버 획득 요청(120)을 설명하였고, 본 논의는 이제 도 4에 자세히 도시된 장치 드라이버 패키지 응답(130)으로 들어간다.
- <69> 도 4는, 장치 드라이버 획득 요청(120)에 대한 응답으로서 제공될 수 있는 장치 드라이버 패키지 응답(130)의 예시적인 콘텐츠를 도시한다. 도 4에 도시된 바와 같이, 장치 드라이버 패키지 응답(130)은 장치 드라이버(112)를 구성하는 하나 이상의 파일(402)을 포함할 수 있다. 도 4는 설명의 편의상 하나의 대표적인 파일(402)만을 도시하지만, 이에 의해 가능한 구현을 제한하지 않는다. 일반적으로, 장치 드라이버(112)는 임의의 수의 파일(402)을 포함할 수 있다.
- <70> 각각의 장치 드라이버 파일(402)에 있어서, 장치 드라이버 패키지 응답(130)은 파일(402)에 대한 관련 경로명을 제공하는 필드(404)를 포함할 수 있다. 각각의 파일(402) 및 대응하는 경로명(404)에 있어서, 장치 드라이버 패키지 응답(128)은 필드(406 또는 408) 중 하나의 필드를 파플레이트할 수 있다. 필드(406)는 장치 드라이버(112)에 대한 데이터를 포함할 수 있으며, 이 데이터는 예를 들면 Base64 리프리젠테이션으로 인코딩된다. 구현에서, 필드(406)는 MTOM(Message Transmission Optimization Mechanism) 또는 유사한 방법을 사용하여 인코딩될 수 있다. 필드(408)는 장치 드라이버 파일이 예를 들면 HTTP GET 명령을 사용하여 폐치될 수 있는 URL을

포함할 수 있다. 또한, 장치 드라이버 패키지 응답(130)은 설치(installation) 정보 필드(410)를 파플레이트할 수 있다. 설치 정보 필드(410)는 소정의 운영 체제 하에서 실행하도록 장치 드라이버(112)를 로드하거나 설치하는 것을 용이하게 하기 위하여 예를 들면, 소정의 운영 체제 공급업자에 의해 제공되는 임의의 정보를 포함할 수 있다. 따라서, 설치 정보 필드(410)의 내용은 구체적인 운영 체제에 특정될 수 있고, 장치 드라이버 패키지 응답(128)의 모든 인스턴스에서 파플레이트되지는 않는다. 따라서, 설치 정보 필드(410)의 선택적인 속성은 블록(410)의 점선으로 된 아웃라인으로 도 4에 나타내었다.

<71> 응답(130)의 예시적인 XML 기반 구현은,

```
<dds:DeviceDriverPackage>
  <dds:Files>
    [<dds:File Name=xs:string>
      [
        <dds:FileURL>xs:anyURI</dds:FileURL>
        |
        <dds:FileData>xs:base64Binary</dds:FileData>
      ]
    </dds:File>]+
  </dds:Files>
  [<dds:InstalationInfo>
    ...
  </dds:InstalationInfo>]?
  ...
</dds:DeviceDriverPackage>
```

<72>

<73> 와 같이 나타난다.

<74> 장치 드라이버 패키지 응답(130)의 상기한 필드(402-410)를 도시하고 설명함에 있어서, 장치 드라이버 패키지 응답(130) 또는 그와 등가의 구조물의 구현은 도 4에 도시된 필드 이외의 필드를 포함할 수 있음을 이해할 것이다. 도 4에 도시된 예가 제한되지 않음을 도 4에 도시된 생략 부호로 나타내었다.

<75> 프로세스 흐름 및 프로토콜

<76> 도 2 내지 도 4와 관련하여 데이터 구조 및 관련 스키마를 설명하였고, 본 논의는 이제 장치 드라이버에 대한 다운로드 서비스를 제공하는 것과 관련하여 실행될 수 있는 다양한 프로세스 흐름 및 관련 프로토콜에 대한 설명으로 들어간다. 이 프로세스 흐름은 도 5 및 도 6과 관련하여 설명된다. 편의상, 도 5 및 도 6은 운영 환경(100)의 일부 컴포넌트와 관련하여 설명된다. 그러나, 도 5 및 도 6에 도시된 프로세스 흐름은 여기에 기재된 설명의 범주 및 사상을 벗어나지 않고도 기타 컴포넌트와 관련하여 구현될 수 있다.

<77> 도 5는 장치 드라이버 정보의 위치를 요청하기 위한 프로세스 흐름(500)을 도시한다. 논의의 편의상, 여기서 프로세스 흐름(500)은 도 1에 도시된 클라이언트(102)와 관련하여 기재된다. 그러나, 프로세스 흐름(500)은 여기에 기재된 설명의 사상 및 범주를 벗어나지 않고도 클라이언트(102) 이외의 장치 또는 컴포넌트, 또는 운영 환경(100)에 도시된 기타 컴포넌트 상에서 구현될 수 있다.

<78> 프로세스 흐름(500)으로 보다 자세히 들어가면, 블록(502)은 소정의 장치에 대한 장치 메타데이터를 요청한다. 하나의 가능한 구현에 불과한 일 구현에서, 도 1은 예시적인 장치(114)의 예를 제공하고, 도 2는 예시적인 메타데이터(118)를 도시한다. 장치(114)가 자신의 메타데이터(118)를 저장하고, 요청 시에 이 메타데이터(118)를 제공할 수 있음을 상기하기 바란다.

<79> 블록(504)은 블록(502)에 도시된 요청에 응답하여 장치 메타데이터(118)를 수신한다. 장치 메타데이터는 장치 드라이버가 획득될 수 있는 적어도 하나의 종점을 특정할 수 있음을 상기하기 바란다.

<80> 블록(506)은 장치 메타데이터(118)에서 특정된 제1 종점으로부터 장치 드라이버(예를 들면, 장치 드라이버(112))를 요청한다. 블록(508)은 블록(506)에서 장치 드라이버를 요청받은 종점이 장치 드라이버를 제공할 수 있는지 여부를 평가한다. 예를 들면, 블록(508)은 블록(506)에서 행해진 요청의 상태를 판단하기 위하여 종점

으로부터 수신된 응답을 조사할 수 있다.

- <81> 만약 현재 종점이 요청 장치 드라이버를 제공하면, 프로세스 흐름(500)은 블록(508)으로부터 예 브랜치(510)를 따라 블록(512)으로 가며, 이 블록(512)에서 장치 드라이버는 예를 들면, 컴퓨터 판독 가능 매체(106)에 설치된다.
- <82> 블록(508)으로 되돌아가서, 만약 현재 종점이 요청 장치 드라이버를 포함하지 않거나 이를 제공할 수 없으면, 프로세스 흐름(500)은 아니오 브랜치(514)를 따라 블록(516)으로 간다. 블록(516)은 블록(504)에서 수신된 메타데이터가 장치 드라이버 파일이 폐치될 수 있는 임의의 다른 종점을 더 제공하는지 여부를 판단한다.
- <83> 만약 메타데이터가 장치 드라이버가 요청될 수 있는 다른 종점을 더 특정하면, 프로세스 흐름(500)은 블록(516)으로부터 예 브랜치(518)를 따라 블록(520)으로 간다. 블록(520)에서, 장치 메타데이터에서 특정된 다음 종점이 현재 종점으로서 선택된다. 그리고, 프로세스 흐름(500)은 블록(506)으로 되돌아가서, 이 새로운 현재 종점으로부터 장치 드라이버를 요청한다. 그리고, 위에서 논의된 바와 같이, 프로세스 흐름(500)은 평가 블록(508)을 반복한다.
- <84> 만약 메타데이터가 장치 드라이버가 요청될 수 있는 종점을 더 이상 특정하지 않으면, 프로세스 흐름(500)은 블록(516)으로부터 아니오 브랜치(522)를 따라 블록(524)으로 간다. 만약 프로세스 흐름(500)이 블록(524)에 도달하면, 장치에 대한 장치 드라이버를 제공할 수 없다. 블록(524)은 적절한 에러 메시지를 보고한다.
- <85> 프로세스 흐름(500)에 대해 상기 설명을 제공하였지만, 장치(예를 들면, 장치(102))에 의해 제공되는 메타데이터는 종점이 장치 드라이버에 대해 쿼리되는 순서를 특정할 수 있음을 주의한다. 또한, 메타데이터는 어느 종점이 쿼리되는 지를 특정할 수 있다.
- <86> 새로운 장치 드라이버 또는 갱신된 장치 드라이버를 요청하기 위한 프로세스 흐름(500)을 설명하였고, 이러한 요청에 응답하기 위한 프로세스를 이제 도 6과 관련하여 설명한다.
- <87> 도 6은 장치 드라이버에 대한 요청에 응답하기 위한 프로세스 흐름(600)을 도시한다. 설명의 편의상, 여기서 프로세스 흐름(600)은 도 1에 도시된 드라이버 다운로드 서비스(120)와 관련하여 설명된다. 일부 구현에서는, 장치 메타데이터에서 특정된 종점에 드라이버 다운로드 서비스(122)의 인스턴스가 제공될 수 있다. 드라이버 다운로드 서비스(122)는 예를 들면, 도 5에 도시된 프로세스 흐름(500)에 의해 제공되는 장치 드라이버에 대한 요청에 응답할 수 있다. 그러나, 프로세스 흐름(600)은 본 발명의 사상 및 범주를 벗어나지 않고도, 드라이버 다운로드 서비스(122) 이외의 장치 또는 컴포넌트, 또는 운영 환경(100)에 도시된 기타 컴포넌트에서 구현될 수 있음을 이해할 것이다. 보다 일반적으로, 프로세스 흐름(600)은 장치 드라이버에 대한 요청을 서비스하기 위하여 장치 메타데이터에서 특정된 임의의 종점에 의해 행해질 수 있음을 이해할 것이다. 장치 메타데이터(118) 및 관련 종점(204, 206 및 208)의 제한되지 않는 예가 도 2에 도시되어 있다.
- <88> 프로세스 흐름(600)으로 보다 자세히 들어 가면, 블록(602)은 소정의 장치에 대한 드라이버 요청을 수신한다. 위에서 언급한 바와 같이, 도 1은 예시적인 장치(114), 예시적인 장치 드라이버(112), 및 예시적인 장치 드라이버 획득 요청(120)을 제공한다.
- <89> 블록(604)은 요청 장치 드라이버가 소정의 종점에서 이용 가능한지 여부를 판단한다. 만약 소정의 종점이 요청 장치 드라이버를 제공할 수 없으면, 프로세스 흐름(600)은 아니오 브랜치(606)를 따라 블록(608)으로 간다. 블록(608)은 소정의 종점이 요청 장치 드라이버를 제공할 수 없다고 보고한다. 블록(608)에서 제공된 응답은 도 5에 도시된 판단 블록(508)으로 입력될 수 있고, 이 판단 블록(508)은 소정의 종점이 장치 드라이버를 제공할 수 있는지 여부를 평가한다.
- <90> 블록(608) 이후에, 프로세스 흐름(600)은 블록(602)으로 되돌아가서, 그 소정의 종점으로 지향된 장치 드라이버에 대한 다음 요청을 기다린다.
- <91> 블록(604)으로 되돌아가서, 만약 요청 장치 드라이버가 이용 가능하면, 프로세스 흐름(600)은 예 브랜치(610)를 따라 블록(612)으로 간다. 블록(612)은 요청 장치 드라이버, 또는 장치 드라이버 파일이 폐치될 수 있는 위치 중 어느 하나를 획득한다.
- <92> 프로세스 흐름(600)의 일부 구현에서, 블록(614)은 블록(612)에서 특정된 위치로부터 실제 장치 드라이버 파일을 폐치하고 로드할 수 있다. 그리고, 실제 장치 드라이버 파일은 블록(602)에서 수신된 요청에 대한 응답의 본문(body)에 로드될 수 있다. 다른 예에서, 요청에 대한 응답은 장치 드라이버 파일이 폐치될 수 있는 위치를 제공하는 포인터 또는 참조를 포함할 수 있다. 따라서, 도 6에서 블록(614)을 점선으로 된 아웃라인으로 나타

낸 것과 같이, 블록(614)은 그 속성상 선택적으로 보일 수 있다.

<93> 블록(616)은 블록(602)에서 수신된 요청에 대한 응답을 제공한다. 도 1은 드라이버 다운로드 서비스(122)로부터 클라이언트(102)로 전송되는 예시적인 장치 드라이버 패키지 응답(130)을 제공한다. 응답의 상이한 예는 장치 드라이버 파일 그 자체, 또는 장치 드라이버 파일이 폐치될 수 있는 위치를 제공하는 포인터 또는 참조를 포함할 수 있다.

<94> 결어

<95> 비록 본 시스템 및 방법이 구조적인 특징 및/또는 방법론적인 단계에 특정된 언어로 기술되었지만, 첨부되는 청구 범위에서 규정되는 본 시스템 및 방법은 설명되는 특정 특징 또는 단계에 제한될 필요가 없음을 이해할 것이다. 오히려, 특정 특징 및 단계를 청구되는 시스템 및 방법을 구현하는 예시적인 형태로서 개시되어 있다.

<96> 또한, 여기에서 설명되고 도시되는 소정의 플로우차트와 관련하여, 여기서 설명되는 프로세스 및 그 하위 프로세스는 본 발명의 사상 및 범주를 벗어나지 않고도 본 명세서에 기재된 순서 이외의 순서로 행해질 수 있음을 주의한다.

도면의 간단한 설명

<9> 장치 드라이버에 대한 다운로드 서비스를 제공하기 위한 툴이 이하의 도면과 관련하여 설명된다. 유사한 구성 요소 및 특징을 참조하기 위하여 명세서 및 도면 전반에 걸쳐 동일한 참조 부호가 사용된다. 참조 부호에서 첫 번째 자리 숫자는 참조 부호가 도입되는 도면을 나타낸다.

<10> 도 1은 장치 드라이버에 대한 다운로드 서비스를 제공하기 위한 운영 환경의 블록도.

<11> 도 2는 다운로드 서비스가 장치 드라이버를 제공할 수 있는 소정의 장치에 관한 예시적인 메타데이터의 블록도.

<12> 도 3은 장치 드라이버에 대한 위치를 획득하기 위한 요청의 예시적인 내용의 블록도.

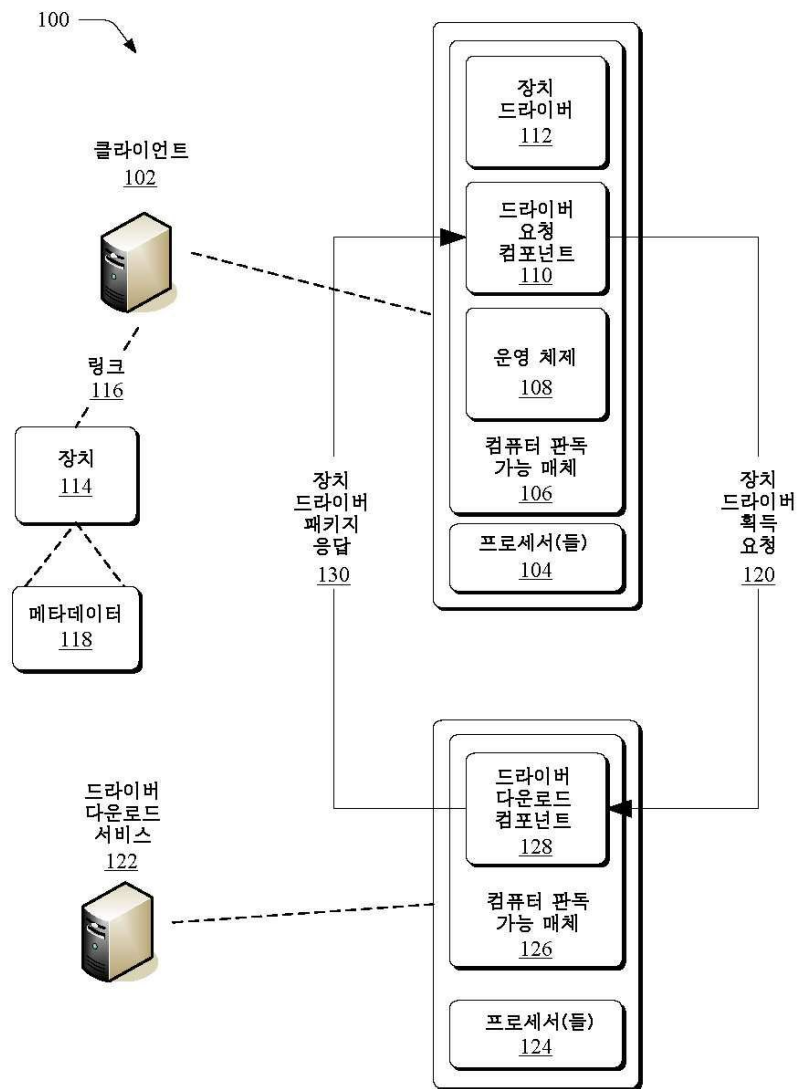
<13> 도 4는 도 1 및 도 3에 도시된 요청에 대한 응답의 예시적인 내용의 블록도.

<14> 도 5는 장치 드라이버의 위치를 요청하기 위한 프로세스의 흐름도.

<15> 도 6은 장치 드라이버 정보의 위치에 대한 요청에 응답하기 위한 프로세스의 흐름도.

도면

도면1



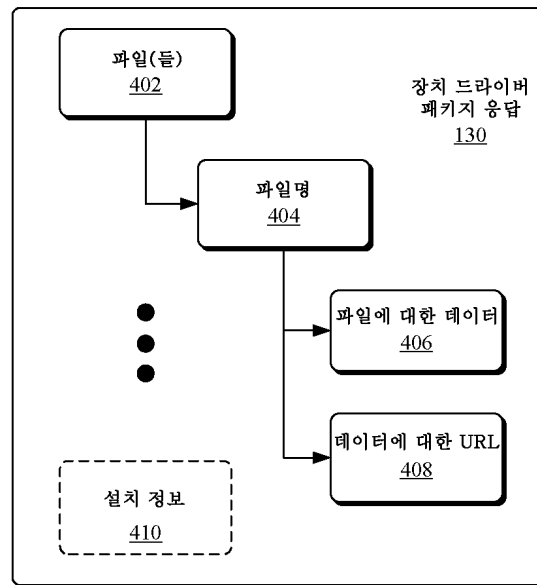
도면2



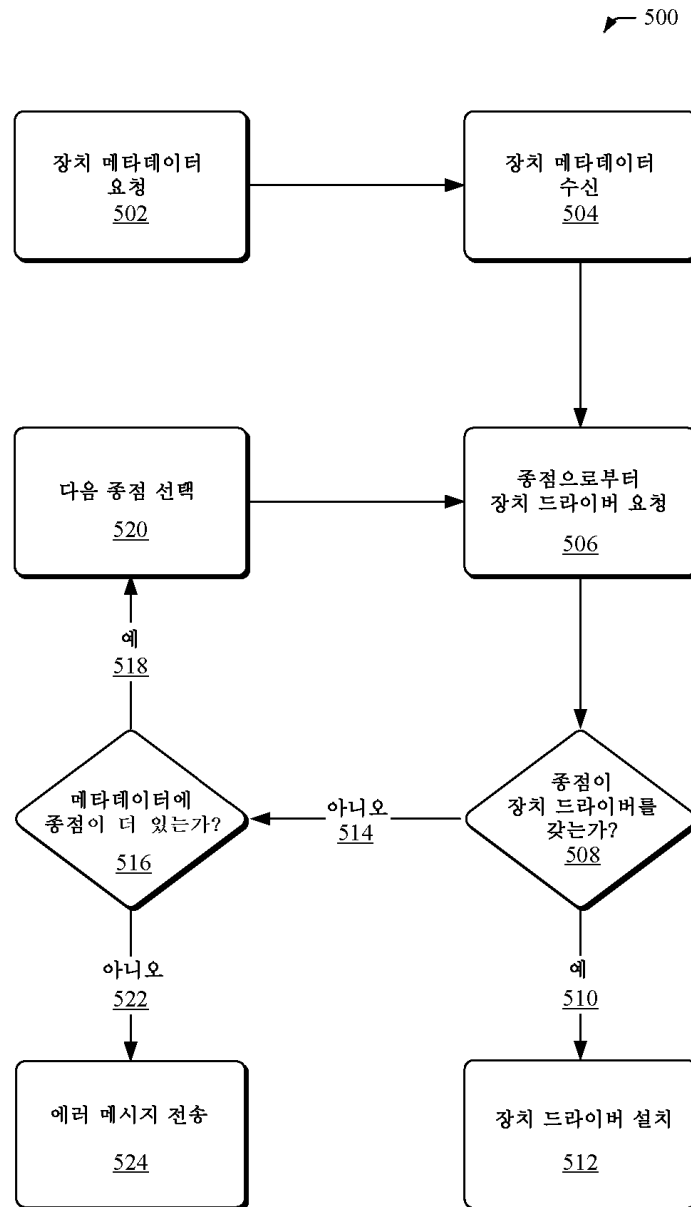
도면3



도면4



도면5



도면6

