



(19) 대한민국특허청(KR)  
 (12) 등록특허공보(B1)

(45) 공고일자 2009년05월07일  
 (11) 등록번호 10-0896541  
 (24) 등록일자 2009년04월29일

(51) Int. Cl.

G06F 13/00 (2006.01) G06F 13/10 (2006.01)

G06F 3/12 (2006.01)

(21) 출원번호 10-2006-7021707

(22) 출원일자 2006년10월19일

심사청구일자 2006년10월19일

번역문제출일자 2006년10월19일

(65) 공개번호 10-2006-0135891

(43) 공개일자 2006년12월29일

(86) 국제출원번호 PCT/JP2005/007282

국제출원일자 2005년04월08일

(87) 국제공개번호 WO 2005/103914

국제공개일자 2005년11월03일

(30) 우선권주장

JP-P-2004-00123443 2004년04월19일 일본(JP)

(56) 선행기술조사문헌

JP15006133 A\*

\*는 심사관에 의하여 인용된 문헌

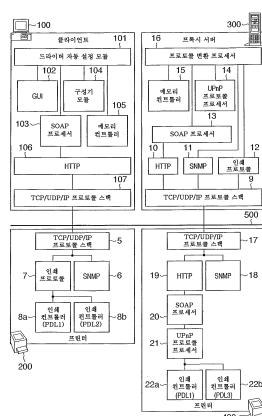
전체 청구항 수 : 총 22 항

심사관 : 한선경

## (54) 네트워크 디바이스 관리 장치 및 그 제어 방법

**(57) 요약**

본 발명은 네트워크 상의 다른 디바이스들이 네트워크 호환성 플러그 앤 플레이 기능을 지원하지 않는 디바이스를, 네트워크 호환성 플러그 앤 플레이 기능을 지원하는 디바이스로 인식하게 하고, 그 디바이스가 기능을 가진 디바이스로 기능하도록 한다. 이를 위해, 프록시 서버는 네트워크 상에 존재하고 어떠한 네트워크 호환성 플러그 앤 플레이 기능을 지원하지 않는 프린터를 검색하고, 발견된 프린터를 지정하는데 이용되는 정보를 메모리 컨트롤러를 통해 하드 디스크에 등록된다. 등록시, 프록시 서버는 마치 네트워크 호환성 플러그 앤 플레이 디바이스인 것처럼 동작하도록 네트워크 호환성 플러그 앤 플레이 DeviceDescription을 생성한다. 네트워크로부터 네트워크 호환성 플러그 앤 플레이 검색 메시지의 수신시, 프록시 서버는 생성된 DeviceDescription을 응답 메시지로서 송신하여, 네트워크 호환성 플러그 앤 플레이 디바이스를 지원하지 않는 프린터 대신에 UPnP 디바이스를 나타낸다. 등록된 프린터에 어드레싱된 인쇄 작업의 수신시, 프록시 서버는 인쇄 작업을 그 프린터에 대한 프로토콜로 변환하고, 변환된 작업 데이터를 프린터에 송신한다.

**대 표 도 - 도1**

## 특허청구의 범위

### 청구항 1

네트워크 접속 수단을 구비하며, 네트워크에 접속되고 복수의 기능을 가진 네트워크 디바이스를 관리하는 네트워크 디바이스 관리 장치로서,

어떠한 네트워크 호환성 플러그 앤 플레이 기능도 지원하지 않고 복수의 기능을 갖는 적어도 하나의 네트워크 디바이스의 네트워크 어드레스와, 상기 네트워크 디바이스의 상기 복수의 기능과 연관된 기능 정보를 기억하는 기억 수단; 및

상기 네트워크 접속 수단을 통해 네트워크 호환성 플러그 앤 플레이 디바이스의 소재 확인 요구를 수신한 경우, 상기 기억 수단에 기억된 복수의 기능 정보가 나타내는 기능들에 대응하는 복수의 독립적인 가상 네트워크 호환성 플러그 앤 플레이 디바이스를 특정하는 식별 정보를 포함하는 메시지를 생성하여 회신하는 응답 수단

을 포함하고,

상기 복수의 독립적인 가상 네트워크 호환성 플러그 앤 플레이 디바이스를 식별하는 상기 식별 정보는 상기 복수의 독립적인 가상 네트워크 호환성 플러그 앤 플레이 디바이스에 대응하는 복수의 디바이스 드라이버를 설치하는데 이용되는 네트워크 디바이스 관리 장치.

### 청구항 2

제1항에 있어서,

상기 기억 수단에 기억된 기능 정보는 기억될 네트워크 디바이스와 통신하는데 필요한 프로토콜 정보를 포함하는 네트워크 디바이스 관리 장치.

### 청구항 3

제2항에 있어서,

상기 네트워크 접속 수단을 통해 가상 네트워크 호환성 플러그 앤 플레이 디바이스에 어드레싱된 작업 정보를 수신한 경우, 상기 기억 수단으로부터 대응하는 네트워크 디바이스의 어드레스 및 프로토콜 정보를 획득하고, 상기 작업 정보를 획득한 프로토콜로 변환하여, 그 변환한 정보를 획득한 어드레스로 송신하는 제어 수단을 더 포함하는 네트워크 디바이스 관리 장치.

### 청구항 4

제1항에 있어서,

상기 기능 정보가 나타내는 기능은 상기 네트워크 디바이스가 처리할 수 있는 인쇄 데이터를 생성할 수 있는 복수의 상이한 프린터 드라이버의 기능들을 포함하는 네트워크 디바이스 관리 장치.

### 청구항 5

제1항에 있어서,

어떠한 네트워크 호환성 플러그 앤 플레이 기능도 지원하지 않는 네트워크 디바이스를 검색하는 검색 수단;

상기 검색 수단이 발견한 네트워크 디바이스의 네트워크 어드레스, 및 상기 검색 수단이 발견한 네트워크 디바이스와의 통신에 이용되는 프로토콜을 특정하는 정보를 상기 기억 수단에 등록하는 등록 수단; 및

상기 등록한 네트워크 디바이스 대신에 상기 응답 수단이 회신할 메시지를 생성하는 생성 수단  
을 더 포함하는 네트워크 디바이스 관리 장치.

### 청구항 6

제5항에 있어서,

상기 검색 수단은 SNMP 프로토콜의 검색에 의해 검출된 네트워크 디바이스 그룹으로부터 UPnP 네트워크 프로토

콜의 검색 결과로서 검출된 네트워크 디바이스를 제외한 후에 남아 있는 네트워크 디바이스 그룹을, 어떠한 네트워크 호환성 플러그 앤 플레이 기능도 지원하지 않는 네트워크 디바이스 그룹으로서 판정하는 네트워크 디바이스 관리 장치.

#### 청구항 7

제1항 내지 제6항 중 어느 한 항에 있어서,

상기 네트워크 디바이스는 네트워크 프린터인 네트워크 디바이스 관리 장치.

#### 청구항 8

제7항에 있어서,

상기 네트워크 디바이스가 복수의 프린터 언어를 지원하는 경우, 상기 응답 수단은 개개의 프린터 언어마다 독립적인 논리적 가상 네트워크 호환성 플러그 앤 플레이 프린터로서 응답하는 네트워크 디바이스 관리 장치.

#### 청구항 9

네트워크 접속 수단, 및 어떠한 네트워크 호환성 플러그 앤 플레이 기능도 지원하지 않고 복수의 기능을 갖는 적어도 하나의 네트워크 디바이스의 네트워크 어드레스와, 상기 네트워크 디바이스와 통신하는데 이용되는 프로토콜 정보를 기억하는 기억 수단을 구비하고, 네트워크에 접속된 네트워크 디바이스를 관리하는 네트워크 디바이스 관리 장치를 제어하는 방법으로서,

상기 네트워크 접속 수단을 통해 네트워크 호환성 플러그 앤 플레이 디바이스의 소재 확인 요구를 수신한 경우, 상기 기억 수단에 기억된 복수의 기능 정보가 나타내는 기능들에 대응하는 복수의 독립적인 가상 네트워크 호환성 플러그 앤 플레이 디바이스를 특정하는 식별 정보를 포함하는 메시지를 생성하여 회신하는 단계

를 포함하고,

상기 복수의 독립적인 가상 네트워크 호환성 플러그 앤 플레이 디바이스를 식별하는 상기 식별 정보는 상기 복수의 독립적인 가상 네트워크 호환성 플러그 앤 플레이 디바이스에 대응하는 복수의 디바이스 드라이버를 설치하는데 이용되는 네트워크 디바이스 관리 장치 제어 방법.

#### 청구항 10

삭제

#### 청구항 11

네트워크 접속 수단, 및 어떠한 네트워크 호환성 플러그 앤 플레이 기능도 지원하지 않고 복수의 기능을 갖는 적어도 하나의 네트워크 디바이스의 네트워크 어드레스와, 상기 네트워크 디바이스와 통신하는데 이용되는 프로토콜 정보를 기억하는 기억 수단을 구비하고, 네트워크에 접속된 네트워크 디바이스를 관리하는 네트워크 디바이스 관리 장치로서 기능하는 컴퓨터 프로그램을 저장하는 컴퓨터 판독가능한 기억 매체로서,

상기 컴퓨터 프로그램은,

상기 네트워크 접속 수단을 통해 네트워크 호환성 플러그 앤 플레이 디바이스의 소재 확인 요구를 수신한 경우, 상기 기억 수단에 기억된 복수의 기능 정보가 나타내는 기능들에 대응하는 복수의 독립적인 가상 네트워크 호환성 플러그 앤 플레이 디바이스를 특정하는 식별 정보를 포함하는 메시지를 생성하여 회신하는 단계

를 포함하고,

상기 복수의 독립적인 가상 네트워크 호환성 플러그 앤 플레이 디바이스를 식별하는 상기 식별 정보는 상기 복수의 독립적인 가상 네트워크 호환성 플러그 앤 플레이 디바이스에 대응하는 복수의 디바이스 드라이버를 설치하는데 이용되는 컴퓨터 판독가능한 기억 매체.

#### 청구항 12

제9항에 있어서,

상기 기억 수단에 기억된 기능 정보는 기억될 네트워크 디바이스와 통신하는데 필요한 프로토콜 정보를 포함하

는 네트워크 디바이스 관리 장치 제어 방법.

### 청구항 13

제12항에 있어서,

상기 네트워크 접속 수단을 통해 가장 네트워크 호환성 플러그 앤 플레이 디바이스에 어드레싱된 작업 정보를 수신하는 경우, 상기 기억 수단으로부터 대응하는 네트워크 디바이스의 어드레스 및 프로토콜 정보를 획득하고, 상기 작업 정보를 획득한 프로토콜로 변환하여, 그 변환한 정보를 상기 획득한 어드레스로 송신하는 제어 단계를 더 포함하는 네트워크 디바이스 관리 장치 제어 방법.

### 청구항 14

제9항에 있어서,

상기 기능 정보가 나타내는 기능들은 상기 네트워크 디바이스가 처리할 수 있는 인쇄 데이터를 생성할 수 있는 복수의 상이한 프린터 드라이버의 기능들을 포함하는 네트워크 디바이스 관리 장치 제어 방법.

### 청구항 15

제9항에 있어서,

어떠한 네트워크 호환성 플러그 앤 플레이 기능도 지원하지 않는 네트워크 디바이스를 검색하는 검색 단계;

상기 검색 단계에서 발견한 네트워크 디바이스의 네트워크 어드레스, 및 상기 검색 단계에서 발견한 네트워크 디바이스와의 통신에 이용되는 프로토콜을 특정하는 정보를 상기 기억 수단에 등록하는 등록 단계; 및

상기 등록한 네트워크 디바이스 대신에 상기 생성 회신 단계에서 회신할 메시지를 생성하는 생성 단계  
를 더 포함하는 네트워크 디바이스 관리 장치 제어 방법.

### 청구항 16

제15항에 있어서,

상기 검색 단계는 SNMP 프로토콜의 검색에 의해 검출된 네트워크 디바이스 그룹으로부터 UPnP 네트워크 프로토콜의 검색 결과로서 검출된 네트워크 디바이스를 제외한 후에 남아 있는 네트워크 디바이스 그룹을, 어떠한 네트워크 호환성 플러그 앤 플레이 기능도 지원하지 않는 네트워크 디바이스 그룹으로서 판정하는 네트워크 디바이스 관리 장치 제어 방법.

### 청구항 17

제9항 또는 제12항 내지 제16항 중 어느 한 항에 있어서,

상기 네트워크 디바이스는 네트워크 프린터인 네트워크 디바이스 관리 장치 제어 방법.

### 청구항 18

제17항에 있어서,

상기 네트워크 디바이스가 복수의 프린터 언어를 지원하는 경우, 상기 생성 회신 단계는 개개의 프린터 언어마다 독립적인 논리적 가상 네트워크 호환성 플러그 앤 플레이 프린터로서 응답하는 네트워크 디바이스 관리 장치 제어 방법.

### 청구항 19

네트워크에 접속되고 복수의 기능을 가진 네트워크 디바이스를 관리하는 네트워크 디바이스 관리 장치로서,

상기 네트워크 디바이스에 관련된 요구를 수신한 경우, 상기 네트워크 디바이스의 복수의 기능 각각에, 각각이 대응하는 복수의 식별 정보를 회신하는 응답 수단

을 포함하고,

상기 응답 수단이 회신할 복수의 식별 정보는 상기 네트워크 디바이스의 복수의 기능에 대응하는 복수의 디바이

스 드라이버를 설치하는데 이용되는 네트워크 디바이스 관리 장치.

### 청구항 20

제19항에 있어서,

상기 네트워크 디바이스는 복수의 인쇄 언어를 지원하고, 상기 복수의 식별 정보의 각각은 상기 복수의 인쇄 언어의 각각에 대응하는 네트워크 디바이스 관리 장치.

### 청구항 21

네트워크에 접속되고 복수의 기능을 가진 네트워크 디바이스를 관리하는 네트워크 디바이스 관리 장치를 제어하는 방법으로서,

상기 네트워크 디바이스에 관련된 요구를 수신한 경우, 상기 네트워크 디바이스의 복수의 기능 각각에, 각각이 대응하는 복수의 식별 정보를 회신하는 응답 단계

를 포함하고,

상기 응답 단계에서 회신할 복수의 식별 정보는 상기 네트워크 디바이스의 복수의 기능에 대응하는 복수의 디바이스 드라이버를 설치하는데 이용되는 네트워크 디바이스 관리 장치 제어 방법.

### 청구항 22

제21항에 있어서,

상기 네트워크 디바이스는 복수의 인쇄 언어를 지원하고, 상기 복수의 식별 정보의 각각은 상기 복수의 인쇄 언어의 각각에 대응하는 네트워크 디바이스 관리 장치 제어 방법.

### 청구항 23

제21항에 기재한 방법을 실행시키기 위한 컴퓨터 프로그램을 저장하는 컴퓨터 판독가능한 기억 매체.

## 명세서

### 기술 분야

<1> 본 발명은 네트워크 디바이스를 관리하기 위한 기술에 관한 것이다.

### 배경 기술

<2> 인쇄 장치는 특정 정보 처리 장치, 정보 처리 장치 상에서 실행되는 오퍼레이팅 시스템(이하에서는, OS로 약칭 함), 또는 그 OS 상에서 실행되는 어플리케이션에 의존하지 않도록, PDL(페이지 디스크립션 언어) 등과 같은 독자의 렌더링 언어 체계인 커맨드 인터페이스를 가지고 있다.

<3> 렌더링 언어 체계는 각각의 개별적인 인쇄 장치에 의존한다. 그러한 의존성을 캡슐화하기 위해, OS는 표준 렌더링 인터페이스를 입력으로 하고, 그 입력을 인쇄 장치-의존 렌더링 언어 체계의 커맨드로 변환하여, 그 커맨드를 출력하는 프린터 드라이버라는 모듈을 정의한다. 프린터 드라이버는 통상 인쇄 장치의 벤더, OS 개발자 등에 의해 생성되고, 정보 처리 장치에 저장된다.

<4> 정보 처리 장치에 저장되는 프린터 드라이버는 다수 존재한다. 이 때문에, 정보 처리 장치에 저장된 모든 프린터 드라이버를 제공하는 것은 사용자를 혼란스럽게 한다. 즉, 프린터 드라이버는 사용자가 이용하는 인쇄 장치에만 효율적으로 관련되어, 사용자가 이용 가능한 상태로 하는 것이 바람직하다. 그러므로, OS의 디폴트 상태에서는 단지 프린터 드라이버가 정보 처리 장치에 저장만 되고, 사용자는 그 프린터 드라이버를 특정 인쇄 장치에 관련시키기 위한 동작으로서 OS에 프린터 드라이버를 설치해야만 한다.

<5> 이러한 설치 프로세스는 종래에는 사용자가 수동으로 행하였다. 그러나, 사용자는 복잡한 프린터 드라이버 설치 프로세스를 행해야 한다. 다른 렌더링 언어 체계의 커맨드가 그 렌더링 언어 체계를 지원하지 않는 인쇄 장치에 전송되는 경우, 인쇄 에러가 발생한다. 그 경우에, 사용자는 해당 인쇄 장치에 대응하는 프린터 드라이버를 정확하게 설치해야 하는데, 그러한 프로세스는 초보자에게는 어려운 것이다.

- <6> 그러한 복잡한 설치 프로세스 시의 부하를 감소시키기 위해, 센트로닉스 데이터컴퓨터 인크(Centronics Datacomputer Inc.)의 사양에 부합하는 병렬 인터페이스, USB 인터페이스 등과 같은 통신 매체를 통해 정보 처리 장치와 인쇄 장치를 접속시키는 것만으로, 어떠한 사용자의 조작 없이도 자동적으로 정보 처리 장치와 인쇄 장치간에 양방향 통신을 행하여, 인쇄 장치에 대응하는 드라이버를 정보 처리 장치에 설치하는 기능을 OS에 도입하고 있다. 이러한 기능을 플러그 앤 플레이(Plug and Play) 기능이라고 한다(예를 들면, 일본 특허 공개 제2003-216378호).
- <7> 플러그 앤 플레이 기능은 직접적인 일대일 접속에 의해 달성된다. 최근, 네트워크 인프라구조의 구축과 함께, 네트워크 호환성 인쇄 장치, 스캐너, 복사기 등과 같은 네트워크 주변 디바이스들이 급속하게 보급되었다. 또한, 네트워크를 통해 네트워크 상에서 다양한 서비스를 제공하는 디바이스를 검색하기 위한 기술들이 개발되었다. 예를 들면, 그러한 기술로서, 마이크로소프트사에 의해 제안된 범용 플러그 앤 플레이(이하에서는, UPnP로 약칭됨)가 적용하다(예를 들면, 일본 특허 공개 제2003-6133호).
- <8> 네트워크 호환성 네트워크 디바이스로서 인쇄 장치를 예로 들면, 이용할 클라이언트(예를 들면, 퍼스널 컴퓨터) 상에서 실행되는 오퍼레이팅 시스템(예를 들면, 윈도우즈(R))이 데이터베이스 상에서 등록되고 관리되는 대응하는 드라이버를 미리 설치하거나, 인쇄 장치 벤더로부터 플렉시블 디스크, CD-ROM 등과 같은 기록 매체를 통해 제공되는 드라이버 소프트웨어(드라이버)를 설치한다. 그런 후, 인쇄 장치에 할당된 IP 어드레스, 프린터 포트, 이용될 인쇄 프로토콜, 디바이스 드라이버 등과 같이 인쇄 장치를 동작시키기 위한 정보를 파악할 필요가 있다.
- <9> 또한, (그 인쇄 장치를 다른 네트워크로 이동시키거나, 그 네트워크에 새로운 인쇄 장치를 추가하는 등에 의해서) 네트워크 시스템에 접속된 디바이스들의 시시각각 변화하는 구성을 감안하여, 이들을 관리하기 위해서 네트워크 상에 접속된 인쇄 장치의 정보를 파악하고 관리하기 위한 기술이 개발되었다.
- <10> 그러나, UPnP 등과 같은 기존의 관리 시스템은 인쇄 장치와 컴퓨터를 접속하기 위한 기술 사양이기는 하지만, 그것은 디바이스 간의 통신에 필요한 프로토콜과 데이터 포맷을 규정하고 있을 뿐이다.
- <11> 그러므로, UPnP 등과 같은 관리 시스템이 관리하고 있는 정보를 이용하는 경우라도, 인쇄 장치를 제어하기 위한 디바이스 드라이버 등과 같은 제어 프로그램을 컴퓨터에 설치하기 위해서는 예전대 복잡한 설정이 필요하다는 문제점이 해결되지 않은 채로 여전히 존재한다.
- <12> 네트워크 호환성 인쇄 장치를 관리하는 복수의 관리 방법이 병존하는 경우, 특정 관리 방법을 지원하는 관리 시스템은 특정 관리 방법을 지원하지 않는, 네트워크 상의 다른 인쇄 장치를 인식할 수 없다.
- <13> 다양한 국가의 사용자 및 시장의 요구를 충족시키기 위해서, 매우 다양한 인쇄 장치용 렌더링 언어 체계가 개발되어, 시장에 출시되었다. 각 렌더링 언어 체계에 대한 인쇄 장치를 처음부터 개발하는 것은 비용이 높다. 그러한 비용을 줄이기 위해, 각 렌더링 언어 체계에 의존하는 부분을 옵션 보드 또는 소프트웨어로서 분리하여, 그 옵션 보드 또는 소프트웨어를 다양한 국가의 사용자 요구 또는 시장 요구에 따라 인쇄 장치에 다시 탑재할 수 있는 인쇄 장치가 가능하게 되었다.
- <14> 게다가, 사용자가 선호하는 복수의 렌더링 언어 체계가 존재하는 환경에서, 각각의 인쇄 장치에서 복수의 렌더링 언어 체계를 지원하는 인쇄 장치들도 가능하다.
- <15> 렌더링 언어 체계가 옵션 보드 또는 소프트웨어에 의해 변경될 수 있는 전술한 인쇄 장치나, 복수의 렌더링 언어 체계를 지원하는 인쇄 장치는 플러그 앤 플레이 기능 탑재 시에는 상정되지 않았다.
- <16> 플러그 앤 플레이 기능은 사용자의 복잡한 설치 프로세스 시의 부하를 줄이기 위해 도입된 것이므로, 검출된 인쇄 장치에 대해 복수의 렌더링 언어 체계로부터 PDL을 선택하는 등 인쇄 장치의 조작에 익숙하지 않은 사용자들을 혼동시키는 선택 프로세스에 대한 필요성을 제거할 수 있는 사용자 인터페이스가 요구된다.
- <17> 상기 2가지 이유 때문에, 소정의 OS 상에 설치된 플러그 앤 플레이 기능은 인쇄 장치로부터 정보 처리 장치에 전달되는 인쇄 장치의 식별 데이터(DeviceID) 중, 렌더링 언어 체계를 나타내는 정보를 무시하고, 벤더명과 프린터명에만 기초하여 그 인쇄 장치에 대응하는 프린터 드라이버를 정보 처리 장치로부터 검색하고, 처음에 검출된 프린터 드라이버를 설치한다.
- <18> 전술한 플러그 앤 플레이 기능은 다음과 같은 결점들을 가지고 있다.
- <19> 첫째로, 예를 들면, 인쇄 장치에 옵션 보드를 탑재하여 지원 렌더링 언어 체계를 전환할 수 있는 인쇄 장치에 2

개의 렌더링 언어 체계(프린터 언어 인터프리터: PDL1, PDL2)를 탑재 가능하지만, 단지 PDL1만이 실제로 인쇄 장치 상에 탑재하는 것으로 한다. 또한, 이들 언어들에 대한 프린터 드라이버들이 이미 정보 처리 장치에 저장되어 있는 것으로 한다. 이 경우에, 플러그 앤 플레이 기능은 인쇄 장치로부터 정보 처리 장치에 전달되는 인쇄 장치의 식별 정보 중, 렌더링 언어 체계를 나타내는 정보를 무시하고, 벤더명과 프린터명만을 이용하여 그 인쇄 장치에 대응하는 프린터 드라이버를 정보 처리 장치로부터 검색한다. 플러그 앤 플레이 기능이 먼저 PDL2에 대한 프린터 드라이버를 검출하는 경우에는, 이 인쇄 장치용으로 PDL2를 OS에 설치한다. PDL1용의 렌더링 언어 체계를 처리하는 옵션 보드는 PDL2를 인식하지 못하고, 따라서 인쇄 에러가 발생하게 된다.

<20> 둘째로, 복수의 렌더링 언어 체계를 지원할 수 있는 인쇄 장치의 경우에, 플러그 앤 플레이 기능은 벤더가 권장하지 않는 에뮬레이션을 위한 렌더링 언어 체계의 프린터 드라이버를 먼저 검출하게 되어, 다른 효율적인 렌더링 시스템이 이용되지 않는다.

### 발명의 상세한 설명

<21> 본 발명의 목적은, 네트워크 호환성 플러그 앤 플레이 기능을 지원하지 않는 디바이스를 네트워크 호환성 플러그 앤 플레이 기능을 지원하는 디바이스로서 네트워크 상의 다른 디바이스들에게 인식시켜, 그 디바이스를 그 기능을 가진 디바이스로서 기능시키기 위한 기술을 제공하는 것이다.

<22> 이러한 목적을 달성하기 위해, 예컨대, 본 발명에 따른 네트워크 디바이스 관리 장치는 이하의 구성을 포함한다. 즉, 네트워크 접속 수단을 구비하며, 네트워크에 접속되는 복수의 기능을 가진 네트워크 디바이스를 관리하는 네트워크 디바이스 관리 장치가 제공되며, 이 네트워크 디바이스 관리 장치는,

<23> 어떠한 네트워크 호환성 플러그 앤 플레이 기능도 지원하지 않는 적어도 하나의 네트워크 디바이스의 네트워크 어드레스와, 상기 네트워크 디바이스의 복수의 기능과 연관된 기능 정보를 기억하는 기억 수단; 및

<24> 상기 네트워크 접속 수단을 통해 네트워크 호환성 플러그 앤 플레이 디바이스의 소재 확인 요구를 수신한 경우, 상기 기억 수단에 기억된 복수의 기능 정보가 나타내는 기능들에 대응하는 복수의 독립적인 가상 네트워크 호환성 플러그 앤 플레이 디바이스로서, 상기 네트워크 호환성 플러그 앤 플레이 기능을 지원하지 않는 네트워크 디바이스를 특정하는 식별 정보를 포함하는 메시지를 생성하여 회신하는 응답 수단을 포함한다.

<25> 본 발명의 다른 특징 및 장점들은 도면 전체에 걸쳐 동일 또는 유사한 부품에는 유사한 참조 번호를 붙인 첨부 도면을 참조로 한 다음의 상세한 설명으로부터 명백하게 될 것이다.

### 실시예

<39> 이하, 첨부 도면을 참조하여 본 발명의 양호한 실시예들을 설명한다.

<40> 이하, 본 발명에 따른 서비스 제공 시스템의 일 실시예인 프로토콜 변환 시스템을 설명한다.

<41> 도 1은 본 발명의 실시예로서 인쇄 시스템의 소프트웨어 구성을 도시하는 블록도이다.

<42> 클라이언트(100)에는, 마이크로소프트사가 제공하는 OS인 윈도우즈(R), 애플컴퓨터사가 제공하는 Mac OS(R) 등과 같은 범용 오피레이팅 시스템, 및 그 OS 상에서 실행되는 범용 어플리케이션이 설치된다.

<43> 범용 오피레이팅 시스템은 통신 기능으로서 TCP/UDP/IP 프로토콜 스택(107), 및 그 프로토콜 스택 상의 HTTP(106)를 포함하여, HTTP 요구의 해석 및 응답 프로세스를 행한다. 범용 오피레이팅 시스템은 HTTP(106) 상에, UPnP와 연관된 통신 수단을 형성하는 심플 오브젝트 액세스 프로토콜(SOAP) 프로세서(103)를 포함한다. 드라이버 자동 다운로드/설정 모듈(101)은 전술한 통신 수단을 이용하여 네트워크 상의 UPnP 호환성 인쇄 장치 및 본 실시예에서 설명할 프록시 응답 디바이스와 통신한다.

<44> 드라이버 자동 다운로드/설정 모듈(101)은 GUI(102)를 이용하여 클라이언트(100) 상에 UPnP에 의해 검출된 인쇄 장치를 표시하고 사용자에게 선택하도록 프롬프트한다.

<45> 구성기(Configurator) 모듈(104)은 네트워크를 통해 인쇄 장치로부터 획득한 정보에 기초하여 하드 디스크에 저장된 것들로부터 대응하는 프린터 드라이버를 OS에 등록(설치)한다.

<46> 설치 시, 메모리 컨트롤러(105)를 통해 메모리 공간의 유무 및 설치처의 관리 제어를 실행한다. 메모리 컨트롤러(105)는 하드 디스크 드라이브(OS, 각종 어플리케이션, 전술한 소프트웨어 모듈, 및 어플리케이션에 의해 생성된 데이터 파일을 저장함)를 제어한다.

- <47> 한편, 본 실시예에서, 네트워크 호환성 인쇄 장치(200)는 통신 기능으로서 TCP/UDP/IP 프로토콜 스택(5), 및 그 프로토콜 스택 상의 심플 네트워크 관리 프로토콜(SNMP) 프로세서(6)를 포함한다. 프로토콜 스택(5) 상에는 인쇄 프로토콜 프로세서(7)가 탑재되어, 클라이언트로부터 발행되는 인쇄 요구를 해석하고 그 인쇄 요구를 인쇄 컨트롤러(8)에 출력하는 기능을 갖는다. 유의할 점은, 인쇄 장치(200)가 동시에 복수의 PDL에 대한 인쇄 컨트롤러(8)를 가질 수 있다는 점이다. 또한, 인쇄 컨트롤러(8)는 탈착 가능하게 설계될 수 있고, 사용자의 요구에 따라 대체될 수 있으므로, 인쇄 장치(200)가 지원하는 PDL을 변경할 수도 있다. 본 실시예에서, 인쇄 장치(200)는 2개의 PDL, 즉 PDL1 및 PDL2에 대한 인쇄 컨트롤러(8a, 8b)를 포함하는 것으로 한다. 이 인쇄 장치는 어떠한 UPnP 프로토콜 프로세서도 포함하지 않으며, 자체적으로는 클라이언트(100)로부터 발행되는 UPnP 프로토콜을 이용하여 디바이스 검색 요구 및 UPnP 인쇄 작업 요구에 응답할 수 없다.
- <48> 다른 네트워크 호환성 디바이스, 즉 본 실시예의 네트워크 호환성 인쇄 장치(400)는 통신 기능으로서의 TCP/UDP/IP 프로토콜 스택(17), 및 그 프로토콜 스택 상의 HTTP(19)를 포함하여, HTTP 요구의 해석 및 응답 프로세스를 행한다.
- <49> 인쇄 장치(400)는 인쇄 장치(200)에서와 같이 그 프로토콜 스택 상에 심플 네트워크 관리 프로토콜(SNMP)(18)을 포함한다.
- <50> 인쇄 장치(400)는 HTTP(19)의 상위층으로서 심플 오브젝트 액세스 프로토콜(SOAP) 프로세서(20) 및 UPnP 프로토콜 프로세서(21)를 포함한다. 네트워크 호환성 인쇄 장치(400)는 UPnP 포럼에 의해 지정된 PrintBasic 서비스를 가지고 있고, UPnP 프로토콜 프로세서는 그 서비스에 의해 정의된 인쇄 작업 및 그의 속성 정보를 해석하고 그 인쇄 요구를 인쇄 컨트롤러(22)에 출력하는 기능을 가지고 있다. 인쇄 장치(400)는 동시에 복수의 PDL에 대한 인쇄 컨트롤러(22)를 포함할 수 있다. 또한, 인쇄 컨트롤러(22)는 탈착 가능하게 설계될 수 있고, 사용자의 요구에 따라 대체될 수 있으므로, 인쇄 장치(400)가 지원하는 PDL을 변경할 수도 있다. 본 실시예에서, 인쇄 장치(400)는 2개의 PDL, 즉 PDL1 및 PDL3에 대한 인쇄 컨트롤러(22a, 22b)를 포함하는 것으로 한다.
- <51> 마찬가지로, 본 실시예의 특징적 구성인 네트워크 관리 장치로서 기능하는 프록시 서버(300)는 통신 기능으로서 TCP/UDP/IP 프로토콜 스택(9) 및 그 프로토콜 스택 상의 HTTP(10)를 포함하여, HTTP 요구의 해석 및 응답 프로세스를 행한다.
- <52> 프록시 서버(300)는 프로토콜 스택(9) 상에 심플 네트워크 관리 프로토콜(SNMP) 프로세서(11)를 포함하고, 이러한 프로토콜에 의해 어떠한 UPnP 프로토콜 프로세스도 포함하지 않는 네트워크 호환성 인쇄 장치(200)의 검색 프로세스 및 정보의 획득 프로세스를 실시한다.
- <53> 프록시 서버(300)는 프로토콜 스택(9) 상에 인쇄 프로토콜 프로세서(12)를 포함한다. 인쇄 프로토콜 프로세서(12)는 어떠한 UPnP 프로토콜 프로세서도 포함하지 않는 네트워크 호환성 인쇄 장치(200)에 인쇄 작업을 발행한다.
- <54> 프록시 서버(300)는 HTTP(10)의 상위층으로서 심플 오브젝트 액세스 프로토콜(SOAP) 프로세서를 포함한다. 복수의 클라이언트(100) 및 프록시 서버(300)가 네트워크 상에 존재하는 경우, UPnP 프로토콜 프로세서(14) 및 프로토콜 변환 프로세서(16)는 프로세서(13)를 통해 확장 가능한 마크업 언어(XML)로 기술된 데이터의 양방향 통신을 실현한다.
- <55> 프로토콜 변환 프로세서(16)는 SNMP 프로세서(11), SOAP 프로세서(13), UPnP 프로세서(14), 인쇄 프로토콜 프로세서(12) 및 메모리 컨트롤러(15)의 상위층으로서 배치된다. 프로토콜 변환 프로세서(16)는 SNMP 프로세서(11)를 통해 획득한 네트워크 호환성 인쇄 장치의 정보를, UPnP 프로토콜에서 이용하는 각종 XML 문서들을 생성한 후, 메모리 컨트롤러(15)가 제어하는 메모리 상에 기록한다. 또한, UPnP 프로토콜로부터의 요구가 있을 때, 프로토콜 변환 프로세서(16)는 대응하는 관리 테이블 상에 기록된 XML 문서를 판독하여, 이를 UPnP 프로세서(14)에 송신한다.
- <56> UPnP 프로토콜에 의한 인쇄 작업 요구를 수신한 경우에, 프로토콜 변환 프로세서(16)는 SOAP 프로세서(14)를 통해 작업 커맨드 및 작업 속성 정보를 획득하고, 그 내용을 출력 지정된 인쇄 장치가 지원하는 인쇄 프로토콜로 변환한 후, 이를 인쇄 프로토콜 프로세서(12)를 통해 지정된 인쇄 장치에 송신한다.
- <57> 프로토콜 변환 프로세서(16)는 메모리 컨트롤러(15)가 제어하는 메모리 상에서 프록시 서버(300)가 관리하는 관리 테이블의 판독/기록 프로세스를 그 컨트롤러(15)를 통해 실행한다. 마찬가지로, 프로토콜 변환 프로세서(16)가 네트워크 상에 존재하는 다른 프록시 서버가 관리하는 관리 테이블을 획득한 경우, 메모리 컨트롤러(15)

5)가 제어하는 메모리 상에서 프록시 서버(300)가 관리하는 관리 테이블의 판독/기록 프로세스를 그 컨트롤러(15)를 통해 실행한다.

<58> 각 장치의 인쇄 프로세스와 관련하여 탑재되는 컴퓨터 프로그램의 구성을 설명하였다. 클라이언트(100) 및 프록시 서버(300)는 퍼스널 컴퓨터 등과 같은 범용 정보 처리 장치에 의해 실현될 수 있다. 하드웨어 구성은 예를 들면, 도 11에 도시된 바와 같다.

<59> 도 11을 참조하면, 참조부호 1001은 전체 장치를 제어하는 CPU를 지칭하고, 1002는 BIOS 및 부트 프로그램을 기억하는 ROM을 지칭한다. 참조부호 1003은 CPU(1001)의 작업 영역으로 이용되는 RAM을 지칭한다. 참조부호 1004는 OS, 어플리케이션 및 각종 프로그램, 데이터 파일 등을 기억하는 하드디스크 드라이브를 지칭한다. 참조부호 1005는 네트워크 인터페이스를 지칭한다. 본 실시예에서, 네트워크 인터페이스(1005)는 이더넷(R) 카드를 포함하지만, 대신에 무선 LAN 카드를 포함할 수도 있다. 참조부호 1006은 표시 컨트롤러를 지칭하는 것으로, 비디오 메모리, 및 CPU(1001)의 제어하에서 렌더링을 수행하고 비디오 메모리에 기억된 이미지 데이터를 비디오 신호로서 출력하는 비디오 컨트롤러를 나타낸다. 참조부호 1007은 액정 디스플레이 또는 CRT로 표시되는 표시 디바이스를 나타낸다. 참조부호 1008은 키보드, 마우스 등과 같은 입력 디바이스를 나타낸다.

<60> 상기 구성에서, 장치의 전원 스위치가 터온되면, CPU(1001)는 ROM(1002)의 부트 프로그램에 따라 하드디스크(1004)로부터 OS를 RAM(1002) 상에 로딩하고, 이어서 다양한 디바이스 드라이버를 로딩한다. 도 11에 도시된 장치가 클라이언트(100)인 경우, CPU(1001)는 도 1에 도시된 구성에 대응하는 프로그램을 로딩하여 실행한다. 프록시 서버(300)의 경우에, OS가 로딩된 후, CPU(1001)는 도 1에 도시된 구성에 대응하는 프로그램을 로딩하고, 후술하는 프로세스들을 실행한다.

<61> 도 12는 네트워크 프린터의 블록도이다. 도 12를 참조하면, 참조부호 2001은 전체 프린터를 제어하는 CPU를 나타내고, 2002는 CPU(2001)가 실행할 인쇄 프로세스 프로그램, 및 폰트 데이터를 기억하는 ROM을 나타낸다. 참조부호 2003은 CPU(2001)의 작업 영역, 수신 버퍼, 및 이미지 렌더링 영역으로서 이용되는 RAM을 나타내고, 참조부호 2004는 다양한 스위치 및 버튼과, 메시지를 표시하는데 이용되는 액정 표시 유닛을 포함하는 제어 패널을 나타낸다. 참조부호 2005는 네트워크를 접속하는데 이용되는 네트워크 인터페이스를 나타내고, 참조부호 2006은 렌더링 유닛을 나타내며, 참조부호 2007은 실제로 인쇄 시트 상에 인쇄하는 프린터 엔진을 나타낸다.

<62> 도 1에 도시된 인쇄 장치(200, 400)는 넓은 관점에서 봤을 때 도 12에 도시된 하드웨어 구성을 가지고 있다. 도 1에 도시된 인쇄 장치들(200, 400)간의 차이는 ROM(2002)에 기억된 펌웨어의 차이이거나, 일부 경우에 다양한 옵션 보드의 차이이다.

<63> 상기 구성을 가지는 실시예의 시스템의 제어 플로우는 도 2에 도시된 플로우차트를 참조하여 이하에 설명된다.

<64> 도 2는 본 실시예에서 프록시 서버(300)의 처리 시퀀스를 도시하는 플로우차트이다. 본 플로우차트에 도시된 처리는 프록시 서버(300)의 전력 스위치가 터온되어 OS가 런칭된 후에, 하드 디스크(1004)로부터 RAM(1003) 상에 로딩되어 실행되는 서버 어플리케이션에 대응한다.

<65> 프록시 서버(300)의 프로토콜 변환 프로세서(16)는 런칭된 후, 하드 디스크(104)에 미리 확보되어 있고, 프로토콜 변환 프로세스가 적용되는 네트워크 디바이스 정보를 기록하는 관리 테이블의 내용을 클리어링한다(단계 S2-1). 관리 테이블에 대한 상세는 이하의 프로세스에서 설명할 것이다.

<66> 네트워크에 로그온하여 서비스를 시작할 때, 네트워크 상에 존재하는 UPnP 호환성 인쇄 장치를 검색한다(단계 S2-2). 이러한 검색 프로세스는 도 3에 도시되어 있고 이하에 설명될 것이다.

<67> 단계 S3-1에서, 프록시 서버는 UPnP에 의해 규정되는 도 4에 도시된 포맷을 가지는 Probe 패킷을 멀티캐스트 어드레스에 발행한다. 즉, 프록시 서버는 네트워크에 로그온하고 UPnP 호환성 네트워크 디바이스의 소재를 획득하는데 이용되는 메시지를 송신한다. 유의할 점은, 도 4의 패킷 메시지내의 uuid(Universally Unique Identifier)가 UPnP 디바이스에 고유한 식별자이고, 프록시 서버(300)가 UPnP 디바이스로서 가능하고 있는 것처럼 나타낸다는 점이다(uuid는 미리 설정되어 유지된다).

<68> 프록시 서버(300)의 프로토콜 변환 프로세서(16)는 도 4에 도시된 Probe 패킷을 발행한 후, 소정 기간 내에 네트워크로부터의 모든 응답들을 해석한다.

<69> 본 실시예에서, 네트워크(500) 상의 인쇄 장치(400)가 UPnP 호환성 인쇄 장치이므로, 프록시 서버(300)로부터의 Probe 패킷에 대한 응답 메시지를 그 프록시 서버에 응답한다. 그러나, 인쇄 장치(200)는 비-UPnP 디바이스이

므로 응답하지 않는다.

- <70> 도 5는 네트워크 디바이스의 일례로서 UPnP 호환성 인쇄 장치(400)로부터의 응답 패킷인 ProbeMatch의 일례를 도시하고 있다. 프록시 서버(300)의 프로토콜 변환 프로세서(16)는 이러한 패킷을 전달한 네트워크 인쇄 장치(UPnP 호환성 인쇄 장치(400))로의 URL을, 메모리 컨트롤러(15)를 통해, 그 메모리 컨트롤러가 제어하는 메모리(본 실시예에서는 하드 디스크)에 기록 보존한다. 이러한 프로세스는 수신된 모든 응답 패킷에 대해 실행되고, 프록시 서버(300)는 네트워크 상에 존재하는 모든 UPnP 호환성 인쇄 장치의 URL을 기록한다(단계 S3-2).
- <71> 상기 프로세스의 완료시, 단계 S3-3에서 어떠한 응답도 검출되지 않는 경우, 프록시 서버(300)의 프로토콜 변환 프로세서(16)는 UPnP 검색 프로세스를 종료시키고, 플로우는 도 2의 단계 S2-3으로 진행하여, 인쇄 장치 정보의 획득 프로세스를 시작한다.
- <72> 유의할 점은, UPnP 호환성 인쇄 장치는 Probe 패킷에 응답하여 그 인쇄 장치가 지원되는 PDL마다 ProbeMatch 포맷을 갖는 하나의 응답을 회신한다는 점이다. 도 5에 도시된 바와 같이, UPnP 호환성 인쇄 장치(400)는 인쇄 컨트롤러(22a, 22b)에 대해 동일한 URL을 가지고 있는 2개의 ProbeMatch 패킷을 발행한다. 각 ProbeMatch 패킷은 각 서비스를 고유하게 지정할 수 있도록 <Address> 태그에 고유 uuid를 갖는다.
- <73> 도 5의 예에서, PDL1에 대한 uuid는 uuid:98190dc2-0890-4ef8-ac9a-5940995e661a이고, PDL3에 대해서는 uuid:98190dc2-0890-4ef8-ac9a-5940995e661b이다.
- <74> 도 6은 도 2의 단계 S2-3에서의 인쇄 장치 정보 획득 프로세스의 상세를 도시하는 플로우차트이다.
- <75> 프록시 서버(300)의 프로토콜 변환 프로세서(16)는 SNMP 제어 모듈(프로세서)(11)로부터 이하의 MIB 오브젝트에 SNMP Get 요구를 브로드캐스팅함으로써, 네트워크 상에 존재하는 인쇄 장치 정보를 획득한다(단계 S6-1).
- <76> PrinterMakeAndModel: 인쇄 장치 벤더/제품명
- <77> PrinterName: 인쇄 장치명
- <78> PrinterLocation: 인쇄 장치 소재
- <79> IPAddress: 인쇄 장치 IP 어드레스
- <80> MACAddress : 인쇄 장치 MAC 어드레스
- <81> SupportedPDL: 지원하는 페이지 기술 언어
- <82> SupportedPrintProtocol : 지원하는 인쇄 프로토콜
- <83> 상기 SNMP Get 요구의 수신시, 네트워크 디바이스인 각 인쇄 장치(200, 400)는 인쇄 장치의 SNMP 프로세서(6) 또는 SNMP 프로세서(18)를 이용하여 각 오브젝트에 대응하는 정보를 생성하고, SNMP 응답으로서 응답 패킷(응답 메시지)을 프록시 서버(300)에 유니캐스팅한다.
- <84> 단계 S6-1-1에서는 응답이 수신되었는지 여부가 체크된다. 응답이 수신된 것으로 결정되면, 플로우는 단계 S6-2로 진행하고, 그렇지 않으면 플로우는 단계 S6-9로 진행한다.
- <85> 각 네트워크 호환성 인쇄 장치로부터 응답을 수신한 프록시 서버(300)의 프로토콜 변환 프로세서(16)는 응답 내용과, 메모리에 이미 등록된 인쇄 장치 관리 테이블의 내용을 비교하여(단계 S6-2), 인쇄 장치가 이미 프로토콜 변환을 실행했는지 여부를 체크한다(단계 S6-3). 이러한 체크 프로세스는 uuid 또는 IP 어드레스를 대조함으로써 탈성된다.
- <86> 단계 S6-3에서, 인쇄 장치가 프로토콜 변환을 실행하지 않았는지, 즉 새로운 인쇄 장치가 발견되었는지 여부가 체크된다. 새로운 인쇄 장치가 발견된 경우, 프록시 서버(300)의 프로토콜 변환 프로세서(16)는 그 인쇄 장치가 UPnP 호환성인지 여부를, 이전에 메모리에 기록된 UPnP 호환성 인쇄 장치의 URL을 비교함으로써 체크한다(단계 S6-4).
- <87> 즉, SNMP Get 요구에 대한 응답으로서 획득한 인쇄 장치 IP 어드레스가 메모리 상에 기록된 URL과 매칭하는 경우, 즉 인쇄 장치가 도 2의 단계 S2-2에서의 검색 프로세스에 대하여 응답하고 단계 S6-1에 대하여도 응답한 경우, 해당 인쇄 장치는 UPnP 호환성 인쇄 장치인 것으로 결정되고, 그 인쇄 장치에 어떠한 프로토콜 변환도 적용되지 않는다. 즉, 그 인쇄 장치는 관리 테이블에 등록되지 않는다.
- <88> 환연하면, 인쇄 장치가 도 2의 단계 S2-2의 검색 프로세스에 대하여 응답하지 않고 단계 S6-1에 대하여 응답한

경우, 해당 인쇄 장치는 비-UPnP 인쇄 장치인 것으로 결정된다. 이 경우에, 플로우는 단계 S6-6으로 진행하여 메모리 컨트롤러(15)를 통해 하드 디스크 상에 확보된 관리 테이블에 그 인쇄 장치를 추가 등록한다.

<89> 이 경우에, 본 실시예에 따르면, SNMP Get 요구에 대한 응답으로서 획득한 정보의 인쇄 장치 벤더/제품명을 이용하여 검색을 수행하고, 그 인쇄 장치가 복수의 언어의 인쇄 컨트롤러를 장착할 수 있거나 탈착 가능한 인쇄 컨트롤러를 지원하는 모델인 경우에, SupportedPDL에 기초하여 지원되는 PDL을 확인하고, 각각의 개별적인 PDL을 지원하는 컨트롤러에 대응하여, 각각 논리적으로 독립적인 인쇄 장치로서, UPnP 호환성 인쇄 장치로서 규정하도록, 관리 테이블에 PrinterDescription 요소를 추가한다.

<90> 본 실시예의 시스템 구성에서, 인쇄 장치(200)는 비-UPnP 인쇄 장치이고, 도 7에 도시된 바와 같이, 인쇄 컨트롤러(8a, 8b)에 대응하여 논리적으로 독립적인 인쇄 장치로서 인쇄 장치 관리 테이블에 2개의 PrinterDescription 요소가 추가된다. 프록시 서버(300)는 클라이언트로부터의 Probe 패킷 또는 소정의 검색 패킷에 응답하여 도 7에 도시된 정보를 네트워크 상에 출력한다. 도 7에 도시된 패킷은 인쇄 컨트롤러(8a)에 대한 디바이스 ID 및 인쇄 컨트롤러(8b)에 대한 디바이스 ID로서 분리되거나, 개별적인 디바이스 ID로 인식될 수 있는 포맷으로 함께 클라이언트 측에서 수신된다. 그리고, 각 디바이스 ID마다 플러그 앤 플레이 프로세스가 실행된다. 즉, 클라이언트 측의 설치기가 인쇄 컨트롤러(8a)에 대한 디바이스 ID를 획득하는 경우, PDL1을 설치하고, 클라이언트 측의 설치기가 인쇄 컨트롤러(8b)에 대한 디바이스 ID를 획득하는 경우, 다른 PDL2에 대한 드라이버를 설치한다. 클라이언트가 디바이스 검색 요구를 발행하는 경우, 인쇄 장치(200)는 하나의 디바이스로 표시되는 것이 아니라, 복수의 논리적 디바이스로 표시된다. 어느 프린터 드라이버가 설치될지 여부는 클라이언트 측의 디바이스 ID 및 드라이버 저장 소재를 나타내는 정보 세트를 나타내는 데이터베이스를 참조함으로써 미리 결정된다. 예를 들면, 클라이언트가 프록시 서버(300)로부터 획득한 인쇄 컨트롤러(8a)에 대한 디바이스 ID 필드에서 MANUFACTURED 및 MODEL 세트를 클라이언트 측의 설치기가 인식하는 경우, 인쇄 컨트롤러(8a)에 대한 MANUFACTURED 및 MODEL 세트를 이용하여 데이터베이스를 참조하여 프린터 드라이버의 저장 소재를 인식하여 그 드라이버를 설치한다. 대안적으로, 설치기는 인쇄 컨트롤러(8a)에 대한 디바이스 ID 필드에서 MANUFACTURED, COMMAND, 및 MODEL의 모든 세트를 인식하고, MANUFACTURED 및 MODEL에 대응하는 복수의 디바이스 드라이버로부터 PDL1에 대한 디바이스 드라이버를 자동으로 인식하여 설치할 수도 있다.

<91> PrinterDescription 요소의 DeviceID로서, 프로토콜 변환 프로세서(16)는 클라이언트(100)에 저장된 PDL에 대응하는 프린터 드라이버를 고유하게 지정할 수 있는 디바이스 ID(도 7에서, 2개의 ID, 즉 "LaserBeamPrinter777 PDL1" 및 "LaserBeamPrinter777 PDL2")를 대응하는 PrinterDescription 요소에 등록한다. 또한, 프로토콜 변환 프로세서(16)는 각각의 논리적 인쇄 장치를 UPnP 패킷을 이용하여 지정할 수 있도록, 상이한 uuid 값들을 가지는 각각의 논리적 인쇄 장치를 <Address> 태그에 등록한다.

<92> 상기한 정보는 메모리 컨트롤러(15)를 통해 하드 디스크 상에 기록된다(단계 S6-7). 단계 S6-8에서, UPnP 프로토콜 프로세서(14)는 관리 테이블에 기록된 모든 인쇄 장치와 연관된 Hello 패킷을 발행하여, 2개의 UPnP 장치(2개의 가상 UPnP 인쇄 장치)가 그 서비스를 개시하고 있는 것처럼 네트워크 상의 디바이스에 통지한다.

<93> 한편, 프록시 서버(300)에 의해 발행된 SNMP Get 요구에 응답하여 어떠한 응답도 얻지 못한 경우, 즉, 단계 S6-1-1에서 아니오로 판정되는 경우, 플로우는 단계 S6-9로 진행하여, 관리 테이블을 탐색하고 이미 등록된 인쇄 장치를 확인한다.

<94> 이미 등록된 인쇄 장치가 어떠한 응답도 회신하지 않았는지 여부가 체크된다(단계 S6-9-1). 이미 등록된 인쇄 장치가 어떠한 응답도 회신하지 않는 경우, 각 인쇄 장치의 전원 스위치가 턴오프되어 있거나 인쇄 장치가 네트워크에서 로그아웃되어 있다는 것을 나타낸다. 이 경우에, 플로우는 단계 S6-10으로 진행하고, 프록시 서버(300)의 프로토콜 변환 프로세서(16)가 관리 테이블로부터 대응하는 인쇄 장치의 정보를 삭제함으로써 관리 테이블을 업데이트한다. 단계 S6-11에서, 프로토콜 변환 프로세서(16)는 대응하는 인쇄 장치의 PrinterDescription 요소들을 삭제한다.

<95> 프록시 서버(300)의 프로토콜 변환 프로세서(16)는 UPnP 프로토콜 프로세서(14)를 이용하여 관리 테이블로부터 삭제된 모든 인쇄 장치와 연관된 Bye 패킷을 발행함으로써, 관리 테이블로부터 삭제된 이들 인쇄 장치들이 네트워크 상에서 그 서비스를 중지한다는 것을 통지한다(단계 S6-12).

<96> 도 2의 프로토콜 변환 프로세스(단계 S2-5)의 상세는 도 8의 플로우차트를 참조하여 이하에 설명된다.

<97> 프록시 서버(300)의 프로토콜 변환 프로세서(16)는 네트워크 상의 클라이언트로부터 발행된 디바이스 검색 프로토콜 Probe 패킷의 수신 메시지를 UPnP 프로토콜 프로세서(14)로부터 수신하는지 여부를 체크한다(단계 S8-1).

즉, 클라이언트가 UPnP 네트워크 디바이스를 검색하고 있는지 여부가 결정된다.

<98> Probe 패킷이 수신된 것으로 결정되는 경우, 플로우는 단계 S8-2로 진행하고, 프록시 서버(300)의 프로토콜 변환 프로세서(16)에 의해 하드 디스크 상에서 관리되는 도 7에 도시한 인쇄 장치 관리 테이블을 메모리 컨트롤러(15)를 통해 검색한다. Probe 패킷의 검색 조건과 일치하는 인쇄 장치가 관리 테이블에 등록되어 있는 경우, ProbeMatch 패킷을 인쇄 장치 관리 테이블로부터 생성하여 UPnP 프로토콜 프로세서(4)를 통해 반송한다.

<99> ProbeMatch 패킷은 도 5에 도시된 구조를 가지고 있고, Probe 패킷의 검색 조건에 일치하는 각 인쇄 컨트롤러에 대해 하나의 패킷이 발행된다. ProbeMatch 패킷은 그 PDL과 연관된 논리적 인쇄 장치를 고유하게 지정할 수 있도록, <Address> 태그에 고유 uuid를 갖는다. 본 실시예의 프록시 서버(300)는 관리 테이블 상에 등록된 모든 대응하는 인쇄 장치에 대해 ProbeMatch 패킷을 반송한다.

<100> 결과적으로, 상기 Probe 패킷을 송신한 클라이언트는 프록시 서버(300)를 UPnP 인쇄 장치(인쇄 장치(200)가 본 실시예에서 2가지 언어의 컨트롤러를 가지고 있으므로 2개의 UPnP 인쇄 장치)로 볼 수 있다.

<101> 플로우가 단계 S8-3으로 진행한 후, UPnP 프로토콜 프로세서(14)가 도 9에 도시된 PrinterDescriptionRequest 패킷을 수신하였는지 여부가 체크된다. 이러한 패킷의 수신시, 프록시 서버(300)는 PrinterDescription 획득 요구의 <To> 태그로 표시되는 uuid를 이용하여 프로토콜 변환 프로세서(16)가 관리하는 인쇄 장치 관리 테이블의 <Address> 태그를 검색하고, 인쇄 장치 관리 테이블로부터 도 10에 도시된 PrinterDescriptionResponse 패킷을 생성한다. 생성된 PrinterDescriptionResponse 패킷은 UPnP 프로토콜 프로세서(14)를 통해 요구원에 반송된다(단계 S8-4).

<102> 단계 S8-5에서, PrinterDescriptionResponse 패킷을 획득한 클라이언트 디바이스로부터 UPnP 인쇄 작업 요구를 수신하였는지 여부가 체크된다. 이러한 요구의 작업 커맨드 및 작업 속성은 XML로 기술되므로, 비-UPnP 디바이스인 인쇄 장치(200)는 이들을 그대로 해석할 수 없다. 그러므로, 프록시 서버(300)의 프로토콜 프로세서(16)가 UPnP 프로토콜 프로세서(14)를 통해 그러한 인쇄 작업을 수신하는 경우(단계 S8-5에서 예), 프로토콜 프로세서(16)는 SOAP 프로세서를 이용하여 커맨드 및 작업 속성을 해석하고, 다음에 메모리 컨트롤러(15)를 통해서 출력 지정된 인쇄 장치에 대응하는 관리 테이블 정보 중, 지원하는 인쇄 프로토콜(도 7에서 <SupportedPrintProtocol>과 </SupportedPrintProtocol>의 사이에 기술된 프로토콜로, 도 7에서는 "LPR"), 및 IP 어드레스를 획득하여, 수신한 커맨드 및 속성 정보를 획득한 인쇄 프로토콜로 변환한다(단계 S8-6). 그리고 나서, 변환된 정보를 출력 지정된 인쇄 장치의 IP 어드레스에 송신한다(단계 S8-8).

<103> 클라이언트로부터의 인쇄 작업은 인쇄 프로세스를 실행해야 하는 인쇄 장치의 uuid(프록시 서버(300)가 인쇄 장치(300)에 대한 것으로 생성한 uuid임)를 포함하므로, 이러한 정보를 키로 이용하여 인쇄 장치의 IP 어드레스를 획득할 수 있다.

<104> 인쇄 작업을 발행한 클라이언트는 순차적으로 작업 데이터(이 경우에는, PDL 데이터)를 HTTP POST 커맨드를 이용하여 프록시 서버(300)에 송신한다. 단계 S8-8-1에서 이러한 작업 데이터가 수신되어 있는지 여부가 체크된다.

<105> 이러한 체크 프로세스는 단계 S8-8-2에서 소정 기간이 경과된 것으로 결정될 때까지 수행된다. 소정 기간이 경과된 후에도 어떠한 작업 데이터도 수신되지 않은 경우, 그 작업 요구가 단계 S8-10에서 폐기된다. 이 때, 폐기 요구는 지정된 인쇄 장치에도 발행된다.

<106> 인쇄 작업 요구를 수신하고 나서 소정 기간 내에 작업 데이터를 수신한 경우, 상기한 단계에서와 같이, 프록시 서버(300)의 프로토콜 변환 프로세서(16)는 수신한 작업 데이터를 지정된 인쇄 장치가 지원하는 인쇄 프로토콜로 변환하고(단계 S8-8-3), 작업 데이터를 이전에 획득한 인쇄 장치 IP 어드레스에 송신한다(단계 S8-9).

<107> 결과적으로, 작업 커맨드, 작업 속성 및 작업 데이터를 수신한 인쇄 장치는 인쇄 프로토콜 프로세서를 이용하여 작업 커맨드 및 작업 속성을 해석하고, 인쇄 작업을 대응하는 인쇄 컨트롤러에 송신함으로써, 인쇄 프로세스를 실행한다.

<108> 도 2로부터 알 수 있는 바와 같이, 프록시 서버(300)는 전원 OFF 지시를 수신하지 않는 한은, 상기한 프로세스, 즉 단계 S2-2 내지 S2-4를 반복하여, 네트워크 인쇄 장치의 동작 상태를 주기적으로 업데이트하고, 그 업데이트한 정보에 따라 프로토콜 변환 프로세스를 실행한다.

<109> 도 2에서 프록시 서버(300)의 전원 OFF 지시가 수신된 것으로 결정된 경우, 플로우는 단계 S2-5에서 단계 S2-6으로 진행한다. 이 경우에, 프로토콜 변환 프로세스를 중지하기 위해, 프록시 서버(300)의 프로토콜 변환 프

로 세서(16)는 메모리 컨트롤러(15)를 통해 모든 관리 테이블을 관리하고, 관리 테이블에 기록된 모든 인쇄 장치에 대한 Bye 패킷(UPnP 네트워크 로그아웃 메시지)을 UPnP 프로토콜 프로세서(14)를 통해 발행함으로써, 이들 인쇄 장치가 네트워크 상에서 그 서비스를 중지한다는 것을 네트워크 상의 다른 디바이스(클라이언트)에 통지한다.

<110> 전술한 바와 같이, 본 실시예에 따르면, 프록시 서버(300)는 SNMP 프로토콜을 이용하여 네트워크 상의 비-UPnP 인쇄 장치(프린터)를 검출하고 등록한다. 그 비-UPnP 프린터 대신에, 프록시 서버(300)는 UPnP 인쇄 장치인 것처럼 행동한다. 인쇄 작업의 수신시, 프록시 서버(300)는 인쇄 작업을 지정된 비-UPnP 인쇄 장치에 출력한다. 결과적으로, 비-UPnP 인쇄 장치는 UPnP 네트워크의 멤버로서 기능한다.

<111> 유의할 점은, 본 실시예의 프록시 서버(300)는 비-UPnP 네트워크 인쇄 장치를 대리하는 서버로서 설명되었다는 점이다. 그러나, 대리하는 대상은 인쇄 장치로 제한되지 않고 다른 디바이스일 수도 있다. 프린터 이외의 네트워크 호환성 디바이스로서는, 통신 기능을 통해 프록시 서버와 속성 정보 및 작업을 교환할 수 있는 한은, 하드 디스크 등과 같은 저장 디바이스, 스캐너, 복사기, 이들 기능들을 함께 가지고 있는 디바이스가 이용될 수 있다. 이 경우에, 프록시 서버와 네트워크 호환성 디바이스간의 통신 프로토콜로서, 표준화되거나 범용 프로토콜 또는 벤더에 고유한 프로토콜이 유사하게 이용될 수 있다.

<112> 본 실시예에서, 네트워크 호환성 디바이스를 일례로 이용하는 실시예가 설명되었다. 디바이스와 프록시 서버간의 통신은 USB, IEEE1394, 병렬 등과 같은 로컬 접속에 의해 구현될 수도 있고, 네트워크는 유선 또는 무선 통신으로 제한되지 않는다.

<113> 본 실시예에서, 프록시 서버는 네트워크 상에서 독립적으로 존재한다. 그러나, 프록시 서버 기능은 네트워크 호환성 디바이스에 물리적으로 또는 논리적으로 통합될 수도 있다.

<114> 또한, 본 실시예의 프록시 서버가 제공하는 프로토콜 변환의 조합으로서, 마이크로소프트사가 주체가 되어 규정한 범용 플러그 앤 플레이, 네트워크 호환성 인쇄 장치의 SNMP, 및 인쇄 프로토콜이 예시되었다. 그러나, 본 발명은 애플 컴퓨터가 제안하는 Rendezvous, JBMIA가 제안하는 BMLinkS 등과 같은 프로토콜에 적용될 수 있다. 또한, 본 발명은 디바이스 검색 및 제어 프로세스를 통합한 프로토콜뿐만 아니라, 서비스 로케이션 프로토콜(SLP), 멀티캐스트 DNS 서비스 디스커버리 등의 디바이스가 제공하는 서비스를 검색하는데 이용되는 프로토콜, 및 (예를 들면, 웹 서비스와 같이) XML/SOAP에 기반한 원격 절차 호출(RPC) 포맷으로 된 디바이스 제어를 종래의 제어 프로토콜로 변환하는 것에 적용될 수 있다.

<115> 본 실시예에서, 프록시 서버와의 정보 프로토콜로서, HTTP/TCP/UDP/IP 프로토콜이 이용된다. 그러나, 본 발명은 어떠한 트랜스포트 수단에도 의존하지 않으며, 양방향 통신이 가능한 한은 다른 범용 프로토콜 또는 고유 프로토콜도 이용 가능하다.

<116> 전술한 바와 같이, 네트워크 호환성 플러그 앤 플레이를 지원하지 않는 인쇄 장치에 프록시 응답 장치를 제공하고, 인쇄 장치가 지원하는 네트워크 호환성 플러그 앤 플레이 이외의 프로토콜에 의해 획득한 디바이스 ID의 모델명이 PDL명 속성을 가지고 있지 않은 경우, 네트워크 호환성 플러그 앤 플레이의 주변 디바이스 검색 프로토콜의 응답은 PDL명을 모델명에 첨부함으로써 회신되므로, 정보 처리 장치에 적절한 프린터 드라이버를 자동으로 설치한다.

<117> 또한, 네트워크 호환성 플러그 앤 플레이를 지원하지 않는 인쇄 장치에 프록시 응답 장치를 제공하고, 인쇄 장치가 지원하는 네트워크 호환성 플러그 앤 플레이 이외의 프로토콜에 의해 획득한 디바이스 ID가 복수의 PDL을 지원하는 경우, 네트워크 호환성 플러그 앤 플레이의 주변 디바이스 검색 프로토콜로의 응답은 디바이스 ID의 모델명에 각각의 PDL명을 첨부하여 얻은 디바이스 ID를 포함하도록 이루어지므로, 인쇄 장치가 PDL의 수에 대응하는 논리적인 인쇄 장치로서 인식된다. 그러므로, 정보 처리 장치는 다른 논리적 실체를 인식할 수 있다.

<118> 전술한 바와 같이, 본 발명에 따르면, 본 발명의 네트워크 디바이스 관리 장치는 네트워크 호환성 플러그 앤 플레이를 지원하지 않는, 네트워크 상에 존재하는 디바이스를 대리하여, 그 디바이스를 가상적으로 네트워크 호환성 플러그 앤 플레이를 지원하는 디바이스로서 네트워크 상의 다른 디바이스들에 인식시켜, 그 디바이스를 그 기능을 가진 디바이스로서 기능시킬 수 있다.

<119> 상기 실시예에 기재된 바와 같이, 본 실시예의 주요 부분들은 프록시 서버(300)의 처리에 있다. 프록시 서버(300)는 전술한 바와 같이, 하드 디스크 등과 같은 기억 디바이스를 필요로 하지만, 퍼스널 컴퓨터 등과 같은 범용 정보 처리 장치 상에서 실행되는 서버 어플리케이션 프로그램에 의해 구현된다. 그러므로, 본 발명은 그 범주내의 컴퓨터 프로그램을 포함한다. 통상, 컴퓨터 프로그램은 CD-ROM과 같은 컴퓨터 판독가능한 기억 매체

를 컴퓨터에 설정함으로써 시스템에 복사되거나 설치될 수 있으므로, 그러한 컴퓨터 관독가능한 기억 매체도 본 발명의 범주에 포함된다.

<120> 본 발명의 다수의 명백하게 다른 실시예들이 그 사상 및 범주에서 벗어나지 않고서도 만들어질 수 있으므로, 본 발명은 청구의 범위에 정의된 것을 제외하고 특정 실시예로 제한되지 않는다는 것은 자명하다.

### 도면의 간단한 설명

<26> 명세서에 포함되고 그 일부를 구성하는 첨부 도면은 본 발명의 실시예를 예시하고, 상세한 설명과 함께 본 발명의 원리를 설명하는 기능을 한다.

<27> 도 1은 본 발명의 실시예에 따른 네트워크 시스템을 형성하는 각 디바이스의 소프트웨어 구성을 도시한 블록도이다.

<28> 도 2는 본 발명의 실시예에서 프록시 서버의 처리 시퀀스를 도시하는 플로우차트이다.

<29> 도 3은 UPnP 호환성 인쇄 장치 검색 프로세스를 도시하는 플로우차트이다.

<30> 도 4는 Probe 패킷의 일례를 도시하고 있다.

<31> 도 5는 Probe 패킷의 응답 패킷인 ProbeMatch 패킷의 일례를 도시하고 있다.

<32> 도 6은 인쇄 장치 정보 획득 프로세스를 도시하는 플로우차트이다.

<33> 도 7은 관리 테이블의 포맷을 도시하고 있다.

<34> 도 8은 프로토콜 변환 프로세스를 도시하는 플로우차트이다.

<35> 도 9는 PrinterDescription의 Request 패킷의 일례를 도시하고 있다.

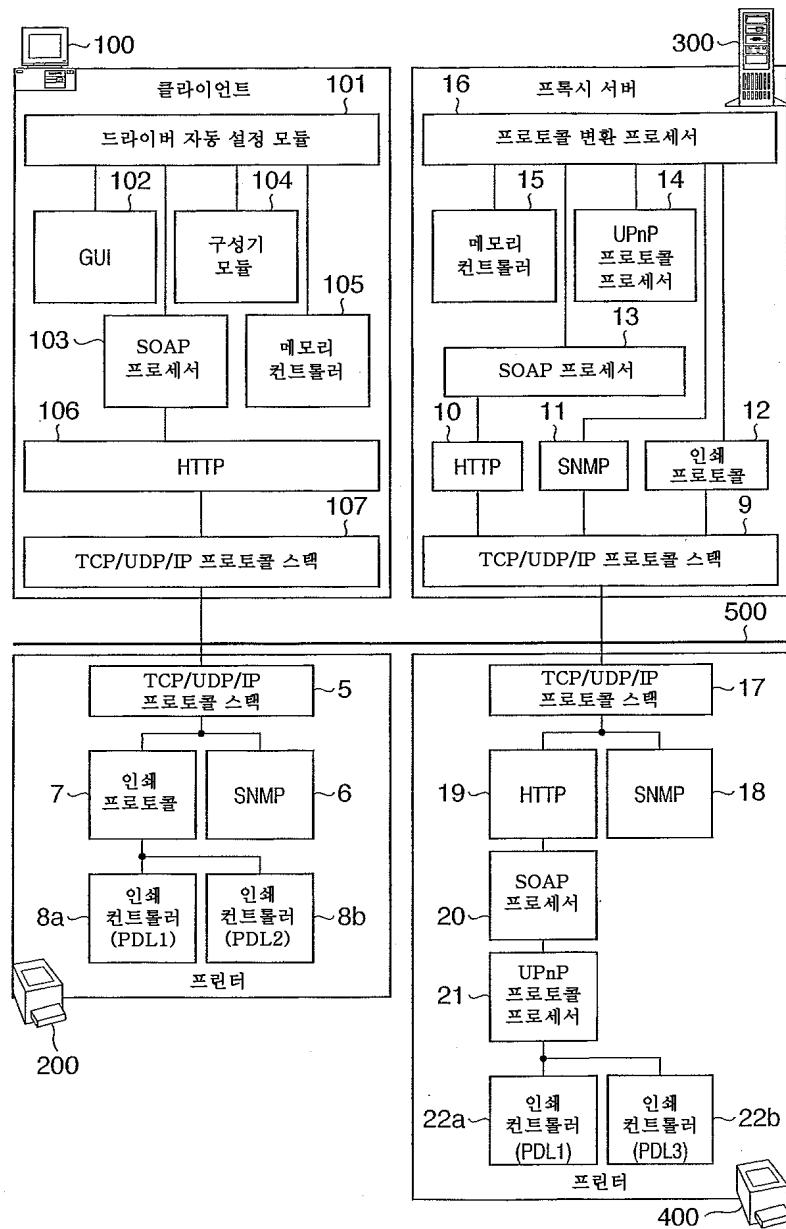
<36> 도 10은 PrinterDescription의 Response 패킷의 일례를 도시하고 있다.

<37> 도 11은 본 발명의 실시예에 따른 클라이언트와 프록시 서버의 하드웨어 구성을 도시하는 블록도이다.

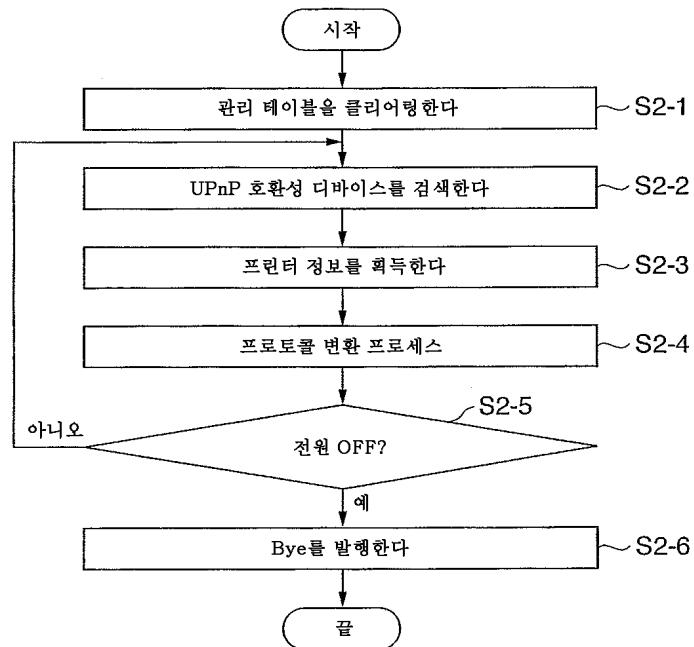
<38> 도 12는 본 발명의 실시예에 따른 프린터(200, 400)의 하드웨어 구성을 도시하는 블록도이다.

## 도면

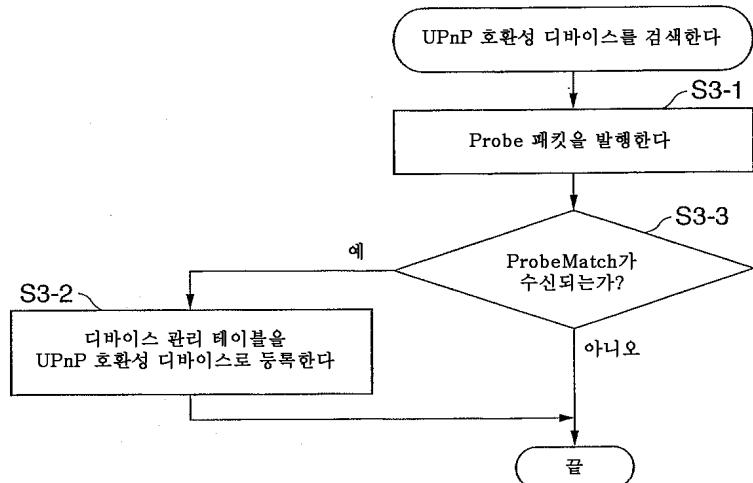
## 도면1



### 도면2



### 도면3



도면4

```
<Envelope>
  <Header>
    <Action>
      http://xxx/Probe
    </Action>
    <MessageID>
      uuid:0a6dc791-2be6-4991-9af1-454778a1917e
    <MessageID>
  </Header>
  <Body>
    <Probe>
      <Type>PrintBasic</Type>
    </Probe>
  </Body>
</Envelope>
```

## 도면5

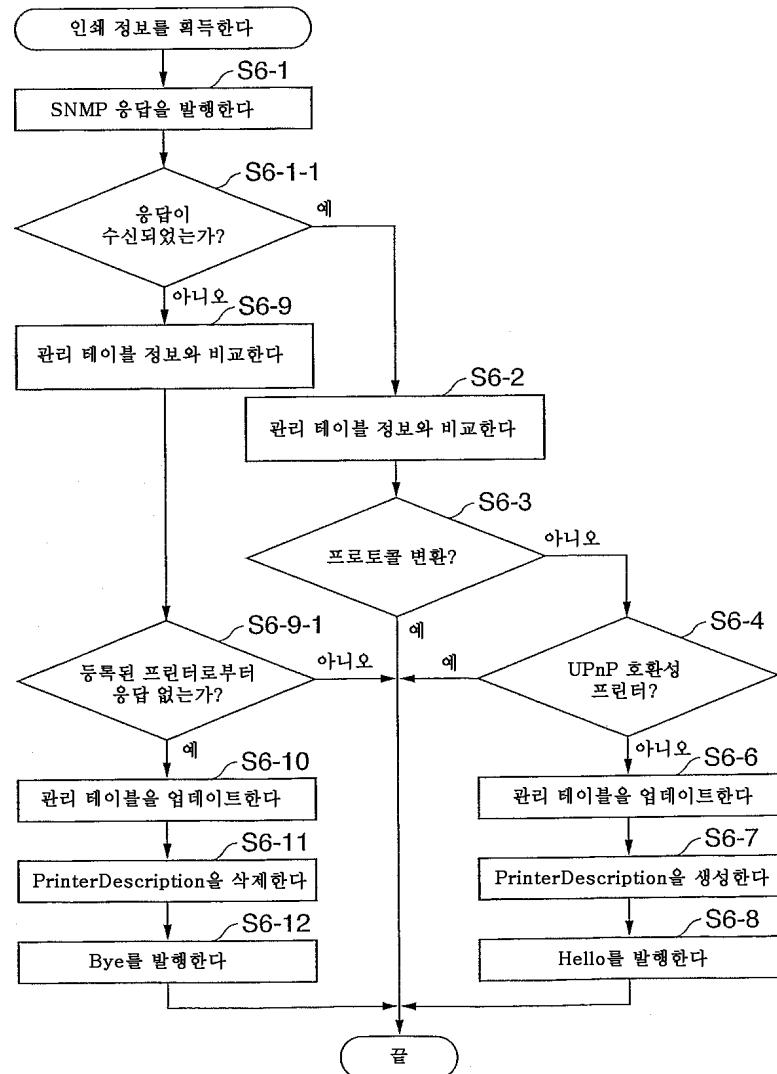
```
<Envelope>
<Header>
<Action>
  http://xxx/ProbeMatch
</Action>
<MessageID>
  uuid:e32e6863-ea5e-4ee4-997e-69539 d1ft2cf
<MessageID>
  uuid:0a6dc791-2be6-4991-9af1-454778a1917e
<RelatesTo>
  uuid:0a6dc791-2be6-4991-9af1-454778a1917e
</RelatesTo>
</Header>
<Body>
<ProbeMatch>
  <EndpointReference>
    <Address>
      uuid:98190dc2-0890-4ef8-ac9a-5940995e611a
    </Address>
    <Types>PrintBasicPrintAdvanced</Types>
  </EndpointReference>
</ProbeMatch>
</Body>
</Envelope>

<Envelope>
<Header>
<Action>
  http://xxx/ProbeMatch
</Action>
<MessageID>
  uuid:e32e6863-ea5e-4ee4-997e-69539 d1ft2cf
<MessageID>
  uuid:0a6dc791-2be6-4991-9af1-454778a1917e
<RelatesTo>
  uuid:0a6dc791-2be6-4991-9af1-454778a1917e
</RelatesTo>
</Header>
<Body>
<ProbeMatch>
  <EndpointReference>
    <Address>
      uuid:98190dc2-0890-4ef8-ac9a-5940995e611a
    </Address>
    <Types>PrintBasicPrintAdvanced</Types>
  </EndpointReference>
</ProbeMatch>
</Body>
</Envelope>
```

22a에 대한  
ProbeMatch

22b에 대한  
ProbeMatch

## 도면6



## 도면7

```

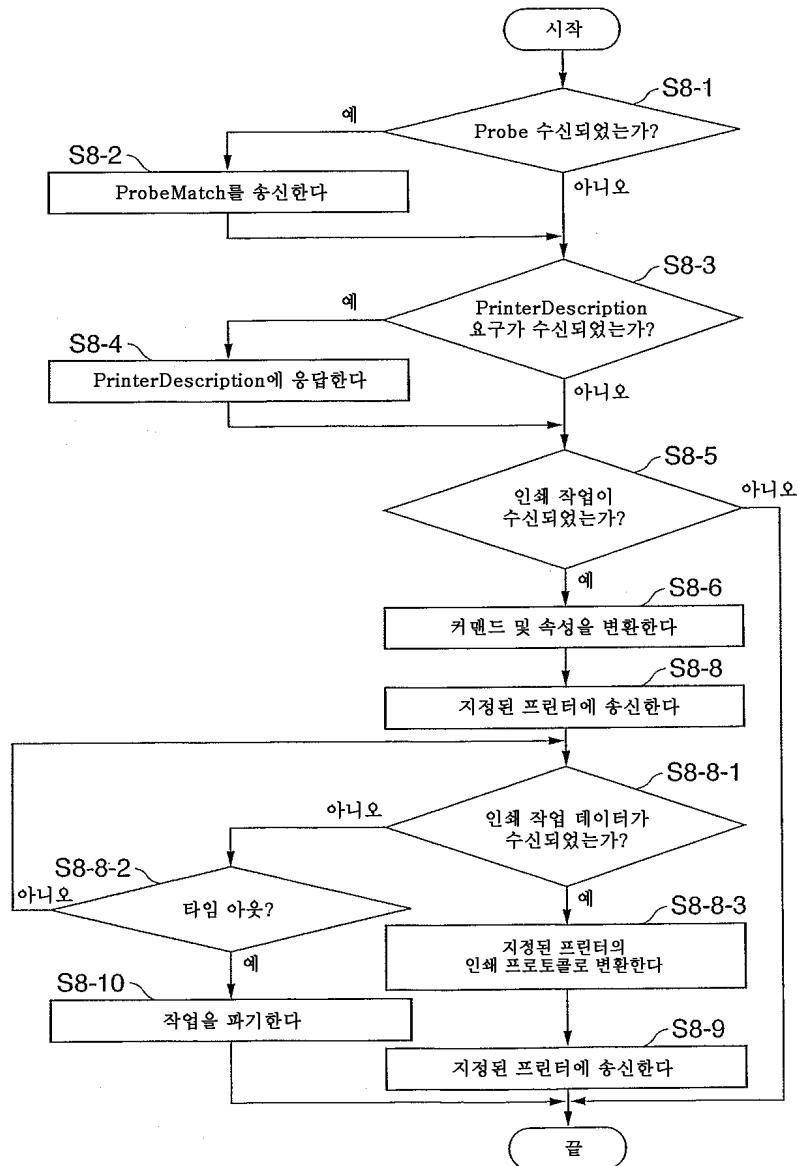
<PrinterTable>
  <PrinterDescription>
    <DeviceId>
      MANUFACTURER:XXXX Inc. ;
      COMMAND SET:PDL1 ;
      MODEL LaserBeamPrinter777 PDL1;
    </DeviceId> ← DeviceID
    <PrinterLocation>Bldg Kosugi, Fl3, Rm3</PrinterLocation>
    <PrinterMakeAndModel> LaserBeamPrinter777 </PrinterMakeAndModel>
    <PrinterName>200 Printer </PrinterName>
    <IPAddress>123.123.123.123</IPAddress>
    <MACAddress>00-02-2D-69-43-89</MACAddress>
    <SupportedPDL>PDL1</SupportedPDL>
    <SupportedPrintProtocol>LPR</SupportedPrintProtocol>
    <Address>
      uuid98190dc2-0890-4ef8-ac9a-5940995e611c
    </Address>
    <Types>PrintBasic</Types>
  </PrinterDescription>
  <PrinterDescription>
    <DeviceId>
      MANUFACTURER:XXXX Inc. ;
      COMMAND SET:PDL2 ;
      MODEL LaserBeamPrinter777 PDL2;
    </DeviceId> ← DeviceID
    <PrinterLocation>Bldg Kosugi, Fl3, Rm3</PrinterLocation>
    <PrinterMakeAndModel> LaserBeamPrinter777 </PrinterMakeAndModel>
    <PrinterName>200 Printer </PrinterName>
    <IPAddress>123.123.123.123</IPAddress>
    <MACAddress>00-02-2D-69-43-89</MACAddress>
    <SupportedPDL>PDL2</SupportedPDL>
    <SupportedPrintProtocol>LPR</SupportedPrintProtocol>
    <Address>
      uuid98190dc2-0890-4ef8-ac9a-5940995e611c
    </Address>
    <Types>PrintBasic</Types>
  </PrinterDescription>
  ...
<PrinterTable>

```

8a에 대한  
디바이스  
관리 테이블

8b에 대한  
디바이스  
관리 테이블

## 도면8



## 도면9

```

<Envelope>
  <Header>
    <MessageId>uuid:09111432-123b-2392-p423-9add28pon3jb</MessageId>
    <To>uuid:98190dc2-0890-4ef8-ac9a-5940995e611c</To> ← Address를 지정한다
    <Types>PrintBasic</Types>
  </Header>
  <Body>
    <GetPrinterElementsRequest Type = PrinterDescription>
      </GetPrinterElementsRequest>
    </Body>
</Envelope>
  
```

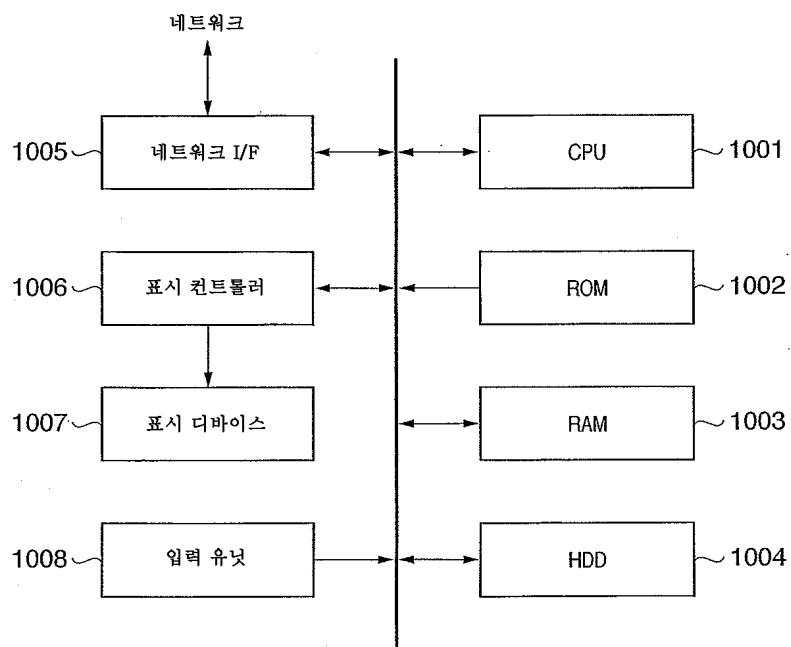
## 도면10

```

<Envelope>
  <Header>
    <MessageId>uuid:09111432-123b-2392-p423-9add28pon3jb</MessageId>
    <ResponseTo>uuid:09111432-123b-5555-c787-2san15pip5ec</ResponseTo>
  </Header>
  <Body>
    <GetPrinterElementsResponse Type = PrinterDescription>
      <DeviceId>
        MANUFACTURER:XXXX inc. ;
        COMMAND SET:PDL1;
        MODEL:LaserBeamPrinter777 PDL1
      </DeviceId>
      <PrinterLocation>Bldg Kosugi, Flr3, Rm3</PrinterLocation>
      <PrinterMakeAndModel>LaserBeamPrinter777</PrinterMakeAndModel>
      <PrinterName>200 Printer</PrinterName>
    </GetPrinterElementsResponse>
  </Body>
</Envelope>

```

## 도면11



도면12

