

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4416563号
(P4416563)

(45) 発行日 平成22年2月17日 (2010.2.17)

(24) 登録日 平成21年12月4日 (2009.12.4)

(51) Int.Cl.

F I

G 0 6 F 13/00 (2006.01)

G 0 6 F 13/00 3 5 7 A

G 0 6 F 13/00 3 5 3 B

請求項の数 10 (全 19 頁)

(21) 出願番号 特願2004-123443 (P2004-123443)
 (22) 出願日 平成16年4月19日 (2004.4.19)
 (65) 公開番号 特開2005-309617 (P2005-309617A)
 (43) 公開日 平成17年11月4日 (2005.11.4)
 審査請求日 平成19年4月18日 (2007.4.18)

(73) 特許権者 000001007
 キヤノン株式会社
 東京都大田区下丸子3丁目30番2号
 (74) 代理人 100076428
 弁理士 大塚 康德
 (74) 代理人 100112508
 弁理士 高柳 司郎
 (74) 代理人 100115071
 弁理士 大塚 康弘
 (74) 代理人 100116894
 弁理士 木村 秀二
 (72) 発明者 大村 宏
 東京都大田区下丸子3丁目30番2号 キ
 ヤノン株式会社内

最終頁に続く

(54) 【発明の名称】 ネットワークデバイス管理装置及びその制御方法、並びに、コンピュータプログラム及びコンピュータ可読記憶媒体

(57) 【特許請求の範囲】

【請求項 1】

ネットワークに接続された複数の機能を備えたネットワークデバイスであって、ネットワーク対応型プラグアンドプレイをサポートしていないネットワークデバイスを管理し、クライアント装置と通信するネットワークデバイス管理装置であって、

ネットワーク上のネットワークデバイスそれぞれから受信した、それぞれの製品種別を識別するための情報を含む受信情報にしたがって、ネットワーク対応型プラグアンドプレイをサポートしていないネットワークデバイスを認識する認識手段と、

前記ネットワーク対応型プラグアンドプレイをサポートしていないネットワークデバイスを前記認識手段が認識した場合、前記ネットワークデバイスが有する複数の機能を制御するための複数のデバイスドライバをクライアント装置においてネットワーク対応型プラグアンドプレイに従ってインストールすることを可能にするため、前記認識手段が、前記ネットワーク対応型プラグアンドプレイをサポートしていないネットワークデバイスとして認識したネットワークデバイスからの前記受信情報の記述にしたがって、前記ネットワークデバイスのモデルを示す情報、メーカー名を示す情報、及び、前記ネットワークデバイスが有する機能を示す情報を少なくとも含むデバイスIDを、前記ネットワークデバイスが有する機能を示す情報ごとに複数生成する生成手段と、

前記生成手段が生成した複数のデバイスIDを前記クライアント装置に対して送信する送信手段と

を備えることを特徴とするネットワークデバイス管理装置。

10

20

【請求項 2】

更に、ネットワークを介して、ネットワーク対応型プラグアンドプレイデバイス宛にジョブ情報を受信した場合、当該ジョブ情報中に記述された識別情報に基づき、ネットワークデバイス管理装置が有する記憶手段から該当するネットワークデバイスのアドレスとプロトコル情報を取得すると共に、前記ジョブ情報を取得したプロトコルに変換し、取得したアドレスに向けて送信する制御手段と

を備えることを特徴とする請求項 1 に記載のネットワークデバイス管理装置。

【請求項 3】

前記ネットワークデバイスが有する機能の種類は、前記ネットワークデバイスが処理可能なページ記述言語の種類であることを特徴とする請求項 1 に記載のネットワークデバイス管理装置。

10

【請求項 4】

前記認識手段は、

ネットワーク対応型プラグアンドプレイの非サポートのネットワークデバイスを探索する探索手段と、

該探索手段で得られたネットワークデバイスのネットワークアドレス、及び、探索して得られたネットワークデバイスとの通信の際に用いるプロトコルを特定する情報をネットワークデバイス管理装置が有する記憶手段に登録する登録手段とを有し、

前記応答手段は、前記記憶手段に登録されたネットワークデバイスに代わって応答する際のメッセージを作成する作成手段と有する

20

ことを特徴とする請求項 1 に記載のネットワークデバイス管理装置。

【請求項 5】

前記探索手段は、SNMP プロトコルでの探索して検出されたネットワークデバイス群から、UPnP ネットワークプロトコルでの探索結果で検出されたネットワークデバイスを除外することで残ったネットワークデバイス群を、ネットワーク対応型プラグアンドプレイの非サポートのネットワークデバイス群とすることを特徴とする請求項 4 に記載のネットワークデバイス管理装置。

【請求項 6】

前記ネットワークデバイスは、ネットワークプリンタとすることを特徴とする請求項 1 乃至 5 のいずれか 1 項に記載のネットワークデバイス管理装置。

30

【請求項 7】

前記ネットワークデバイスが複数のプリンタ言語をサポートしている場合、前記応答手段は、個々のプリンタ言語毎に独立した論理的な仮想ネットワーク対応型プラグアンドプレイプリンタとして応答することを特徴とする請求項 6 に記載のネットワークデバイス管理装置。

【請求項 8】

ネットワークに接続された複数の機能を備えたネットワークデバイスであって、ネットワーク対応型プラグアンドプレイをサポートしていないネットワークデバイスを管理し、クライアント装置と通信するネットワークデバイス管理装置の制御方法であって、

ネットワーク上のネットワークデバイスそれぞれから受信した、それぞれの製品種別を識別するための情報を含む受信情報にしたがって、ネットワーク対応型プラグアンドプレイをサポートしていないネットワークデバイスを認識する認識工程と、

40

前記ネットワーク対応型プラグアンドプレイをサポートしていないネットワークデバイスを前記認識工程で認識した場合、前記ネットワークデバイスが有する複数の機能を制御するための複数のデバイスドライバをクライアント装置においてネットワーク対応型プラグアンドプレイに従ってインストールすることを可能にするため、前記認識工程で、前記ネットワーク対応型プラグアンドプレイをサポートしていないネットワークデバイスとして認識されたネットワークデバイスからの前記受信情報の記述にしたがって、前記ネットワークデバイスのモデルを示す情報、メーカー名を示す情報、及び、前記ネットワークデバイスが有する機能を示す情報を少なくとも含むデバイス ID を、前記ネットワークデバ

50

スが有する機能を示す情報ごとに複数生成する生成工程と、

前記生成工程で生成した複数のデバイスIDを前記クライアント装置に対して送信する送信工程と

を備えることを特徴とするネットワークデバイス管理装置の制御方法。

【請求項9】

コンピュータが読み込み実行することで、前記コンピュータを、

ネットワークに接続された複数の機能を備えたネットワークデバイスであって、ネットワーク対応型プラグアンドプレイをサポートしていないネットワークデバイスを管理し、クライアント装置と通信するネットワークデバイス管理装置

として機能させるコンピュータプログラムであって、

ネットワーク上のネットワークデバイスそれぞれから受信した、それぞれの製品種別を識別するための情報を含む受信情報にしたがって、ネットワーク対応型プラグアンドプレイをサポートしていないネットワークデバイスを認識する認識手段と、

前記ネットワーク対応型プラグアンドプレイをサポートしていないネットワークデバイスを前記認識手段が認識した場合、前記ネットワークデバイスが有する複数の機能を制御するための複数のデバイスドライバをクライアント装置においてネットワーク対応型プラグアンドプレイに従ってインストールすることを可能にするため、前記認識手段が、前記ネットワーク対応型プラグアンドプレイをサポートしていないネットワークデバイスとして認識したネットワークデバイスからの前記受信情報の記述にしたがって、前記ネットワークデバイスのモデルを示す情報、メーカー名を示す情報、及び、前記ネットワークデバイスが有する機能を示す情報を少なくとも含むデバイスIDを、前記ネットワークデバイスが有する機能を示す情報ごとに複数生成する生成手段と、

前記生成手段が生成した複数のデバイスIDを前記クライアント装置に対して送信する送信手段と

して機能させることを特徴とするコンピュータプログラム。

【請求項10】

請求項9に記載のコンピュータプログラムを格納したことを特徴とするコンピュータ可読記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ネットワークデバイスを管理する技術に関するものである。

【背景技術】

【0002】

特定の情報処理装置や、情報処理装置内で動作するオペレーションシステム（以下OSと略す）、またそのOS上で動作するアプリケーションに依存しないように、印刷装置はPDL (Page Description Language)などの独自の描画言語体系であるコマンドインターフェースを有している。

【0003】

その描画言語体系は個々の印刷装置に依存する。よってその依存性をカプセル化するために、OSは、標準的な描画インターフェースを入力とし、その印刷装置依存の描画言語体系のコマンドに変換し出力するプリンタドライバというモジュールを定義している。このプリンタドライバは、一般には、印刷装置の製造元やOS開発者などによって作成され、情報処理装置内に格納される。

【0004】

情報処理装置内に格納されるプリンタドライバは多数存在する。そのため情報処理装置に存在する全てのプリンタドライバをユーザに示すのは混乱の元である。つまり、ユーザの使用する印刷装置のみだけ効率的に結びつけ、ユーザが使用可能な状態にしたい。よってOSの初期状態ではプリンタドライバは情報処理装置内に格納されるだけであり、ユーザは特定の印刷装置にプリンタドライバを結びつける作業であるOSへのプリンタドライ

10

20

30

40

50

バのインストールを行う必要がある。

【 0 0 0 5 】

このインストール作業は従来ユーザが手動で行っていた。しかしユーザは煩雑な印刷装置ドライバのインストール作業を行う必要があり、また異なる描画言語体系のコマンドを、その描画体系をサポートしていない印刷装置に送った場合、印刷エラーとなるため、該当の印刷装置に対応したプリンタドライバを正しくインストールする必要があり、初心者には難しい作業であった。

【 0 0 0 6 】

この煩雑なインストール作業を軽減するために、米国セントロニクス社使用のパラレルインタフェースやU S B (Universal Serial Bus) インタフェースなどの通信媒介を用いて情報処理装置と印刷装置を接続するだけで、ユーザが操作を行わなくても自動的に情報処理装置と印刷装置で双方向通信を行い、その印刷装置に該当するドライバを情報処理装置にインストールする機能をO Sに導入している。この機能をプラグアンドプレイと呼ぶ(特許文献1)。

【 0 0 0 7 】

上記プラグアンドプレイは直接的な一対一の接続であった。近年、ネットワークのインフラが整備されるに従い、ネットワーク対応型の印刷装置、スキャナ、複写機等のネットワーク周辺デバイスが急速に普及しつつある。ネットワーク上の各種サービスを提供するデバイスを、ネットワークを介して探索する技術が開発されつつある。例えば、このような技術には米国マイクロソフト社の提唱するUn i v e r s a l P l u g a n d P l a y (以下、UPnPと略す)が存在する(特許文献2)。

【 0 0 0 8 】

これらのネットワーク対応型のネットワークデバイスとしての印刷装置を例にとると、事前に使用するクライアント、例えばパーソナルコンピュータ上に、そのクライアント上で稼動するオペレーティングシステム、例えばWindows(登録商標)によりデータベース上に登録管理された該当ドライバをインストールするか、あるいは、印刷装置ベンダからフレキシブルディスク、CD-ROM等の記録媒体を介して提供されるドライバソフトウェア(ドライバ)をインストールした上で、印刷装置に割り当てられたIPアドレス、印刷装置ポート、使用するプリントプロトコル、デバイスドライバ等の印刷装置を動作させるための情報を把握する必要がある。

【 0 0 0 9 】

また、その印刷装置を他のネットワークへの移動したり、そのネットワークへの新規加入などにより、ネットワークシステムに接続されている装置の構成は刻々と変化していることを鑑み、ネットワーク上に存在する印刷装置を管理するためには、該印刷装置の情報を把握して管理する技術が開発されつつある。

【 0 0 1 0 】

しかし、UPnPなどの既存の管理システムは、印刷装置とコンピュータを接続するための技術仕様であるが、機器同士が通信するためのプロトコルとデータフォーマットについて規定しているだけである。

【 0 0 1 1 】

従って、UPnPなどの管理システムが管理している情報を用いたとしても、デバイスドライバなど、印刷装置を制御する制御プログラムをコンピュータにインストールするには、煩雑な設定が必要であるという問題は依然として存在する。

【 0 0 1 2 】

さらに、複数のネットワーク対応の印刷装置を管理する管理方式がネットワーク上に並存している場合、ある特定の管理方式に対応した管理システムからは、該特定の管理方式には対応していないネットワーク上の他の印刷装置を認識することができないという問題もある。

【 0 0 1 3 】

またユーザや各国市場の要求を満たすために、多種多様な印刷装置のための描画言語体

10

20

30

40

50

系が作られ、市場に送り出されている。それぞれの描画言語体系のために印刷装置を一から開発するのは費用がかさむ。このコストを軽減するために各描画言語体系に依存する部分を拡張ボードやソフトウェアとして分離し、ユーザの要求や各国市場要求に応じて、印刷装置に、その拡張ボードやソフトウェアを搭載し直すことが可能な印刷装置が出てきた。

【 0 0 1 4 】

加えて、ユーザが好む描画言語体系が複数個存在する環境のために、一つの印刷装置で複数個の描画言語体系をサポートする印刷装置も登場している。

【 0 0 1 5 】

上記、描画言語体系が拡張ボードやソフトウェアによって変更可能な印刷装置や、複数個の描画言語体系がサポートする印刷装置は、プラグアンドプレイ搭載時には想定されていないかった。

【 0 0 1 6 】

またプラグアンドプレイは、ユーザの煩雑なインストール作業を軽減するために導入されたので、発見された一つの印刷装置に対して複数の描画言語体系からのPDLの選択などの、印刷装置に詳しくないユーザが混乱する選択作業を不要するユーザインターフェイスが望まれている。

【 0 0 1 7 】

上記2つの理由により、あるOSで搭載されているプラグアンドプレイでは、印刷装置から情報処理装置に受け渡される印刷装置の識別データ (DeviceID) のうち、描画言語体系を示す情報を無視して、製造者名と印刷装置名でのみ、その印刷装置に該当するプリンタドライバを情報処理装置内から検索し、最初に発見したものをインストールする動作を行う。

【特許文献1】特開2003-216378公報

【特許文献1】特開2003-6133公報

【発明の開示】

【発明が解決しようとする課題】

【 0 0 1 8 】

以上、プラグアンドプレイについて説明したが、これには以下のような欠点がある。

【 0 0 1 9 】

第1には、例えば、印刷装置に拡張ボードを搭載することによりサポート描画体系を切り替え可能な印刷装置に、PDL1、PDL2という2つの描画言語体系 (プリンタ言語インタプリタ) が搭載可能だが、PDL1のみが実際に印刷装置に搭載されているものとする。また、そのプリンタドライバが既に情報処理装置に格納されているものとする。この場合プラグアンドプレイは、印刷装置から情報処理装置に受け渡される印刷装置の識別データのうち、描画言語体系を示す情報を無視して、製造者名と印刷装置名でのみ、その印刷装置に該当するプリンタドライバを情報処理装置から検索する。もしPDL2のプリンタドライバが先にプラグアンドプレイによって発見された場合、この印刷装置のためにPDL2がOSにインストールされてしまう。当然、PDL1用の描画言語系を処理する拡張ボードは、PDL2を認識しないので、印刷エラーとなってしまう。

【 0 0 2 0 】

第2には、複数個の描画言語体系をサポート可能な印刷装置の場合、製造元があまりお奨めできないエミュレーションのための描画体系用のプリンタドライバが先にプラグアンドプレイに発見されてしまい、他の効率的な描画体系があまり利用されないということが存在する。

【 0 0 2 1 】

本発明は、ネットワーク対応型プラグアンドプレイをサポートしないデバイスを、ネットワーク対応型プラグアンドプレイに対応するデバイスとしてネットワーク上の他のデバイスに認識させ、機能させる技術を提供しようとするものである。

【課題を解決するための手段】

【 0 0 2 2 】

この課題を解決するため、本発明の、例えばネットワークデバイス管理装置は以下の構成を備える。すなわち、

ネットワークに接続された複数の機能を備えたネットワークデバイスであって、ネットワーク対応型プラグアンドプレイをサポートしていないネットワークデバイスを管理し、クライアント装置と通信するネットワークデバイス管理装置であって、

ネットワーク上のネットワークデバイスそれぞれから受信した、それぞれの製品種別を識別するための情報を含む受信情報にしたがって、ネットワーク対応型プラグアンドプレイをサポートしていないネットワークデバイスを認識する認識手段と、

前記ネットワーク対応型プラグアンドプレイをサポートしていないネットワークデバイスを前記認識手段が認識した場合、前記ネットワークデバイスが有する複数の機能を制御するための複数のデバイスドライバをクライアント装置においてネットワーク対応型プラグアンドプレイに従ってインストールすることを可能にするため、前記認識手段が、前記ネットワーク対応型プラグアンドプレイをサポートしていないネットワークデバイスとして認識したネットワークデバイスからの前記受信情報の記述にしたがって、前記ネットワークデバイスのモデルを示す情報、メーカー名を示す情報、及び、前記ネットワークデバイスが有する機能を示す情報を少なくとも含むデバイスIDを、前記ネットワークデバイスが有する機能を示す情報ごとに複数生成する生成手段と、

前記生成手段が生成した複数のデバイスIDを前記クライアント装置に対して送信する送信手段とを備える。

【 発明の効果 】

【 0 0 2 3 】

本発明によれば、本発明のネットワークデバイス管理装置が、ネットワーク上に存在するネットワーク対応型プラグアンドプレイをサポートしないデバイスを代理し、そのデバイスを仮想的にネットワーク対応型プラグアンドプレイに対応するデバイスとしてネットワーク上の他のデバイスに認識させ、機能させることが可能になる。

【 発明を実施するための最良の形態 】

【 0 0 2 4 】

以下、添付図面に従って本発明に係る実施形態を詳細に説明する。

【 0 0 2 5 】

本発明に係るサービス提供システムの一実施形態としてのプロトコル変換システムについて説明する。

【 0 0 2 6 】

図1は本発明の一実施形態としてのプリントシステムの構成をソフトウェア面からみたブロック図である。

【 0 0 2 7 】

クライアント100には、不図示の米国マイクロソフト社が提供するOSであるwindows(登録商標)や米国Apple社が提供するMac OS(登録商標)等の汎用オペレーティングシステム、およびその上で実行可能な汎用アプリケーションがインストールされている。

【 0 0 2 8 】

上記汎用オペレーティングシステムは、通信機能としてTCP/UDP/IPプロトコルスタック107を備え、そのプロトコルスタック上にHTTP106を備え、HTTPリクエストの解析、およびレスポンス処理を行い、HTTP106上には、Simple Object Access Protocol(SOAP)処理部103を備え、UPnPに関する通信手段を形成している。UPnPのためのドライバ自動ダウンロード・設定モジュール部101は、ネットワーク上のUPnP対応印刷装置および実施形態で述べる代理応答装置と通信を行うために、上記通信手段を用いることになる。

【 0 0 2 9 】

クライアント100にUPnPで発見した印刷装置を表示し、ユーザに選択させるため

10

20

30

40

50

に、ドライバ自動ダウンロード・設定モジュール部 101 は GUI 部 102 を利用する。

【0030】

また Configurator モジュール部 (Configurator) 104 は、ネットワークを介して印刷装置より取得した情報に基づいて、予めハードディスクに格納された中から該当するプリンタドライバを OS に登録 (インストール) する。

【0031】

なお、イントール実行の際には、記憶装置制御部 105 を介してメモリスパースの有無、およびイントール先の管理制御を実行する。記憶装置制御部 105 は、後述するようにハードディスクドライブ (OS、各種アプリケーション、上記ソフトウェアモジュール、並びに各種アプリケーションで作成されたデータファイルが格納される) を制御するものである。

10

【0032】

一方、本実施の形態ではネットワーク対応型印刷装置 200 は、通信機能として TCP / UDP / IP プロトコルスタック 5 を備え、そのプロトコルスタック上に Simple Network Management Protocol (SNMP) 処理部 6 を備え、また、プロトコルスタック 5 上にはプリントプロトコル処理部 7 が実装され、クライアントから発行されるプリント要求を解析し、印刷装置コントローラ 8 に対し、そのプリント要求を送出する機能を備える。なお印刷装置 200 は、複数の PDL に対する印刷装置コントローラ 8 を同時に有することが出来る。また印刷装置コントローラ 8 を脱着可能にし、ユーザの要求に応じて差し替えて、印刷装置 200 がサポートする PDL を変更することも可能である。本実施形態では、PDL 1、PDL 2 の 2 つの PDL のための印刷装置コントローラ 8a、8b が印刷装置 200 内部に存在しているものとする。また、この印刷装置は UPnP プロトコル処理部を備えておらず、該印刷装置単独では、クライアント 100 から発行される UPnP プロトコルを使用したデバイス検索要求、および UPnP 印刷ジョブ要求に対し応じることはできない。

20

【0033】

もうひとつのネットワーク対応型デバイス、本実施形態ではネットワーク対応型印刷装置 400 は、通信機能として TCP / UDP / IP プロトコルスタック 17 を備え、そのプロトコルスタック上に HTTP 19 を備え、HTTP リクエストの解析、およびレスポンス処理を行う。

30

【0034】

プロトコルスタック 17 上には、ネットワーク印刷装置 200 同様、Simple Network Management Protocol (SNMP) 18 処理部を備える。

【0035】

HTTP 19 の上位層には Simple Object Access Protocol (SOAP) 処理部 20 を備え、UPnP プロトコル処理部 21 を備える。ネットワーク対応型印刷装置 400 は UPnP Forum で策定された Print Basic サービスを実装しており、UPnP プロトコル処理部は該サービスで定義されたプリントジョブ、および属性情報を解析し、印刷装置コントローラ 22 に対し、そのプリント要求を送出する機能を備える。印刷装置 400 は、複数の PDL に対する印刷装置コントローラ 22 を同時に有することが出来る。また印刷装置コントローラ 22 を脱着可能にし、ユーザの要求に応じて差し替えて、印刷装置 400 がサポートする PDL を変更することも可能である。本実施形態におけるこの印刷装置 400 は、PDL 1、PDL 3 の 2 つの PDL のための印刷装置コントローラ 22a、22b を搭載しているものとする。

40

【0036】

実施形態における特徴となるネットワークデバイス管理装置として機能するプロキシサーバ (代理サーバ) 300 も同様で、通信機能として TCP / UDP / IP プロトコルスタック 9 を備え、そのプロトコルスタック上に HTTP 10 を備え、HTTP リクエストの解析、およびレスポンス処理を行う。

50

【 0 0 3 7 】

プロトコルスタック 9 上には Simple Network Management Protocol (SNMP) 11 処理部を備え、UPnP プロトコル処理部を備えていないネットワーク対応型印刷装置 200 の検索、および情報の取得を該プロトコルにより実施する。

【 0 0 3 8 】

プロトコルスタック 9 上には Print Protocol 処理部 12 を備える。この Print Protocol 部 12 は、UPnP プロトコル処理部を備えていないネットワーク対応型印刷装置 200 に対し、プリントジョブの発行するものである。

【 0 0 3 9 】

HTTP 10 の上位層には Simple Object Access Protocol (SOAP) 処理部 13 を備え、UPnP プロトコル処理部 14、およびプロトコル変換処理部 16 がそれぞれ、該処理部 13 を介してそれぞれ、クライアント 200、およびプロキシサーバ 300 が複数ネットワーク上に存在する場合、eXtensible Markup Language (XML) で記述されたデータの双方向通信を実現する。

【 0 0 4 0 】

プロトコル変換処理部 16 は、SNMP 処理部 11、SOAP 処理部 13、UPnP 処理部 14、プリントプロトコル処理部 12、および記録装置制御部 15 の上位層にあり、SNMP 処理部 11 を介して取得したネットワーク対応型印刷装置の情報を、UPnP プロトコルで使用される各種 XML ドキュメントを作成した上で、記録装置制御部 15 が制御する記録装置に記録、あるいは UPnP プロトコルから要求があった際に該当する管理テーブルに記録された XML ドキュメントを、記録制御装置部 15 を介して読み出し、UPnP プロトコル処理部 14 に送信するなどの処理を実行する。

【 0 0 4 1 】

またプロトコル変換処理部 16 は、UPnP プロトコルによるプリントジョブのリクエストを受信した場合、SOAP 処理部 14 を介して、ジョブコマンド、ジョブ属性情報を取得し、その内容を出力指定された印刷装置がサポートするプリントプロトコルに変換したのち、プリントプロトコル制御部 12 を介して、指定印刷装置にジョブを送信する。

【 0 0 4 2 】

また、プロトコル変換処理部 16 は、プロキシサーバ 300 が管理する管理テーブルを記録装置制御部 15 を介して、該制御部が制御する記録装置に対し書き込み、読み出し処理を行なう。また同様に、プロトコル変換処理部 16 は、ネットワーク上に存在する他のプロキシサーバが管理する管理テーブルを取得した場合、記録装置制御部 15 を介して、該制御部が制御する記録装置に対し書き込み、読み出し処理を行なう。

【 0 0 4 3 】

以上は、各装置のプリントに関する実装されるコンピュータプログラムの構成であるが、クライアント 100、プロキシサーバ 300 は、通常のパーソナルコンピュータ等の汎用情報処理装置で実現できる。ハードウェアとしての構成は例えば図 11 のようになる。

【 0 0 4 4 】

図示において、1001 は装置全体の制御を司る CPU であり、1002 は BIOS やブートプログラムを記憶する ROM である。1003 は CPU 1001 のワークエリアとして使用される RAM である。1004 はハードディスク装置であって、ここには OS、アプリケーションや各種プログラム、並びに、データファイル等が格納される。1005 はネットワークインタフェースであり、実施形態ではイーサネット(登録商標)カードとするが、無線 LAN カードでも構わない。1006 は表示制御部であり、ビデオメモリ及び CPU 1001 の制御により描画を行うと共にビデオメモリに記憶された画像データをビデオ信号として出力するビデオコントローラが内蔵される。1007 は表示装置であって、液晶表示器や CRT に代表されるものである。1008 はキーボードやマウス等の入力装置である。

10

20

30

40

50

【 0 0 4 5 】

上記構成において、装置の電源がONになると、CPU 1 0 0 1はROM 1 0 0 2のブートプログラムに従って、ハードディスク 1 0 0 4からOSをRAM 1 0 0 2にロードし、各種デバイスドライバをロードすることになる。図 1 1の装置が上記クライアント 1 0 0であるとするなら、図 1に示す構成に該当するプログラムをRAM 1 0 0 2にロードし、実行することになる。これはプロキシサーバ 3 0 0でも同様である。プロキシサーバ 3 0 0の場合には、OSがロードされた後、図 1に示す構成に該当するプログラムをRAM 1 0 0 2にロードし、且つ、後述する処理を行うことになる。

【 0 0 4 6 】

図 1 2は、ネットワークプリンタのブロック構成図である。図示において、2 0 0 1はプリンタ全体の制御を司るCPUであり、2 0 0 2はCPU 2 0 0 1が実行する印刷処理プログラムやフォントデータを格納するROMである。2 0 0 3はCPU 2 0 0 1のワークエリア、受信バッファ、画像描画に使用されるRAMであり、2 0 0 4は各種スイッチやボタン、並びに、メッセージ表示用の液晶表示部で構成される操作パネルである。2 0 0 5はネットワークに接続するためのネットワークインタフェース、2 0 0 6はレンダリング部、2 0 0 7は実際に記録紙に印刷を行うプリンタエンジンである。

10

【 0 0 4 7 】

図 1における印刷装置 2 0 0、4 0 0はハードウェアを巨視的に見た場合の構成は図 1 2のようになる。図 1の印刷装置 2 0 0、4 0 0の構成の違いは、ROM 2 0 0 2に格納されるファームウェアの違いであると言える。場合によっては、各種拡張ボードによる違いであるとも見ても可能である。

20

【 0 0 4 8 】

以上の構成の元、実施形態におけるシステムの制御の流れを図 2フローチャートに従い説明する。

【 0 0 4 9 】

図 2は、実施形態におけるプロキシサーバ 3 0 0の処理手順を示すフローチャートである。このフローチャートに示す処理は、プロキシサーバ 3 0 0の電源がONになってOSが起動した後に、ハードディスク 1 0 0 4からRAM 1 0 0 3にロードされ実行されるサーバアプリケーションでもある。

【 0 0 5 0 】

プロキシサーバ 3 0 0のプロトコル変換処理部 1 6は起動後、記録装置制御部 1 5を介し、ハードディスク 1 0 0 4内に予め確保された、プロトコル変換処理を実施しているネットワークデバイス情報を記録する管理テーブルの内容をクリアする（ステップ S 2 - 1）。該管理テーブルの詳細に関しては以下のプロセスで説明する。

30

【 0 0 5 1 】

次にネットワークに参加しサービスを開始するにあたり、同ネットワーク上に存在するUPnP対応印刷装置を検索する（ステップ S 2 - 2）。この探索処理は図 3に示し、以下に説明する。

【 0 0 5 2 】

まず、ステップ S 3 - 1にて、マルチキャストアドレスに対し、UPnPで規定される図 4に示すフォーマットのProbeパケットを発行する。つまり、ネットワーク上に参加し、UPnP対応のネットワークデバイスの所在を取得するメッセージを送信する。なお、図 4中のパケットメッセージ中のu u i d（Universally Unique Identifier）は、UPnPデバイスに固有の識別子であって、プロキシサーバ 3 0 0があたかもUPnPデバイスとして機能することを示すためのものでもある（予め設定されて、保持されている）。

40

プロキシサーバ 3 0 0のプロトコル変換処理部 1 6は、図 4に示すProbeパケットを発行した後、予め規定された所定時間内に、ネットワークからの全レスポンスに対し解析を実行する。

【 0 0 5 3 】

実施形態の場合、ネットワーク 5 0 0上の印刷装置 4 0 0は、UPnP対応印刷装置で

50

あるものとしているので、上記プロキシサーバ300からのProbeパケットに対するレスポンスメッセージをプロキシサーバ300に回答することになるが、印刷装置200は非UPnPであるので、回答しないことになる。

【0054】

図5は、ネットワークデバイスの一例としてのUPnP対応印刷装置400からのレスポンスパケットであるProbeMatchの例を示している。プロキシサーバ300のプロトコル変換処理部16は、このパケットを渡してきたネットワーク印刷装置（UPnP対応印刷装置400）へのURLを、記録装置制御部15を介して、該制御部が制御する記録装置（実施形態ではハードディスク）に対し記録保存する。該処理は受信した全レスポンスパケットに対して実施され、プロキシサーバ300は、ネットワーク上に存在する全てのUPnP対応印刷装置のURLを記録することになる（ステップS3-2）。

10

【0055】

以上のプロセスが完了した場合、ステップS3-3において応答が無い場合、プロキシサーバ300のプロトコル変換処理部16は、UPnP検索処理を終了して、図2のステップS2-3に処理を進め、印刷装置情報の取得処理を開始する。

【0056】

ここで、UPnP対応印刷装置は、Probeパケットに対してその印刷装置がサポートするPDL毎に1つのProbeMatchフォーマットの応答を返す。図5に示すように、UPnP対応印刷装置400は、印刷装置コントローラ22a、22bのために同一のURLを有する2つのProbeMatchパケットを発行する。それぞれのProbeMatchパケットは、それぞれのサービスを一意に指定可能なように<Address>タグ内にユニークなuuidを有する。

20

【0057】

図5の例では、それぞれPDLに対するのuuidは、PDL1ではuuid:98190dc2-0890-4ef8-ac9a-5940995e611a、PDL3ではuuid:98190dc2-0890-4ef8-ac9a-5940995e611bである。

【0058】

図6は、図2のステップS2-3の印刷装置情報取得の処理の詳細を示すフローチャートである。

【0059】

プロキシサーバ300の持つプロトコル変換処理部16はSNMP制御モジュール部11より、以下のMIBオブジェクトに対してSNMP Getリクエストをブロードキャストすることで、ネットワーク上に存在する印刷装置情報を取得する（ステップS6-1）。

30

PrinterMakeAndModel：印刷装置ベンダ・製品名称

PrinterName：印刷装置名

PrinterLocation：印刷装置設置場所

IPAddress：印刷装置 IP アドレス

MACAddress：印刷装置MAC アドレス

SupportedPDL：サポートするページ記述言語

SupportedPrintProtocol：サポートするプリントプロトコル

40

ネットワークデバイスである、印刷装置200および400は、上記のSNMP Getリクエストを受信すると、それぞれの印刷装置が実装するSNMP処理部6又はSNMP処理部18において、各オブジェクトに該当する情報を生成した後に、SNMPレスポンスとしてプロキシサーバ300に対し、ユニキャストで応答パケット（応答メッセージ）を送信する。

【0060】

ステップS6-1-1では、レスポンスを受信したかどうかを判定し、レスポンスがあった判定した場合にはステップS6-2へ、レスポンスが無かったと判断した場合にはステップS6-9へ進む。

【0061】

50

各ネットワーク対応印刷装置からレスポンスを受け取ったプロキシサーバ300のプロトコル変換処理部16は、各レスポンス内容を記録装置に既に登録済みの印刷装置管理テーブルの内容と比較し(ステップS6-2)、既にプロトコル変換を実施している印刷装置か否かを判断する(ステップS6-3)。この判定は、u u i dもしくはI Pアドレスを照合することで判定する。

【0062】

ステップS6-3では、プロトコル変換を実施していない印刷装置、即ち、新規に発見された印刷装置であるか否かを判断する。新規発見の印刷装置である場合には、プロキシサーバ300のプロトコル変換処理部16は、該印刷装置がU P n P対応する印刷装置であるか否を、先に記録装置に記録したU P n P対応印刷装置のU R L Lと比較することで判定処理を行なう(ステップS6-4)。

10

【0063】

つまり、S N M P G E Tリクエストのレスポンスとして取得した印刷装置I Pアドレスと、記録装置に記録されたU R Lが一致する場合、図2のステップS2-2の探索処理に対して応答し、且つ、ステップS6-1に対しても応答した場合、その印刷装置はU P n P対応印刷装置であると判断し、該印刷装置に対してはプロトコル変換を実施しない。つまり、管理テーブルには登録しない。

【0064】

換言すれば、図2のステップS2-2の探索処理に対して応答せず、ステップS6-1に対して応答した印刷装置は、非U P n P対応印刷装置であると判断することになる。この場合、ステップS6-6に進んで、記録装置制御部15を介してハードディスクに確保された管理テーブルに追加登録する。

20

【0065】

この場合、実施形態では、S N M P G E Tリクエストのレスポンスとして取得した情報の印刷装置ベンダ・製品名称に対して検索をかけて、もしその印刷装置が複数言語のプリントコントローラを搭載可能または、脱着可能な印刷装置コントローラに対応しているモデルである場合、S u p p o r t e d P D LよりサポートしているP D Lを確認し、個々のP D Lをサポートするコントローラに対して、それぞれ論理的に別の印刷装置として、管理テーブルにU P n P対応の印刷装置であるかのようにするため、PrinterDescriptionエレメントを追加する。

30

【0066】

実施形態のシステム構成の場合、印刷装置200が非U P n P印刷装置であり、図7に示すように、印刷装置コントローラ8a, 8bに対して、論理的な別の印刷装置として印刷装置管理テーブルにPrinterDescriptionエレメントが2つ追加登録される。図7に示す情報は、クライアントからのProbeパケットや所定の探索パケットに応じて、プロキシサーバ300がネットワーク上に、送出する。図7に示すパケットは、プリンタ8a用のデバイスI D, プリンタ8b用のデバイスI Dとして、分割して、又は、ひとまとまりにして、別個のデバイスI Dとして認識可能な形態で、クライアント側で受信される。そして、それぞれのデバイスI Dごとに、プラグアンドプレイが実行することになる。つまり、プリンタ8a用のデバイスI Dをクライアント側のインストーラが取得することにより、P D L 1をインストールし、プリンタ8b用のデバイスI Dをクライアント側のインストーラが取得することにより、別のP D L 2のドライバがインストールされる。また、クライアントが、該デバイスの探索要求を発行した場合は、一つのデバイスではなく、複数の論理的な装置として表示されることになる。どのプリンタドライバをインストールするかは、予めクライアント側の、デバイスI Dとドライバの格納位置を示す情報の組を示すデータベースを引くことにより決定される。例えば、クライアントがプロキシサーバ300から取得したプリンタ8a用のデバイスI Dのうち、MANUFACTURED、MODELの組をクライアント側のインストーラが認識すると、当該8a用のMANUFACTURED、MODELの組をキーにしてデータベースを引いて、プリンタドライバの格納位置を認識して、インストールを行う。別の実施形としては、8a用のプリンタドライバ部MANUFACTURED、COMMAND、MODELの全ての

40

50

組を認識し、MANUFACTURED、MODELに対応する複数のデバイスドライバの中から、PDL 1用のデバイスドライバを自動的に認識してインストールしても良い。

【0067】

PrinterDescriptionエレメントのDeviceIDとして、クライアント100に格納されているそれぞれのPDLに対応するプリンタドライバを一意に指定できるデバイスID(図示の場合には、「LaserBeamPrinter777 PDL1」及び「LaserBeamPrinter777 PDL2」の2つ)をプロトコル変換処理部16は、このPrinterDescriptionエレメントに登録する。またそれぞれの論理印刷装置をUPnPパケットで指定可能なように、<Address>タグ内に異なるuuidを持つように登録する。

【0068】

以上の情報を、記録装置制御部15を介してハードディスクに記録し(ステップS6-7)、ステップS6-8においてUPnPプロトコル処理部14より、管理テーブルに登録された全ての印刷装置に関するHelloパケットを発行し、あたかも2つのUPnP印刷装置(仮想的UPnP印刷装置が2つ)がそのサービスを開始していることをネットワーク上に通知する。

【0069】

一方、プロキシサーバ300より発行されたSNMPGet要求に対し、何のレスポンスが得られなかった場合、つまり、ステップS6-1-1がNoと判定された場合、処理はステップS6-9へ進み、管理テーブルを検索し、登録済み印刷装置を確認する。

【0070】

そして、既に管理テーブルに登録済みの印刷装置がありながら、レスポンスが得られなかった印刷装置があるか否かを判断する(ステップS6-9-1)。登録済みでレスポンスがなかった場合、該当する印刷装置は、電源がOFFになった、もしくはネットワークから離脱したことを示す。この場合、ステップS6-10に進み、プロキシサーバ300のプロトコル変換処理部16は管理テーブルより該印刷装置の情報の削除更新し、ステップS6-11にて、該当する印刷装置のPrinterDescriptionエレメントを削除する。

【0071】

そして、プロキシサーバ300のプロトコル変換処理部16は、UPnPプロトコル処理部14より、管理テーブルから削除された全ての印刷装置に関するByeパケットを発行し、これら管理テーブルから削除された印刷装置がネットワーク上でのサービスを停止したことを通知する(ステップS6-12)。

【0072】

次に、図2におけるプロトコル変換処理(ステップS2-5)の詳細を図8のフローチャートに従って説明する。

【0073】

プロキシサーバ300のプロトコル変換処理部16、ネットワーク上のクライアントから発行されるデバイス検索プロトコルProbeパケットの受信通知をUPnPプロトコル処理部14から受けか否かを判断する(ステップS8-1)。つまり、クライアントがネットワーク上のUPnPネットワークデバイスの探索を行っているか否かを判断する。

【0074】

Probeパケットを受信したと判断した場合には、処理はステップS8-2に進み、プロキシサーバ300のプロトコル変換処理部16が管理する、図7に示す印刷装置管理テーブルを、記録装置制御部15を介してハードディスクから検索する。そして、Probeパケットの検索条件に該当する印刷装置がその管理テーブルに登録されている場合、Probe Matchパケットを印刷装置管理テーブルより生成しUPnPプロトコル処理部14を介して返送する。

【0075】

Probe Matchパケットは図5のような構造をしており、Probeパケットの検索条件に該当するそれぞれの印刷装置コントローラに対して一つ発行される。またProbe Matchパケットは、そのPDLに関する論理的な印刷装置を一意に指定できるように、<Address>タ

10

20

30

40

50

グにユニークなuuidをもつ。実施形態におけるプロキシサーバ300は、管理テーブルに登録されている該当する印刷装置のすべてについて上記Probe Matchパケットを返答する。

【0076】

この結果、上記のProbeパケットを送信したクライアントは、プロキシサーバ300UPnP印刷装置（実施形態では、印刷装置200が2つの言語のコントローラを有することになるので、2つのUPnP印刷装置）であるかのように見えることになる。

【0077】

処理がステップS8-3に進むと、UPnPプロトコル処理部14が図9に示すようなPrinterDescriptionRequestパケットを受信したか否かを判断する。このパケットを受信すると、プロキシサーバ300は、プロトコル変換処理部16が管理する印刷装置管理テーブルの<Address>タグを、PrinterDescriptionの取得要求中の<To>タグで示されるuuidで検索し、図10に示すようなPrinterDescriptionResponseパケットを、印刷装置管理テーブルより生成する。そして、生成したPrinterDescriptionResponseパケットをUPnPプロトコル処理部14を介して要求元に返送する（ステップS8-4）。

【0078】

ステップS8-5では、上記のPrinterDescriptionResponseを取得したクライアントデバイスから、UPnPによるプリントジョブ要求を受信したか否かを判断する。この要求のジョブコマンド、およびジョブ属性はXMLの形式で記述されているので、そのままでは非UPnPデバイスである印刷装置200は解釈できない。そこで、プロキシサーバ300のプロトコル変換処理部16が、UPnPプロトコル処理部14を介して該プリントジョブを受信した場合（ステップS8-5がYESの場合）、プロトコル変換処理部16はSOAP処理部において該コマンド、およびジョブ属性を解析し、次に記録装置制御部15を介して出力指定された印刷装置に該当する管理テーブル情報のうち、サポートするプリントプロトコル（図7の<SupportedPrintProtocol>～</SupportedPrintProtocol>で記述されたプロトコルで、図示の場合には「LPR」となる）、およびIPアドレスを取得し、受信したコマンド、属性情報を該プリントプロトコルに変換する（ステップS8-6）。そして、出力指定された印刷装置のIPアドレスに向けて、変換後の情報を送信する（ステップS8-8）。

【0079】

なお、クライアントからのプリントジョブには、印刷すべき印刷装置のuuid（プロキシサーバ300が印刷装置200のために作成したuuidでもある）が含まれているので、これをキーにして、本来の印刷装置のIPアドレスを取得することができる。

【0080】

プリントジョブを発行したクライアントは、引き続きジョブデータ（この場合PDLデータ）をHTTP POSTコマンドを使用してプロキシサーバ300に対し送信することになる。ステップS8-8-1では、このジョブデータを受信したか否かを判断する。

【0081】

そして、この判断を、ステップS8-8-2で所定時間経過するまで行い、所定時間経過してもジョブデータを受信しない場合には、ステップS8-10にてそのジョブ要求を破棄する。このとき、指定された印刷装置にも破棄要求を行う。

【0082】

また、プリントジョブ要求を受信してから所定時間以内に、ジョブデータを受信した場合、プロキシサーバ300のプロトコル変換処理部16は、先のステップ同様に受信したジョブデータを指定された印刷装置がサポートするプリントプロトコルに変換し（ステップS8-8-3）、先に取得した印刷装置IPアドレスに対しジョブデータを送信する（ステップS8-9）。

【0083】

以上の結果、ジョブコマンド、ジョブ属性、およびジョブデータを受信した印刷装置は、プリント制御部においてジョブコマンド、ジョブ属性を解析した後、印刷装置コントロ

10

20

30

40

50

ーラに対しプリントジョブを送信しプリントを実行する。

【 0 0 8 4 】

なお、図 2 からわかるように、プロキシサーバ 3 0 0 は電源断の指示が成されない限り、以上のプロセス、すなわち、ステップ S 2 - 2 乃至 S 2 - 4 を繰り返し実行することで、定期的にネットワーク印刷装置の稼動状況を更新し、その更新情報に従いプロトコル変換処理を実行する。

【 0 0 8 5 】

そして、図 2 において、プロキシサーバ 3 0 0 の電源断の指示が与えられたと判断した場合、処理はステップ S 2 - 5 からステップ S 2 - 6 に進む。この場合、プロキシサーバ 3 0 0 のプロトコル変換処理部 1 6 はプロトコル変換処理を停止するため、プロトコル変換処理部 1 6 は、記録装置制御部 1 5 を介して全ての管理テーブルを読み出し、管理テーブルに記録された全ての印刷装置に関する Bye パケット (U P n P ネットワーク離脱メッセージ) を、 U P n P プロトコル処理部 1 4 を介して発行し、これら印刷装置がネットワーク上でのサービスを停止したことネットワークの他のデバイス (クライアント) に通知する。

10

【 0 0 8 6 】

以上説明したように、本実施形態によると、プロキシサーバ 3 0 0 は、ネットワーク上の非 U P n P 印刷装置プリンタを、 S N M P プロトコルを用いて検出し登録する。そして、その非 U P n P プリンタに代わってあたかも U P n P 印刷装置であるかのように振る舞い、印刷ジョブを受信した場合には、その印刷ジョブを、指定された非 U P n P 印刷装置に出力する。この結果、非 U P n P 印刷装置が、 U P n P ネットワークのメンバとして機能することになる。

20

【 0 0 8 7 】

なお、実施形態におけるプロキシサーバ 3 0 0 は、非 U P n P のネットワーク印刷装置を代理するサーバとして説明したが、代理する対象は印刷装置に限らず、他のデバイスでも構わない。プリンタ以外のネットワーク対応型デバイスとしてはハードディスク等のストレージデバイス、スキャナ、複写機、およびそれら複合機能を備えるデバイスであって、通信機能を介してプロキシサーバと属性情報の交換、ジョブの送受信が可能な装置であればいずれの場合においても実現可能である。また、この場合、プロキシサーバとネットワーク対応型デバイス間の通信プロトコルは標準化された、あるいは汎用プロトコルであっても、ベンダ固有のプロトコルであっても同様に実現可能である。

30

【 0 0 8 8 】

また本実施形態においてはネットワーク対応型デバイスを例とした実施の形態を示しているが、デバイスとプロキシサーバ間は、 U S B 、 I E E E 1 3 9 4 、パラレルなどのローカル接続による通信によっても実現可能であるし、ネットワークは有線、無線を問わない。

【 0 0 8 9 】

また本実施形態では、独立した形態でプロキシサーバがネットワーク上に存在していたが、該プロキシサーバ機能は、ネットワーク対応型デバイスの内部に物理的に、あるいは論理的に組み込まれている場合においても実現可能である。

40

【 0 0 9 0 】

さらにまた、実施形態におけるプロキシサーバが提供するプロトコル変換の組み合わせとして M i c r o s o f t が主体となって策定している U n i v e r s a l P l u g a n d P l a y と、ネットワーク対応型印刷装置が実装する S N M P 、およびプリントプロトコルとの例を示しているが、 A p p l e の提案する R e n d e z v o u s 、 J B M I A が提案する B M L i n k S などのプロトコルに対しても実現可能であり、また、これらデバイスの検索、デバイスの制御が統合化されたプロトコルのみならず、 S e r v i c e L o c a t i o n P r o t o c o l (S L P) 、 M u l t i c a s t D N S S e r v i c e D i s c o v e r y 等のデバイスが提供するサービスを検索するためのプロトコルへの利用、および W e b S e r v i c e のように X M L / S O A P をベ-

50

スとしたRemote Procedure Call (RPC) 形式によるデバイスの制御を、従来の制御プロトコルに変換するために利用することも可能である。

【0091】

本実施形態においてプロキシサーバ間の情報通知プロトコルとしてHTTP/ TCP / UDP / IPプロトコルを使用した例を示しているが、本発明はトランスポート手段に依存するものではなく、双方向の通信が可能であれば他の汎用プロトコル、あるいは独自プロトコルを使用した場合でも実現可能である。

【0092】

以上説明したように、ネットワーク対応型プラグアンドプレイをサポートしていない印刷装置に対して代理応答装置を設け、上記印刷装置がサポートしているネットワーク対応型プラグアンドプレイ以外のプロトコルで取得したデバイスIDのモデル名称がPDL名称の属性を持たない場合、ネットワーク対応型プラグアンドプレイの周辺装置検索プロトコルの応答として、PDL名称をモデル名に付加して返すことにより、適切なプリンタドライバを情報処理装置に自動インストールすることが可能である。

【0093】

またネットワーク対応型プラグアンドプレイをサポートしていない印刷装置に対して代理応答装置を設け、上記印刷装置がサポートしているネットワーク対応型プラグアンドプレイ以外のプロトコルで取得したデバイスIDが複数のPDLをサポートしている場合、PDLの個数分論理的な印刷装置と認識させるように、デバイスIDのモデル名称に、それぞれのPDL名称を付加したデバイスIDを含むネットワーク対応型プラグアンドプレイの周辺装置検索プロトコルへ応答を行うことにより、情報処理装置に別の論理的な実態であると認識させることが可能になる。

【0094】

なお、実施形態で説明したように、本実施形態の主要部分はプロキシサーバ300の処理にある。プロキシサーバ300は、先に説明したように、ハードディスク等の記憶装置を必要とするものの、パーソナルコンピュータ等の汎用情報処理装置上で動かすサーバアプリケーションプログラムで実現しているわけであるから、本発明は、コンピュータプログラムをその範疇とするのは明らかである。また、通常、コンピュータプログラムは、CDROM等のコンピュータ可読記憶媒体をコンピュータにセットし、システムにコピーもしくはインストールすることで実行可能になるわけであるから、当然、そのようなコンピュータ可読記憶媒体も本発明の範疇に含まれる。

【図面の簡単な説明】

【0095】

【図1】実施形態におけるネットワークシステムを構成する各デバイスのソフトウェアの構成を示す図である。

【図2】実施形態におけるプロキシサーバの処理手順を示すフローチャートである。

【図3】UPnP対応印刷装置検索のプロセスを示すフローチャート図である。

【図4】Probeパケットの一例を示す図である。

【図5】ProbeパケットのレスポンスパケットであるProbe Matchパケットの一例を示す図である。

【図6】印刷装置情報取得のプロセスを示すフローチャート図である。

【図7】管理テーブルのフォーマットを示す図である。

【図8】プロトコル変換処理のプロセスを示すフローチャート図である。

【図9】PrinterDescriptionのRequestパケットの一例を示す図である。

【図10】PrinterDescriptionのResponseパケットの一例を示す図である。

【図11】実施形態におけるクライアント及びプロキシサーバのハードウェアブロック構成図である。

【図12】実施形態におけるプリンタ200、400のハードウェアブロック構成図である。

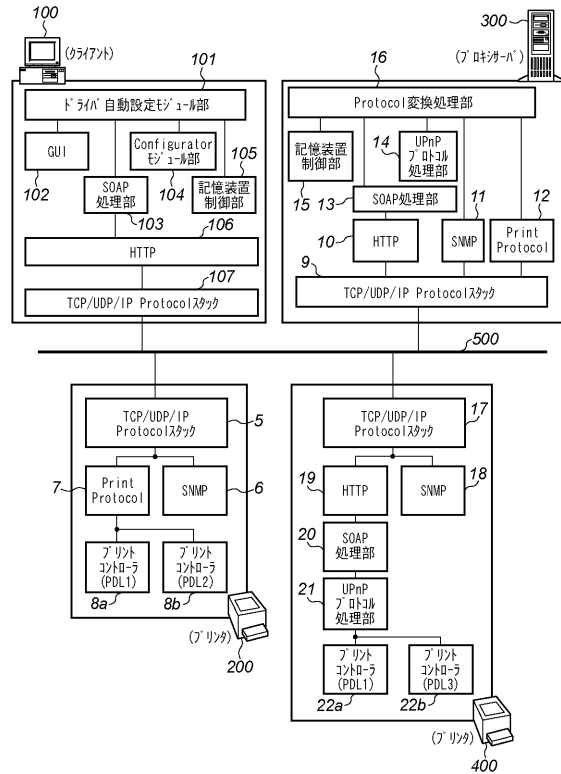
10

20

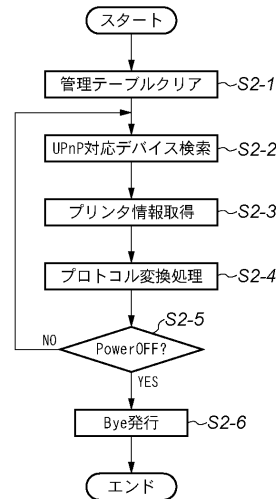
30

40

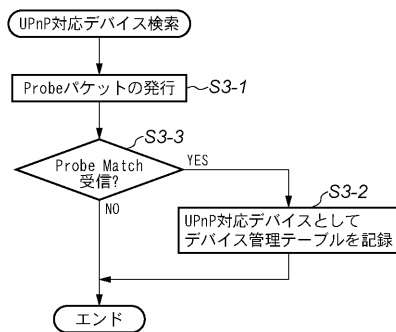
【図 1】



【図 2】



【図 3】



【図 5】

```

<Envelope>
<Header>
<Action>
  http://xxx/ProbeMatch
</Action>
<MessageID>
  uuid:e32e6863-ea5e-4ee4-997e-69539 d1ft2cf
</MessageID>
<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>
  
```

22a用のProbe Match

【図 4】

```

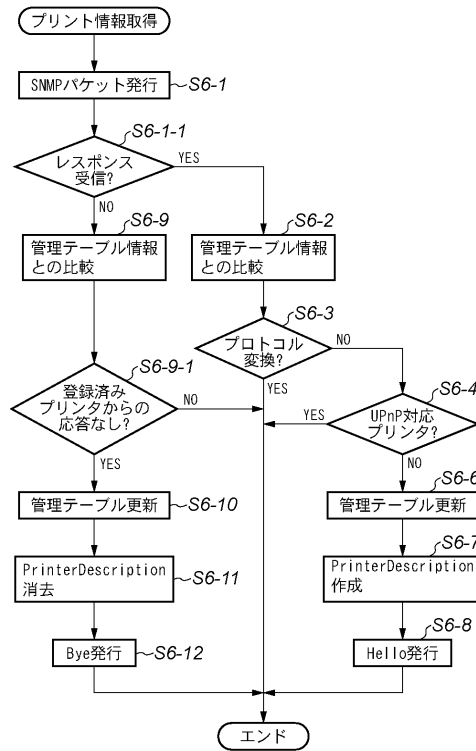
<Envelope>
<Header>
<Action>
  http://xxx/Probe
</Action>
<MessageID>
  uuid:0a6dc791-2be6-4991-9af1-454778a1917e
</MessageID>
</Header>
<Body>
<Probe>
  <Type>PrintBasic</Type>
</Probe>
</Body>
</Envelope>
  
```

```

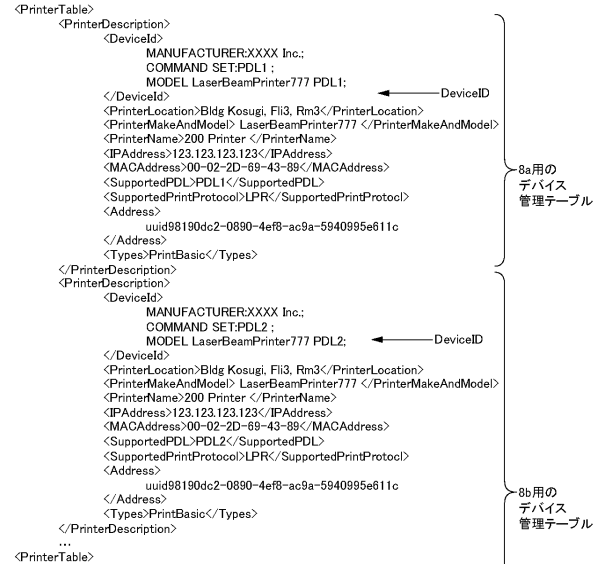
<Envelope>
<Header>
<Action>
  http://xxx/ProbeMatch
</Action>
<MessageID>
  uuid:e32e6863-ea5e-4ee4-997e-69539 d1ft2cf
</MessageID>
<RelatesTo>
  uuid:0a6dc791-2be6-4991-9af1-454778a1917e
</RelatesTo>
</Header>
<Body>
<ProbeMatch>
<EndpointReference>
<Address>
  uuid:98190dc2-0890-4ef8-ac9a-5940995e611a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

22b用のProbe Match

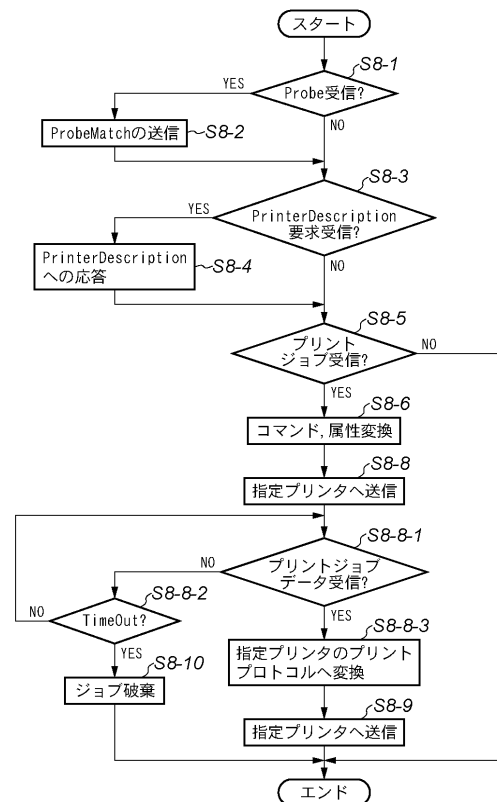
【図 6】



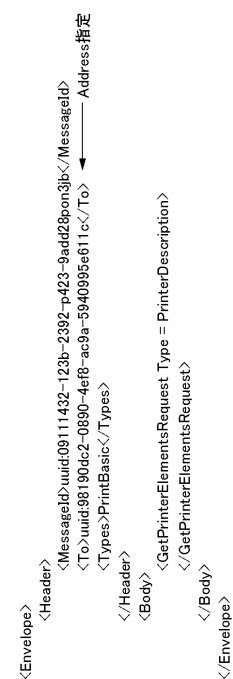
【図 7】



【図 8】



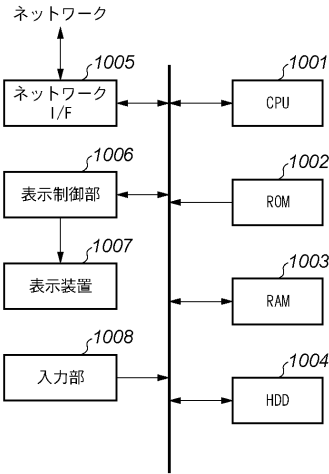
【図 9】



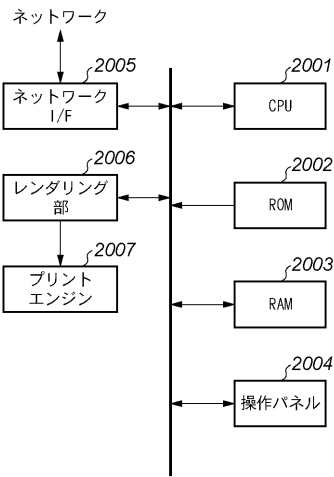
【図 10】

```
<Envelope>
<Header>
  <MessageId>uuid:09111432-123b-2392-p423-9add28pon3jb</MessageId>
  <ResponseTo>uuid:09111432-123b-5555-c787-2san15sip5ec</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>
```

【図 11】



【図 12】



フロントページの続き

審査官 須藤 竜也

(56)参考文献 特開2003-006133(JP,A)
特開2004-078359(JP,A)
特開2000-181656(JP,A)
特表2002-501244(JP,A)

(58)調査した分野(Int.Cl., DB名)
G06F 13/00