

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4630783号  
(P4630783)

(45) 発行日 平成23年2月9日 (2011.2.9)

(24) 登録日 平成22年11月19日 (2010.11.19)

(51) Int. Cl.	F I
<b>G 0 6 F 3/12 (2006.01)</b>	G O 6 F 3/12 K
<b>B 4 1 J 29/38 (2006.01)</b>	B 4 1 J 29/38 Z

請求項の数 25 (全 54 頁)

(21) 出願番号	特願2005-280107 (P2005-280107)	(73) 特許権者	000001007
(22) 出願日	平成17年9月27日 (2005.9.27)		キヤノン株式会社
(65) 公開番号	特開2006-195953 (P2006-195953A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成18年7月27日 (2006.7.27)	(74) 代理人	100126240
審査請求日	平成20年9月16日 (2008.9.16)		弁理士 阿部 琢磨
(31) 優先権主張番号	特願2004-364441 (P2004-364441)	(74) 代理人	100124442
(32) 優先日	平成16年12月16日 (2004.12.16)		弁理士 黒岩 創吾
(33) 優先権主張国	日本国 (JP)	(72) 発明者	本田 秀樹
			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
		(72) 発明者	安部 孝一
			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
		審査官	内田 正和
			最終頁に続く

(54) 【発明の名称】 周辺装置制御システム、印刷装置、周辺装置制御方法、及びプログラム

(57) 【特許請求の範囲】

【請求項 1】

周辺装置制御部とアプリケーションとを有する情報処理装置と、前記情報処理装置に接続された周辺装置とを有する周辺装置制御システムにおいて、

前記周辺装置制御部は、前記周辺装置の消耗品の残量及び並び順、及び並び方向に関する情報をアプリケーションに渡し、

前記アプリケーションは、周辺装置制御部から受け取った残量及び並び順、及び並び方向に関する情報に基づき消耗品の残量を消耗品の並び方向に従った並び順で表示することを特徴とする周辺装置制御システム。

【請求項 2】

周辺装置制御部とアプリケーションとを有する情報処理装置と、前記情報処理装置に接続された周辺装置とを有する周辺装置制御システムにおいて、

前記周辺装置は、前記周辺装置制御部からの要求に応じて残量及び消耗品の並び順、及び並び方向に関する情報を前記周辺装置制御部に渡し、

前記周辺装置制御部は、前記周辺装置から取得した消耗品の残量及び並び順、及び並び方向に関する情報をアプリケーションに渡し、

前記アプリケーションは、前記周辺装置制御部から受け取った残量及び並び順、及び並び方向に関する情報に基づき消耗品の残量を消耗品の並び方向に従った並び順で表示することを特徴とする周辺装置制御システム。

【請求項 3】

10

20



前記周辺装置制御部は、消耗品の並び順、及び並び方向に関する情報を、前記周辺装置から取得できた場合は前記周辺装置から取得した消耗品の並び順、及び並び方向に関する情報をアプリケーションに渡し、前記周辺装置から取得できない場合は前記周辺装置制御部で保持している消耗品の並び順、及び並び方向に関する情報をアプリケーションに返却することを特徴とする請求項 1 記載の周辺装置制御システム。

【請求項 4】

前記周辺装置は、消耗品の並び順、及び並び方向に関する情報を、前記周辺装置のメモリから取得することを特徴とする請求項 2 または 3 記載の周辺装置制御システム。

【請求項 5】

前記周辺装置は、消耗品の並び順、及び並び方向に関する情報を、消耗品の装着状態を判断して取得することを特徴とする請求項 2 または 3 記載の周辺装置制御システム。

10

【請求項 6】

前記周辺装置は、前記周辺装置制御部からの要求に応じて消耗品の残量に関する情報を前記周辺装置制御部に返却することを特徴とする請求項 1 乃至 5 いずれか記載の周辺装置制御システム。

【請求項 7】

前記消耗品の並びに関する情報は、消耗品の画像を表示するための情報を含むことを特徴とする請求項 1 乃至 6 いずれか記載の周辺装置制御システム。

【請求項 8】

前記消耗品の並びに関する情報は、消耗品のビットマップまたは色情報を含むことを特徴とする請求項 1 乃至 6 いずれか記載の周辺装置制御システム。

20

【請求項 9】

前記周辺装置は、前記周辺装置制御部からの要求に応じて消耗品の画像を表示するための情報を前記周辺装置制御部に返却することを特徴とする請求項 1 乃至 8 いずれか記載の周辺装置制御システム。

【請求項 10】

前記周辺装置は、前記周辺装置制御部からの要求に応じて消耗品のビットマップまたは色情報を前記周辺装置制御部に返却することを特徴とする請求項 1 乃至 8 いずれか記載の周辺装置制御システム。

【請求項 11】

30

周辺装置に接続され、周辺装置制御部とアプリケーションとを有する情報処理装置における周辺装置制御方法において、

前記周辺装置制御部は、前記周辺装置の消耗品の残量及び並び順、及び並び方向に関する情報をスキーマの定義に変換してアプリケーションに渡し、

前記アプリケーションは、周辺装置制御部から受け取ったスキーマの定義に変換された残量及び並び順、及び並び方向に関する情報に基づき消耗品の残量を消耗品の並び方向に従った並び順で表示することを特徴とする周辺装置制御方法。

【請求項 12】

周辺装置に接続され、周辺装置制御部とアプリケーションとを有する情報処理装置における周辺装置制御方法において、

40

前記周辺装置は、前記周辺装置制御部からの要求に応じて残量及び消耗品の並び順、及び並び方向に関する情報を前記周辺装置制御部に渡し、

前記周辺装置制御部は、前記周辺装置から取得した消耗品の残量及び並び順、及び並び方向に関する情報をスキーマの定義に変換してアプリケーションに渡し、

前記アプリケーションは、前記周辺装置制御部から受け取ったスキーマの定義に変換された残量及び並び順、及び並び方向に関する情報に基づき消耗品の残量を消耗品の並び方向に従った並び順で表示することを特徴とする周辺装置制御方法。

【請求項 13】

周辺装置に接続される情報処理装置で実行されるプログラムにおいて、

前記周辺装置から消耗品の残量及び並び順、及び並び方向に関する情報を取得し、

50



前記周辺装置の消耗品の残量及び並び順、及び並び方向に関する情報をアプリケーションに送信することを特徴とするプログラム。

【請求項 1 4】

周辺装置に接続される情報処理装置で実行されるアプリケーションプログラムにおいて、

周辺装置の消耗品の残量及び並び順、及び並び方向に関する情報を取得し、取得した残量及び並び順、及び並び方向に関する情報に基づき周辺装置の消耗品の残量を消耗品の並び方向に従った並び順で表示することを特徴とするアプリケーションプログラム。

【請求項 1 5】

前記消耗品の並び順、及び並び方向に関する情報を、前記周辺装置から取得できた場合は前記周辺装置から取得した消耗品の並び順、及び並び方向に関する情報をアプリケーションに送信し、前記周辺装置から取得できない場合はあらかじめ保持している消耗品の並び順、及び並び方向に関する情報をアプリケーションに送信することを特徴とする請求項 1 3 記載のプログラム。

10

【請求項 1 6】

前記消耗品の並びに関する情報は、消耗品の画像を表示するための情報を含むことを特徴とする請求項 1 3 乃至 1 5 いずれか記載のプログラム。

【請求項 1 7】

前記消耗品の並びに関する情報は、消耗品のビットマップまたは色情報を含むことを特徴とする請求項 1 3 乃至 1 5 いずれか記載のプログラム。

20

【請求項 1 8】

前記周辺装置から消耗品の画像を表示するための情報を取得し、アプリケーションに送信することを特徴とする請求項 1 3 記載のプログラム。

【請求項 1 9】

前記周辺装置から消耗品のビットマップまたは色情報を取得し、アプリケーションに送信することを特徴とする請求項 1 3 記載のプログラム。

【請求項 2 0】

前記消耗品の画像を表示するための情報を取得し、前記消耗品に関する表示を行なうことを特徴とする請求項 1 4 記載のプログラム。

【請求項 2 1】

前記消耗品のビットマップまたは色情報を取得し、前記消耗品に関する表示を行なうことを特徴とする請求項 1 4 記載のプログラム。

30

【請求項 2 2】

周辺装置に接続され、周辺装置制御部とアプリケーションとを有する情報処理装置における周辺装置制御方法において、

前記周辺装置制御部は、前記周辺装置の消耗品の残量及び並び順、及び並び方向に関する情報をアプリケーションに渡し、

前記アプリケーションは、周辺装置制御部から受け取った残量及び並び順、及び並び方向に関する情報に基づき消耗品の残量を消耗品の並び方向に従った並び順で表示することを特徴とする周辺装置制御方法。

40

【請求項 2 3】

周辺装置に接続され、周辺装置制御部とアプリケーションとを有する情報処理装置における周辺装置制御方法において、

前記周辺装置は、前記周辺装置制御部からの要求に応じて消耗品の並び順、及び並び方向に関する情報を前記周辺装置制御部に渡し、

前記周辺装置制御部は、前記周辺装置から取得した消耗品の並び順、及び並び方向に関する情報をアプリケーションに渡し、

前記アプリケーションは、前記周辺装置制御部から受け取った情報に基づき消耗品の残量を消耗品の並び方向に従った並び順で表示することを特徴とする周辺装置制御方法。

【請求項 2 4】

50



周辺装置に接続される情報処理装置で実行されるプログラムにおいて、  
前記周辺装置から消耗品の残量及び並び順、及び並び方向に関する情報を取得し、  
前記周辺装置の消耗品の残量及び並び順、及び並び方向に関する情報をアプリケーションに送信することを特徴とするプログラム。

【請求項 25】

周辺装置に接続される情報処理装置で実行されるアプリケーションプログラムにおいて、  
周辺装置の消耗品の残量及び並び順、及び並び方向に関する情報を取得し、  
取得した周辺装置の消耗品の残量及び並び順、及び並び方向に関する情報に基づき周辺装置の消耗品の残量を消耗品の並び方向に従った並び順で表示することを特徴とするアプリケーションプログラム。

10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ネットワークを介してプリンタ等の周辺装置に対する制御が可能な技術に関する。

【背景技術】

【0002】

近年、ネットワークの基盤が整備されるに従い、ネットワーク対応型のプリンタ、複写機、ファクシミリ、スキャナ、デジタルカメラ、およびこれらの複合機能を備える画像処理装置等の周辺装置が急速に普及しつつある。これに伴い、これらの周辺装置の製造元とは全く関係の無いサードパーティから、これらの周辺装置を制御したり、状態をモニタしたりするような前記サードパーティ製のアプリケーションが、ユーザに配布（販売）されるようになってきている。

20

【0003】

例えば、特許文献 1 には、周辺装置であるプリンタ用のインク情報をネットワークを介して当該プリンタと接続される情報処理装置内のアプリケーション（ステータスマニタ）でリアルタイムに表示できるものが開示されている。

【0004】

また、例えば、特許文献 2 には、周辺装置であるプリンタがインク残量を検出し、それを制御コマンドに変換して情報処理装置に送り、情報処理装置上でインク残量を表示するような情報処理装置及び情報処理方法が提案されている。

30

【特許文献 1】特開平 09 - 152946 号公報

【特許文献 2】特開平 08 - 118675 号公報

【発明の開示】

【発明が解決しようとする課題】

【0005】

上記従来の例では、アプリケーション（ステータスマニタ）では、インクの画像のビットマップ等のリソースをアプリケーション内に持つことで、この画像を使って、インクの形状、色、大きさ、並び方向、並び順、残量等を、細部までグラフィカルに、かつ、ビジュアル的にリアルに表示することができる。しかしながら、その為には、インクの形状、色、大きさ、並び方向、並び順がどのようなものであるかを予め知っておき、各インクの残量を表す画像のビットマップ等のリソースを予め残量の各段階分複数持っておく必要があった。ここでいうリアルな表示とは、具体的には交換用インクタンクの形状が各ベンダーや各製品毎に異なったり、同じシアンという色でも各ベンダー毎に色味が多少異なったりするので、正確かつ忠実な画像を使うことで、これらの部分も細部までグラフィカルに、かつ、ビジュアル的にリアルに表示することである。

40

【0006】

自社（プリンタベンダー）製のアプリケーション（ステータスマニタ）の場合は、対応するプリンタの前述したような各必要情報を入手して制御に活用することが比較的容易で

50



ある。よって、前述したような各必要情報を入手できたプリンタに対して、前記仕様を備えるアプリケーション（ステータスマニタ）を実現できた。しかしながら、このような自社（プリンタベンダー）製のアプリケーション（ステータスマニタ）の場合でも、新しいプリンタに対応する際には、その新しいプリンタの前述したような各必要情報を再度入手し、それをもとにアプリケーションを作り直す必要があった。

【 0 0 0 7 】

また、上記従来例では、サードパーティ製等のアプリケーションでは、プリンタベンダーから前述したような各必要情報を入手することが困難な場合、アプリケーション内に各必要情報を持つことができない。よって、例えば後述するような Windows（登録商標）XP の Bidirectional Communications Schema を使うことでインク残量を文字列で表示することはできた。しかしながら、インクの形状、色、大きさ、並び方向、並び順、残量等を、細部までグラフィカルに、かつ、ビジュアル的にリアルに表示することはできなかった。このように、対応する全プリンタの前述したような各必要情報を入手することができず、インクの形状、色、大きさ、並び方向、並び順、残量等を、細部までグラフィカルに、かつ、ビジュアル的にリアルに表現できなかった。また、仮に表現できたとしても、対象のプリンタを限定せざるを得ないという問題もあった。

【 0 0 0 8 】

さらに、上記従来例では、前述したような各必要情報を入手して、アプリケーション内の制御に活用することで、インクの形状、色、大きさ、並び方向、並び順、残量等を、細部までグラフィカルに、かつ、ビジュアル的にリアルに表示することができる場合もある。しかしながら、インクの画像のビットマップ等のリソースを使っていたので、2次元の表現に留まっており、3次元の表現を行おうとする場合、処理が非常に複雑になるので、実現することが困難であるという問題もあった。

【 0 0 0 9 】

本発明では、これらの問題点に着目し、周辺装置の製造元とは全く関係のないサードパーティ製のアプリケーション等の任意のアプリケーション上に、周辺装置に関する情報を表現することを目的とする。例えば、プリンタに搭載されているインクの状態を表す情報、すなわち、インクの形状、色、大きさ、並び方向、並び順、残量等を、細部までグラフィカルに、かつ、ビジュアル的にリアルに表現することを目的とする。さらに、3次元の表現を容易に行うことができる周辺装置制御システムを実現することも目的とする。

【課題を解決するための手段】

【 0 0 1 0 】

上記の課題を解決するに、本発明の周辺装置制御システムは、  
周辺装置制御部とアプリケーションとを有する情報処理装置と、前記情報処理装置に接続された周辺装置とを有する周辺装置制御システムにおいて、

前記周辺装置制御部は、前記周辺装置の消耗品の残量及び並び順、及び並び方向に関する情報をアプリケーションに渡し、

前記アプリケーションは、周辺装置制御部から受け取った残量及び並び順、及び並び方向に関する情報に基づき消耗品の残量を消耗品の並び方向に従った並び順で表示することを特徴とする。

【 0 0 1 1 】

また、周辺装置制御部とアプリケーションとを有する情報処理装置と、前記情報処理装置に接続された周辺装置とを有する周辺装置制御システムにおいて、

前記周辺装置は、前記周辺装置制御部からの要求に応じて残量及び消耗品の並び順、及び並び方向に関する情報を前記周辺装置制御部に渡し、

前記周辺装置制御部は、前記周辺装置から取得した消耗品の残量及び並び順、及び並び方向に関する情報をアプリケーションに渡し、

前記アプリケーションは、前記周辺装置制御部から受け取った残量及び並び順、及び並び方向に関する情報に基づき消耗品の残量を消耗品の並び方向に従った並び順で表示することを特徴とする。



## 【 0 0 1 2 】

また、周辺装置に接続され、周辺装置制御部とアプリケーションとを有する情報処理装置における周辺装置制御方法において、

前記周辺装置制御部は、前記周辺装置の消耗品の残量及び並び順、及び並び方向に関する情報をスキーマの定義に変換してアプリケーションに渡し、

前記アプリケーションは、周辺装置制御部から受け取ったスキーマの定義に変換された残量及び並び順、及び並び方向に関する情報に基づき消耗品の残量を消耗品の並び方向に従った並び順で表示することを特徴とする。

## 【 0 0 1 3 】

また、周辺装置に接続され、周辺装置制御部とアプリケーションとを有する情報処理装置における周辺装置制御方法において、

前記周辺装置は、前記周辺装置制御部からの要求に応じて残量及び消耗品の並び順、及び並び方向に関する情報を前記周辺装置制御部に渡し、

前記周辺装置制御部は、前記周辺装置から取得した消耗品の残量及び並び順、及び並び方向に関する情報をスキーマの定義に変換してアプリケーションに渡し、

前記アプリケーションは、前記周辺装置制御部から受け取ったスキーマの定義に変換された残量及び並び順、及び並び方向に関する情報に基づき消耗品の残量を消耗品の並び方向に従った並び順で表示することを特徴とする。

## 【 0 0 1 4 】

また、周辺装置に接続される情報処理装置で実行されるプログラムにおいて、

前記周辺装置から消耗品の残量及び並び順、及び並び方向に関する情報を取得し、

前記周辺装置の消耗品の残量及び並び順、及び並び方向に関する情報をアプリケーションに送信することを特徴とする。

## 【 0 0 1 5 】

また、周辺装置に接続される情報処理装置で実行されるアプリケーションプログラムにおいて、

周辺装置の消耗品の残量及び並び順、及び並び方向に関する情報を取得し、取得した残量及び並び順、及び並び方向に関する情報に基づき周辺装置の消耗品の残量を消耗品の並び方向に従った並び順で表示することを特徴とする。

## 【 0 0 1 7 】

また、本発明の周辺装置制御方法は、

周辺装置に接続され、周辺装置制御部とアプリケーションとを有する情報処理装置における周辺装置制御方法において、

前記周辺装置制御部は、前記周辺装置の消耗品の残量及び並び順、及び並び方向に関する情報をアプリケーションに渡し、

前記アプリケーションは、周辺装置制御部から受け取った残量及び並び順、及び並び方向に関する情報に基づき消耗品の残量を消耗品の並び方向に従った並び順で表示することを特徴とする。

## 【 0 0 1 8 】

また、周辺装置に接続され、周辺装置制御部とアプリケーションとを有する情報処理装置における周辺装置制御方法において、

前記周辺装置は、前記周辺装置制御部からの要求に応じて消耗品の並び順、及び並び方向に関する情報を前記周辺装置制御部に渡し、

前記周辺装置制御部は、前記周辺装置から取得した消耗品の並び順、及び並び方向に関する情報をアプリケーションに渡し、

前記アプリケーションは、前記周辺装置制御部から受け取った情報に基づき消耗品の残量を消耗品の並び方向に従った並び順で表示することを特徴とする。

## 【 0 0 1 9 】

また、本発明のプログラムは、

周辺装置に接続される情報処理装置で実行されるプログラムにおいて、



前記周辺装置から消耗品の残量及び並び順、及び並び方向に関する情報を取得し、  
前記周辺装置の消耗品の残量及び並び順、及び並び方向に関する情報をアプリケーションに送信することを特徴とする。

【0020】

また、本発明のプログラムは、

周辺装置に接続される情報処理装置で実行されるアプリケーションプログラムにおいて

、  
周辺装置の消耗品の残量及び並び順、及び並び方向に関する情報を取得し、  
取得した周辺装置の消耗品の残量及び並び順、及び並び方向に関する情報に基づき周辺装置の消耗品の残量を消耗品の並び方向に従った並び順で表示することを特徴とする。

10

【発明の効果】

【0021】

本発明によれば、アプリケーションが周辺装置の消耗品の並びを考慮して消耗品に関する表示を行うことが容易になる。

【発明を実施するための最良の形態】

【0022】

以下、本発明を適用した好適な実施形態を、添付図面を参照しながら説明する。

【0023】

< 第1の実施の形態 >

図1は、本発明の第1の実施形態に係る情報処理装置及び周辺装置からなる周辺装置制御システムの構成を示すブロック図である。

20

【0024】

同図において、1は情報処理装置であり、一般的なパーソナルコンピュータ（以下、PCと略す場合がある）で構成される。PC1は図3で後述するようなハードウェアで構成され、OS（Operating System）として米国マイクロソフト社のWindows（登録商標）XPがインストールされている。3はプリンタであり、カラーインクジェットプリンタで構成され、本実施形態における周辺装置である。プリンタ3はXYZ社製のkmmnというモデル名のプリンタである。尚、本発明における周辺装置としては、プリンタ、複写機、ファクシミリ、またはこれらの複合機などの画像形成装置、スキャナ、デジタルカメラであってもよい。

30

【0025】

プリンタ3は図4で後述するようなハードウェアで構成され、PC1とUSBインタフェース9を介して接続されており、互いに双方向通信が可能である。USBとはUniversal Serial Busの略であり、双方向通信が可能な公知のインタフェースである。

【0026】

36は図5で後述するランゲージモニタ（以降、LMと略す場合がある）であり、Windows（登録商標）用のダイナミックリンクライブラリで構成される。6はランゲージモニタ用リソースファイル（以降、LMリソースと略す場合がある）であり、Windows（登録商標）用のダイナミックリンクライブラリで構成され、ビットマップリソースや文字列リソース等が格納されている。30はアプリケーションであり、Windows（登録商標）用の実行可能形式のファイル（\*.EXE）で構成される。アプリケーション30の一例としては、プリンタ3の状態を表示するようなステータスモニタ等が挙げられる。

40

【0027】

アプリケーション30は前記MSDNのサイトで公開されているPrinting and Print Spooler Interfacesを利用して、LM36ヘータ（情報）を送ったり、LM36からデータ（情報）を受け取ったりすることができる。これはWindows（登録商標）XPの公知の機能である。

【0028】

50



図2は本発明に係る情報処理装置及び周辺装置からなる周辺装置制御システムをネットワーク環境下で実現した時のシステムの構成部分を表すブロック図である。同図において、2は情報処理装置であり、一般的なパーソナルコンピュータ（以降、PCと略す場合がある）で構成される。PC2は図3で後述するようなハードウェアで構成され、OS（Operating System）として米国マイクロソフト社のWindows（登録商標）XPがインストールされている。PC1とPC2はEthernet（登録商標）で構成されるネットワーク4を介して接続され、互いに双方向通信が可能である。本発明の周辺装置制御システムでは、PC1がサーバ、PC2がクライアントとなるような関係を持ち、PC1はプリンタ3を共有プリンタとしてネットワーク4を介して他の情報処理装置から印刷することができるようなプリントサーバの機能を備えている。200はアプリケーションであり、Windows（登録商標）用の実行可能形式のファイル（\*.EXE）で構成される。アプリケーション200の一例としては、プリンタ3の状態を表示するようなステータスマニタ等が挙げられる。

10

#### 【0029】

アプリケーション200は前記MSDNのサイトで公開されているPrinting and Print Spooler Interfacesを利用して、PC1内のLM36ヘータ（情報）を送ることができる。また、LM36からデータ（情報）を受け取ったりすることができる。これはWindows（登録商標）XPの公知の機能であるので、その詳細説明は省略する。以下の説明で引用される関数（情報）の内、特に詳しい説明を付加していないものに関しては、インターネット上の下記のURLで公開されている。

20

Microsoft Developer Network（MSDN）のサイト<http://msdn.microsoft.com/library/default.asp>で公開されている。

#### 【0030】

図3はPCのハードウェア構成の一例を表すブロック図である。PC1及びPC2は何れも同図に示すようなハードウェア構成を持つが、ここではPC1を例にして説明する。PC1はランダムアクセスメモリ部（RAM1201）、記憶部であるハードディスクドライブ部（HDD1202）、入力部の一例であるキーボード部（KBD1203）、制御部のCPU1204、表示部の一例である表示用ディスプレイ（LCD1205）、通信制御部の一例であるネットワークボード（NB1207）、以上のPC1の構成要素を互いに接続するバス1206から構成される。なお、記憶部は、可搬性CD-ROMまたは内部据付のROMなどであってもよい。図1に示すPC1の各モジュール（アプリケーション30、LM366）は、HDD1202に記憶され、必要に応じてRAM1201に読み出されてCPU1204により実行される。これにより、CPU1204が、図1に示す各モジュールの機能を実現する。

30

#### 【0031】

図4はプリンタのハードウェア構成を表すブロック図である。プリンタ3は同図に示すようなハードウェア構成を持つ。同図において、15はマイクロプロセッサ等から構成されるCPUであり、プリンタ3の中央処理装置として、ROM16に記憶されているプログラムに従って、RAM17、通信部18、記録部19を制御する。

40

#### 【0032】

ROM16にはプリンタドライバ50（図5で後述する）の制御に従ってプリンタ3が記録（印刷）処理や、状態をPC1へ伝える処理を行うプログラムが記憶されている。

#### 【0033】

RAM17は主にPC1から送られて、それをもとに記録部19によって印刷される印字データが一時的に記憶される。通信部18にはUSBインタフェース9用の接続ポートが含まれており、USBの通信を制御する。記録部19は、インクジェット方式の記録ヘッド、各カラーインク、キャリッジ、記録紙搬送機構等から構成される記録ユニットと、前記印字データをもとに前記記録ヘッドにて印字用パルスが発生させる為のASIC等か

50



ら構成される電気回路とから構成される。印刷可能なアプリケーション上での印刷操作によって、アプリケーションで開かれているファイルの表示内容（画像データ）が、E M F形式のスプールファイルとしてP C 1のH D D 1 2 0 2に一時的に格納される。そして、プリンタドライバ5 0を介してプリンタ3 制御用コマンドを含む印字データに変換された後、U S Bインタフェース9を介してプリンタ3に送られる。プリンタ3にて受信された印字データは、記録部1 9によって印字用パルスに変換されて、記録紙上に印刷される。

#### 【0034】

図5はP C 1におけるプリンタドライバの構成を表すブロック図である。同図において、プリンタドライバ5 0はP C 1にインストールされているドライバであり、3 2～3 8で示す複数のモジュールを備えている。アプリケーション3 0は印刷指令やプリンタの状態表示が可能なアプリケーションソフトウェアであり、例えば、Windows（登録商標）X P O Sに標準で同梱されているテキストエディタであるNotepad（Notepad.exe）等に相当する。尚、本実施形態では、アプリケーション3 0はステータスモニタ等である。3 1はG D I（Graphics Device Interface）であり、Windows（登録商標）X P O Sの一部である。3 2はプリンタキューで、Windows（登録商標）X P O SのSpoolerの一部であり、印刷ジョブをキューイングする。次にプリンタドライバ5 0の構成を説明する。3 3はプリントプロセッサであり、印刷レイアウトの変更や印刷画像の特殊処理を行う。3 4はグラフィックスドライバで、プリンタドライバ5 0の画像処理のコアとして動作し、G D I 3 1から送られて来る描画命令に基づき印刷用の画像処理を行い、印刷制御コマンドを作成する。3 5はU Iモジュールで、プリンタドライバ5 0のユーザインタフェースの提供及び制御を行う。3 6は図1で説明したランゲージモニタで、データの通信I / Fとしてデータの送受信を制御する。6は図1で説明したランゲージモニタ用リソースファイルである。3 7はポートモニタであり、L M 3 6から送られてくるデータを適切なポートに対して送信したり、プリンタ3から送られてくるデータを、クラスドライバ3 8を介して受信したりする。クラスドライバ3 8は、最もポートに近いローレベルのモジュールであり、本発明ではU S Bのプリンタクラスのドライバに相当し、ポート（本発明ではU S Bポート）を制御する。プリンタドライバ5 0はマルチ言語対応のドライバであるので、マルチ言語対応のO SであるWindows（登録商標）X Pがサポートしている全ての言語に対応しており、これら全ての言語用の文字列リソース等を備えている。

#### 【0035】

尚、プリンタドライバ5 0はプリンタ3の製造元であるX Y Z社製のものである。

#### 【0036】

図6はアプリケーション3 0（ステータスモニタ等）がプリンタ3からインクの情報及び状態を取得する時に使用するスキーマを表す図である。

#### 【0037】

これらのスキーマは、アプリケーション3 0がPrinting and Print Spooler Interfacesを介してL M 3 6経由でプリンタ3からインクの情報及び状態を取得する時に、Windows（登録商標）X P O Sで利用できるC O MインタフェースI B i d i S p lのA P I関数I B i d i S p l::SendRecv（）をコールする際に引数として指定される。

#### 【0038】

同図において、InkInfoはインクの情報を表すPropertyであり、スキーマのフルパスの指定は¥Printer.InkInfoである。[Color]は色の情報を表すPropertyである。スキーマのフルパスの指定は¥Printer.InkInfo.[Color]である。設定可能な値は、黒を表すBlack、シアンを表すCyan、マゼンタを表すMagenta、黄を表すYellow、一体型のフォトインクを表すPhoto、一体型のカラーインクを表すColorの何れかである。

#### 【0039】

例えば、黒インクの情報を取得したい場合は、¥Printer.InkInfo.B

10

20

30

40

50



lackと指定する。

【0040】

Installedは[Color]色のインクが搭載されているか否かを表すValueである。データ型はBooleanである。スキーマのフルパスの指定は¥Printer.InkInfo.[Color]:Installedである。設定可能な値は、搭載されていること表すTrue、搭載されていないを表すFalseの何れかである。

【0041】

Stateは[Color]色のインクの残量の状態を表すValueである。データ型はStringである。スキーマのフルパスの指定は¥Printer.InkInfo.[Color]:Stateである。設定可能な値は、インク満タンを表すFull、インク残りわずかを表すLow、インク無しを表すOut、インク残量不明を表すUnknownの何れかである。

10

【0042】

ModelNameは[Color]色のインクの型名を表すValueであり、データ型はString、スキーマのフルパスの指定は¥Printer.InkInfo.[Color]:ModelNameである。

【0043】

設定可能な値の例は、同図のExamplesのようなASCIIの文字列である。

【0044】

Directionは[Color]色のインクの並び方向を表すValueである。データ型はString、スキーマのフルパス指定は¥Printer.InkInfo.[Color]:Directionである。設定可能な値は、原点をプリンタの左下として、左から右並びを表すLeftRight、手前から後ろの並びを表すFrontBack、下から上の並びを表すBottomTopの何れかである。

20

【0045】

Orderは[Color]色のインクの原点からの順番を表すValueである。原点はプリンタの左下である。データ型はInt、スキーマのフルパス指定は¥Printer.InkInfo.[Color]:Orderである。設定される値は、例えば、[Color]色のインクの原点からの順番が1番目の場合は、1となる。

30

【0046】

Installed、State、ModelName、Direction、Order等のValueはLM36またはプリンタ3からアプリケーション30へ返される値である。

【0047】

このように、同図で定義されているスキーマを使って、アプリケーション30はプリンタ3に搭載されているインクの情報及び状態、並び方向、並び順を取得することができる。

【0048】

スキーマとは一般的には、データベース全体の構造のことまたは、それらを記述したファイルを意味するが、ここでは周辺装置の状態を表現する表現方法・形式のこと意味するものとする。

40

【0049】

図7は図6で定義されているスキーマを使ってインクの情報及び状態、並び方向、順番を列挙(Enumeration)する時のスキーマ及び値を表す図である。同図において、アプリケーション30(ステータスマニタ等)から¥Printer.InkInfoのスキーマを指定してIBidiSpl::SendRecv()関数をコールする。すると(Query(Schema))、プリンタ3の全インクの情報及び状態、並び方向、並び順のスキーマ(Retrieve(Schema))と値(Retrieve(Value))がセットで返される。

50



## 【 0 0 5 0 】

同図の例では、プリンタ 3 に黒、シアン、マゼンタ、黄のインクが搭載されていて、インクの残量の状態がそれぞれ、インク満タン ( F u l l )、インク残りわずか ( L o w )、インク無し ( O u t )、インク満タン ( F u l l ) である。インクの型名がそれぞれ、C I - B、C I - C、C I - M、C I - Y、インクの並び方向は左から右方向で、順番は左から順に、黒、シアン、マゼンタ、黄の順で並んでいることを表している。

## 【 0 0 5 1 】

図 8 は、図 7 のスキーマを使ってステータスアプリケーション 3 0 (ステータスマニタ等) がインクの残量状態をグラフィカルに表示する一例である。4 0 に現在のプリンタの状態が文字列で表現される。4 1 にはインクの並び方向、並び順および残量がグラフィックを用いて表現されている。インクは左から、黒 ( 4 2 )、シアン ( 4 3 )、マゼンタ ( 4 4 )、黄 ( 4 5 ) の順で並んでおり、残量は、黒は満タン ( F u l l )、シアンは残りわずか ( L o w )、マゼンタはインク無し ( O u t )、黄は満タン ( F u l l ) となっていることがわかる。なお不図示ではあるが、黒 ( 4 2 )、シアン ( 4 3 )、マゼンタ ( 4 4 )、黄 ( 4 5 ) のインク残量と共にそれぞれのインク型名 ( C I - B、C I - C、C I - M、C I - Y ) や色 ( 黒、シアン、マゼンタ、黄 ) を表示しても良い。

10

## 【 0 0 5 2 】

図 9 は、インクが前後に並んでいるプリンタのインクの情報及び状態、並び方向、順番を列挙 ( E n u m e r a t i o n ) する時のスキーマ及び値を表す図である。図 9 の例では、プリンタ 3 に黒 ( B l a c k ) および一体型のカラーインク ( C o l o r ) が搭載されていて、インクの残量の状態は、黒は満タン ( F u l l )、カラーインクは残りわずか ( L o w ) である。インクの型名がそれぞれ、C I - B、C I - C o l o r、インクの並び方向はプリンタの手前から後ろで、順番は手前から、カラーインク、黒の順で並んでいることを表している。

20

## 【 0 0 5 3 】

図 1 0 は、図 9 のスキーマを使ってステータスアプリケーション 3 0 (ステータスマニタ等) がインクの残量状態をグラフィカルに表示する一例である。1 0 4 に現在のプリンタの状態が文字列で表現される。1 0 3 にはインクの並び方向、並び順および残量がグラフィックを用いて表現されている。インクは前から、カラーインク ( 1 0 1 )、黒 ( 1 0 2 ) の順で並んでおり、残量は、カラーは残りわずか ( L o w )、黒は満タン ( F u l l ) となっていることがわかる。

30

## 【 0 0 5 4 】

図 1 1 は、インクが上下に並んでいるプリンタのインクの情報及び状態、並び方向、順番を列挙 ( E n u m e r a t i o n ) する時のスキーマ及び値を表す図である。図 1 1 の例では、プリンタ 3 に一体型のカラーインク ( C o l o r ) および一体型のフォトインク ( P h o t o ) が搭載されていて、インクの残量の状態は、カラーインクは満タン ( F u l l )、フォトインクは残りわずか ( L o w ) である。インクの型名がそれぞれ、C I - C o l o r、C I - P h o t o、インクの並び方向はプリンタの下から上で、順番は下から、カラーインク、フォトインクの順で並んでいることを表している。

## 【 0 0 5 5 】

図 1 2 は、図 1 1 のスキーマを使ってステータスアプリケーション 3 0 (ステータスマニタ等) がインクの残量状態をグラフィカルに表示する一例である。4 6 に現在のプリンタの状態が文字列で表現される。4 7 にはインクの並び方向、並び順および残量がグラフィックを用いて表現されている。インクは下から、カラーインク ( 4 8 )、フォトインク ( 4 9 ) の順で並んでおり、残量は、カラーインクは満タン ( F u l l )、フォトインクは残りわずか ( L o w ) となっていることがわかる。

40

## 【 0 0 5 6 】

図 1 3 は P C 1 がプリンタ 3 からインク情報を取得する時のデータを表す図である。実際に P C 1 とプリンタ 3 との間で送受信されるデータはバイナリデータであるが、同図では理解しやすくすることを考えて、A S C I I 文字コードでエンコードした後のテキスト

50



データとして表している。同図において、P C 1 が要求コマンドをプリンタ 3 に U S B インタフェース 9 を介して発行すると、インク情報がプリンタ 3 から P C 1 に U S B インタフェース 9 を介して返される。このインク情報は次の内容を表している。

<色>	<型名>	<状態>
黒	C I - B	満タン
シアン	C I - C	残りわずか
マゼンタ	C I - M	無し
黄	C I - Y	満タン

図 1 4 も図 1 3 と同様に P C 1 がプリンタ 3 からインク情報を取得する時のデータを表す図である。図 1 3 と異なり、プリンタ 3 からインクの並び方向、並び順も取得している

10

<色>	<型名>	<状態>	<並び方向>	<並び順>
黒	C I - B	満タン	左から右	1
シアン	C I - C	残りわずか	左から右	2
マゼンタ	C I - M	無し	左から右	3
黄	C I - Y	満タン	左から右	4

このようにプリンタによって、インクの並び方向、並び順の情報まで返せるプリンタ ( 図 1 4 ) 、返せないプリンタ ( 図 1 3 ) の二通りが存在する。

【 0 0 5 7 】

以下、図 1 5 ~ 図 1 9 のフローチャートを用いて、本実施形態の動作について説明する

20

【 0 0 5 8 】

図 1 5 はアプリケーション 3 0 ( ステータスマニタ等 ) がプリンタ 3 のインク情報を取得してインク情報の表示を更新する処理を表すフローチャートである。

【 0 0 5 9 】

ここでは、プリンタ 3 はインクの並び方向、並び順の情報は返せないものとする。同図において、アプリケーション 3 0 がプリンタ 3 のインク情報を取得してインク情報の表示を更新する処理をスタートさせる ( S 1 5 0 1 ) 。次に、アプリケーション 3 0 が図 6 で定義されている ¥ P r i n t e r . I n k I n f o のスキーマを引数として C O M インタフェース I B i d i S p l の A P I 関数 I B i d i S p l : : S e n d R e c v ( ) をコールしてインク情報の取得を開始する ( S 1 5 0 2 ) 。次に、P r i n t i n g a n d P r i n t S p o o l e r I n t e r f a c e s を介して、L M 3 6 がエクスポートしている S e n d R e c v B i d i D a t a F r o m P o r t ( ) 関数がスプーラ ( W i n d o w s ( 登録商標 ) X P O S の公知の機能の一部なので、図や詳細説明は省略 ) からコールされて、この関数における処理が開始される ( S 1 5 0 3 ) 。

30

【 0 0 6 0 】

L M 3 6 の S e n d R e c v B i d i D a t a F r o m P o r t ( ) 関数内において、図 1 3 に示すように、P C 1 から要求コマンドを発行してプリンタ 3 から返されるインク情報を取得する ( S 1 5 0 4 ) 。

【 0 0 6 1 】

さらに自分自身の内部で保持しているインクの並び方向および並び順を取得しする ( S 1 5 0 5 ) 。

40

【 0 0 6 2 】

これら取得したインク情報を S e n d R e c v B i d i D a t a F r o m P o r t ( ) 関数の引数として指定されている ¥ P r i n t e r . I n k I n f o のスキーマの定義に従って変換する ( S 1 5 0 6 ) 。

【 0 0 6 3 】

S e n d R e c v B i d i D a t a F r o m P o r t ( ) 関数の引数としてスキーマの定義に変換されたインク情報を返し、L M 3 6 の S e n d R e c v B i d i D a t a F r o m P o r t ( ) 関数の処理を終了する ( S 1 5 0 7 ) 。

50



## 【 0 0 6 4 】

アプリケーション 3 0 において、I B i d i S p l : : S e n d R e c v ( ) 関数がリターンされ、引数としてスキーマの定義で返されたインク情報を取得する ( S 1 5 0 8 ) 。

## 【 0 0 6 5 】

このインク情報を使ってインク情報を更新し ( S 1 5 0 9 ) 、終了する ( S 1 5 1 0 ) 。

## 【 0 0 6 6 】

通常、アプリケーション 3 0 は、ステップ S 1 5 0 1 からスタートするプリンタ 3 のインク情報を取得してインク情報の表示を更新する処理を、例えば 5 秒に 1 回等、ある一定時間の間隔で繰り返すことで、プリンタ 3 に搭載されているインクの情報及び状態をリアルタイムに表示することができる。

10

## 【 0 0 6 7 】

図 1 6 は、図 1 5 と同様にアプリケーション 3 0 ( ステータスマニタ等 ) がプリンタ 3 のインク情報を取得してインク情報の表示を更新する処理を表すフローチャートである。ここでは、プリンタ 3 はインクの並び方向、並び順の情報を返せるものとする。

## 【 0 0 6 8 】

同図において、アプリケーション 3 0 がプリンタ 3 のインク情報を取得してインク情報の表示を更新する処理をスタートさせる ( S 1 6 0 1 ) 。

## 【 0 0 6 9 】

20

アプリケーション 3 0 が図 6 で定義されている ¥ P r i n t e r . I n k I n f o のスキーマを引数として C O M インタフェース I B i d i S p l の A P I 関数 I B i d i S p l : : S e n d R e c v ( ) をコールしてインク情報の取得を開始する ( S 1 6 0 2 ) 。開始すると、P r i n t i n g a n d P r i n t S p o o l e r I n t e r f a c e s を介して、L M 3 6 がエクスポートしている S e n d R e c v B i d i D a t a F r o m P o r t ( ) 関数がスプーラ ( W i n d o w s ( 登録商標 ) X P O S の公知の機能の一部なので、図や詳細説明は省略 ) からコールされて、この関数における処理が開始される ( S 1 6 0 3 ) 。

## 【 0 0 7 0 】

L M 3 6 の S e n d R e c v B i d i D a t a F r o m P o r t ( ) 関数内において、図 1 4 に示すように、P C 1 から要求コマンドを発行してプリンタ 3 から、インク残量、並び方向、並び順などのインク情報を取得する ( S 1 6 0 4 ) 。

30

## 【 0 0 7 1 】

次に取得したインク情報を S e n d R e c v B i d i D a t a F r o m P o r t ( ) 関数の引数として指定されている ¥ P r i n t e r . I n k I n f o のスキーマの定義に従って変換する ( S 1 6 0 5 ) 。

## 【 0 0 7 2 】

次に S e n d R e c v B i d i D a t a F r o m P o r t ( ) 関数の引数としてスキーマの定義に変換されたインク情報を返し、L M 3 6 の S e n d R e c v B i d i D a t a F r o m P o r t ( ) 関数の処理を終了する ( S 1 6 0 6 ) 。

40

## 【 0 0 7 3 】

アプリケーション 3 0 において、I B i d i S p l : : S e n d R e c v ( ) 関数がリターンされ、引数としてスキーマの定義で返されたインク情報を取得し ( S 1 6 0 7 ) 、このインク情報を使ってインク情報を更新する ( S 1 6 0 8 ) 。次にアプリケーション 3 0 がプリンタ 3 のインク情報を取得してインク情報の表示を更新する処理を終了する ( S 1 6 0 9 ) 。

## 【 0 0 7 4 】

図 1 7 は、図 1 5 、図 1 6 と同様にアプリケーション 3 0 ( ステータスマニタ等 ) がプリンタ 3 のインク情報を取得してインク情報の表示を更新する処理を表すフローチャートである。

50



## 【0075】

ここでは、LM36は、プリンタ3から取得したインク情報(S1704)にインクの並び方向、並び順の情報が含まれているかどうかを判定する(S1705)。含まれていない場合は自分自身の内部のデータベースからインクの並び方向および並び順を取得する(S1706)ことを特徴とする。

## 【0076】

同図において、アプリケーション30がプリンタ3のインク情報を取得してインク情報の表示を更新する処理をスタートさせる(S1701)。次にアプリケーション30が図6で定義されている¥Printer.InkInfoのスキーマを引数としてCOMインタフェースIBidiSplのAPI関数IBidiSpl::SendRecv()をコールしてインク情報の取得を開始する(S1702)。開始すると、Printing and Print Spooler Interfacesを介して、LM36がエクスポートしているSendRecvBidiDataFromPort()関数がスプーラ(Windows(登録商標)XP OSの公知の機能の一部なので、図や詳細説明は省略)からコールされて、この関数における処理が開始される(S1703)。

## 【0077】

LM36のSendRecvBidiDataFromPort()関数内において、図13、図14に示すように、PC1から要求コマンドを発行してプリンタ3から返されるインク情報を取得する(S1704)。次にインク情報にインクの並び方向、並び順の情報が含まれているかどうかを判定する(S1705)。

## 【0078】

プリンタ3から取得した情報が、図13のようにインクの並び方向、並び順情報を含んでいない場合は自分自身の内部で保持しているインクの並び方向および並び順を取得する(S1706)。

## 【0079】

プリンタ3から取得した情報が、図14に示すようにインクの並び方向、並び順情報を含んでいる場合は、S1707に進む。

## 【0080】

S1707で、S1704およびS1706で取得したインク情報をSendRecvBidiDataFromPort()関数の引数として指定されている¥Printer.InkInfoのスキーマの定義に従って変換する。次にSendRecvBidiDataFromPort()関数の引数としてスキーマの定義に変換されたインク情報を返す。そして、LM36のSendRecvBidiDataFromPort()関数の処理を終了する(S1708)。

## 【0081】

アプリケーション30において、IBidiSpl::SendRecv()関数がリターンされ、引数としてスキーマの定義で返されたインク情報を取得しする(S1709)。次にこのインク情報を使ってインク情報を更新する(S1710)。そしてアプリケーション30がプリンタ3のインク情報を取得してインク情報の表示を更新する処理を終了する(S1711)。

## 【0082】

図18、図19は、プリンタ3がインクの並び方向、並び順を取得する2つの方法を表したフローチャートである。

## 【0083】

第1の方法を図18を用いて説明する。第1の方法は、インクタンクの取り付け位置が固定のプリンタが実行する。プリンタ3は、USBインタフェース9を通してPC1から図14で表されるインク情報を要求するコマンドを受け取る(S1801)。ROM16からインクの並び方向、並び順の情報の情報を読み出し(S1802)、図14で表されるコマンドの形式に変換して、インク情報をPC1に返し終了する(S1803)。

## 【0084】



第2の方法を図19を用いて説明する。第2の方法は、インクタンクの取り付け位置が任意であるプリンタが実行する。プリンタ3は、USBインタフェース9を通してPC1から図14で表されるインク情報を要求するコマンドを受け取る(S1901)。次に現在、プリンタ3に実際に装着されているインクの種類および並び順をインクカートリッジまたはインクタンクのメモリから識別情報(色情報)を読み出してインクがどのように並んでいるか調べる(S1902)。そしてインクの並び順情報を生成する(S1903)。ROM16からインクの並び方向の情報を読み出し(S1904)、図14で表されるコマンドの形式に変換して、インク情報をPC1に返し終了する(S1905)。

【0085】

なお、図18, 19のフローに関わるプログラムは、ROM16に記憶されており、CPU15が適時読み出して実行する。

【0086】

< 第2の実施の形態 >

以降、本発明の第2の実施の形態に関して、第1の実施の形態の図1、図2で説明した周辺装置制御システムの例を用いて説明する。尚、以下における具体的な説明は、図1の周辺装置制御システムの例を用いて行う。

【0087】

図20及び図21はアプリケーションがプリンタからインクの情報及び状態を取得する時に使用するスキーマを表す図である。これらのスキーマは、アプリケーション30がPrinting and Print Spooler Interfacesを介してLM36経由でプリンタ3からインクの情報及び状態を取得する時に、Windows(登録商標)XP OSで利用できるCOMインタフェースIBidiSplのAPI関数IBidiSpl::SendRecv()をコールする際に引数として指定される。

【0088】

図20において、InkInfoはインクの情報を表すPropertyであり、スキーマのフルパスの指定は¥Printer.InkInfoである。

【0089】

[Color]は色の情報を表すPropertyである。スキーマのフルパスの指定は¥Printer.InkInfo.[Color]である。設定可能な値は、黒を表すBlack、シアンを表すCyan、マゼンタを表すMagenta、黄を表すYellow、一体型のカラーインクを表すColor、一体型のフォトインクを表すPhotoの何れかである。

【0090】

例えば、黒インクの情報を取得したい場合は、¥Printer.InkInfo.Blackと指定する。

【0091】

Installedは[Color]色のインクが搭載されているか否かを表すValueである。データ型はBoolean、スキーマのフルパスの指定は¥Printer.InkInfo.[Color]:Installedである。設定可能な値は、搭載されていること表すTrue、搭載されていないを表すFalseの何れかである。

【0092】

Levelは[Color]色のインクの残量を表すValueである。データ型はIntである。スキーマのフルパスの指定は¥Printer.InkInfo.[Color]:Levelである。設定可能な値は、インク残量無しを表す0(単位:%)からインク残量満タンを表す100(単位:%)までの間の整数、または、インク残量不明を表す-1である。

【0093】

ModelNameは[Color]色のインクの型名を表すValueである。データ型はStringである。スキーマのフルパスの指定は¥Printer.InkInfo.[Color]:ModelNameである。設定可能な値の例は、同図のExa



mplesのようなASCIIの文字列である。

【0094】

Directionは[Color]色のインクの並び方向を表すValueである。データ型はStringである。スキーマのフルパス指定は¥Printer.InkInfo.[Color]:Directionである。設定可能な値は、原点をプリンタの左下前として、左から右並びを表すLeftRight、手前から後ろの並びを表すFrontBack、下から上の並びを表すBottomTopの何れかである。

【0095】

Orderは[Color]色のインクの原点からの順番(並び順)を表すValueである。原点はプリンタの左下である。データ型はIntである。スキーマのフルパス指定は¥Printer.InkInfo.[Color]:Orderである。設定される値は、例えば、[Color]色のインクの原点からの順番が1番目の場合は、1となる。

【0096】

図21において、Imageは[Color]色のインクの画像を表すValueである。データ型はBitmapである。スキーマのフルパス指定は¥Printer.InkInfo.[Color]:Imageである。設定可能な値はインクの画像を表すビットマップファイル(バイナリデータ)である。このインクの画像を表すビットマップファイル(バイナリデータ)は、インクの残量が満タンの時のインクの画像である。

【0097】

図20及び図21において、Installed、State、ModelName、Direction、Order、Image等のValueはLM36またはプリンタ3からアプリケーション30へ返される値である。このように、これらの図で定義されているスキーマを使って、アプリケーション30はプリンタ3に搭載されているインクの情報及び状態、具体的な例としては、残量、並び方向、順番(並び順)、インクの画像を取得することができる。

【0098】

図22は図20及び図21で定義されているスキーマを使ってインクの情報及び状態を列挙(Enumeration)する時のスキーマ及び値を表す図である。同図において、アプリケーション30(ステータスマニタ等)から¥Printer.InkInfoのスキーマを指定してIBidiSpl::SendRecv()関数をコールすると(Query(Schema))、プリンタ3の全インクの情報及び状態、並び方向、順番(並び順)、インクの画像のスキーマ(Retrieve(Schema))と値(Retrieve(Value))がセットで返される。

【0099】

同図の例では、プリンタ3に黒、黄、マゼンタ、シアンのインクが搭載されていて、インクの残量がそれぞれ、80%、50%、0%(インク残量無し)、100%(インク残量満タン)である。インクの型名がそれぞれ、CI-B、CI-Y、CI-M、CI-Cである。インクの並び方向が左から右方向である。順番(並び順)が左から順に、黒、黄、マゼンタ、シアンの順で並んでいて、インクの画像がそれぞれ、CI-B.bmp、CI-Y.bmp、CI-M.bmp、CI-C.bmpのビットマップファイルであることを表している。

【0100】

このようにインクの情報及び状態を列挙(Enumeration)することにより、インクの情報及び状態、具体的な例としては、残量、並び方向、順番(並び順)、インクの画像を取得することができる。

【0101】

尚、第1の実施の形態の図7の説明と比較すると、インクの並び順が異なっているが、プリンタ3には複数のタイプのインクヘッドを搭載することが可能である。この例では図7の例とは異なるタイプのインクヘッドが搭載されているケースを表している。



## 【 0 1 0 2 】

図 2 3 はプリンタの状態をモニタするステータスマニタを表す図である。ここでは、同図のステータスマニタが、P C 1 にインストールされているアプリケーション 3 0 であると仮定する。

## 【 0 1 0 3 】

同図において、2 0 1 はステータスマニタのメインウィンドウであり、プリンタ 3 ( X Y Z 社製の k m m n というモデル名のプリンタ ) の現在の状態を表している。2 0 2 はプリンタ情報 ( P r i n t e r I n f o r m a t i o n ) 表示部であり、プリンタが O n l i n e ( オンライン ) の状態であることを表している。2 0 3 はインク情報 ( I n k I n f o r m a t i o n ) 表示部であり、プリンタ 3 のインクの情報を表す。2 0 4 はインク残量表示部であり、プリンタ 3 に搭載されている各インクの残量を表示する。

10

## 【 0 1 0 4 】

本発明の特徴としては、任意のアプリケーションにおいて、プリンタ 3 に搭載されているインクの状態、例えば、インクの形状、色、大きさ、並び方向、並び順、残量等を、細部までグラフィカルに、かつ、ビジュアル的にリアルに表現することで、ユーザの操作性を大幅に向上させている。同図に示す通り、プリンタ 3 には大容量の黒 ( B l a c k )、標準容量の黄 ( Y e l l o w )、標準容量のマゼンタ ( M a g e n t a )、標準容量のシアン ( C y a n ) の 4 色のインクが左から右方向にこの順で装着されている。さらに、それぞれの色のインクの残量が 8 0 %、5 0 %、0 % ( インク残量無し )、1 0 0 % ( インク残量満タン ) であることをグラフィカルに、かつ、ビジュアル的にリアルに表している。

20

## 【 0 1 0 5 】

図 2 4 は図 2 2 における各インクの画像のビットマップファイルの表示内容を表す図である。同図において、2 0 5 はインクの残量が満タンの時の黒インクの画像を表すビットマップファイル ( C I - B . b m p ) の表示内容がある。2 0 6 はインクの残量が満タンの時の黄インクの画像を表すビットマップファイル ( C I - Y . b m p ) の表示内容である。2 0 7 はインクの残量が満タンの時のマゼンタインクの画像を表すビットマップファイル ( C I - M . b m p ) の表示内容である。2 0 8 はインクの残量が満タンの時のシアンインクの画像を表すビットマップファイル ( C I - C . b m p ) の表示内容である。

## 【 0 1 0 6 】

30

図 2 5 は各インクの残量の画像のビットマップファイルの表示内容を表す図である。同図において、2 0 9 はインクの残量が 8 0 % の黒インクの画像を表すビットマップファイルの表示内容である。2 1 0 はインクの残量が 5 0 % の黄インクの画像を表すビットマップファイルの表示内容である。2 1 1 はインクの残量が 0 % のマゼンタインクの画像を表すビットマップファイルの表示内容である。2 1 2 はインクの残量が 1 0 0 % のシアンインクの画像を表すビットマップファイルの表示内容である。図中の破線で記されている部分は、インクの残量が満タンの時の外形を表すものであり、実際には表示されない部分である。

## 【 0 1 0 7 】

図 2 6 は図 2 0 及び図 2 1 で定義されているスキーマを使ってインクの情報及び状態を列挙 ( E n u m e r a t i o n ) する時のスキーマ及び値を表す図である。同図において、アプリケーション 3 0 ( ステータスマニタ等 ) から ¥ P r i n t e r . I n k I n f o のスキーマを指定して I B i d i S p l : : S e n d R e c v ( ) 関数をコールすると ( Q u e r y ( S c h e m a ) )、プリンタ 3 の全インクの情報及び状態、並び方向、順番 ( ならば順 )、インクの画像のスキーマ ( R e t r i e v e ( S c h e m a ) ) と値 ( R e t r i e v e ( V a l u e ) ) がセットで返される。

40

## 【 0 1 0 8 】

同図の例では、プリンタ 3 に、黄、マゼンタ、シアンの 3 色のインクが一体となっているカラーインクと、写真印刷用の特殊な 3 色のインクが一体となっているのフォトインクが搭載されていることを表す。さらに、インクの残量がそれぞれ、4 0 %、8 0 % で

50



あり、インクの型名がそれぞれ、C I - C o l o r、C I - P h o t o、インクの並び方向が下から上方向で、順番（並び順）が下から順に、カラー、フォトの順で並んでいることを表す。またインクの画像がそれぞれ、C I - C o l o r . b m p、C I - P h o t o . b m pのビットマップファイルであることを表している。

【 0 1 0 9 】

このようにインクの情報及び状態を列挙（E n u m e r a t i o n）することにより、インクの情報及び状態、具体的な例としては、残量、並び方向、順番（並び順）、インクの画像を取得することができる。

【 0 1 1 0 】

尚、第1の実施の形態の図7の説明、第2の実施の形態の図22と比較すると、インクの種類、並び方向、並び順が異なっているが、プリンタ3には複数のタイプのインクヘッドを搭載することが可能である。この例では図7の例とは異なるタイプのインクヘッドが搭載されているケースを表している。

【 0 1 1 1 】

図27はプリンタの状態をモニタするステータスマニタを表す図である。ここでは、同図のステータスマニタが、P C 1にインストールされているアプリケーション30であると仮定する。

【 0 1 1 2 】

同図において、201、202、203は図23で前述したものと同一ものである。

【 0 1 1 3 】

216はカラーインクの残量表示部であり、プリンタ3に搭載されているカラーインクの残量を表示する。

【 0 1 1 4 】

217はフォトインクの残量表示部であり、プリンタ3に搭載されているフォトインクの残量を表示する。

【 0 1 1 5 】

本発明の特徴の一つとして、任意のアプリケーションにおいて、プリンタ3に搭載されているインクの状態を、細部までグラフィカルに、かつ、ビジュアル的にリアルに表現することができる。例えば、インクの形状、色、大きさ、並び方向、並び順、残量等を、細部までグラフィカルに、かつ、ビジュアル的にリアルに表現することで、ユーザの操作性を大幅に向上させている。

【 0 1 1 6 】

同図に示す通り、プリンタ3には、黄、マゼンタ、シアンの3色のインクが一体となっているカラーインク（C o l o r）、写真印刷用の特殊な3色のインクが一体となっているのフォトインク（P h o t o）の2種類の一体型インクが下から上方向にこの順で装着されていることを表し。それぞれの色のインクの残量が40%、80%であることをグラフィカルに、かつ、ビジュアル的にリアルに表している。

【 0 1 1 7 】

図28は図26における各インクの画像のビットマップファイルの表示内容を表す図である。同図において、218はインクの残量が満タンの時のカラーインクの画像を表すビットマップファイル（C I - C o l o r . b m p）の表示内容である。219はインクの残量が満タンの時のフォトインクの画像を表すビットマップファイル（C I - P h o t o . b m p）の表示内容である。

【 0 1 1 8 】

図29は各インクの残量の画像のビットマップファイルの表示内容を表す図である。同図において、220はインクの残量が40%のカラーインクの画像を表すビットマップファイルの表示内容、221はインクの残量が80%のフォトインクの画像を表すビットマップファイルの表示内容である。図中の破線で記されている部分は、インクの残量が満タンの時の外形を表すものであり、実際には表示されない部分である。

【 0 1 1 9 】

10

20

30

40

50



図20及び図30はアプリケーションプリンタからインクの情報及び状態を取得する時に使用するスキーマを表す図である。これらのスキーマは、アプリケーション30がPrinting and Print Spooler Interfacesを介してLM36経由でプリンタ3からインクの情報及び状態を取得する時に、Windows（登録商標）XP OSで利用できるCOMインタフェースIBidSplのAPI関数IBidSpl::SendRecv（）をコールする際に引数として指定される。尚、図20は前述したのでここではその説明を省略する。

#### 【0120】

図30において、ImageDataは[Color]色のインクの画像情報を表すPropertyであり、スキーマのフルパスの指定は¥Printer.InkInfo.[Color].ImageDataである。

Redは[Color]色のインクの画像の赤色成分を表すValueである。データ型はIntである。スキーマのフルパス指定は¥Printer.InkInfo.[Color].ImageData:Redである。設定可能な値は0～255の整数である。

Greenは[Color]色のインクの画像の緑色成分を表すValueである。データ型はIntである。スキーマのフルパス指定は¥Printer.InkInfo.[Color].ImageData:Greenである。設定可能な値は0～255の整数である。

Blueは[Color]色のインクの画像の青色成分を表すValueである。データ型はIntである。スキーマのフルパス指定は¥Printer.InkInfo.[Color].ImageData:Blueである。設定可能な値は0～255の整数である。

Widthは[Color]色のインクの画像の幅を表すValueである。データ型はIntである。スキーマのフルパス指定は¥Printer.InkInfo.[Color].ImageData:Widthである。設定可能な値は0～100（単位：ピクセル）の整数である。

Depthは[Color]色のインクの画像の奥行きを表すValueである。データ型はIntである。スキーマのフルパス指定は¥Printer.InkInfo.[Color].ImageData:Depthである。設定可能な値は0～100（単位：ピクセル）の整数である。

Heightは[Color]色のインクの画像の高さを表すValueである。データ型はIntである。スキーマのフルパス指定は¥Printer.InkInfo.[Color].ImageData:Heightである。設定可能な値は0～100（単位：ピクセル）の整数である。

#### 【0121】

このインクの画像情報は、インクの残量が満タンの時のインクの画像情報である。

#### 【0122】

図20及び図30において、Installed、State、ModelName、Direction、Order、Red、Green、Blue、Width、Depth、Height等のValueはLM36またはプリンタ3からアプリケーション30へ返される値である。このように、これらの図で定義されているスキーマを使って、アプリケーション30はプリンタ3に搭載されているインクの情報及び状態、具体的な例としては、残量、並び方向、順番（並び順）、インクの画像情報を取得することができる。

#### 【0123】

図21と図30を比較した場合、図21では[Color]色のインクの画像をImageのスキーマを使って、DataTypeがBitmapであるビットマップファイル（バイナリデータ）で定義している。それに対し、図30では[Color]色のインクの画像情報をImageDataのRed、Green、Blue、Width、Depth、Heightの各スキーマを使って、DataTypeがIntである整数で

10

20

30

40

50



定義している。

【0124】

図21の長所としては、任意のアプリケーションにおいて、プリンタ3に搭載されているインクの状態を、細部までグラフィカルに、かつ、ビジュアル的に、最もリアルに表現することができる。例えば、インクの形状、色、大きさ、並び方向、並び順、残量等を、細部までグラフィカルに、かつ、ビジュアル的に、最もリアルに表現することができる点である。

【0125】

短所としては、[Color]色のインクの画像を表すImageのスキーマのData TypeがBitmapであり、実際にLM36またはプリンタ3からアプリケーション30に送られるデータがビットマップファイルとなるので、データ量が多い点である。

10

【0126】

図30の長所としては、[Color]色のインクの画像情報を表すImageDataのRed、Green、Blue、Width、Depth、Heightの各スキーマのData TypeがIntである。実際にLM36またはプリンタ3からアプリケーション30に送られるデータがRed、Green、Blue、Width、Depth、Heightを表す6つの整数であるので、データ量が非常に少ない点である。

【0127】

また、本発明の特徴の一つとして、Width、Depth、Heightの定義を使うことで3次元の表示の制御を容易に行うことができる点も長所として挙げられる。

20

【0128】

短所としては、インクの形状、色、大きさ、並び方向、並び順、残量等を、細部までグラフィカルに、かつ、ビジュアル的に表現しようとする場合、図21のインクの画像を表すImageのスキーマを使った場合に比べて、表現が劣る点である。

【0129】

図21のImage、図30のImageDataの両者それぞれに一長一短があるので、どちらのスキーマを使うかは、使用する周辺装置制御システムの環境に依存するので、その環境に合わせて最適な方を選択するシステムを構築するのが理想である。

【0130】

30

例えば、図1に示す周辺装置制御システムのように、LM36またはプリンタ3からアプリケーション30へのデータ転送速度が速いシステムにおいては、インクの形状、色、大きさ、並び方向、並び順、残量等を、細部までグラフィカルに、かつ、ビジュアル的に、最もリアルに表現することができる図21のImageのスキーマを使う方がよいと考えられる。

【0131】

図31は図20及び図30で定義されているスキーマを使ってインクの情報及び状態を列挙(Enumeration)する時のスキーマ及び値を表す図である。同図において、アプリケーション30(ステータスマニタ等)から¥Printer.InkInfoのスキーマを指定してIBidiSpl::SendRecv()関数をコールする。コールすると(Query(Schema))、プリンタ3の全インクの情報及び状態、並び方向、順番(ならび順)、インクの画像情報のスキーマ(Retrieve(Schema))と値(Retrieve(Value))がセットで返される。

40

【0132】

同図の例では、プリンタ3に黒、黄、マゼンタ、シアンのインクが搭載されていて、インクの残量がそれぞれ、80%、50%、0%(インク残量無し)、100%(インク残量満タン)であることを表している。そしてインクの型名がそれぞれ、CI-B、CI-Y、CI-M、CI-C、インクの並び方向が左から右方向であることを表している。また順番(並び順)が左から順に、黒、黄、マゼンタ、シアンの順で並んでいることを表している。そしてインクの画像情報が同図のImageDataのPropertyのRed

50



、Green、Blue、Width、Depth、Heightの各値(Retrieve( Value ))で図示されている内容であることを表している。

【0133】

このようにインクの情報及び状態を列挙(Enumeration)することにより、インクの情報及び状態、具体的な例としては、残量、並び方向、順番(並び順)、インクの画像情報を取得することができる。

【0134】

尚、第1の実施の形態の図7の説明と比較すると、インクの並び順が異なっているが、プリンタ3には複数のタイプのインクヘッドを搭載することが可能である。この例では図7の例とは異なるタイプのインクヘッドが搭載されているケースを表している。

10

【0135】

図32はインク情報のデータベースを表す図である。

【0136】

同図に示す通り、各インクの色(Ink Color)、すなわち、黒(Black)、黄(Yellow)、マゼンタ(Magenta)、シアン(Cyan)に対して、インクの並び方向(Direction)、インクの並び順(Order)、インクの画像(Image)、インクの画像情報(Image Data)が、このデータベースに登録されている。

【0137】

Image Dataには、図30に示すRed、Green、Blue、Width、Depth、Heightのそれぞれの値(Value)が順に格納されている。このデータベースはLMリソース6に格納されている。

20

【0138】

図33はインク情報のデータベースを表す図である。

【0139】

同図に示す通り、各インクの色(Ink Color)、すなわち、黄、マゼンタ、シアンの3色のインクが一体となっているカラーインク(Color)、写真印刷用の特殊な3色のインクが一体となっているのフォトインク(Photo)に対して、インクの並び方向(Direction)、インクの並び順(Order)、インクの画像(Image)が、このデータベースに登録されている。

30

【0140】

このデータベースはLMリソース6に格納されている。

【0141】

図34はPCがプリンタからインク情報を取得する時のデータを表す図である。実際にPC1とプリンタ3との間で送受信されるデータはバイナリデータであるが、同図では理解しやすくすることを考えて、ASCII文字コードでエンコードした後のテキストデータとして表している。同図において、PC1が要求コマンドをプリンタ3にUSBインタフェース9を介して発行すると、インク情報がプリンタ3からPC1にUSBインタフェース9を介して返される。このインク情報は次の内容を表している。

<色>	<型名>	<残量>
黒	CI - B	80 %
黄	CI - Y	50 %
マゼンタ	CI - M	0 %
シアン	CI - C	100 %

40

図35はPCがプリンタからインク情報を取得する時のデータを表す図である。実際にPC1とプリンタ3との間で送受信されるデータはバイナリデータであるが、同図では理解しやすくすることを考えて、ASCII文字コードでエンコードした後のテキストデータとして表している。同図において、PC1が要求コマンドをプリンタ3にUSBインタフェース9を介して発行すると、インク情報がプリンタ3からPC1にUSBインタフェース9を介して返される。このインク情報は次の内容を表している。

50



【 0 1 4 2 】

【表 1】

＜色＞	＜型名＞	＜残量＞	＜並び方向＞	＜並び順＞	＜画像＞
黒	CI-B	80%	左から右	1	CI-B. bmpのビットマップファイル
黄	CI-Y	50%	左から右	2	CI-Y. bmpのビットマップファイル
マゼンタ	CI-M	0%	左から右	3	CI-M. bmpのビットマップファイル
シアン	CI-C	100%	左から右	4	CI-C. bmpのビットマップファイル

【 0 1 4 3 】

図 3 6 は P C がプリンタからインク情報を取得する時のデータを表す図である。実際に P C 1 とプリンタ 3 との間で送受信されるデータはバイナリデータであるが、同図では理解しやすくすることを考えて、A S C I I 文字コードでエンコードした後のテキストデータとして表している。同図において、P C 1 が要求コマンドをプリンタ 3 に U S B インタフェース 9 を介して発行すると、インク情報がプリンタ 3 から P C 1 に U S B インタフェース 9 を介して返される。このインク情報は次の内容を表している。

【 0 1 4 4 】

10

20

30

40



【表 2】

＜色＞	＜型名＞	＜残量＞	＜並び方向＞	＜並び順＞	＜画像情報＞
黒	CI-B	80%	左から右	1	0 0 0 65 100 75
黄	CI-Y	50%	左から右	2	255 255 0 25 100 75
マゼンタ	CI-M	0%	左から右	3	255 0 255 25 100 75
シアン	CI-C	100%	左から右	4	0 255 255 25 100 75

## 【0145】

尚、画像情報には、図30に示すRed、Green、Blue、Width、Depth、Heightのそれぞれに相当する値(Value)が順に格納されている。

## 【0146】

図37はPCがプリンタからインク情報を取得する時のデータを表す図である。実際にPC1とプリンタ3との間で送受信されるデータはバイナリデータであるが、同図では理解しやすくすることを考えて、ASCII文字コードでエンコードした後のテキストデータとして表している。同図において、PC1が要求コマンドをプリンタ3にUSBインタフェース9を介して発行すると、インク情報がプリンタ3からPC1にUSBインタフェ

10

20

30

40

50



ース 9 を介して返される。このインク情報は次の内容を表している。

<色>	<型名>	<残量>
カラー	C I - C o l o r	4 0 %
フォト	C I - P h o t o	8 0 %

図 3 8 は P C がプリンタからインク情報を取得する時のデータを表す図である。実際に P C 1 とプリンタ 3 との間で送受信されるデータはバイナリデータであるが、同図では理解しやすくすることを考えて、A S C I I 文字コードでエンコードした後のテキストデータとして表している。同図において、P C 1 が要求コマンドをプリンタ 3 に U S B インタフェース 9 を介して発行すると、インク情報がプリンタ 3 から P C 1 に U S B インタフェース 9 を介して返される。このインク情報は次の内容を表している。

【 0 1 4 7 】



【表 3】

＜色＞	＜型名＞	＜残量＞	＜並び方向＞	＜並び順＞	＜画像＞
カラー	CI-Color	40%	下から上	1	CI-Color. bmpのビットマップファイル
フォト	CI-Photo	80%	下から上	2	CI-Photo. bmpのビットマップファイル

【 0 1 4 8 】

図 3 4 ~ 図 3 8 のデータは、プリンタ 3 が搭載されているインクの情報を取得し、取得したデータをこれらの図に示すような所定のデータフォーマットに変換するという一連の処理によって行われる。プリンタ 3 は、P C 1 からの要求コマンドを受信すると、前述した一連の処理によりインク情報を表すデータを作成し、P C 1 へそのデータを送る。プリンタ 3 が行うこのような処理は、一般的なプリンタ等の周辺装置において行われている公知の処理であるので、ここでは図面を用いた詳細説明を省略する。



## 【0149】

以下、図39～41のフローチャートを用いて、第2の実施の形態の動作について説明する。

## 【0150】

図39はアプリケーションがプリンタのインク情報を取得してインク情報の表示を更新する処理を表すフローチャートである。

## 【0151】

同図において、アプリケーション30がプリンタ3のインク情報を取得してインク情報の表示を更新する処理をスタートさせる(S3901)。スタートさせると、アプリケーション30が図20及び図21、または、図20及び図30で定義されている¥Printer.InkInfoのスキーマを引数としてCOMインタフェースIBidisp1のAPI関数IBidisp1::SendRecv()をコールしてインク情報の取得を開始する(S3902)。開始すると、Printing and PrintSpooler Interfacesを介して、LM36がエクスポートしているSendRecvBidiDataFromPort()関数がスプーラ(Windows(登録商標)XP OSの公知の機能の一部なので、図や詳細説明は省略)からコールされて、この関数における処理が開始される(S3903)。

10

## 【0152】

LM36のSendRecvBidiDataFromPort()関数内において、図34～図38に示すように、PC1から要求コマンドを発行してプリンタ3から返されるインク情報を取得する(S3904)。次にこの取得したインク情報をSendRecvBidiDataFromPort()関数の引数として指定されている¥Printer.InkInfoのスキーマの定義に従って変換(図40で後述)する(S3905)。次にSendRecvBidiDataFromPort()関数の引数として図22または図31に示すようなスキーマの定義に変換されたインク情報を返す。そして、LM36のSendRecvBidiDataFromPort()関数の処理を終了する(S3906)。

20

## 【0153】

アプリケーション30において、IBidisp1::SendRecv()関数がリターンされ、引数として図22または図31に示すようなスキーマの定義で返されたインク情報を取得する(S3907)。次に、インク情報表示用のデータを作成(図41で後述)する(S3908)。次に、このデータを使ってインク情報を更新し(S3909)、アプリケーション30がプリンタ3のインク情報を取得してインク情報の表示を更新する処理を終了する(S3910)。

30

## 【0154】

通常、アプリケーション30は、ステップS3901からスタートするプリンタ3のインク情報を取得してインク情報の表示を更新する処理を、例えば5秒に1回等、ある一定時間の間隔で繰り返すことで、プリンタ3に搭載されているインクの情報及び状態をリアルタイムに表示することができる。

## 【0155】

このように、上記一連の処理を行うことで、アプリケーション30(図23または図27に示すようなステータスマニタ等の任意のアプリケーション)において、プリンタ3に搭載されているインクの状態をグラフィカルに、かつ、ビジュアル的にリアルに表現することができる。例えば、インクの形状、色、大きさ、並び方向、並び順、残量等を、細部までグラフィカルに、かつ、ビジュアル的にリアルに表現することができる。

40

## 【0156】

図40はLMにおけるインク情報をスキーマの定義に変換する処理を表すフローチャートである。

## 【0157】

同図において、LM36におけるインク情報をスキーマの定義に変換する処理が開始さ

50



れると (S 4 0 0 1)、図 3 9 のステップ S 3 9 0 4 で取得したプリンタ 3 から返されるインク情報からインクの色を取得する (S 4 0 0 2)。次に、色が有る (見つかった) 場合 (S 4 0 0 3)、その色をステップ S 4 0 1 9 におけるリターン時の戻り値格納用バッファ (メモリ) にセットする (S 4 0 0 4)。次に、その色のインクの型名を取得してその型名をステップ S 4 0 1 9 におけるリターン時の戻り値格納用バッファ (メモリ) にセットする (S 4 0 0 5)。次に、その色のインクの残量を取得してその残量をステップ S 4 0 1 9 におけるリターン時の戻り値格納用バッファ (メモリ) にセットする (S 4 0 0 6)。次に、その色のインクの並び方向を取得し (S 4 0 0 7)、並び方向が有る (見つかった) 場合 (S 4 0 0 8)、その色のインクの並び方向をステップ S 4 0 1 9 におけるリターン時の戻り値格納用バッファ (メモリ) にセットする (S 4 0 0 9)。次に、その色のインクの並び順を取得してその並び順をステップ S 4 0 1 9 におけるリターン時の戻り値格納用バッファ (メモリ) にセットする (S 4 0 1 0)。次に、その色のインクの画像を取得する (S 4 0 1 1)。次に、画像が有る (見つかった) 場合 (S 4 0 1 2)、その色のインクの画像をステップ S 4 0 1 9 におけるリターン時の戻り値格納用バッファ (メモリ) にセットし (S 4 0 1 3)、ステップ S 4 0 0 2 に戻る。ステップ S 4 0 1 2 において、画像が無い (見つからなかった) 場合、その色のインクの画像情報を取得する (S 4 0 1 4)。画像情報が有る (見つかった) 場合 (S 4 0 1 5)、その色のインクの画像情報をステップ S 4 0 1 9 におけるリターン時の戻り値格納用バッファ (メモリ) にセットし (S 4 0 1 6)、ステップ S 4 0 0 2 に戻る。

【 0 1 5 8 】

ステップ S 4 0 1 5 において、画像情報が無い (見つからなかった) 場合、図 3 2 または図 3 3 のインク情報のデータベースを使ってその色のインクの画像またはインクの画像情報を取得する。そして、その画像情報をステップ S 4 0 1 9 におけるリターン時の戻り値格納用バッファ (メモリ) にセットし (S 4 0 1 7)、ステップ S 4 0 0 2 に戻る。

【 0 1 5 9 】

ステップ S 4 0 1 7 において、インク情報のデータベース内にインクの画像とインクの画像情報の両方が存在する場合、その時の周辺装置制御システムの環境に最適な方を選択するような制御を行う。

【 0 1 6 0 】

この例では図 1 に示す周辺装置制御システムであるので、LM 3 6 またはプリンタ 3 からアプリケーション 3 0 へのデータ転送速度が速いので、インクの画像とインクの画像情報の両方が存在する場合はインクの画像を優先してセットする。

【 0 1 6 1 】

ステップ S 4 0 0 8 において、並び方向が無い (見つからなかった) 場合、図 3 2 または図 3 3 のインク情報のデータベースを使ってその色のインクの並び方向と並び順を取得する。そして、その並び方向と並び順をステップ S 4 0 1 9 におけるリターン時の戻り値格納用バッファ (メモリ) にセットし (S 4 0 1 8)、ステップ S 4 0 0 2 に戻る。

【 0 1 6 2 】

ステップ S 4 0 0 3 において、色が無い (見つからなかった) 場合、全ての色のインクに対する処理が完了したことになる。よって、戻り値格納用バッファにセットされている色 (図 2 0 の [ C o l o r ])、及び、その色のインクの型名 (図 2 0 の M o d e l N a m e)、残量 (図 2 0 の L e v e l)、並び方向 (図 2 0 の D i r e c t i o n)、並び順 (図 2 0 の O r d e r)、画像 (図 2 1 の I m a g e)、画像情報 (図 3 0 の I m a g e D a t a の R e d、G r e e n、B l u e、W i d t h、D e p t h、H e i g h t) を返して、リターンする (S 4 0 1 9)。

【 0 1 6 3 】

図 4 1 はアプリケーションにおけるインク情報表示用データの作成処理を表すフローチャートである。

【 0 1 6 4 】

同図において、アプリケーション 3 0 におけるインク情報表示用データの作成処理が開

10

20

30

40

50



始される ( S 4 1 0 1 )。開始されると、図 3 9 のステップ S 3 9 0 7 で取得した I B i d i S p l : : S e n d R e c v ( ) 関数の引数として図 2 2 または図 3 1 に示すようなスキーマの定義で返されたインク情報からインクの色を取得する ( S 4 1 0 2 )。次に、色が有る ( 見つかった ) 場合 ( S 4 1 0 3 )、その色をステップ S 4 1 1 4 におけるリターン時の戻り値格納用バッファ ( メモリ ) にセットする ( S 4 1 0 4 )。次に、その色のインクの残量を取得してその残量をステップ S 4 1 1 4 におけるリターン時の戻り値格納用バッファ ( メモリ ) にセットする ( S 4 1 0 5 )。次に、その色のインクの並び方向を取得してその並び方向をステップ S 4 1 1 4 におけるリターン時の戻り値格納用バッファ ( メモリ ) にセットする ( S 4 1 0 6 )。次に、その色のインクの並び順を取得してその並び順をステップ S 4 1 1 4 におけるリターン時の戻り値格納用バッファ ( メモリ ) にセ  
 ットする ( S 4 1 0 7 )。次に、その色のインクの画像を取得する ( S 4 1 0 8 )。画像が有る ( 見つかった ) 場合 ( S 4 1 0 9 )、その色のインクの画像とステップ S 4 1 0 5 で取得してバッファ ( メモリ ) にセットされているその色のインクの残量から、その色のインクの画像を加工してその色のインクの残量の画像を生成する ( S 4 1 1 0 )。次に、その色のインクの残量の画像をステップ S 4 1 1 4 におけるリターン時の戻り値格納用バッファ ( メモリ ) にセットし ( S 4 1 1 1 )、ステップ S 4 1 0 2 に戻る。

10

【 0 1 6 5 】

ステップ S 4 1 1 0 において、その色のインクの残量の画像を生成する際は、図 2 4 または図 2 8 に示すような各インクの画像のビットマップファイルに対して、次のような計算式によりその色のインクの画像の高さを変更することで、図 2 5 または図 2 9 に示す各  
 インクの残量の画像を得ることができる。

20

[ 計算式 ]

( その色のインクの画像の高さ ) × ( その色のインクの残量 ( 図 2 0 の L e v e l ) ) / 1 0 0 . . . ( 1 )

ステップ S 4 1 0 9 において、画像が無い ( 見つからなかった ) 場合、その色のインクの画像情報を取得する ( S 4 1 1 2 )。次に、その画像情報とステップ S 4 1 0 5 で取得してバッファ ( メモリ ) にセットされているその色のインクの残量から、その色のインクの画像情報を加工してその色のインクの残量の画像を生成する ( S 4 1 1 3 )。そして、ステップ S 4 1 1 1 に進む。

【 0 1 6 6 】

30

ステップ S 4 1 1 3 において、その色のインクの残量の画像を生成する際は、まず初めに図 3 0 に示すインクの画像情報 ( I m a g e D a t a ) の R e d、G r e e n、B l u e、W i d t h、D e p t h、H e i g h t の各値 ( V a l u e ) をもとに、具体的には図 3 1 に示すような値 ( V a l u e ) をもとに、W i n d o w s ( 登録商標 ) G D I の関数を使ってインクの残量が満タンの時のインクの画像を生成する。

【 0 1 6 7 】

ここで P C 1 内のディスプレイで表現可能な 2 次元のユーザインタフェースを考え、D e p t h の値を無視してインクの画像を生成すると、図 2 4 または図 2 8 に示すような各インクの画像とほぼ同等な各インクの画像が得られる。

【 0 1 6 8 】

40

ここでは D e p t h の値を無視するケースで説明する。生成されたインクの画像に対して、次のような計算式によりその色のインクの画像の高さを変更することで、図 2 5 または図 2 9 に示すような各インクの残量の画像とほぼ同等な各インクの残量の画像を得ることができる。

[ 計算式 ]

( その色のインクの画像の高さ ) × ( その色のインクの残量 ( 図 2 0 の L e v e l ) ) / 1 0 0 . . . ( 2 )

ステップ S 4 1 0 3 において、色が無い ( 見つからなかった ) 場合、全ての色のインクに対する処理が完了した。よって、戻り値格納用バッファにセットされている色 ( 図 2 0 の [ C o l o r ] )、及び、その色のインクの残量 ( 図 2 0 の L e v e l )、並び方向 (

50



図20のDirection)、並び順(図20のOrder)、画像(図21のImage)、画像情報(図30のImageDataのRed、Green、Blue、Width、Depth、Height)、残量の画像(計算式(1)または(2)により高さを変更することで得られる画像)を返して、リターンする(S4114)。

【0169】

上記例では、アプリケーション30において、[Color]色のインクの画像の奥行きを表すDepth(図31の例ではDepth=100)を無視して処理を行った。しかしながら、これは、この例がWindows(登録商標)XP上で動作している周辺装置制御システムの例であり、PC1内のディスプレイで表現可能な2次元のユーザインタフェースを想定している為である。

10

【0170】

もし3次元のユーザインタフェースを想定する場合は、[Color]色のインクの画像の奥行きを表すDepthを無視せずにこれを使って制御することで、3次元のユーザインタフェースを有効に活用することができる。

【0171】

<第3の実施の形態>

以降、本発明の第3の実施の形態に関して、第1の実施の形態の図1、図2で説明した周辺装置制御システムの例を用いて説明する。尚、以下における具体的な説明は、図1の周辺装置制御システムの例を用いて行う。

【0172】

20

図43はアプリケーション30(ステータスマニタ等)がプリンタ3からインクの情報及び状態を取得する時に使用するスキーマを表す図である。

【0173】

これらのスキーマは、アプリケーション30がPrinting and Print Spooler Interfacesを介してLM36経由でプリンタ3からインクの情報及び状態を取得する時に、Windows(登録商標)XP OSで利用できるCOMインタフェースIBidiSplのAPI関数IBidiSpl::SendRecv()をコールする際に引数として指定される。

【0174】

同図において、InkInfoはインクの情報を表すPropertyであり、スキーマのフルパスの指定は¥Printer.InkInfoである。

30

【0175】

[Color]は色の情報を表すPropertyである。スキーマのフルパスの指定は¥Printer.InkInfo.[Color]である。設定可能な値は、黒を表すBlack、シアンを表すCyan、マゼンタを表すMagenta、黄を表すYellow、一体型のフォトインクを表すPhoto、一体型のカラーインクを表すColorの何れかである。

【0176】

例えば、黒インクの情報を取得したい場合は、¥Printer.InkInfo.Blackと指定する。

40

【0177】

Installedは[Color]色のインクが搭載されているか否かを表すValueである。データ型はBooleanである。スキーマのフルパスの指定は¥Printer.InkInfo.[Color]:Installedである。設定可能な値は、搭載されていること表すTrue、搭載されていないを表すFalseの何れかである。

【0178】

Stateは[Color]色のインクの残量の状態を表すValueである。データ型はStringである。スキーマのフルパスの指定は¥Printer.InkInfo.[Color]:Stateである。設定可能な値は、インク満タンを表すFull

50



、インク残りわずかを表すLow、インク無しを表すOut、インク残量不明を表すUnknownの何れかである。

【0179】

ModelNameは[Color]色のインクの型名を表すValueである。データ型はStringである。スキーマのフルパスの指定は¥Printer.InkInfo.[Color]:ModelNameである。設定可能な値の例は、同図のExamplesのようなASCIIの文字列である。

【0180】

Directionは[Color]色のインクの並び方向を表すValueである。データ型はStringである。スキーマのフルパス指定は¥Printer.InkInfo.[Color]:Directionである。設定可能な値は、原点をプリンタの左下として、左から右並びを表すLeftRight、手前から後ろの並びを表すFrontBack、下から上の並びを表すBottomTopの何れかである。

10

【0181】

Orderは[Color]色のインクの原点からの順番を表すValueである。原点はプリンタの左下である。データ型はIntである。スキーマのフルパス指定は¥Printer.InkInfo.[Color]:Orderである。設定される値は、例えば、[Color]色のインクの原点からの順番が1番目の場合は、1となる。

【0182】

InsertDirectionは[Color]色のインクの挿入方向を表すValueである。データ型はStringである。スキーマのフルパス指定は¥Printer.InkInfo.[Color]:InsertDirectionである。設定可能な値は、原点をプリンタの左下として、左から右の方向を表すLeftToRight、右から左の方向を表すRightToLeft、手前から後ろの方向を表すFrontToBack、後ろから手前の方向を表すBackToFront、下から上の方向を表すBottomToTop、上から下の方向を表すTopToBottomの何れかである。

20

【0183】

Installed、State、ModelName、Direction、Order、InsertDirection等のValueはLM36またはプリンタ3からアプリケーション30へ返される値である。

30

【0184】

このように、同図で定義されているスキーマを使って、アプリケーション30はプリンタ3に搭載されているインクの情報及び状態、並び方向、並び順、インクの挿入方向を取得することができる。

【0185】

図44は図43で定義されているスキーマを使ってインクの情報及び状態、並び方向、順番、挿入方向を列挙(Enumeration)する時のスキーマ及び値を表す図である。同図において、アプリケーション30(ステータスマニタ等)から¥Printer.InkInfoのスキーマを指定してIBidiSpl::SendRecv()関数をコールすると(Query(Schema))、プリンタ3の全インクの情報及び状態、並び方向、並び順、挿入方向のスキーマ(Retrieve(Schema))と値(Retrieve(Value))がセットで返される。

40

【0186】

同図の例では、プリンタ3に黒、一体型のカラーインクが搭載されていて、インクの残量の状態がいずれも、インク無し(Out)であることを表している。さらに、インクの型名がそれぞれ、CI-B、CI-Color、インクの並び方向は左から右方向で、順番は左から順に、黒、カラー、の順で並んでいて、インクの挿入方向が、下から上の方向であることを表している。

【0187】

50



図45は、図43のスキーマを使ってステータスアプリケーション30（ステータスマニタ等）がインク無しエラーをグラフィカルに表示する一例である。300にインク無しエラーになっているインクが文字列で表現される。

【0188】

301にはインクの並び方向、並び順および残量およびインクの挿入方向がグラフィックを用いて表現されている。

【0189】

インクは左から、黒（302）、カラー（303）の順で並んでおり、残量は、黒／カラーともにインク無し（Out）、インクの挿入方向は黒／カラーともに下から上方向となっていることがわかる。

【0190】

図46は、インクの挿入方向が横向きなプリンタのインクの情報进行列挙（Enumeration）する時のスキーマ及び値を表す図である。

【0191】

図46の例では、プリンタ3に黒（Black）および一体型のカラーインク（Color）が搭載されていて、インクの残量の状態は、黒は満タン（Full）、カラーインクはインク無し（Out）であることを表している。さらにインクの型名がそれぞれ、CI-B、CI-Color、インクの並び方向は下から上で、順番は下から、カラーインク、黒の順で並んでいて、インクの挿入方向は黒／カラーインクとも左から右の方向であることを表している。

【0192】

図47は、図46のスキーマを使ってステータスアプリケーション30（ステータスマニタ等）がインク無しエラーをグラフィカルに表示する一例である。

【0193】

304にインク無しエラーになっているインクが文字列で表現される。305にはインクの並び方向、並び順および残量およびインクの挿入方向がグラフィックを用いて表現されている。

【0194】

インクは下から、カラーインク（101）、黒（102）の順で並んでおり、残量は、黒は満タン（Full）、カラーインクはインク無し（Out）、インク無しになっているカラーインクの挿入方向は左から右となっていることがわかる。

【0195】

図48はPC1がプリンタ3からインク情報を取得する時のデータを表す図である。実際にPC1とプリンタ3との間で送受信されるデータはバイナリデータであるが、同図では理解しやすくすることを考えて、ASCII文字コードでエンコードした後のテキストデータとして表している。

【0196】

同図において、PC1が要求コマンドをプリンタ3にUSBインタフェース9を介して発行すると、インク情報がプリンタ3からPC1にUSBインタフェース9を介して返される。このインク情報は次の内容を表している。

<色>	<型名>	<状態>	<並び方向>	<並び順>
黒	CI-B	インク無し	左から右	1
カラー	CI-Color	インク無し	左から右	2

図49も図48と同様にPC1がプリンタ3からインク情報を取得する時のデータを表す図である。図48と異なり、プリンタ3からインクの挿入方向も取得している。

<色>	<型名>	<状態>	<並び方向>	<並び順>	<インクの挿入方向>
黒	CI-B	インク無し	左から右	1	下から上
カラー	CI-Color	インク無し	左から右	2	下から上

このようにプリンタによって、インクの挿入方向の情報まで返せるプリンタ（図49）



、返せないプリンタ（図４８）の二通りが存在する。

【０１９７】

以下、図５０～図５２のフローチャートを用いて、本実施形態の動作について説明する。

【０１９８】

図５０はアプリケーション３０（ステータスマニタ等）がプリンタ３のインク情報を取得してインク情報の表示を更新する処理を表すフローチャートである。

【０１９９】

ここでは、プリンタ３はインクの並び方向、並び順の情報は返せるが、インクの挿入方向は返せないものとする。

【０２００】

同図において、アプリケーション３０がプリンタ３のインク情報を取得してインク情報の表示を更新する処理をスタートさせる（Ｓ５００１）。アプリケーション３０が図４３で定義されている¥Printer.InkInfoのスキーマを引数としてCOMインタフェースIBidiSplのAPI関数IBidiSpl::SendRecv()をコールしてインク情報の取得を開始する（Ｓ５００２）。開始すると、Printing and Print Spooler Interfacesを介して、LM36がエクスポートしているSendRecvBidiDataFromPort()関数がスプーラ（Windows（登録商標）XP OSの公知の機能の一部なので、図や詳細説明は省略）からコールされて、この関数における処理が開始される（Ｓ５００３）。

【０２０１】

LM36のSendRecvBidiDataFromPort()関数内において、図４８に示すように、PC1から要求コマンドを発行してプリンタ３から返されるインク情報を取得する（Ｓ５００４）。さらに自分自身の内部で保持しているインクの挿入方向を取得する（Ｓ５００５）。次に、これら取得したインク情報をSendRecvBidiDataFromPort()関数の引数として指定されている¥Printer.InkInfoのスキーマの定義に従って変換する（Ｓ５００６）。次に、SendRecvBidiDataFromPort()関数の引数としてスキーマの定義に変換されたインク情報を返し、LM36のSendRecvBidiDataFromPort()関数の処理を終了する（Ｓ５００７）。

【０２０２】

アプリケーション３０において、IBidiSpl::SendRecv()関数がリターンされ、引数としてスキーマの定義で返されたインク情報を取得する（Ｓ５００８）。次に、このインク情報を使ってインク情報を更新し（Ｓ５００９）、終了する（Ｓ５０１０）。

【０２０３】

通常、アプリケーション３０は、ステップＳ５００１からスタートするプリンタ３のインク情報を取得してインク情報の表示を更新する処理を、例えば５秒に１回等、ある一定時間の間隔で繰り返すことで、プリンタ３に搭載されているインクの情報及び状態をリアルタイムに表示することができる。

【０２０４】

図５１は、図５０と同様にアプリケーション３０（ステータスマニタ等）がプリンタ３のインク情報を取得してインク情報の表示を更新する処理を表すフローチャートである。

【０２０５】

ここでは、プリンタ３はインクの並び方向、並び順、インクの挿入方向の情報を返せるものとする。同図において、アプリケーション３０がプリンタ３のインク情報を取得してインク情報の表示を更新する処理をスタートさせる（Ｓ５１０１）。スタートさせるとアプリケーション３０が図４３で定義されている¥Printer.InkInfoのスキーマを引数としてCOMインタフェースIBidiSplのAPI関数IBidiSpl::SendRecv()をコールしてインク情報の取得を開始すると（Ｓ５１０２）



、`Printing and Print Spooler Interfaces`を介して、`LM36`がエクスポートしている`SendRecvBidiDataFromPort()`関数がスプーラ(`Windows`(登録商標)`XP OS`の公知の機能の一部なので、図や詳細説明は省略)からコールされて、この関数における処理が開始される(`S5103`)。

#### 【0206】

`LM36`の`SendRecvBidiDataFromPort()`関数内において、図49に示すように、`PC1`から要求コマンドを発行してプリンタ3から、インク残量、並び方向、並び順、インクの挿入方向などのインク情報を取得する(`S5104`)。次に、取得したインク情報を`SendRecvBidiDataFromPort()`関数の引数として指定されている`¥Printer.InkInfo`のスキーマの定義に従って変換する(`S5105`)。次に、`SendRecvBidiDataFromPort()`関数の引数としてスキーマの定義に変換されたインク情報を返し、`LM36`の`SendRecvBidiDataFromPort()`関数の処理を終了する(`S5106`)。

#### 【0207】

アプリケーション30において、`IBidiSpl::SendRecv()`関数がリターンされ、引数としてスキーマの定義で返されたインク情報を取得する(`S5107`)。次に、このインク情報を使ってインク情報を更新する(`S5108`)。次に、アプリケーション30がプリンタ3のインク情報を取得してインク情報の表示を更新する処理を終了する(`S5109`)。

#### 【0208】

図52は、図50、図51と同様にアプリケーション30(ステータスマニタ等)がプリンタ3のインク情報を取得してインク情報の表示を更新する処理を表すフローチャートである。

#### 【0209】

ここでは、`LM36`は、プリンタ3から取得したインク情報(`S5204`)にインクの挿入方向の情報が含まれているかどうかを判定する(`S5205`)。判定の結果、含まれていない場合は自分自身の内部のデータベースからインクの挿入方向を取得する(`S5206`)ことを特徴とする。

#### 【0210】

同図において、アプリケーション30がプリンタ3のインク情報を取得してインク情報の表示を更新する処理をスタートさせる(`S5201`)。スタートさせると、アプリケーション30が図43で定義されている`¥Printer.InkInfo`のスキーマを引数として`COMインタフェースIBidiSpl`のAPI関数`IBidiSpl::SendRecv()`をコールしてインク情報の取得を開始する(`S5202`)。開始すると、`Printing and Print Spooler Interfaces`を介して、`LM36`がエクスポートしている`SendRecvBidiDataFromPort()`関数がスプーラ(`Windows`(登録商標)`XP OS`の公知の機能の一部なので、図や詳細説明は省略)からコールされて、この関数における処理が開始される(`S5203`)。

#### 【0211】

`LM36`の`SendRecvBidiDataFromPort()`関数内において、図48、図49に示すように、`PC1`から要求コマンドを発行してプリンタ3から返されるインク情報を取得する(`S5204`)。次に、インク情報にインクの挿入方向の情報が含まれているかどうかを判定する(`S5205`)。

#### 【0212】

プリンタ3から取得した情報が、図48のようにインクの挿入方向の情報を含んでいない場合は自分自身の内部で保持しているインクの挿入方向を取得する(`S5206`)。

#### 【0213】

プリンタ3から取得した情報が、図49に示すようにインクの挿入方向の情報を含んで

10

20

30

40

50



いる場合は、S 5 2 0 7に進む。

【 0 2 1 4 】

S 5 2 0 7で、S 5 2 0 4およびS 5 2 0 6で取得したインク情報をSendRecvBidiDataFromPort()関数の引数として指定されている¥Printer.InkInfoのスキーマの定義に従って変換する。そして、SendRecvBidiDataFromPort()関数の引数としてスキーマの定義に変換されたインク情報を返し、LM36のSendRecvBidiDataFromPort()関数の処理を終了する(S 5 2 0 8)。

【 0 2 1 5 】

アプリケーション30において、IBidiSpl::SendRecv()関数がリターンされ、引数としてスキーマの定義で返されたインク情報を取得する(S 5 2 0 9)。次に、このインク情報を使ってインク情報を更新する(S 5 2 1 0)。次に、アプリケーション30がプリンタ3のインク情報を取得してインク情報の表示を更新する処理を終了する(S 5 2 1 1)。

10

【 0 2 1 6 】

以下、図42に示すメモリマップを参照して、上記実施形態に係る情報処理装置及び周辺装置からなる周辺装置制御システムで読み出し可能なデータ処理プログラムの構成について説明する。

【 0 2 1 7 】

図42は、本実施形態に係る周辺装置制御システムで読み出し可能な各種データ処理プログラムを格納する記憶媒体のメモリマップを示す図である。

20

【 0 2 1 8 】

なお、特に図示しないが、記憶媒体に記憶されるプログラム群を管理する情報、例えばバージョン情報、作成者等もこの記憶媒体に記憶される。かつ、プログラム読み出し側のOS等に依存する情報、例えばプログラムを識別表示するアイコン等も記憶される場合もある。

【 0 2 1 9 】

同図において、64は記憶媒体であり、ここではハードディスク1202で構成されているものとする。65はディレクトリ情報管理部であり、各種プログラムに従属するデータがこのディレクトリ情報管理部65で管理されている。

30

【 0 2 2 0 】

66はプログラム格納部であり、各種プログラムを情報処理装置にインストールするためのプログラムや、インストールするプログラムが圧縮されている場合に解凍するためのプログラム等も記憶される場合がある。

【 0 2 2 1 】

上記第1の実施の形態における図15、図16、図17、上記第2の実施の形態における図39、図40、図41、上記第3の実施の形態における図50、図51、図52にそれぞれ示す各フローチャートで実現される処理は、このプログラム格納部66に記憶されており、それぞれ実行される。

【 0 2 2 2 】

これらの各フローチャートの実行によって実現できる各機能が、外部からインストールされるプログラムによって、情報処理装置によって実現されるようにしてもよい。

40

【 0 2 2 3 】

そして、その場合、CD-ROMやフラッシュメモリやフレキシブルディスク等の記憶媒体により、あるいはネットワークを介して外部の記憶媒体から、プログラムを含む情報群が情報処理装置や周辺装置に供給される場合でも、本発明は適用されるものである。

【 0 2 2 4 】

上記第1の実施形態、上記第2の実施の形態、上記第3の実施の形態では、アプリケーション30の一例としてステータスマニタを挙げたが、この例に限られることなく、例えば周辺装置から情報を取得して、それを利用するような任意のアプリケーションで実現可

50



能であり、有効である。

【 0 2 2 5 】

また、上記第 1 の実施形態、上記第 2 の実施の形態、上記第 3 の実施の形態では、アプリケーション（ステータスマニタ）30 でプリンタ 3 に搭載されているインクの情報及び状態をモニタする例を説明した。しかしながら、この例に限られることなく、例えば、周辺装置の動作状態や、警告、エラーの状態、オプションの装着状態等の任意の情報や状態の取得に有効活用できる。

【 0 2 2 6 】

また、上記第 1 の実施形態、上記第 2 の実施の形態、上記第 3 の実施の形態では、プリンタの例としてカラーインクジェットプリンタを使用した。しかしながら、この例に限られることなく、例えば、カラー L B P 等の電子写真プリンタのトナーの表示に使用することができる。

10

【 0 2 2 7 】

また、本実施の形態では、消耗品としてはインクやトナーで説明したが、用紙に適用することもできる。用紙に適用することによりアプリケーションはプリンタの複数のカセットにどのようなサイズや種類の用紙が入っているか表示することができるようになる。

【 0 2 2 8 】

また、上記第 1 の実施形態、上記第 2 の実施の形態、上記第 3 の実施の形態では、情報処理装置、及び情報処理装置から構成されるクライアントやサーバとしてパーソナルコンピュータを想定したが、この例に限られることはない。例えば D V D ビデオプレーヤー、ゲーム、セットトップボックス、インターネット家電等、同様な使用方法が可能な任意の端末に対して実現することができ、有効である。

20

【 0 2 2 9 】

また、上記第 1 の実施形態、上記第 2 の実施の形態、上記第 3 の実施の形態では、周辺装置としてプリンタを例示している。しかしながら、周辺装置として他に、複写機、ファクシミリ、スキャナ、デジタルカメラ、およびこれらの複合機能を備える装置などのいずれかが、本発明の適用対象となり得る。

【 0 2 3 0 】

また、上記第 1 の実施形態、上記第 2 の実施の形態、上記第 3 の実施の形態では、OS に例として W i n d o w s（登録商標）X P を使用したが、これらの OS に限られることなく、任意の OS を使用することができる。

30

【 0 2 3 1 】

また、上記第 1 の実施形態、上記第 2 の実施の形態、上記第 3 の実施の形態では、ネットワーク 4 の構成例として E t h e r n e t（登録商標）を用いたが、この例に限られることなく、他の任意のネットワーク構成であってもよい。

【 0 2 3 2 】

また、上記第 1 の実施形態、上記第 2 の実施の形態、上記第 3 の実施の形態では、P C 1 とプリンタ 3 との間のインタフェースとして、U S B インタフェースを用いたが、このインタフェースに限られることはない。例えば、E t h e r n e t（登録商標）、無線 L A N、I E E E 1 3 9 4、B l u e t o o t h、I r D A、パラレル、シリアル等の任意のインタフェースを用いるようにしてもよい。

40

【 0 2 3 3 】

また、上記第 1 の実施形態、上記第 2 の実施の形態、上記第 3 の実施の形態では、P C 1 上のアプリケーション 30 からプリンタ 3 に搭載されているインク情報を取得する例や、ネットワーク 4 を介して接続された P C 2（クライアント）上のアプリケーション 200 から P C 1（サーバ）を介してプリンタ 3 に搭載されているインク情報を取得する例を挙げて説明した。しかしながら、これらの例に限られることない。例えば、P C 1 の OS がマルチユーザをサポートし、各ユーザがそれぞれ異なるアプリケーションを起動している場合においても、P C 1 上のアプリケーション 30 や P C 2 上のアプリケーション 200 が P r i n t i n g a n d P r i n t S p o o l e r I n t e r f a c e s を

50



介してランゲージモニタ36経由でプリンタに搭載されているインク情報を取得することができる。よって、このようなマルチユーザの環境においても、実現可能であり、有効である。

【0234】

また、上記第2の実施の形態では、インクの画像を表すスキーマとしてビットマップファイル（バイナリデータ）を用いたが、この例に限られることなく、例えばJ P E G、T I F F等の任意のフォーマットの画像を用いてもよい。

【0235】

また、上記第2の実施の形態では、インクの画像の各色成分を表すスキーマとして赤色成分（R e d）、緑色成分（G r e e n）、青色成分（B l u e）を用いた。しかしながら、この例に限られることなく、例えば黒色成分（B l a c k）、白色成分（W h i t e）、グレー色成分（G r a y）、黄色成分（Y e l l o w）、マゼンタ色成分（M a g e n t a）、シアン色成分（C y a n）等の任意の成分を用いてもよい。

10

【0236】

また、色成分とは別に、色合い、鮮やかさ、明るさ等の色を表す情報も用いることで、より一層リアルな表現が可能となる。

【0237】

また、上記第2の実施の形態では、インクの画像を表すスキーマとして幅（W i d t h）、奥行き（D e p t h）、高さ（H e i g h t）を用いたが、この例に限られることなく、任意の形状を表す定義を用いてもよい。

20

【0238】

また、これらとは別に、例えば形状の定義として、直方体、円柱等の定義も合わせることで、より一層リアルな表現が可能となる。

【0239】

また、本発明の目的は、前述した第1の実施形態、上記第2の実施の形態、上記第3の実施の形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システム或いは装置に供給する。そして、そのシステム或いは装置のコンピュータ（またはC P UやM P U）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。

【0240】

30

この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、プログラムコード自体及びそのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0241】

プログラムコードを供給するための記憶媒体としては、例えば、フレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、C D - R O M、C D - R、磁気テープ、不揮発性のメモ리카ード、R O M等を用いることができる。

【0242】

また、コンピュータが読み出したプログラムコードを実行することにより、前述した第1の実施形態や第2の実施の形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼動しているO S（基本システム或いはオペレーティングシステム）などが実際の処理の一部又は全部を行い、その処理によって前述した第1の実施形態や第2の実施の形態の機能が実現される場合も含まれることは言うまでもない。

40

【0243】

さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれる。その後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるC P U等が実際の処理の一部又は全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

50



## 【 0 2 4 4 】

以上説明したように本実施の形態によれば、周辺装置の製造元とは全く関係のないサードパーティ製のアプリケーション等の任意のアプリケーション上に、周辺装置に関する情報を細部までグラフィカルに、かつ、ビジュアル的にリアルに表現することができる。例えば、プリンタに搭載されているインクの状態を表す情報、すなわち、インクの形状、色、大きさ、並び方向、並び順、残量、挿入方向等を、細部までグラフィカルに、かつ、ビジュアル的にリアルに表現することができる。

## 【 0 2 4 5 】

また、正確に表示し、操作性に優れた周辺装置制御システムを実現することができる。

## 【図面の簡単な説明】

10

## 【 0 2 4 6 】

【図 1】本発明の第 1 の実施形態に係る情報処理装置（PC）及び周辺装置からなる周辺装置制御システムの構成を示すブロック図である。

【図 2】本発明に係る情報処理装置及び周辺装置からなる周辺装置制御システムをネットワーク環境下で実現した時のシステムの構成部分を表すブロック図である。

【図 3】PC のハードウェア構成を示す図である。

【図 4】プリンタのハードウェア構成を示す図である。

【図 5】PC におけるプリンタドライバの構成を示すブロック図である。

【図 6】アプリケーションがプリンタからインクの情報及び状態を取得する時に使用するスキーマを表す図である。

20

【図 7】図 6 で定義されているスキーマを使ってインクの情報及び状態を列挙（Enumeration）する時のスキーマ及び値を表す図である。

【図 8】図 7 のスキーマを使ってアプリケーションがインク残量をグラフィカルに表示している図である。

【図 9】図 6 で定義されているスキーマを使って、インクタンクが前後の並びで取り付けられるプリンタの情報及び状態を取得する時に使用するスキーマを表す図である。

【図 10】図 9 のスキーマで表現されるプリンタの外観図である。

【図 11】図 6 で定義されているスキーマを使って、インクタンクが上下の並びで取り付けられるプリンタの情報及び状態を取得する時に使用するスキーマを表す図である。

【図 12】図 11 のスキーマを使ってアプリケーションがインク残量をグラフィカルに表示している図である。

30

【図 13】PC がプリンタからインク情報を取得する時のデータを表す図である。

【図 14】PC がプリンタからインク情報を取得する時のデータを表す図である。

【図 15】アプリケーションがプリンタのインク情報を取得してインク情報の表示を更新する処理を表すフローチャートである。

【図 16】アプリケーションがプリンタのインク情報を取得してインク情報の表示を更新する処理を表すフローチャートである。

【図 17】アプリケーションがプリンタのインク情報を取得してインク情報の表示を更新する処理を表すフローチャートである。

【図 18】プリンタがインクの並び方向、並び順を取得する方法を表したフローチャートである。

40

【図 19】プリンタがインクの並び方向、並び順を取得する方法を表したフローチャートである。

【図 20】アプリケーションがプリンタからインクの情報及び状態を取得する時に使用するスキーマを表す図である。

【図 21】アプリケーションがプリンタからインクの情報及び状態を取得する時に使用するスキーマを表す図である。

【図 22】図 20 及び図 21 で定義されているスキーマを使ってインクの情報及び状態を列挙（Enumeration）する時のスキーマ及び値を表す図である。

【図 23】プリンタの状態をモニタするステータスマニタを表す図である。

50



【図 2 4】図 2 2 における各インクの画像のビットマップファイルの表示内容を表す図である。

【図 2 5】各インクの残量の画像のビットマップファイルの表示内容を表す図である。

【図 2 6】図 2 0 及び図 2 1 で定義されているスキーマを使ってインクの情報及び状態を列挙 ( E n u m e r a t i o n ) する時のスキーマ及び値を表す図である。

【図 2 7】プリンタの状態をモニタするステータスマニタを表す図である。

【図 2 8】図 2 6 における各インクの画像のビットマップファイルの表示内容を表す図である。

【図 2 9】各インクの残量の画像のビットマップファイルの表示内容を表す図である。

【図 3 0】アプリケーションがプリンタからインクの情報及び状態を取得する時に使用するスキーマを表す図である。

10

【図 3 1】図 2 0 及び図 3 0 で定義されているスキーマを使ってインクの情報及び状態を列挙 ( E n u m e r a t i o n ) する時のスキーマ及び値を表す図である。

【図 3 2】インク情報のデータベースを表す図である。

【図 3 3】インク情報のデータベースを表す図である。

【図 3 4】P C がプリンタからインク情報を取得する時のデータを表す図である。

【図 3 5】P C がプリンタからインク情報を取得する時のデータを表す図である。

【図 3 6】P C がプリンタからインク情報を取得する時のデータを表す図である。

【図 3 7】P C がプリンタからインク情報を取得する時のデータを表す図である。

【図 3 8】P C がプリンタからインク情報を取得する時のデータを表す図である。

20

【図 3 9】アプリケーションがプリンタのインク情報を取得してインク情報の表示を更新する処理を表すフローチャートである。

【図 4 0】L M におけるインク情報をスキーマの定義に変換する処理を表すフローチャートである。

【図 4 1】アプリケーションにおけるインク情報表示用データの作成処理を表すフローチャートである。

【図 4 2】周辺装置制御システムで読み出し可能な各種データ処理プログラムを格納する記憶媒体のメモリマップを示す図である。

【図 4 3】アプリケーションがプリンタからインクの情報及び状態、挿入方向を取得する時に使用するスキーマを表す図である。

30

【図 4 4】図 4 3 で定義されているスキーマを使ってインクの情報及び状態を列挙 ( E n u m e r a t i o n ) する時のスキーマ及び値を表す図である。

【図 4 5】図 4 4 のスキーマを使ってアプリケーションがインク残量及び挿入方向をグラフィカルに表示している図である。

【図 4 6】図 4 3 で定義されているスキーマを使って、インクの挿入方向が左から右であるプリンタの情報及び状態を取得する時に使用するスキーマを表す図である。

【図 4 7】図 4 6 のスキーマを使ってアプリケーションがインク残量及び挿入方向をグラフィカルに表示している図である。

【図 4 8】P C がプリンタからインク情報を取得する時のデータを表す図である。

【図 4 9】P C がプリンタからインク情報を取得する時のデータを表す図である。

40

【図 5 0】アプリケーションがプリンタのインク情報を取得してインク情報の表示を更新する処理を表すフローチャートである。

【図 5 1】アプリケーションがプリンタのインク情報を取得してインク情報の表示を更新する処理を表すフローチャートである。

【図 5 2】アプリケーションがプリンタのインク情報を取得してインク情報の表示を更新する処理を表すフローチャートである。

【符号の説明】

【 0 2 4 7 】

1 情報処理装置 ( P C )

2 情報処理装置 ( P C )

50

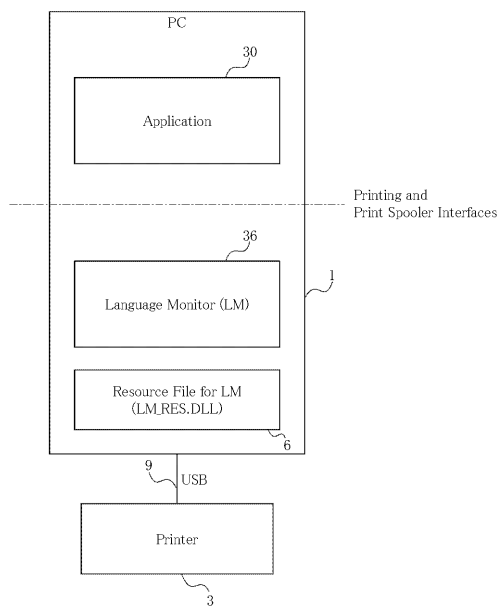


3	プリンタ	
6	ランゲージモニタ用リソースファイル ( L M リソース )	
9	U S B インタフェース	
3 0	アプリケーション ( ステータスモニタ )	
3 6	ランゲージモニタ	
4 0	プリンタ情報表示部	
4 1	インク残量表示部	
4 2	黒	
4 3	シアン	
4 4	マゼンタ	10
4 5	黄	
4 6	プリンタ情報表示部	
4 7	インク残量表示部	
4 8	カラーインク	
4 9	フォトインク	
2 0 0	アプリケーション ( ステータスモニタ )	
2 0 1	ステータスモニタのメインウィンドウ	
2 0 2	プリンタ情報表示部	
2 0 3	インク情報表示部	
2 0 4	インク残量表示部	20
2 0 5	インクの残量が満タンの時の黒インクの画像を表すビットマップファイル ( C I - B . b m p ) の表示内容	
2 0 6	インクの残量が満タンの時の黄インクの画像を表すビットマップファイル ( C I - Y . b m p ) の表示内容	
2 0 7	インクの残量が満タンの時のマゼンタインクの画像を表すビットマップファイル ( C I - M . b m p ) の表示内容	
2 0 8	インクの残量が満タンの時のシアンインクの画像を表すビットマップファイル ( C I - C . b m p ) の表示内容	
2 0 9	インクの残量が 8 0 % の黒インクの画像を表すビットマップファイルの表示内容	30
2 1 0	インクの残量が 5 0 % の黄インクの画像を表すビットマップファイルの表示内容	
2 1 1	インクの残量が 0 % のマゼンタインクの画像を表すビットマップファイルの表示内容	
2 1 2	インクの残量が 1 0 0 % のシアンインクの画像を表すビットマップファイルの表示内容	
2 1 6	カラーインクの残量表示部	
2 1 7	フォトインクの残量表示部	
2 1 8	インクの残量が満タンの時のカラーインクの画像を表すビットマップファイル ( C I - C o l o r . b m p ) の表示内容	40
2 1 9	インクの残量が満タンの時のフォトインクの画像を表すビットマップファイル ( C I - P h o t o . b m p ) の表示内容	
2 2 0	インクの残量が 4 0 % のカラーインクの画像を表すビットマップファイルの表示内容	
2 2 1	インクの残量が 8 0 % のフォトインクの画像を表すビットマップファイルの表示内容	
3 0 0	インク残量表示部 ( テキスト表示 )	
3 0 1	インク残量表示部 ( グラフィック表示 )	
3 0 2	黒	
3 0 3	カラー	50

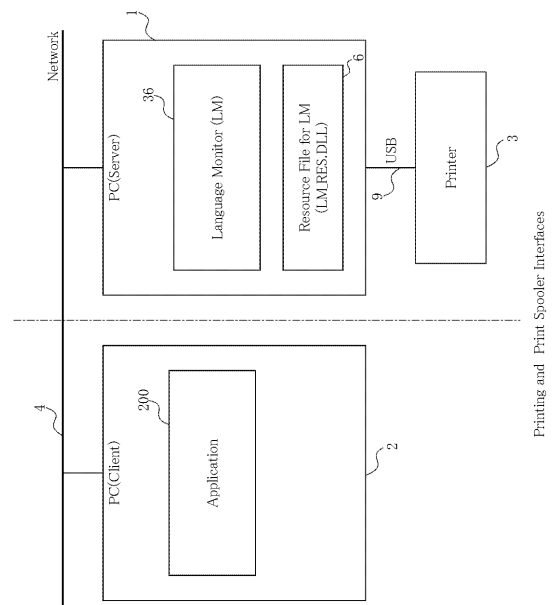


- 3 0 4 インク残量表示部（テキスト表示）
- 3 0 5 インク残量表示部（グラフィック表示）
- 3 0 6 黒
- 3 0 7 カラー

【図 1】

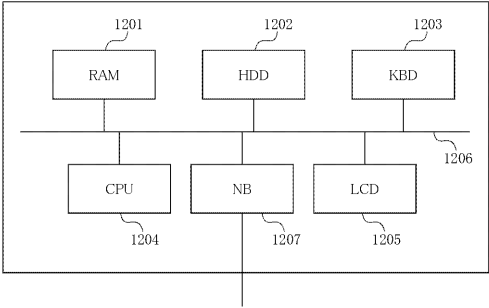


【図 2】

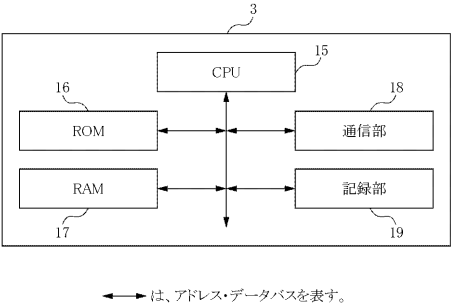




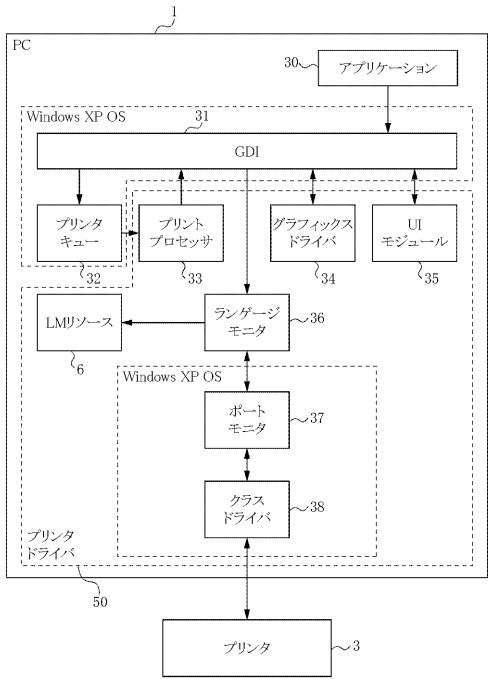
【図 3】



【図 4】



【図 5】



【図 6】

Definition: InkInfo

Node Type: Property

Description: インクの情報

Full Schema Path: ¥Printer.InkInfo

Definition: [Color]

Node Type: Property

Description: 色の情報

Full Schema Path: ¥Printer.InkInfo.[Color]

Allowed Values:

Black	# 黒	Yellow	# 黄
Cyan	# シアン	Photo	# フォト
Magenta	# マゼンタ	Color	# カラー

Definition: Installed

Node Type: Value

Data Type: Boolean

Description: [Color]色のインクが搭載されているか否か

Full Schema Path: ¥Printer.InkInfo.[Color].Installed

Allowed Values:

True	# 搭載されている
False	# 搭載されていない

Definition: State

Node Type: Value

Data Type: String

Description: [Color]色のインクの残量の状態

Full Schema Path: ¥Printer.InkInfo.[Color].State

Allowed Values:

Full	# インク満タン	Out	# インク無し
Low	# インク残りわずか	Unknown	# インク残量不明

Definition: ModelName

Node Type: Value

Data Type: String

Description: [Color]色のインクの型名

Full Schema Path: ¥Printer.InkInfo.[Color].ModelName

Examples:

CI-B	# 黒インク	CI-Y	# 黄インク
CI-C	# シアンインク	CI-Color	# 一体型のカラーインク
CI-M	# マゼンタインク	CI-Photo	# 一体型のフォトインク

Definition: Direction

Node Type: Value

Data Type: String

Description: インクの並び方向

Full Schema Path: ¥Printer.InkInfo.[Color].Direction

Allowed Values:

原点は、プリンタの左下とする。	
LeftRight	# 左から右
FrontBack	# 手前から後ろ
BottomTop	# 下から上

Definition: Order

Node Type: Value

Data Type: Int

Description: 原点からインクの順番

Full Schema Path: ¥Printer.InkInfo.[Color].Order

Examples:

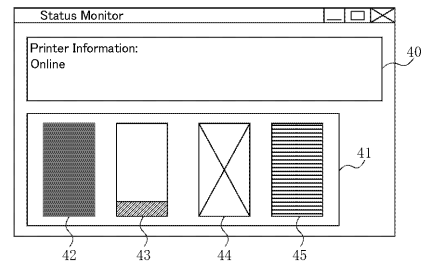
1	# 1番目
2	# 2番目



【図 7】

<Enumeration> -Query(Schema)- ¥Printer.InkInfo	-Retrieve(Schema)-	-Retrieve(Value)-
¥Printer.InkInfo.Black:Installed		True
¥Printer.InkInfo.Black:State		Full
¥Printer.InkInfo.Black:ModelName		CI-B
¥Printer.InkInfo.Black:Direction		LeftRight
¥Printer.InkInfo.Black:Order		1
¥Printer.InkInfo.Cyan:Installed		True
¥Printer.InkInfo.Cyan:State		Low
¥Printer.InkInfo.Cyan:ModelName		CI-C
¥Printer.InkInfo.Cyan:Direction		LeftRight
¥Printer.InkInfo.Cyan:Order		2
¥Printer.InkInfo.Magenta:Installed		True
¥Printer.InkInfo.Magenta:State		Out
¥Printer.InkInfo.Magenta:ModelName		CI-M
¥Printer.InkInfo.Magenta:Direction		LeftRight
¥Printer.InkInfo.Magenta:Order		3
¥Printer.InkInfo.Yellow:Installed		True
¥Printer.InkInfo.Yellow:State		Full
¥Printer.InkInfo.Yellow:ModelName		CI-Y
¥Printer.InkInfo.Yellow:Direction		LeftRight
¥Printer.InkInfo.Yellow:Order		4

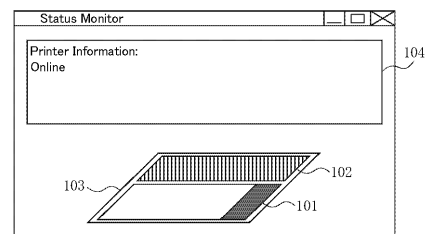
【図 8】



【図 9】

<Enumeration> -Query(Schema)- ¥Printer.InkInfo	-Retrieve(Schema)-	-Retrieve(Value)-
¥Printer.InkInfo.Color:Installed		True
¥Printer.InkInfo.Color:State		Low
¥Printer.InkInfo.Color:ModelName		CI-Color
¥Printer.InkInfo.Color:Direction		FrontBack
¥Printer.InkInfo.Color:Order		1
¥Printer.InkInfo.Black:Installed		True
¥Printer.InkInfo.Black:State		Full
¥Printer.InkInfo.Black:ModelName		CI-B
¥Printer.InkInfo.Black:Direction		FrontBack
¥Printer.InkInfo.Black:Order		2

【図 10】

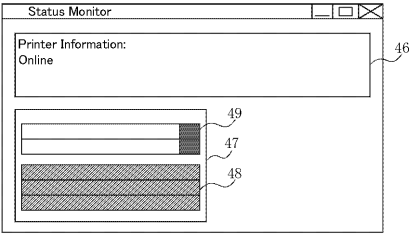




【図 1 1】

<Enumeration>		
-Query(Schema)-		
¥Printer.InkInfo		
-Retrieve(Schema)-		
¥Printer.InkInfo.Color:Installed	True	
¥Printer.InkInfo.Color:State	Full	
¥Printer.InkInfo.Color:ModelName	CI-Color	
¥Printer.InkInfo.Color:Direction	BottomTop	
¥Printer.InkInfo.Color:Order	1	
¥Printer.InkInfo.Photo:Installed	True	
¥Printer.InkInfo.Photo:State	Low	
¥Printer.InkInfo.Photo:ModelName	CI-Photo	
¥Printer.InkInfo.Photo:Direction	BottomTop	
¥Printer.InkInfo.Photo:Order	2	
-Retrieve(Value)-		

【図 1 2】



【図 1 3】

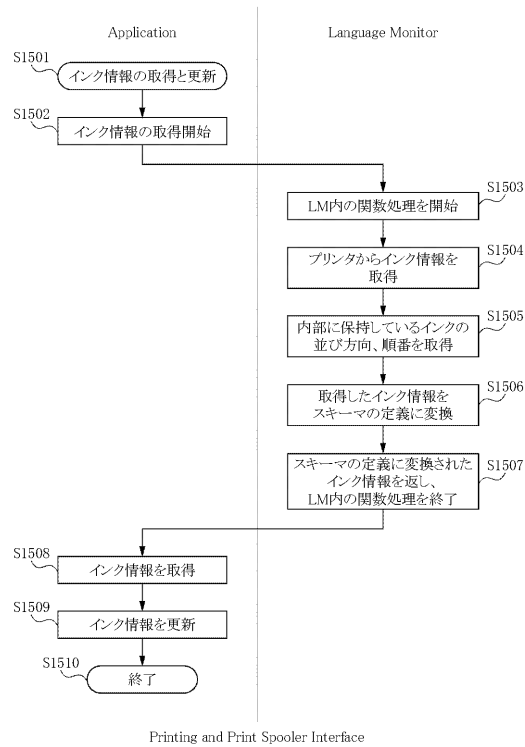
-PCから発行する要求コマンド-  
Ink;  
-プリンタから返されるインク情報-  
Ink:Black,CI-B,Full,Cyan,CI-C,Low,Magenta,CI-M,Out,Yellow,CI-Y,Full;

【図 1 4】

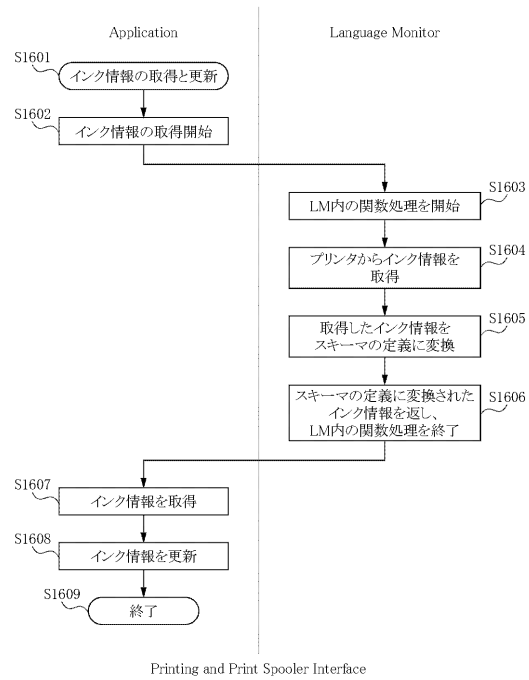
-PCから発行する要求コマンド-  
Ink;  
-プリンタから返されるインク情報-  
Ink:Black,CI-B,Full,LeftRight,1,Cyan,CI-C,Low,LeftRight,2,Magenta,CI-M,Out,LeftRight,3,Yellow,CI-Y,Full,LeftRight,4;



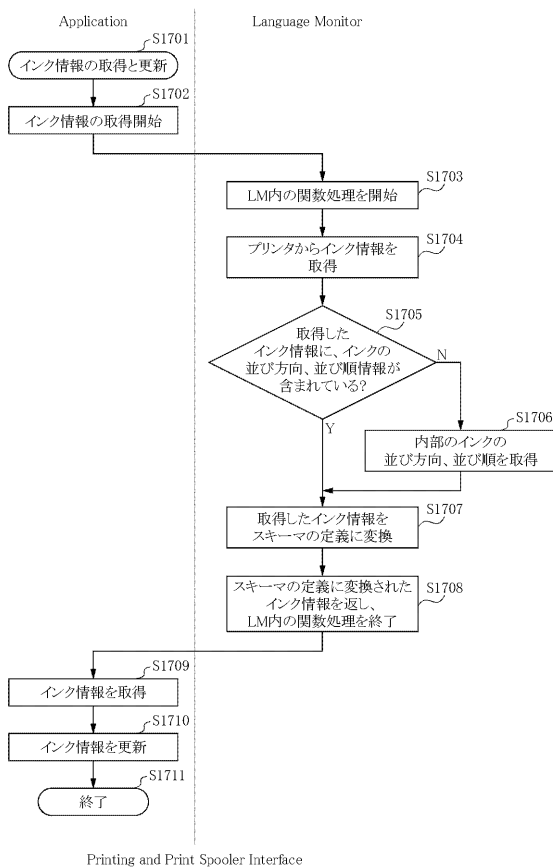
【図 15】



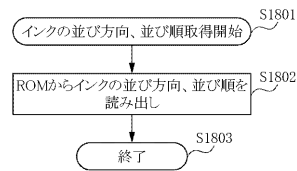
【図 16】



【図 17】

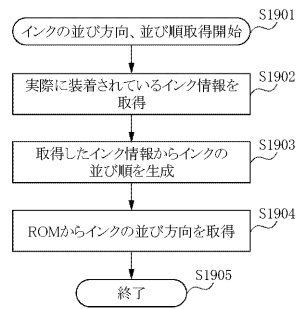


【図 18】





【図 19】



【図 20】

Definition:	InkInfo
Node Type:	Property
Description:	インクの情報
Full Schema Path:	¥Printer.InkInfo
Definition:	[Color]
Node Type:	Property
Description:	色の情報
Full Schema Path:	¥Printer.InkInfo.[Color]
Allowed Values:	Black Cyan Magent Yellow Photo Color
	# 黒 # シアン # マゼンタ # 黄 # フォト # カラー
Definition:	Installed
Node Type:	Value
Data Type:	Boolean
Description:	[Color]色のインクが搭載されているかどうか
Full Schema Path:	¥Printer.InkInfo.[Color]:Installed
Allowed Values:	True False
	# 搭載されている # 搭載されていない
Definition:	Level
Node Type:	Value
Data Type:	Int
Description:	[Color]色のインクの残量
Full Schema Path:	¥Printer.InkInfo.[Color]:Level
Allowed Values:	0~100 -1
	# インク残量をパーセント表示(単位:%) # インク残量不明
Definition:	ModelName
Node Type:	Value
Data Type:	String
Description:	[Color]色のインクの型名
Full Schema Path:	¥Printer.InkInfo.[Color]:ModelName
Examples:	CI-B # 黒インク CI-Y # 黄インク CI-C # シアンインク CI-Color # カラーインク CI-M # マゼンタインク CI-Photo # フォトインク
Definition:	Direction
Node Type:	Value
Data Type:	String
Description:	[Color]色のインクの並び方向
Full Schema Path:	¥Printer.InkInfo.[Color]:Direction
Allowed Values:	原点は、プリンタの左下に設定している。 LeftRight FrontBack BottomTop
	# 左から右 # 手前から後ろ # 下から上
Definition:	Order
Node Type:	Value
Data Type:	Int
Description:	原点からの[Color]色のインクの順番(並び順)
Full Schema Path:	¥Printer.InkInfo.[Color]:Order
Examples:	1 2
	# 1番目 # 2番目

【図 21】

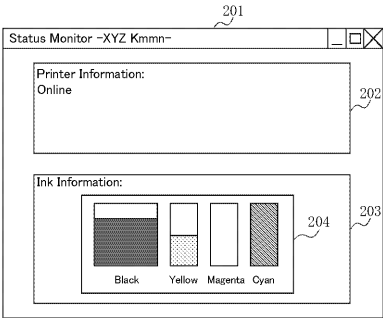
Definition:	Image
Node Type:	Value
Data Type:	Bitmap
Description:	[Color]色のインクの画像
Full Schema Path:	¥Printer.InkInfo.[Color]:Image
Allowed Values:	ビットマップファイル(バイナリデータ)

【図 22】

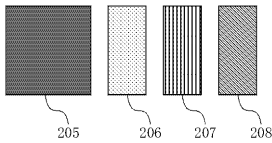
<Enumeration>	-Retrieve(Schema)-	-Retrieve(Value)-
-Query(Schema)- ¥Printer.InkInfo	¥Printer.InkInfo.Black:Installed ¥Printer.InkInfo.Black:Level ¥Printer.InkInfo.Black:ModelName ¥Printer.InkInfo.Black:Direction ¥Printer.InkInfo.Black:Order ¥Printer.InkInfo.Black:Image ¥Printer.InkInfo.Yellow:Installed ¥Printer.InkInfo.Yellow:Level ¥Printer.InkInfo.Yellow:ModelName ¥Printer.InkInfo.Yellow:Direction ¥Printer.InkInfo.Yellow:Order ¥Printer.InkInfo.Yellow:Image ¥Printer.InkInfo.Magenta:Installed ¥Printer.InkInfo.Magenta:Level ¥Printer.InkInfo.Magenta:ModelName ¥Printer.InkInfo.Magenta:Direction ¥Printer.InkInfo.Magenta:Order ¥Printer.InkInfo.Magenta:Image ¥Printer.InkInfo.Cyan:Installed ¥Printer.InkInfo.Cyan:Level ¥Printer.InkInfo.Cyan:ModelName ¥Printer.InkInfo.Cyan:Direction ¥Printer.InkInfo.Cyan:Order ¥Printer.InkInfo.Cyan:Image	True 80 CI-B LeftRight 1 CI-B.bmp True 50 CI-Y LeftRight 2 CI-Y.bmp True 0 CI-M LeftRight 3 CI-M.bmp True 100 CI-C LeftRight 4 CI-C.bmp



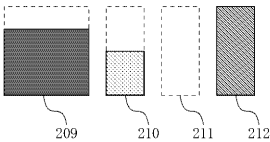
【図 2 3】



【図 2 4】



【図 2 5】

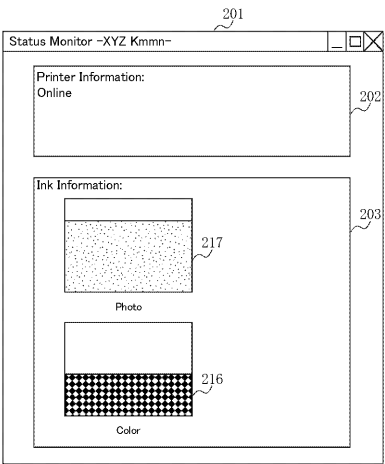


【図 2 6】

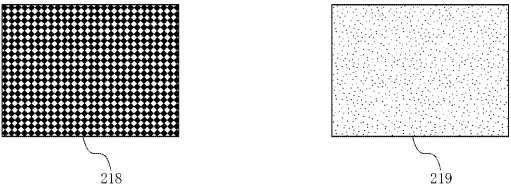
<Enumeration>	-Retrieve(Schema)-	-Retrieve(Value)-
-Query(Schema)-		
¥Printer.InkInfo		
¥Printer.InkInfo.Color:Installed		True
¥Printer.InkInfo.Color:Level		40
¥Printer.InkInfo.Color:ModelName		CI-Color
¥Printer.InkInfo.Color:Direction		BottomTop
¥Printer.InkInfo.Color:Order		1
¥Printer.InkInfo.Color:Image		CI-Color.bmp
¥Printer.InkInfo.Photo:Installed		True
¥Printer.InkInfo.Photo:Level		80
¥Printer.InkInfo.Photo:ModelName		CI-Photo
¥Printer.InkInfo.Photo:Direction		BottomTop
¥Printer.InkInfo.Photo:Order		2
¥Printer.InkInfo.Photo:Image		CI-Photo.bmp



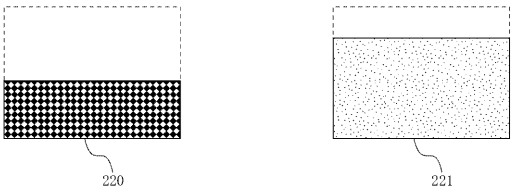
【図 27】



【図 28】



【図 29】



【図 30】

Definition:	ImageData
Node Type:	Property
Description:	[Color]色のインクの画像情報
Full Schema Path:	¥Printer.InkInfo.[Color].ImageData

Definition:	Red
Node Type:	Value
Data Type:	Int
Description:	[Color]色のインクの画像の赤色成分
Full Schema Path:	¥Printer.InkInfo.[Color].ImageData:Red
Allowed Values:	0~255

Definition:	Green
Node Type:	Value
Data Type:	Int
Description:	[Color]色のインクの画像の緑色成分
Full Schema Path:	¥Printer.InkInfo.[Color].ImageData:Green
Allowed Values:	0~255

Definition:	Blue
Node Type:	Value
Data Type:	Int
Description:	[Color]色のインクの画像の青色成分
Full Schema Path:	¥Printer.InkInfo.[Color].ImageData:Blue
Allowed Values:	0~255

Definition:	Width
Node Type:	Value
Data Type:	Int
Description:	[Color]色のインクの画像の幅
Full Schema Path:	¥Printer.InkInfo.[Color].ImageData:Width
Allowed Values:	0~100

Definition:	Depth
Node Type:	Value
Data Type:	Int
Description:	[Color]色のインクの画像の奥行き
Full Schema Path:	¥Printer.InkInfo.[Color].ImageData:Depth
Allowed Values:	0~100

Definition:	Height
Node Type:	Value
Data Type:	Int
Description:	[Color]色のインクの画像の高さ
Full Schema Path:	¥Printer.InkInfo.[Color].ImageData:Height
Allowed Values:	0~100



【図 3 1】

<Enumeration>		
-Query(Schema)-		
¥Printer.InkInfo		
¥Printer.InkInfo.Black:Installed	True	
¥Printer.InkInfo.Black:Level	80	
¥Printer.InkInfo.Black:ModelName	CI-B	
¥Printer.InkInfo.Black:Direction	LeftRight	
¥Printer.InkInfo.Black:Order	1	
¥Printer.InkInfo.Black:ImageData:Red	0	
¥Printer.InkInfo.Black:ImageData:Green	0	
¥Printer.InkInfo.Black:ImageData:Blue	0	
¥Printer.InkInfo.Black:ImageData:Width	65	
¥Printer.InkInfo.Black:ImageData:Depth	100	
¥Printer.InkInfo.Black:ImageData:Height	75	
¥Printer.InkInfo.Yellow:Installed	True	
¥Printer.InkInfo.Yellow:Level	50	
¥Printer.InkInfo.Yellow:ModelName	CI-Y	
¥Printer.InkInfo.Yellow:Direction	LeftRight	
¥Printer.InkInfo.Yellow:Order	2	
¥Printer.InkInfo.Yellow:ImageData:Red	255	
¥Printer.InkInfo.Yellow:ImageData:Green	255	
¥Printer.InkInfo.Yellow:ImageData:Blue	0	
¥Printer.InkInfo.Yellow:ImageData:Width	25	
¥Printer.InkInfo.Yellow:ImageData:Depth	100	
¥Printer.InkInfo.Yellow:ImageData:Height	75	
¥Printer.InkInfo.Magenta:Installed	True	
¥Printer.InkInfo.Magenta:Level	0	
¥Printer.InkInfo.Magenta:ModelName	CI-M	
¥Printer.InkInfo.Magenta:Direction	LeftRight	
¥Printer.InkInfo.Magenta:Order	3	
¥Printer.InkInfo.Magenta:ImageData:Red	255	
¥Printer.InkInfo.Magenta:ImageData:Green	0	
¥Printer.InkInfo.Magenta:ImageData:Blue	255	
¥Printer.InkInfo.Magenta:ImageData:Width	25	
¥Printer.InkInfo.Magenta:ImageData:Depth	100	
¥Printer.InkInfo.Magenta:ImageData:Height	75	
¥Printer.InkInfo.Cyan:Installed	True	
¥Printer.InkInfo.Cyan:Level	100	
¥Printer.InkInfo.Cyan:ModelName	CI-C	
¥Printer.InkInfo.Cyan:Direction	LeftRight	
¥Printer.InkInfo.Cyan:Order	4	
¥Printer.InkInfo.Cyan:ImageData:Red	0	
¥Printer.InkInfo.Cyan:ImageData:Green	255	
¥Printer.InkInfo.Cyan:ImageData:Blue	255	
¥Printer.InkInfo.Cyan:ImageData:Width	25	
¥Printer.InkInfo.Cyan:ImageData:Depth	100	
¥Printer.InkInfo.Cyan:ImageData:Height	75	

【図 3 2】

Ink Color	Direction	Order	Image	ImageData
Black	LeftRight	1	CI-B.bmp	0,0,0,65,100,75
Yellow	LeftRight	2	CI-Y.bmp	255,255,0,25,100,75
Magenta	LeftRight	3	CI-M.bmp	255,0,255,25,100,75
Cyan	LeftRight	4	CI-C.bmp	0,255,255,25,100,75

【図 3 3】

Ink Color	Direction	Order	Image
Color	BottomTop	1	CI-Color.bmp
Photo	BottomTop	2	CI-Photo.bmp

【図 3 4】

-PCから発行する要求コマンド-

Ink;

-プリンタから返されるインク情報-

Ink:Black,CI-B,80,Yellow,CI-Y,50,Magenta,CI-M,0,Cyan,CI-C,100;



【図 3 5】

-PCから発行する要求コマンド-	-プリンタから返されるインク情報-
Ink;	Ink:Black,CI-B,80,LeftRight,1,"CI-B.bmp", Yellow,CI-Y,50,LeftRight,2,"CI-Y.bmp", Magenta,CI-M,0,LeftRight,3,"CI-M.bmp", Cyan,CI-C,100,LeftRight,4,"CI-C.bmp";

【図 3 6】

-PCから発行する要求コマンド-	-プリンタから返されるインク情報-
Ink;	Ink:Black,CI-B,80,LeftRight,1,0,0,0,65,100,75, Yellow,CI-Y,50,LeftRight,2,255,255,0,25,100,75, Magenta,CI-M,0,LeftRight,3,255,0,255,25,100,75, Cyan,CI-C,100,LeftRight,4,0,255,255,25,100,75;

【図 3 7】

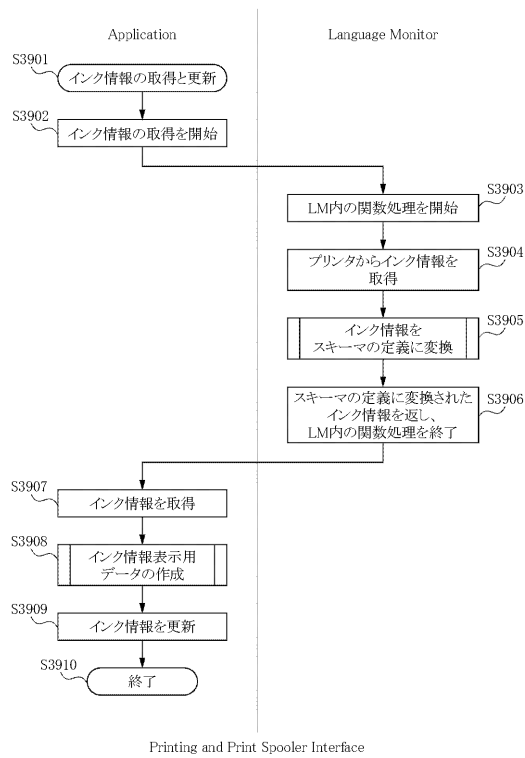
-PCから発行する要求コマンド-	-プリンタから返されるインク情報-
Ink;	Ink:Color,CI-Color,40,Photo,CI-Photo,80;

【図 3 8】

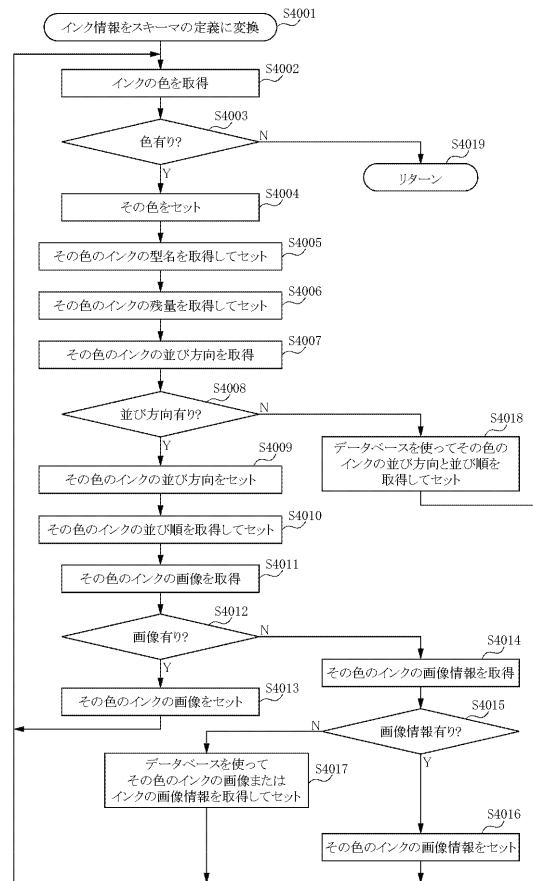
-PCから発行する要求コマンド-	-プリンタから返されるインク情報-
Ink;	Ink:Color,CI-Color,40,BottomTop,1,"CI-Color.bmp", Photo,CI-Photo,80,BottomTop,2,"CI-Photo.bmp";



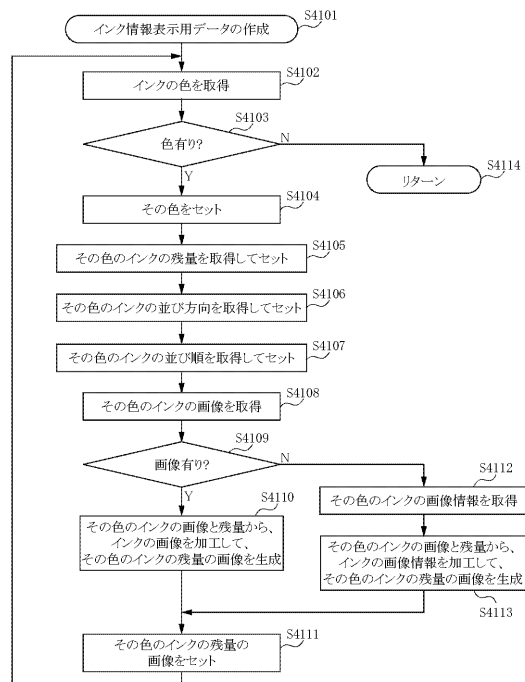
【図 39】



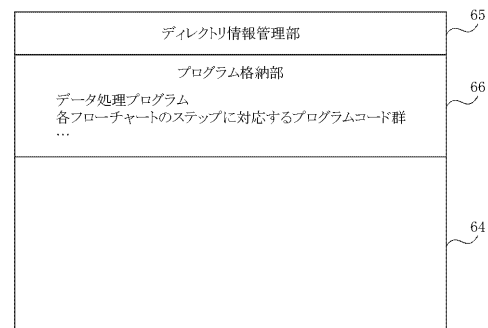
【図 40】



【図 41】



【図 42】





【図 4 3】

Definition: InkInfo  
Node Type:Property  
Description: インクの情報  
Full Schema Path: \Printer.InkInfo  
Definition: [Color]  
Node Type:Property  
Description: 色の情報  
Full Schema Path: \Printer.InkInfo.[Color]  
Allowed Values: Black //黒 Yellow //黄  
Cyan //シアン Photo //フォト  
Magenta //マゼンタ Color //カラー

Definition: Installed  
Node Type:Value  
Data Type:Boolean  
Description: [Color]色のインクが搭載されているか否か  
Full Schema Path: \Printer.InkInfo.[Color].Installed  
Allowed Values: True //搭載されている  
False //搭載されていない

Definition: State  
Node Type:Value  
Data Type:String  
Description: [Color]色のインクの残量の状態  
Full Schema Path: \Printer.InkInfo.[Color].State  
Allowed Values: Full //インク満タン Out //インク無し  
Low //インク残りわずか Unknown //インク残量不明

Definition: ModelName  
Node Type:Value  
Data Type:String  
Description: [Color]色のインクの型名  
Full Schema Path: \Printer.InkInfo.[Color].ModelName  
Examples: CI-B //黒インク CI-Y //黄インク  
CI-C //シアンインク CI-Color //一体型のカラーインク  
CI-M //マゼンタインク CI-Photo //一体型のフォトインク

Definition: Direction  
Node Type:Value  
Data Type:String  
Description: インクの並び方向  
Full Schema Path: \Printer.InkInfo.[Color].Direction  
Allowed Values: LeftRight //左から右  
FrontBack //手前から後ろ  
BottomTop //下から上

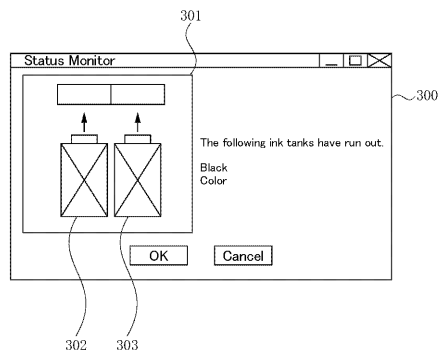
Definition: Order  
Node Type:Value  
Data Type:Int  
Description: 原点からのインクの順番  
Full Schema Path: \Printer.InkInfo.[Color].Order  
Examples: 1 //1番目  
2 //2番目

Definition: InsertDirection  
Node Type:Value  
Data Type:String  
Description: インクの挿入方向  
Full Schema Path: \Printer.InkInfo.[Color].InsertDirection  
Allowed Values: LeftToRight //左から右へ  
RightToLeft //右から左へ  
FrontToBack //手前から後ろへ  
BackToFront //後ろから手前へ  
BottomToTop //下から上へ  
TopToBottom //上から下へ

【図 4 4】

<Enumeration> -Query(Schema)- \Printer.InkInfo	-Retrieve(Schema)-	-Retrieve(Value)-
	\Printer.InkInfo.Black:Installed	True
	\Printer.InkInfo.Black:State	Out
	\Printer.InkInfo.Black:ModelName	CI-B
	\Printer.InkInfo.Black:Direction	LeftRight
	\Printer.InkInfo.Black:Order	1
	\Printer.InkInfo.Black:InsertDirection	BottomToTop
	\Printer.InkInfo.Color:Installed	True
	\Printer.InkInfo.Color:State	Out
	\Printer.InkInfo.Color:ModelName	CI-Color
	\Printer.InkInfo.Color:Direction	LeftRight
	\Printer.InkInfo.Color:Order	2
	\Printer.InkInfo.Color:InsertDirection	BottomToTop

【図 4 5】

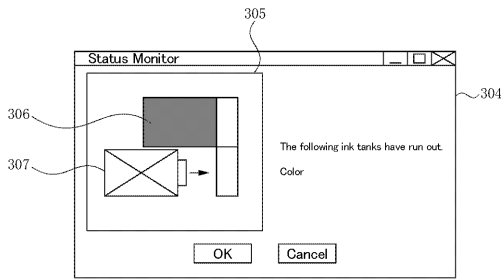


【図 4 6】

<Enumeration> -Query(Schema)- \Printer.InkInfo	-Retrieve(Schema)-	-Retrieve(Value)-
	\Printer.InkInfo.Black:Installed	True
	\Printer.InkInfo.Black:State	Full
	\Printer.InkInfo.Black:ModelName	CI-B
	\Printer.InkInfo.Black:Direction	BottomTop
	\Printer.InkInfo.Black:Order	2
	\Printer.InkInfo.Black:InsertDirection	LeftToRight
	\Printer.InkInfo.Color:Installed	True
	\Printer.InkInfo.Color:State	Out
	\Printer.InkInfo.Color:ModelName	CI-Color
	\Printer.InkInfo.Color:Direction	BottomTop
	\Printer.InkInfo.Color:Order	1
	\Printer.InkInfo.Color:InsertDirection	LeftToRight



【図 47】



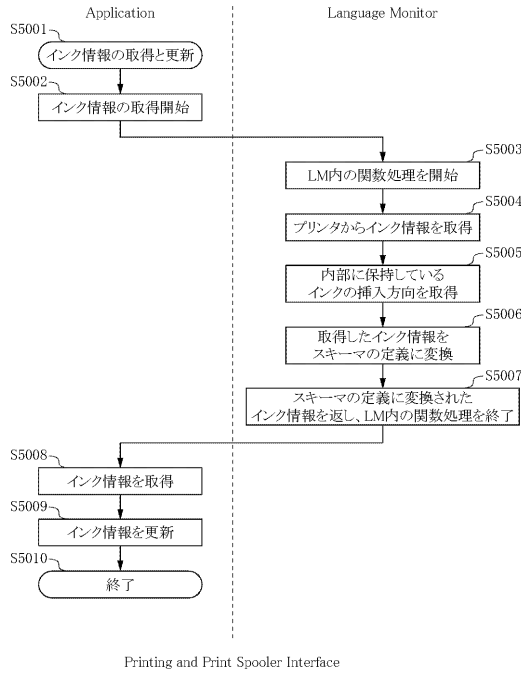
【図 48】

—PCから発行する要求コマンド—プリンタから返されるインク情報—  
Ink:Black,CI-B,Out,LeftRight,1,Color:CI-Color, Out, LeftRight,2;  
Ink;

【図 49】

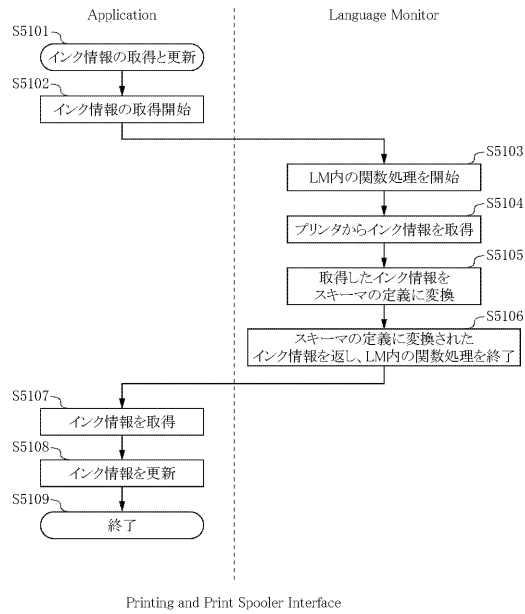
—PCから発行する要求コマンド—プリンタから返されるインク情報—  
Ink:Black,CI-B,Out,LeftRight,1,BottomToTop,Color:CI-Color, Out, LeftRight,1,BottomToTop;  
Ink;

【図 50】

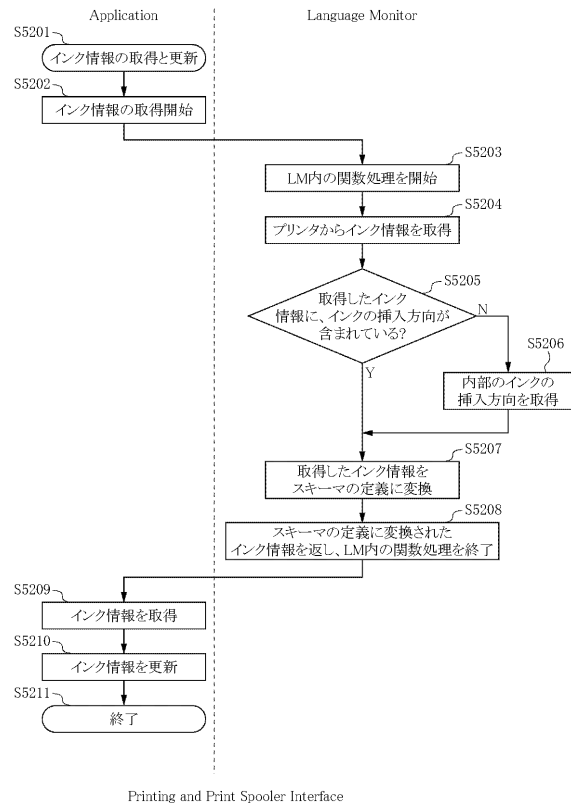




【図 5 1】



【図 5 2】





---

フロントページの続き

(56)参考文献 特開平 09 - 131897 (JP, A)  
特開 2004 - 170810 (JP, A)  
特開 2002 - 108597 (JP, A)  
特開 2003 - 271321 (JP, A)  
特開 2004 - 058645 (JP, A)  
特開 2003 - 187123 (JP, A)

(58)調査した分野(Int.Cl., DB名)  
G 0 6 F      3 / 1 2  
B 4 1 J      2 9 / 3 8