

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2006-268752
(P2006-268752A)

(43) 公開日 平成18年10月5日(2006.10.5)

(51) Int. Cl. F I テーマコード(参考)
G06F 9/445 (2006.01) G06F 9/06 650C 5B076
 5B176

審査請求 未請求 請求項の数 10 O L (全 17 頁)

(21) 出願番号	特願2005-89606 (P2005-89606)	(71) 出願人	000002369 セイコーエプソン株式会社 東京都新宿区西新宿2丁目4番1号
(22) 出願日	平成17年3月25日(2005.3.25)	(74) 代理人	100096703 弁理士 横井 俊之
		(74) 代理人	100117466 弁理士 岩上 涉
		(72) 発明者	齋藤 雅典 長野県松本市芳川村井町1059番地 株 式会社エプソンソフト開発センター内
		Fターム(参考)	5B076 AB17 AC07 5B176 AB17 AC07

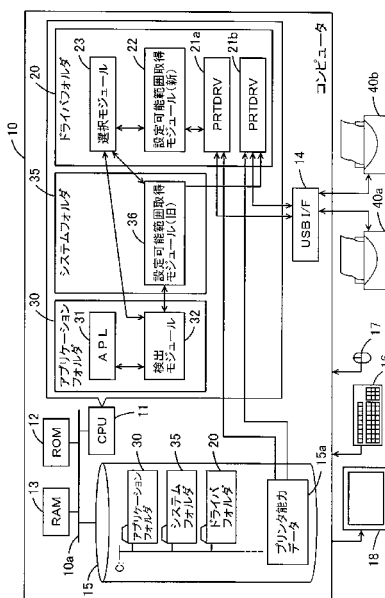
(54) 【発明の名称】 複数のバージョンに対応したモジュールの起動

(57) 【要約】

【課題】 複数のプログラムモジュールが協働して所定の機能を実現する構成において容易にバージョンを整合させることができるように改善を行うことが望まれていた。

【解決手段】 アプリケーションプログラムから指示された処理をプログラムモジュールに実行させるにあたり、上記アプリケーションプログラムから、所定の処理の実行指示を取得し、上記所定の処理を実行する複数のバージョンのプログラムモジュールから、上記実行指示がなされた処理を実行するプログラムモジュールを選択し、当該選択されたプログラムモジュールに上記所定の処理を実行させる。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

アプリケーションプログラムから指示された処理をプログラムモジュールに実行させるための中間処理装置であって、

上記アプリケーションプログラムから、所定の処理の実行指示を取得する実行指示取得手段と、

上記所定の処理を実行する複数のバージョンのプログラムモジュールから、上記実行指示がなされた処理を実行するプログラムモジュールを選択するモジュール選択手段と、

当該選択されたプログラムモジュールに上記所定の処理を実行させる実行制御手段とを備えることを特徴とする中間処理装置。

10

【請求項 2】

上記複数のバージョンのプログラムモジュールは、あるバージョンのプログラムのモジュールとそれより古いバージョンのプログラムモジュールとの少なくとも 2 つのバージョンが存在し、これらのいずれかまたは組み合わせが実行可能な状態であることを特徴とする上記請求項 1 に記載の中間処理装置。

【請求項 3】

上記所定の処理は、コンピュータの周辺機器に関する所定の問い合わせとこの問い合わせに対する返答を行う処理であることを特徴とする上記請求項 1 または請求項 2 のいずれかに記載の中間処理装置。

【請求項 4】

上記プログラムモジュールは、コンピュータの周辺機器の制御プログラムに対して問い合わせを行い、この制御プログラムからの返答を取得して上記アプリケーションプログラムに対して返答することを特徴とする上記請求項 1 ~ 請求項 3 のいずれかに記載の中間処理装置。

20

【請求項 5】

上記実行指示取得手段はコンピュータの周辺機器の指定とともに上記実行指示を受け付け、上記モジュール選択手段は指定された周辺機器に対応づけられたプログラムモジュールを選択することを特徴とする上記請求項 1 ~ 請求項 4 のいずれかに記載の中間処理装置。

【請求項 6】

所定の処理を他のプログラムモジュールに実行させる情報処理装置であって、

複数のバージョンのプログラムモジュールから実行すべきプログラムモジュールを選択するための処理が実行可能であるか否かを判断する実行可否判断手段と、

実行可能であると判断されないときに、特定のバージョンのプログラムモジュールが実行可能であるか否かを判断し、当該特定のバージョンのプログラムモジュールに上記所定の処理を実行させる特定モジュール起動手段とを備えることを特徴とする情報処理装置。

30

【請求項 7】

アプリケーションプログラムから指示された処理をプログラムモジュールに実行させるための中間処理方法であって、

上記アプリケーションプログラムから、所定の処理の実行指示を取得する実行指示取得工程と、

上記所定の処理を実行する複数のバージョンのプログラムモジュールから、上記実行指示がなされた処理を実行するプログラムモジュールを選択するモジュール選択工程と、

当該選択されたプログラムモジュールに上記所定の処理を実行させる実行制御工程とを備えることを特徴とする中間処理方法。

40

【請求項 8】

所定の処理を他のプログラムモジュールに実行させる情報処理方法であって、

複数のバージョンのプログラムモジュールから実行すべきプログラムモジュールを選択するための処理が実行可能であるか否かを判断する実行可否判断工程と、

実行可能であると判断されないときに、特定のバージョンのプログラムモジュールが実行可能であるか否かを判断し、当該特定のバージョンのプログラムモジュールに上記所定

50

の処理を実行させる特定モジュール起動工程とを備えることを特徴とする情報処理方法。

【請求項 9】

アプリケーションプログラムから指示された処理をプログラムモジュールに実行させるための中間処理プログラムであって、

上記アプリケーションプログラムから、所定の処理の実行指示を取得する実行指示取得機能と、

上記所定の処理を実行する複数のバージョンのプログラムモジュールから、上記実行指示がなされた処理を実行するプログラムモジュールを選択するモジュール選択機能と、

当該選択されたプログラムモジュールに上記所定の処理を実行させる実行制御機能とをコンピュータに実現させることを特徴とする中間処理プログラム。

10

【請求項 10】

所定の処理を他のプログラムモジュールに実行させる情報処理プログラムであって、

複数のバージョンのプログラムモジュールから実行すべきプログラムモジュールを選択するための処理が実行可能であるか否かを判断する実行可否判断機能と、

実行可能であると判断されないときに、特定のバージョンのプログラムモジュールが実行可能であるか否かを判断し、当該特定のバージョンのプログラムモジュールに上記所定の処理を実行させる特定モジュール起動機能とをコンピュータに実現させることを特徴とする情報処理プログラム。

【発明の詳細な説明】

20

【技術分野】

【0001】

本発明は、複数のバージョンに対応したモジュールの起動に関する。

【背景技術】

【0002】

コンピュータで利用されるアプリケーションプログラムは、その機能追加やアプリケーションプログラムによって制御する周辺機器の機能追加等に伴ってバージョンアップがなされる。従来、バージョンアップに際しては、古いバージョンのアプリケーションプログラムを一旦削除して、新しいバージョンのプログラムをインストールしたり、古いバージョンのプログラムを新しいバージョンのプログラムで行使するなどしていた（例えば、特許文献 1）。

30

【特許文献 1】特開 2003 - 256212 号

【発明の開示】

【発明が解決しようとする課題】

【0003】

従来技術においては、複数のプログラムモジュールが協働して所定の機能を実現する構成において容易にバージョンを整合させることができるように改善を行うことが望まれていた。すなわち、あるアプリケーションプログラムのみに着目した場合、バージョンの変更は、インストーラ等によって容易に実行可能であるが、アプリケーションプログラムと周辺機器のドライバプログラムと両者の仲介を行う中間プログラムとを利用する場合など、複数のプログラムモジュールが協働する場合には、これらのバージョンを整合させることが困難であった。

40

【0004】

例えば、周辺機器を提供するメーカーが上記ドライバプログラムと中間プログラムとを提供する際には、両者におけるバージョンの整合は容易である。しかし、複数のメーカーが提供する複数のアプリケーションプログラムの処理について中間プログラムが仲介を行う場合には、両者の整合を行うことが困難である。すなわち、ドライバプログラムおよび中間プログラムのバージョン改訂に伴って、総てのアプリケーションプログラムのメーカーにアプリケーションプログラムの変更を要請することは困難である。また、新しいバージョンのアプリケーションプログラムを利用するとしても、ユーザがドライバプログラム

50

および中間プログラムのバージョンアップをしない場合には、やはり両者は整合しない。このように、バージョンの整合がなされていないと、プログラムの実行が不可能になる場合がある。

本発明は、上記課題にかんがみてなされたもので、複数のプログラムモジュールが協働して所定の機能を実現する構成においてバージョンにかかわらずプログラムの実行を可能にすることを目的とする。

【課題を解決するための手段】

【0005】

上記目的を達成するため本発明では、アプリケーションプログラムから所定の処理の実行指示がなされたときに、複数のバージョンのプログラムモジュールからプログラムモジュールを選択して当該所定の処理を実行させる。従って、複数のバージョンのプログラムモジュールが混在し得る状況であっても、当該所定の処理を実行可能なプログラムモジュールを選択し、所定の処理を実行させることができる。この結果、適切なバージョンのプログラムモジュールを起動し、確実に上記所定の処理を実行することが可能になる。

【0006】

ここで、アプリケーションプログラムは、コンピュータに各種の処理を実行させるプログラムであれば良く、処理の目的や動作は特に限定されない。但し、本発明においては、各種処理の一つとして他のプログラムモジュールに指示を行い、その処理を行わせるような処理が存在する。すなわち、本発明にかかる中間処理装置は、アプリケーションプログラムと協働する他のプログラムモジュールを起動する。このとき、アプリケーションプログラムのバージョンとプログラムモジュールのバージョンとが整合していない場合には、上記所定の処理を実行できない場合がある。

【0007】

例えば、バージョンアップによって新しい処理が追加された場合、この処理を実行するためにはアプリケーションプログラムから当該新しい処理の実行指示を行い、バージョンアップされたプログラムモジュールでこの処理を実行する。ここで、古いバージョンのアプリケーションプログラムから新しい処理の実行指示を行うことができなければ、新しいバージョンのプログラムモジュールによって上記新しい処理を実行することはできない。

【0008】

しかし、最新のバージョンにおける動作を保証するために、古いバージョンのプログラムを削除し、最新のバージョンのアプリケーションプログラムと最新のバージョンのプログラムモジュールとを実行可能にするのみでは、それ以前のバージョンにおける処理が実行可能であることは補償されない。そこで、複数のバージョンのプログラムモジュールを併用可能に構成すると、アプリケーションプログラムとプログラムモジュールとのバージョンに不整合が生じ得る。そこで、複数のバージョンのプログラムモジュールが併存し得るように構成し、上記実行指示がなされた所定の処理を実行可能なプログラムモジュールをモジュール選択手段によって選択することにより、新旧、複数のバージョンのプログラムモジュールを併用することが可能になる。

【0009】

すなわち、古いバージョンのプログラムモジュールのみにて実現される処理が存在する場合、新しいバージョンのプログラムモジュールと古いバージョンのプログラムモジュールとを併存させなければ古いバージョンのプログラムモジュールによって実現される処理を実行することができない。例えば、各バージョンのプログラムモジュールによって各種の周辺機器を制御し、各周辺機器の機種に各バージョンのプログラムモジュールが対応している場合、最新の機種に対応しているプログラムモジュールのみを実行可能にすると、古いバージョンのプログラムモジュールに対応している周辺機器を使用することができない。従って、以上の効果は、複数のバージョンのプログラムモジュールが実行可能に構成されている場合に特に顕著に現れる。

【0010】

また、アプリケーションプログラムをバージョンアップしたが、プログラムモジュール

をバージョンアップしていない場合にも有効である。すなわち、各プログラムモジュールのバージョンアップにおいて、旧バージョンの総ての処理および新たな処理を実施可能なプログラムモジュールを作成することを想定する。この場合、新旧バージョンで共通の処理を実施可能であるが、当該共通の処理の実行指示がなされたときに、バージョンアップ後のプログラムモジュールが実行可能であることを要請すると、新たに追加された処理のみならず共通の処理も実施することができない。

【0011】

しかし、プログラムモジュールのバージョンアップをしていないとしても、旧バージョンのプログラムモジュールに処理を実行させれば、当該共通の処理を実行させることができる。従って、新たなバージョンのプログラムモジュールが存在しないときに、モジュール選択手段によって旧バージョンのプログラムモジュールを選択し、処理を実行させることで、プログラムモジュールをバージョンアップしていない場合であっても、上記共通の処理を実行することが可能である。

10

【0012】

上記実行指示取得手段においては、上記プログラムモジュールで実行可能な処理のいずれかを実行させるための指示を取得すればよい。この指示は特に限定されず、各種の指示を採用可能である。例えば、アプリケーションプログラムによって印刷時のUIを表示するために、UIに表示すべき項目について表示すべき選択肢を取得するための処理等を採用可能である。本発明では、アプリケーションプログラムと別のプログラムモジュールにて処理を実行することを想定しているため、本発明における実体的な処理はアプリケーションプログラムとは別に完結されることが好ましい。その一例としては、当該プログラムモジュールと他のプログラムモジュール（例えば、周辺機器の制御プログラム（ドライバプログラム））とが協働して処理を実行するような場合が挙げられる。

20

【0013】

モジュール選択手段は、複数のバージョンのプログラムモジュールから、実行指示がなされた処理を実行するプログラムモジュールを選択することができれば良い。ここで、プログラムモジュールを選択するための基準は、各種の基準を採用可能である。例えば、アプリケーションプログラムから取得する実行指示とともに処理の実行対象（例えば、周辺機器）の指示を受け付け、この実行対象に対応したプログラムモジュールを選択する構成を採用可能である。

30

【0014】

他にも、実行可能なプログラムモジュールの中で最新のものを選択したり、アプリケーションプログラムのバージョンに対応したバージョンのプログラムモジュールを検索して選択しても良い。さらに、各処理の内容に応じて、各処理を実行可能なプログラムモジュールを抽出して選択しても良く、種々の手法を採用可能である。実行制御手段においては、このようにして選択したプログラムモジュールに対して処理を実行させる指示を行えばよい。

【0015】

本発明の適用対象としては、種々の状況が想定されるが、特に好ましい状況として、あるバージョンのプログラムのモジュールとそれより古いバージョンのプログラムモジュールとの少なくとも2つのバージョンが存在し、これらのいずれかまたは組み合わせが実行可能な状態である場合を想定可能である。すなわち、新旧、少なくとも2つのバージョンが存在し、双方が実行可能であれば、少なくとも2つのバージョンが併存している状況下で、モジュール選択手段にて適切なバージョンを選択して実行させることができる。

40

【0016】

旧バージョンのみが実行可能であれば、旧バージョンに対応したアプリケーションプログラムからの実行指示に対応して処理を実行させることが可能であるし、新バージョンに対応したアプリケーションプログラムからの実行指示であっても、新旧バージョンに共通の処理については、実行させることが可能である。新バージョンのみが実行可能であれば、新バージョンに対応したアプリケーションプログラムからの実行指示に対応して処理を

50

実行させることが可能である。この場合、旧バージョンに対応したアプリケーションプログラムからの実行指示に対応して処理を実行させることはできないが、エラー等を入力することにより、実施不可能であることを明示することは可能である。

【0017】

上記所定の処理の好ましい例としては、コンピュータの周辺機器に関する所定の問い合わせとこの問い合わせに対する返答を行う処理が挙げられる。すなわち、アプリケーションプログラムにおいて周辺機器に関する処理を実行する際には、周辺機器にて実行可能な処理を把握した上でその中の処理を選択することがある。この場合、アプリケーションプログラムにて予め当該実行可能な処理の総てを把握することは困難であるため、実行可能な処理を指示する際に問い合わせを行うことが多い。

10

【0018】

例えば、所定のUIを表示させるに際して、UIに表示すべき項目について表示すべき選択肢を取得するために問い合わせを行う。この問い合わせの返答は周辺機器毎に異なり得るので、プログラムモジュールを介して問い合わせを行うことにより、周辺機器毎に正確な情報を取得することが可能である。また、周辺機器とプログラムモジュールのバージョンは対応していることが多い。すなわち、ある周辺機器の機能向上に伴って機種が変更されたときにプログラムモジュールがバージョンアップされることがある。従って、本発明のようにプログラムモジュールを選択可能に構成することにより、周辺機器における種々の機種に対応した処理を実行可能である。

【0019】

上記周辺機器としては、コンピュータに接続されて駆動される機器であればよく、種々の機器がこの周辺機器に該当する。例えば、コンピュータに接続されて画像の出力動作を行うプリンタやディスプレイ、プロジェクト等の出力機器や、コンピュータに接続されて画像の入力動作を行うスキャナやデジタルカメラ等の入力機器や、コンピュータに接続されたデータの記録動作を行うハードディスクドライブ等の記録機器等、種々の機器が周辺機器となり得る。

20

【0020】

上述の例においてUIに表示すべき項目は、周辺機器にて実行可能な処理に対応しており、通常は、周辺機器を制御するための制御プログラム(ドライバプログラム)に対して問い合わせを行うか、周辺機器と通信を行うことによって上記項目に関する情報を取得可能である。従って、プログラムモジュールにおいては、ドライバプログラムや周辺機器に対して問い合わせを行い、その返答を取得し、さらにアプリケーションプログラムに対して返答を返すように構成すればよい。

30

【0021】

尚、アプリケーションプログラムから、直接、ドライバプログラムに対する問い合わせを行うことも想定し得るが、この構成においては、アプリケーションプログラムに対する問い合わせのフォーマットや問い合わせに対する返答のフォーマットが常に一致している必要がある。従って、ドライバプログラムのバージョンアップに伴ってアプリケーションプログラムの変更が必要になり得る。

【0022】

しかし、本発明のように、プログラムモジュールによってアプリケーションプログラムとドライバプログラムとの仲介を行うことにより、アプリケーションプログラムからプログラムモジュールへの問い合わせおよび返答のフォーマットを常に同じフォーマットにすることができる。すなわち、ドライバプログラムが変更された場合には、プログラムモジュールの変更によって対応することができる。この構成は、ドライバプログラムとプログラムモジュールとを作成するメーカーが同じメーカーであり、アプリケーションプログラムは多数のメーカーによって提供される場合に好適である。

40

【0023】

さらに、プログラムモジュールが周辺機器に関する処理を行う場合に好適な構成として、上記実行指示取得手段においてコンピュータの周辺機器の指定とともに上記実行指示を

50

受け付ける構成を採用可能である。かかる構成によれば、上記モジュール選択手段は、容易に指定された周辺機器に対応づけられたプログラムモジュールを選択することができる。すなわち、プログラムモジュールのバージョンが周辺機器と一対一に対応している場合に、周辺機器に基づいてプログラムモジュールを選択すれば、確実に適切なプログラムモジュールを選択することができる。尚、アプリケーションプログラムによって周辺機器に関する処理を実施する際には、処理の実行指示とともに周辺機器を指定するのが通常であるため、通常の処理において実施される情報のみを利用して上述の構成を実現することが可能である。

【0024】

さらに、本発明にかかる中間処理装置と協働するアプリケーションプログラムを実行する情報処理装置として、上述のモジュール選択手段に相当する処理が実行可能であるか否かを判定し、実行不能であるときに、既存のバージョンのプログラムモジュールに対して処理の実行指示を与える構成を採用することが可能である。すなわち、本発明にかかるモジュール選択手段が実施可能でない場合（モジュール選択手段に相当するプログラムがインストールされていない場合等）であっても、古いバージョンのプログラムモジュールが実行可能である場合がある。このような状況は古いバージョンのプログラムモジュールが実行可能であり、アプリケーションプログラムから直接呼び出して実行可能である場合に発生し得る。

10

【0025】

そこで、実行可否判断手段において、複数のバージョンのプログラムモジュールから実行すべきプログラムモジュールを選択するモジュール選択手段が実行可能であるか否かを判別する。モジュール選択手段が実行可能でない場合には、特定のバージョンのプログラムモジュールが実行可能であるか否かを判断し、当該特定のバージョンのプログラムモジュールに上記所定の処理を実行させる。この結果、モジュール選択手段が実行不可能であるが、特定のバージョンのプログラムモジュールが実行可能であれば、そのプログラムモジュールに所定の処理を実行させることが可能である。

20

【0026】

ところで、上述した中間処理装置や情報処理装置は、単独で実施される場合もあるし、ある機器に組み込まれた状態で他の方法とともに実施されることもあるなど、発明の思想としては各種の態様を含むものであって、適宜、変更可能である。また、複数のバージョンのプログラムモジュールのいずれかを選択して起動する手順に発明が存在するとも言える。したがって、本発明は方法としても適用可能であり、請求項7、請求項8にかかる発明においても、同様の作用となる。また、本発明を実施しようとする際に、コンピュータにて所定のプログラムを実行させる場合もある。従って、本発明は、そのプログラムとしても適用可能であり、請求項9、請求項10にかかる発明においても、同様の作用となる。

30

【0027】

むしろ、請求項2～請求項5に記載された構成を上記方法やプログラムに対応させることも可能である。また、いかなる記憶媒体もプログラムを提供するために使用可能である。例えば、磁気記録媒体や光磁気記録媒体であってもよいし、今後開発されるいかなる記録媒体においても全く同様に考えることができる。また、一部がソフトウェアであって、一部がハードウェアで実現される場合においても本発明の思想において全く異なるものではなく、一部を記録媒体上に記録しておいて必要に応じて適宜読み込む形態のものも含まれる。さらに、一次複製品、二次複製品などの複製段階についても同等である。

40

【発明を実施するための最良の形態】**【0028】**

ここでは、下記の順序に従って本発明の実施の形態について説明する。

(1) 印刷制御装置の構成：

(2) 印刷制御処理およびUI表示例：

(2-1) 設定可能範囲取得処理：

50

(3) 他の実施形態：

【0029】

(1) 印刷制御装置の構成：

図1は、本発明の一実施形態にかかる印刷制御装置の概略構成を示している。本実施形態ではコンピュータ10の機能の一部によって印刷制御装置が実現される。コンピュータ10は演算処理の中枢をなすCPU11を備えており、このCPU11はシステムバス10aを介してコンピュータ10全体の制御を行う。同システムバス10aには、ROM12、RAM13、USB I/F14やハードディスクドライブ(HDD)15、図示しないCRT I/Fや入力機器I/F等が接続されている。

【0030】

ハードディスクドライブ15には、ソフトウェアとしてオペレーティングシステム(OS)等のプログラムが格納されており、また、必要に応じて各種プログラムをインストールして格納することができる。これらのソフトウェアは、実行時にCPU11によって適宜RAM13に転送される。CPU11は、RAM13を一時的なワークエリアとして適宜アクセスしながらOSの制御下で種々のプログラムを実行する。

【0031】

本実施形態においては、HDD15に格納するプログラムやデータがツリー構造のフォルダによって管理され、予め決められたフォルダに対してプログラムやデータを格納する。図1に示す例では、アプリケーションプログラム(APL31)に関するプログラムおよびデータがアプリケーションフォルダ30に格納され、プリンタドライバ(PRTDRV21a, 21b)に関するプログラムおよびデータがドライバフォルダ20に格納され、OSに関するプログラムおよびデータがシステムフォルダ35に格納されている例を示している。

【0032】

各プログラムおよびデータの格納場所は、予め決められていれば良く、各プログラムおよびデータの格納場所は限定されるわけではない。例えば、OSの提供者が設定した規則に従って格納場所を決めたり、プログラムの提供者が予め格納場所を決めればよい。いずれにしても、プログラムおよびデータが必要とされるときに適切なプログラムおよびデータを読み出すことができればよい。

【0033】

入力機器I/Fには、キーボード16やマウス17が操作入力機器として接続される。また、CRT I/Fには、表示用のディスプレイ18が接続されている。従って、コンピュータ10では、キーボード16やマウス17による操作内容を受け付け、また、ディスプレイ18に各種情報を表示することが可能である。さらに、USB I/F14には、プリンタ40a, 40bが接続されており、コンピュータ10が出力するデータに基づいて画像を印刷することが可能である。むしろ、プリンタ40a, 40bとの接続I/FはUSB I/Fに限られる必要もなく、パラレルI/F, シリアルI/F, SCSI接続など種々の接続態様を採用可能であるし、今後開発されるいかなる接続態様であっても同様である。

【0034】

APL31は予め決められた各種の処理を実行するプログラムであり、その処理の一つとして、APL31で扱う画像や文書等をプリンタ40a, 40bにて印刷させるための処理を行う。本実施形態においては、印刷実行を指示する前に所定のUIを表示し、UIでの指示に従って印刷を実行させる。このUIでは、プリンタ40a, 40bにおける各種印刷条件を設定可能であり、印刷条件として設定可能な項目(印刷条件項目)を表示するとともに各項目の設定可能範囲内で印刷条件を変更可能である。

【0035】

印刷条件項目は、印刷を実行するために必要な複数の条件の項目であり、印刷解像度、インクの種類(カラー、モノクロの選択やインクセット(インク色の組み合わせ)の選択)、印刷媒体の種類やサイズ、印刷時のレイアウト、縁なし印刷の可否、印刷品質(きれ

10

20

30

40

50

い、はやい等)、カラーマネジメントの種類(自動調整の可否やディスプレイの色域を超えた色の使用可否等)、双方向印刷の可否、ドライバによる画像処理の可否等、PRTDRV21a, 21bに対して設定可能な種々の印刷条件項目がある。尚、本実施形態においては、上述のように印刷解像度やインクの種類など、項目毎に印刷条件を設定するようになっているため異なる印刷条件を項目として区別する。また、各印刷条件の設定内容はその内容に対応した数値(設定値)で特定するようになっており、この数値を項目値と呼ぶ。

【0036】

印刷条件項目において設定可能な項目値は他の印刷条件項目の項目値に依存している場合があり、この場合、ある項目値を設定したことによって他の印刷条件項目で設定可能な項目値の範囲が変動する(依存関係がある)。そこで、APL31は、各項目値が設定された状態での設定可能範囲を取得し、ある項目値を設定したことによって設定可能範囲に影響がある他の印刷条件項目についてUIの表示を更新する。すなわち、取得した設定可能範囲のデータに従ってUI表示上の設定可能範囲を変動させる。

10

【0037】

APL31においては、上記設定可能範囲を取得するため、他のモジュールに設定可能範囲を問い合わせることが可能であり、この処理を行うためにアプリケーションフォルダ30には検出モジュール32が格納されている。尚、検出モジュール32はAPL31から呼び出されることによって実行され、APL31からの問い合わせに対して返答を行うためのモジュールが存在するか否かを検出するモジュールである。本実施形態においては、APL31と検出モジュール32とを別個のプログラムモジュールとして構成しているが、むろん、両者を一つのモジュールとして構成しても良い。

20

【0038】

本実施形態において、コンピュータ10には、異なる機種のプロンタ40a, 40bが接続可能であり、各プリンタ40a, 40bは、各機種に対応したPRTDRV21a, 21bによって制御される。PRTDRV21a, 21bは、プリンタ40a, 40bのそれぞれを制御するとともに、各プリンタにおける設定可能範囲を取得することが可能である。ここでは、各プリンタにおける設定可能範囲を取得することができればよいが、本実施形態においては、PRTDRV21a, 21bのインストール時に予め作成されたプリンタ能力データ15aを参照して設定可能範囲を取得するようになっている。

30

【0039】

図2は、プリンタ能力データ15aのデータ記述例を示す図である。同図に示すようにプリンタ能力データ15aにはプリンタ名、ドライババージョン、インクセット、設定可能範囲を示すデータが記述されている。設定可能範囲は"対象項目n, 上位項目の項目数m, 上位項目n1およびその項目値, , , 上位項目m1およびその項目値, 設定可能範囲"を示すデータを順に記述する書式となっている(m, n, lは0もしくは自然数)。

【0040】

これにより、上位依存関係(上位項目の項目値によって設定可能範囲が変動し得ること)にある項目値が特定の値であるときの対象項目の設定可能範囲を特定しており、例えば、対象項目nが印刷品質、上位項目が印刷媒体の種類、上位項目数mが一つの場合であって印刷媒体の種類項目値が"1"のときに印刷品質の項目値が"2, 3"である場合、"印刷品質, 1, 印刷媒体の種類=1, 2, 3"と記述することになる。尚、各項目値については、予めその値によって設定内容(例えば、印刷品質の項目値"1"のとき、高品質など)が決められている。むろん、このプリンタ能力データ15aの構成は一例であり、設定可能範囲を取得することができる限りにおいて、各種の構成を採用可能である。

40

【0041】

また、プリンタ能力データ15aがPRTDRV21a, 21bのインストール時点で作成されることは必須ではないし、プリンタ能力データ15aを予め作成するのではなく、設定可能範囲を取得する際にPRTDRV21a, 21bが各プリンタ40a, 40bに問い合わせる構成としても良い。

50

【 0 0 4 2 】

本実施形態において、設定可能範囲の問い合わせは、設定可能範囲取得モジュール 2 2 および設定可能範囲取得モジュール 3 6 によって実施される。すなわち、設定可能範囲取得モジュール 2 2 および設定可能範囲取得モジュール 3 6 が起動されると、U I の表示に必要な印刷条件項目の設定可能範囲を P R T D R V 2 1 a , 2 1 b のそれぞれに問い合わせ、当該設定可能範囲を取得する。取得した設定可能範囲は、U I の表示に利用される。

【 0 0 4 3 】

設定可能範囲取得モジュール 2 2 および設定可能範囲取得モジュール 3 6 は、プリンタの機種に対応しており、各プリンタを制御するための P R T D R V 2 1 a , 2 1 b に対して問い合わせを行う。本実施形態においては、プリンタの機種が P R T D R V 2 1 a , 2 1 b のバージョンおよび設定可能範囲取得モジュールのバージョンに対応している。図 1 において、設定可能範囲取得モジュール 2 2 および設定可能範囲取得モジュール 3 6 が実行可能に構成されているのは、異なる機種 of プリンタ 4 0 a , 4 0 b がコンピュータ 1 0 によって併用されているからである。

10

【 0 0 4 4 】

また、本実施形態において、設定可能範囲取得モジュール 3 6 は設定可能範囲取得モジュール 2 2 より古いバージョン、すなわち、発売時期が古いプリンタに対応したモジュールであり、設定可能範囲を取得するという意味では同じ機能を備えている。但し、P R T D R V 2 1 b が P R T D R V 2 1 a にバージョンアップされることによって新しい処理を実施することができるように構成する場合には、その点について新しいバージョンである設定可能範囲取得モジュール 2 2 は設定可能範囲取得モジュール 3 6 で実施不可能な処理を実施することが可能である。例えば、プリンタの機能向上によって設定可能範囲が増加した場合や印刷条件の項目が増加した場合、当該増加分の印刷条件の内容や項目値を示すデータを扱えるように構成される。

20

【 0 0 4 5 】

尚、本実施形態において、設定可能範囲取得モジュール 2 2 および設定可能範囲取得モジュール 3 6 は、対応する P R T D R V 2 1 a , 2 1 b をインストールする際に予め決められたフォルダに格納される。従って、図 1 に示すように設定可能範囲取得モジュール 2 2 および設定可能範囲取得モジュール 3 6 の双方が実行可能である場合には、P R T D R V 2 1 a , 2 1 b の双方がインストール済である。通常、インストールはプリンタ 4 0 a , 4 0 b のメーカーが提供する C D - R O M 等の媒体を利用して行われるが、O S の機能によって自動で P R T D R V 2 1 a , 2 1 b をインストールすることも可能である。この場合は、設定可能範囲取得モジュール 2 2 および設定可能範囲取得モジュール 3 6 がインストールされないこともあり得る。

30

【 0 0 4 6 】

上記設定可能範囲取得モジュール 3 6 と設定可能範囲取得モジュール 2 2 とでは設定可能範囲を問い合わせる対象の P R T D R V が異なり、また、P R T D R V 2 1 a , 2 1 b ではその処理が異なり得る。このため、本実施形態において、異なるプリンタに対する問い合わせは、異なる設定可能範囲取得モジュールによって実施されるように構成してある。但し、本実施形態において、P R T D R V 2 1 a , 2 1 b は下位互換性を持ち、古いバージョンの設定可能範囲取得モジュールからの問い合わせに対して、より新しいバージョンの P R T D R V は適切に設定可能範囲を返答するようになっている。

40

【 0 0 4 7 】

いずれにしても、本実施形態において、P R T D R V のバージョンと設定可能範囲取得モジュールのバージョンとが整合している場合には、対応しているバージョン同士で設定可能範囲の授受を行う。ドライバフォルダには、このようにバージョンを整合させるため、選択モジュール 2 3 が格納されている。すなわち、選択モジュール 2 3 は、設定可能範囲の問い合わせが行われたときに指定されたプリンタに対応した設定可能範囲取得モジュールを選択して実行するようになっている。

【 0 0 4 8 】

50

また、指定されたプリンタに対応した設定可能範囲取得モジュールが検出されない場合、古いバージョンの設定可能範囲取得モジュールを検索し、実行可能であればそのモジュールを起動する。この構成により、プリンタの機種に対応した適切なバージョンの設定可能範囲取得モジュールが存在する場合にはそのモジュールを起動し、バージョンが整合しないときには、その下位互換性を利用して古いバージョンで実現されていた処理を最低限実現するようにしてある。

【 0 0 4 9 】

むろん、選択モジュール 2 3 においては、A P L 3 1 にて実行指示がなされた処理を実行可能な設定可能範囲取得モジュールを選択することができればよいので、プリンタの機種名に基づいて対応する設定可能範囲取得モジュールを選択しても良い。但し、上述のように下位互換性を利用する場合には、最新のバージョンについてのみプリンタの機種名に基づいて対応する設定可能範囲取得モジュールを検出し、古いモジュールについてはモジュールの存在有無を検出する構成を採用することが好ましい。

10

【 0 0 5 0 】

上述の検出モジュール 3 2 は、上記選択モジュール 2 3 が実行可能であるか否かを検出するモジュールであり、選択モジュール 2 3 が実行可能であることを検出したときには、当該選択モジュール 2 3 を起動する。また、選択モジュール 2 3 が実行可能であることを検出しないときには、設定可能範囲取得モジュール 3 6 が実行可能であるか否かを検出する。すなわち、選択モジュール 2 3 が実行可能でない場合には、最新のバージョンである設定可能範囲取得モジュール 2 2 が存在しない場合であるので、それより古いバージョンである設定可能範囲取得モジュール 3 6 を検索し、設定可能範囲取得モジュール 3 6 が存在すれば起動する。

20

【 0 0 5 1 】

この構成により、設定可能範囲取得モジュール 2 2 がインストールされてない場合であっても、古いバージョンで実現されていた処理は最低限実現するようにしてある。尚、以上の構成において、各モジュールが実行可能であるか否かを判別するためには、予め決められたフォルダに予め決められたプログラムモジュールが格納されているか否かを判別するなど、種々の構成を採用可能である。

【 0 0 5 2 】

(2) 印刷制御処理および U I 表示例 :

30

次に、上記構成における印刷制御処理および U I 表示例を説明する。印刷制御処理は図 3 に示すフローに従って実施され、図 4 はこのときに表示される U I の例を示している。利用者は、A P L 3 1 によって扱う画像や文書等の印刷指示を行うことができ、A P L 3 1 が印刷指示を受け付けると (ステップ S 1 0 0)、A P L 3 1 は図示しない U I を表示し、当該指示にかかる印刷にて使用するプリンタのプリンタ名を取得する (ステップ S 1 0 5)。尚、プリンタ名としては O S 2 0 によって通常使用されるプリンタとして設定されているプリンタの機種名を取得すればよいが、むろん、印刷指示に際して利用者が特定のプリンタを指定したときにはその機種名を取得する。

【 0 0 5 3 】

次に、各印刷条件項目について現在の項目値 (現項目値) を取得する (ステップ S 1 1 0)。ここで、現項目値とは、ステップ S 1 1 0 ~ S 1 3 0 におけるループを実施している間に各印刷条件項目について設定されている項目値である。ループの初回であれば、前回の印刷実行時における印刷条件項目の項目値あるいはデフォルト設定されている項目値を現項目値として取得する。また、2 回目以降のループ処理であれば U I の操作によって適宜変更された項目値を現項目値として取得する。

40

【 0 0 5 4 】

現項目値を取得したら、U I 表示対象項目の設定可能範囲を取得する (ステップ S 1 1 5)。すなわち、U I 表示対象の全印刷条件項目について上位項目があるか否かを判別し、上位項目が存在する場合にはその上位項目について現項目値を設定し、このように上位項目の項目値を設定した状態で U I 表示対象の印刷条件項目がとり得る設定可能範囲を取

50

得する。設定可能範囲を取得するためのモジュールを選択する処理については後述する。

【0055】

UI表示対象項目の設定可能範囲を取得したら、当該取得した設定可能範囲を選択可能な選択肢にするとともに、上記現項目値が現在の設定値として設定された状態でUIを表示する(ステップS120)。このUIの表示においては、マウス17等の操作によって項目値の変更を受け付けたか否かを判別する(ステップS125)。項目値の変更を受け付けたと判別されたときには、その項目の項目値を変更し(ステップS130)、ステップS110以降の処理を繰り返す。

【0056】

この繰り返し処理においては、ステップS130の項目値の変更によってUI表示対象の印刷条件項目の設定可能範囲が変動したとしても、ステップS110、S115の処理により適切な設定可能範囲が取得され、ステップS120のUI表示では選択不能の印刷条件項目を選択可能に表示させることなくUIを更新することができる。

10

【0057】

本実施形態にかかるUIでは、利用者に選択を促す全印刷条件項目のうちの一部のみを一画面に表示し、一画面で少数の印刷条件項目を入力しながら画面を逐次切り替えて最終的に印刷実行指示を受け付けるようにしている。そこで、ステップS135では、画面の切替指示を受け付けたか否かを判別し、画面の切替指示を受け付けたと判別したときにはUI画面を切り替える(ステップS140)。そして、切り替え後のUI表示対象の印刷条件項目について適切な表示をするため、ステップS110以降を繰り返す。

20

【0058】

ステップS135にて画面の切替指示を受け付けたと判別しないときには、さらに、印刷実行指示を受け付けたか否かを判別する(ステップS145)。印刷実行指示を受け付けたと判別したときには、印刷対象の画像データをPRTDRV21aあるいはPRTDRV21bに対して出力する(ステップS150)。むろん、このとき、上記UIで設定された印刷条件項目の項目値で印刷を実行する。この結果、PRTDRV21a、21bは画像データに対して印刷条件に従った画像処理を施すとともに印刷条件に従って印刷を実行するための印刷データを作成し、プリンタ40a、40bに出力し、印刷を実行する。

【0059】

図4は、本実施形態におけるUI画面が切り替わる様子を示している。本実施形態にかかるUIでは、利用者に選択を促す全印刷条件項目のうちの一部のみを一画面に表示し、一画面で少数の印刷条件項目を入力しながら画面を逐次切り替えて最終的に印刷実行指示を受け付けるようにしている。図4に示す例においては印刷媒体のレイアウトを選択するUIをUI1と示している。当該UI1においては左側に印刷条件項目としての印刷媒体サイズを選択するフレーム1aを表示し、右側に印刷条件項目としてのレイアウトを選択するフレーム1bを表示している。また、図4のUI0は縁なし印刷(媒体の四辺に余白を設けない印刷)を実施するか否かを選択するUIである。

30

【0060】

尚、各UIにおいてはラジオボタンやアイコンのクリックによって各印刷条件項目を指定するようになっており、画面右下の「戻る」ボタンによって一つ前のUI画面に戻り、「次へ」ボタンによって次のUI画面に進むようになっており、最後のUIにおいては印刷実行指示ボタンによって印刷実行指示を実施できるようになっている。この例において、レイアウトは印刷媒体上での画像配置を規定しており、印刷媒体サイズはレイアウトの上位項目である。また、プリンタ40a、40bにおいては縁なし印刷を実施可能な印刷媒体のサイズが限られており、この意味で縁なし印刷の可否は印刷媒体の上位項目である。図4の例ではプリンタ40a、40bにおいて、はがき、L版、2L版、A3~A6の各印刷媒体サイズで印刷を実行可能であり、このうち、はがき、L版、2L版、A3、A4の各印刷媒体サイズで縁なし印刷を実施可能である。

40

【0061】

50

従って、UI 0において設定可能範囲として縁ありと縁なしを表示し、項目値として縁なしを選択しつつ次のUI 1に進んだときには、フレーム1 aにおいて選択可能範囲としてはがき、L版、2 L版、A 3、A 4を表示する。UI 0において項目値として縁ありを選択しつつ次のUI 1に進んだときには、フレーム1 aとして選択可能範囲としてはがき、L版、2 L版、A 3～A 6を表示する。また、UI 1のフレーム1 aにて印刷媒体サイズが選択されたときには、フレーム1 bにてレイアウトが自動的に更新され、当該選択中の印刷媒体サイズにて印刷実行可能なレイアウトのみが設定可能範囲として提示される。

【0062】

以上のように、本実施形態にかかるUIにおいてはAPL 31の制作者が所望の印刷条件項目を選択し、UIの表示対象として設定しておくのみで、UIを構成することができる。すなわち、APL 31においては、所望の印刷条件項目の設定可能範囲を問い合わせるようにしておけば、設定可能範囲取得モジュール22あるいは設定可能範囲取得モジュール36がその返答を行うので、APL 31の制作者は非常に容易にUIを構成することができる。さらに、将来、バージョンアップがなされたときであっても、上記選択モジュール23が適切な設定可能範囲取得モジュールを選択するので、APL 31が呼び出すモジュールのバージョンが不適切であることによって処理が実施できなくなることはない。むしろ、図4に示すUIの構成は一例であり、一画面で多数の印刷条件項目を表示させるUIを構成してもよいし、選択不可能な項目値をグレースアウトさせる構成を採用してもよい。

10

【0063】

(2-1) 設定可能範囲取得処理：

図5は上記ステップS 115に示す設定可能範囲取得処理を示すフローチャートであり、以下、このフローチャートに沿って設定可能範囲取得処理を説明する。設定可能範囲取得処理において、APL 31は、まず検出モジュール32を呼び出して実行する(ステップS 200)。検出モジュール32が実行されると、当該検出モジュール32は、ドライバフォルダ20に選択モジュール23が実行可能に格納されているか否かを判別する(ステップS 205)。

20

【0064】

ステップS 205にて選択モジュール23が実行可能であると判別されたときには、当該選択モジュール23を呼び出して実行する(ステップS 210)。選択モジュール23が実行されると、当該選択モジュール23は、上記ステップS 105で取得したプリンタ名に基づいて当該プリンタ名のプリンタが新しいバージョンの設定可能範囲取得モジュール22に対応しているか否かを判別する(ステップS 215)。すなわち、設定可能範囲取得モジュールが対応しているプリンタは予め決められているので、印刷実行時に指定されたプリンタが対応している新しいバージョンの設定可能範囲取得モジュールが存在するか否かを判別する。

30

【0065】

ステップS 215にて、指定されたプリンタが新しいバージョンの設定可能範囲取得モジュール22に対応していると判別されたときには、この設定可能範囲取得モジュール22を呼び出して実行する(ステップS 220)。尚、選択モジュール23においては、APL 31で実施しようとしている処理がUI表示対象の印刷条件項目について設定可能範囲を取得するための処理であることを検出し、当該設定可能範囲を取得するための設定可能範囲取得モジュールを呼び出している。従って、本実施形態においては、選択モジュール23が指示された処理を把握しており、この処理内容の把握が請求項における実行指示取得手段に相当する。むしろ、APL 31から取得する実行指示は設定可能範囲の取得処理に限られず、他の処理についてAPL 31から実行指示を取得することができるし、この場合は、当該他の処理を実施するためのプログラムモジュールを起動することになる。

40

【0066】

上記ステップS 220にて設定可能範囲取得モジュール22が実行されると、指定されたプリンタを制御するためのPRTDRV 21 aに対して上記UIの表示対象項目につい

50

て設定可能範囲を問い合わせる（ステップS 2 2 5）。この結果、P R T D R V 2 1 aは、プリンタ能力データ1 5 aを参照し、設定可能範囲を取得して設定可能範囲取得モジュール2 2に受け渡す。設定可能範囲取得モジュール2 2は、A P L 3 1に対してこの設定可能範囲を示すデータを返答として受け渡す（ステップS 2 3 0）。以上の処理によって、A P L 3 1が表示対象項目としている印刷条件についての設定可能範囲が取得され、適切にU Iが表示される。

【0 0 6 7】

上記ステップS 2 0 5にて選択モジュール2 3が実行可能であると判別されないとき、上記検出モジュール3 2は古いバージョンの設定可能範囲取得モジュールが実行可能であるか否か判別する。また、上記ステップS 2 1 5にてプリンタが新しいバージョンの設定可能範囲取得モジュール2 2に対応していると判別されないとき、選択モジュール2 3は古いバージョンの設定可能範囲取得モジュールが実行可能であるか否か判別する（ステップS 2 4 0）。このステップS 2 4 0にて古いバージョンの設定可能範囲取得モジュールが実行可能であると判別されないときには、設定可能範囲を取得することができないので、U I上にエラーである旨を表示して（ステップS 2 6 0）処理を終了する。むろん、ここでは、種々のエラー表示を行うことができ、適正なプリンタドライバをインストールするようにメッセージを表示しても良い。

10

【0 0 6 8】

ステップS 2 4 0にて、古いバージョンの設定可能範囲取得モジュールが実行可能であると判別されたときには、設定可能範囲取得モジュールを呼び出して実行する（ステップS 2 4 5）。尚、本実施形態において、当該古いバージョンの設定可能範囲取得モジュールは設定可能範囲取得モジュール3 6である。また、ステップS 2 1 5，S 2 4 0の処理が請求項におけるモジュール選択手段の処理に相当し、ステップS 2 2 0，S 2 4 5の処理が請求項における実行制御手段に相当する。

20

【0 0 6 9】

設定可能範囲取得モジュール3 6が実行されると、指定されたプリンタを制御するためのP R T D R V 2 1 bに対して上記U Iの表示対象項目について設定可能範囲を問い合わせる（ステップS 2 5 0）。この結果、P R T D R V 2 1 bは、プリンタ能力データ1 5 aを参照し、設定可能範囲を取得して設定可能範囲取得モジュール3 6に受け渡す。設定可能範囲取得モジュール3 6は、A P L 3 1に対してこの設定可能範囲を示すデータを返答として受け渡す（ステップS 2 5 5）。以上の処理によって、A P L 3 1が表示対象項目としている印刷条件についての設定可能範囲が取得され、適切にU Iが表示される。

30

【0 0 7 0】

選択モジュール2 3と設定可能範囲取得モジュール2 2とは同時にインストールされるので、以上の処理において、ステップS 2 0 5にて選択モジュール2 3が実行可能であると判別されない場合は、新しいバージョンの設定可能範囲取得モジュール2 2がインストールされていないことを意味する。しかし、この場合であっても、各種のプリンタに対するドライバがインストールされていることはあり得る。例えば、P R T D R VはO Sが提供する機能によってインストールできることも多い。

【0 0 7 1】

この場合、新しいバージョンに対応した設定可能範囲取得モジュール2 2は実行不能であるが、新しいバージョンのP R T D R Vに対応したプリンタにて印刷を実施可能であることになる。そこで、ステップS 2 0 5にて選択モジュール2 3が実行可能であると判別されない場合に、旧バージョンの設定可能範囲取得モジュール3 6を検索して実行すれば、設定可能範囲取得モジュール3 6で実行可能な処理は最低限実施することができる。すなわち、本実施形態において、P R T D R V 2 1 a，2 1 bは下位互換性を有するので、ステップS 1 0 5にてプリンタ4 0 a，4 0 bのいずれが指定されていたとしても、設定可能範囲取得モジュール3 6にて設定可能範囲を取得することが可能である。

40

【0 0 7 2】

また、ステップS 2 1 5にてプリンタが新しいバージョンの設定可能範囲取得モジュール

50

ル 2 2 に対応していると判別されない場合には、選択モジュール 2 3 が実行可能であるが、指定されたプリンタが新バージョンの設定可能範囲取得モジュール 2 2 に対応していない場合であるので、旧バージョンの設定可能範囲取得モジュール 3 6 を検索して実行する。この処理により、旧バージョンの設定可能範囲取得モジュール 3 6 に対応したプリンタ 4 0 b と新バージョンの設定可能範囲取得モジュール 2 2 に対応したプリンタ 4 0 a とを併用し、かつバージョンを整合させることが可能である。

【 0 0 7 3 】

以上の構成において、A P L 3 1 は検出モジュール 3 2 を備え、検出モジュール 3 2 によって選択モジュール 2 3 が実行可能であるか否かを判断している。従って、この A P L 3 1 は、選択モジュール 2 3 と設定可能範囲取得モジュール 2 2 とのバージョンに対応したアプリケーションプログラムであるといえる。しかし、上述のようにステップ S 2 0 5 , S 2 1 5 の判別を経て旧バージョンである設定可能範囲取得モジュール 3 6 を実行可能に構成している。従って、A P L 3 1 がコンピュータ 1 0 に対してプレインストールされており、設定可能範囲取得モジュール 3 6 についてインストールするか否かをエンドユーザの判断にゆだねることとしても、A P L 3 1 で U I を表示するにあたり、最低限の処理を実施することが可能である。

10

【 0 0 7 4 】

本実施形態においては、さらに、A P L において検出モジュール 3 2 に相当する処理を実施しなくても古いバージョンでの設定可能範囲取得モジュール 3 6 を実行することは可能である。すなわち、古いバージョンの A P L において、選択モジュール 2 3 を呼び出すことなく設定可能範囲取得モジュール 3 6 を直接呼び出す構成であれば、仮に、プリンタ 4 0 a , 4 0 b に対応する P R T D R V 2 1 a , 2 1 b がインストール済であって、古いバージョンの A P L が実行されたとしても、適切な設定可能範囲取得モジュール 3 6 を呼び出して実行することができる。

20

【 0 0 7 5 】

(3) 他の実施形態 :

本発明においては、モジュール選択手段によって適切なモジュールを選択してアプリケーションプログラムからの実行指示に対応するモジュールを呼び出すことができればよく、上記実施形態の他にも種々の構成を採用可能である。例えば、上述の設定可能範囲取得モジュールや選択モジュールの格納場所は上述の例に限定されないし、選択モジュール 2 3 と同様の処理を行うモジュールをシステムフォルダに格納しても良い。

30

【 0 0 7 6 】

さらに、P R T D R V 2 1 a , 2 1 b の下位互換性により、P R T D R V 2 1 a , 2 1 b の双方が設定可能範囲取得モジュール 3 6 とデータの授受を実行可能である実施形態について説明したが、P R T D R V と設定可能範囲取得モジュールと選択モジュールと A P L とのインストール状況および実行可能なバージョンの組み合わせとしては、多数の場合が発生し得る。そこで、各場合について適切な処理を行うため、設定可能範囲取得モジュール 3 6 と略同等の処理を行うモジュールを複数個実行可能にしても良い。

【 0 0 7 7 】

さらに、本発明は、プリンタ以外の周辺機器について本発明を適用することも可能である。すなわち、U I によって駆動条件を設定可能な総ての周辺機器について本発明を適用することができる。また、上記実施形態においてはコンピュータ 1 0 によって印刷制御装置を構成しているが、分散処理によって本発明にかかる印刷制御処理を実施するなど種々の構成を採用可能である。むろん、画像を取り込むスキャナと画像を印刷するプリンタとが一体となったいわゆる複合機において本発明にかかる印刷制御処理を行ってもよい。

40

【 図面の簡単な説明 】

【 0 0 7 8 】

【 図 1 】 印刷制御装置の概略構成を示すブロック図である。

【 図 2 】 プリンタ能力データの記述例を示す図である。

【 図 3 】 印刷制御処理のフローチャートである。

50

【図4】UIの表示例を示す図である。

【図5】設定可能範囲取得処理を示すフローチャートである。説明図である。

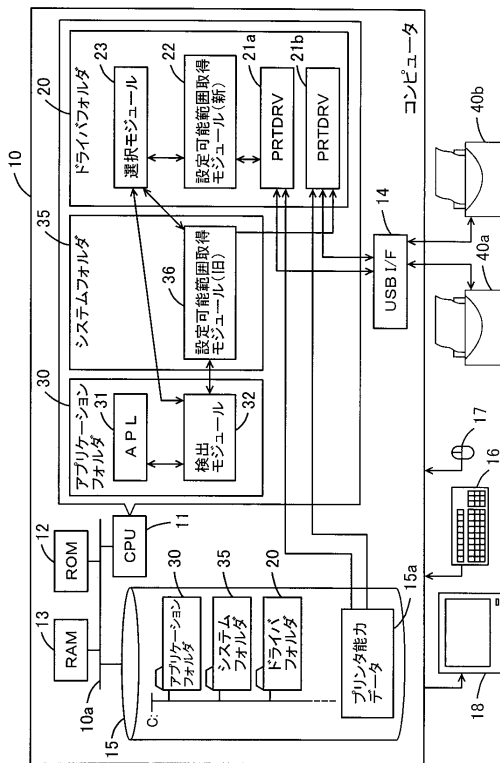
【符号の説明】

【0079】

10...コンピュータ、11...CPU、12...ROM、13...RAM、15...ハードディスクドライブ、15a...プリンタ能力データ、21a, 21b...PRTDRV、22...設定可能範囲取得モジュール、23...選択モジュール、31...APL、32...検出モジュール、36...設定可能範囲取得モジュール、40a, 40b...プリンタ

【図1】

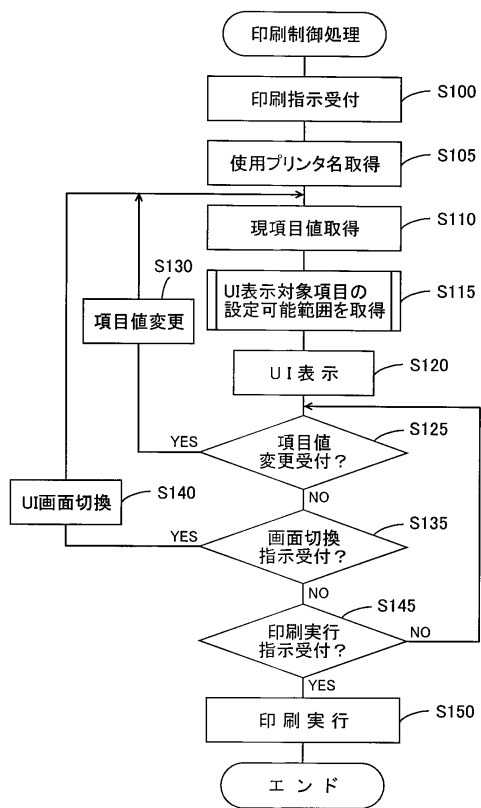
【図2】



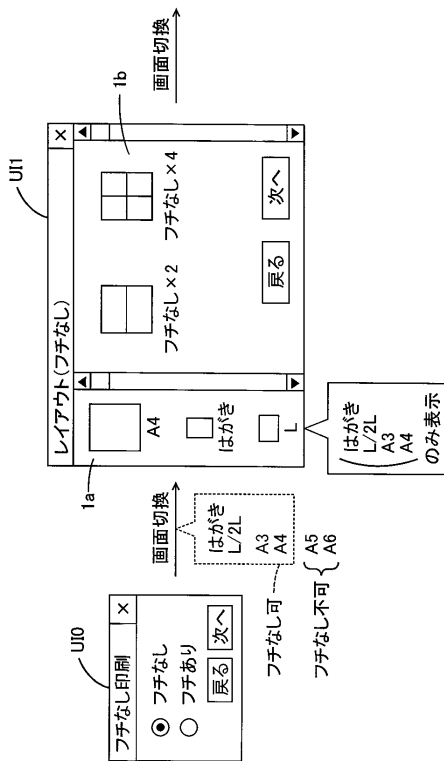
プリンタ能力データ

- ・プリンタ名: PM - * * * *
- ・ドライババージョン: 5.3e
- ・インクセット: 顔料, CMYK1clm
- ・設定可能範囲
 - ・対象項目1,2,上位項目1=2,上位項目2=1,1
 - ⋮
 - ・対象項目n,m,上位項目n1=1...上位項目m1=3,2,3

【 図 3 】



【 図 4 】



【 図 5 】

