

(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2009-9493  
(P2009-9493A)

(43) 公開日 平成21年1月15日(2009.1.15)

(51) Int.Cl.	F I	テーマコード (参考)
<b>G06F 3/12 (2006.01)</b>	G06F 3/12	2C061
<b>B41J 29/38 (2006.01)</b>	G06F 3/12	5B021
	B41J 29/38	Z

審査請求 未請求 請求項の数 13 O L (全 25 頁)

(21) 出願番号 特願2007-172288 (P2007-172288)  
(22) 出願日 平成19年6月29日 (2007. 6. 29)

(71) 出願人 000001007  
キヤノン株式会社  
東京都大田区下丸子3丁目30番2号  
(74) 代理人 100090538  
弁理士 西山 恵三  
(74) 代理人 100096965  
弁理士 内尾 裕一  
(72) 発明者 杉山 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  
Fターム(参考) 2C061 AP01 BB10 HH03 HJ06 HK03  
HK11 HN05 HN23 HR01  
5B021 AA01 AA02 NN00

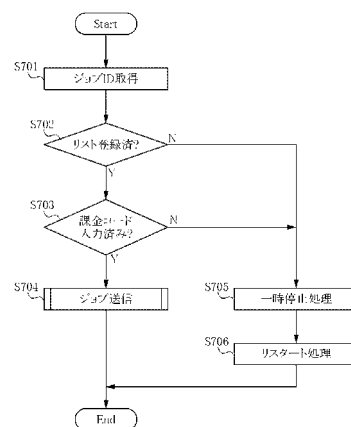
(54) 【発明の名称】 印刷制御装置、印刷制御方法、及びプログラム

(57) 【要約】

【課題】 特定の条件で印刷ジョブの実行を停止する環境において、印刷ジョブに対して、印刷再開できるように効率的に追加情報を設定し得る印刷システムは提供できていない。

【解決手段】 上記課題を解決するために本発明は、スプールされた印刷データのうち、追加情報（課金コード）の入力されている印刷データから印刷装置に送信可能に制御し、また、追加情報を入力するためのスプールされた印刷データの一覧を表示する。また、前記印刷データに関連する情報をログとして外部に送信する。

【選択図】 図7



**【特許請求の範囲】****【請求項 1】**

印刷アプリケーションから受信した順番で印刷データを記憶領域に保持し、保持された印刷データのうち、受信した順番が先頭の印刷データから印刷装置に送信する印刷制御装置において、

前記記憶領域で受信した順番が先頭の印刷データに課金コードが設定されている場合、当該印刷データを印刷装置に送信し、前記記憶領域で受信した順番が先頭の印刷データの課金コードが設定されていない場合、当該印刷データより前記記憶領域で受信した順番が後続となる印刷データが先頭になるように印刷データの順番を制御する制御手段と、

前記印刷アプリケーションから受信した印刷データの一覧を表示するための表示制御手段と、

前記表示制御手段により表示された印刷データの一覧から、ユーザの入力に基づき、少なくとも1つ以上の印刷データを選択し、印刷データに対して課金コードを設定する設定手段とを備えることを特徴とする印刷制御装置。

**【請求項 2】**

少なくとも課金コードを含む前記印刷データに関連する情報をログとして送信する送信手段を備えることを特徴とする請求項 1 記載の印刷制御装置。

**【請求項 3】**

前記設定手段により課金コードが設定された印刷データを優先して処理させるために、前記設定手段により課金コードが設定された印刷データの処理の優先度を高く設定する優先度設定手段とを備えることを特徴とする請求項 1 乃至請求項 2 のいずれか 1 項に記載の印刷制御装置。

**【請求項 4】**

前記記憶領域に存在する課金コードが設定されていない印刷データの記憶領域に存在した時間が閾値を超えると、当該印刷データを削除する削除手段を備えることを特徴とする請求項 1 乃至請求項 3 のいずれか 1 項に記載の印刷制御装置。

**【請求項 5】**

前記印刷装置が備える第 2 の記憶領域に存在する印刷データの情報を受信する受信手段と、

前記表示制御手段は、前記受信手段で受信した情報に基づき、前記印刷装置が備える第 2 の記憶領域に存在する印刷データの一覧を表示することを特徴とする請求項 1 乃至請求項 4 のいずれか 1 項に記載の印刷制御装置。

**【請求項 6】**

受信した順番で印刷データを記憶領域に保持し、保持された印刷データのうち、受信した順番が先頭の印刷データから印刷装置に送信する印刷制御方法であって、

前記記憶領域で受信した順番が先頭の印刷データに課金コードが設定されている場合、当該印刷データを前記印刷装置に送信し、前記記憶領域で受信した順番が先頭の印刷データの課金コードが設定されていない場合、当該印刷データより前記記憶領域で受信した順番が後続となる印刷データが先頭になるように印刷データの順番を制御する制御工程と、

前記受信した印刷データの一覧を表示するための表示制御工程と、

前記表示制御工程により表示された印刷データの一覧から、ユーザの入力に基づき、印刷データを選択し、印刷データに対して課金コードを設定する設定工程とを備えることを特徴とする印刷制御方法。

**【請求項 7】**

少なくとも課金コードを含む前記印刷データに関連する情報をログとして送信する送信工程とを備えることを特徴とする請求項 6 記載の印刷制御方法。

**【請求項 8】**

前記設定工程により課金コードが設定された印刷データを優先して処理させるために、前記設定工程により課金コードが設定された印刷データの処理の優先度を高く設定する優先度設定工程とを備えることを特徴とする請求項 6 乃至請求項 7 のいずれか 1 項に記載の

10

20

30

40

50

印刷制御方法。

【請求項 9】

前記記憶領域に存在する課金コードが設定されていない印刷データの記憶領域に存在した時間が閾値を超えると、当該印刷データを削除する削除工程を備えることを特徴とする請求項 6 乃至請求項 8 のいずれか 1 項に記載の印刷制御方法。

【請求項 10】

請求項 6 乃至請求項 9 のいずれか 1 項に記載の印刷制御工程をコンピュータに実行させるためのプログラム。

【請求項 11】

請求項 6 乃至請求項 9 のいずれか 1 項に記載の印刷制御工程をコンピュータに実行させるためのプログラムを格納したコンピュータが読み取り可能な記憶媒体。

10

【請求項 12】

ユーザにより印刷を指示された場合に、当該指示に基づく印刷データに対して課金コードが設定可能なコンピュータにおいて動作する印刷制御装置であって、

印刷アプリケーションから受信した印刷データを印刷装置に送信する前に一時的に記憶する記憶手段と、

前記記憶手段において受信した印刷データを印刷装置に送信する送信手段と、

前記記憶手段において受信した印刷データに課金コードが設定されているか否かを判断する判断手段とを備え、

前記送信手段は、前記判断手段により前記記憶手段において受信した印刷データに課金コードが設定されていないと判断された場合、当該印刷データを印刷装置に送信せず、前記判断手段により前記記憶手段において受信した印刷データに課金コードが設定されていると判断された場合、当該印刷データを印刷装置に送信することを特徴とする印刷制御装置。

20

【請求項 13】

ユーザにより印刷を指示された場合に、当該指示に基づく印刷データに対して課金コードが設定可能なコンピュータにおいて動作する印刷制御方法であって、

印刷アプリケーションから受信した印刷データを印刷装置に送信する前に一時的に記憶する記憶工程と、

前記記憶工程において受信した印刷データを印刷装置に送信する送信工程と、

前記記憶工程において受信した印刷データに課金コードが設定されているか否かを判断する判断工程とを備え、

前記送信工程は、前記判断工程により前記記憶工程において受信した印刷データに課金コードが設定されていないと判断された場合、当該印刷データを印刷装置に送信せず、前記判断工程により前記記憶工程において受信した印刷データに課金コードが設定されていると判断された場合、当該印刷データを印刷装置に送信することを特徴とする印刷制御方法。

30

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、印刷システムにおいて印刷ジョブに対して課金コードなどの追加情報を付加し、そのログを収集することでジョブのコストなどを管理する技術に関する。

40

【背景技術】

【0002】

近年ユーザ毎に印刷文書名、印刷日時、用紙、部数、カラー/モノクロ等を集計する印刷履歴管理システムが実現している。さらに、印刷アプリケーションやプリンタドライバでは付加されない情報（例えば、顧客情報や印刷単価）を印刷ジョブに紐付け、印刷ログ（履歴）の一部として集計している。集計結果から印刷傾向を調査するなど、精度の高い課金に利用するシステムが実現されている。

【0003】

50

また、無駄な印刷の抑制のため、ユーザ毎に印刷可能数を設定しておき、ユーザの印刷上限を制限するシステムが提供されている。

【0004】

印刷ジョブに追加情報を紐付けさせる手段として、ジョブ作成時に印刷設定ダイアログから情報入力し、プリンタドライバで印字情報を格納する領域とは別の領域（例えば、ヘッダー）に書き込み、ジョブを生成する方法があった。しかしながら、ジョブを作成するドライバを変更する必要があるとあり、既存システムへの適用ができなかった。そこで、クライアントPCからジョブ生成後に、スプーラに格納後、画像処理装置へジョブ送信時に情報入力ダイアログをさせる方法が考え出された。しかし、ジョブはスプールされた順に、1度に1ジョブが送信されるのが普通である。該当ジョブが先頭ジョブ以外の時は送信順番が回ってくるまでには時間がかかり、情報入力ダイアログが表示される時にはユーザがログオフし、入力できないことがあった。そのため、先頭ジョブの情報入力待ちのために後続ジョブが止まってしまうことがあった。これを防ぐために、入力ダイアログに一定期間入力がないときは、タイムアウトとしてジョブをキャンセルしていた。それにより、印刷実行したユーザが印刷物を受け取ることができないという問題があった。

10

【0005】

ここで例えば、特許文献1に開示された発明がある。この発明では、画像処理装置が、キューの先頭ジョブが資源待ち（例えば、特定用紙切れ）の場合は、該当ジョブを一時停止しておき、キューの次ジョブを実行するものである。

【特許文献1】特開2003-131831号公報

20

【発明の開示】

【発明が解決しようとする課題】

【0006】

上記従来技術によると、一般的に画像処理装置の記憶装置は情報処理装置に比べ低容量であり、大量のジョブまたは、大きなジョブを止めておくことができない。止めてしまうとシステムの停止に陥ることがある。大容量の記憶装置を持った画像処理装置があったとしても、すぐに印刷できないジョブをネットワーク経由で情報処理装置から送信することにより、非効率である。また、印刷実行したユーザが画像処理装置に出向かないと、特定条件（資源待ち）を解除することができない。

【0007】

30

本発明は、上記問題に鑑みてなされたもので、ユーザの指示により生成された印刷ジョブに対して、後に印刷ログの一部として有効な追加情報を設定することをユーザに促せる印刷制御装置の提供を目的としている。

【課題を解決するための手段】

【0008】

上記課題を解決するために本発明は、印刷アプリケーションから受信した順番で印刷データを記憶領域に保持し、保持された印刷データのうち、受信した順番が先頭の印刷データから印刷装置に送信する印刷制御装置において、前記記憶領域で受信した順番が先頭の印刷データに課金コードが設定されている場合、当該印刷データを印刷装置に送信し、前記記憶領域で受信した順番が先頭の印刷データの課金コードが設定されていない場合、当該印刷データより前記記憶領域で受信した順番が後続となる印刷データが先頭になるように印刷データの順番を制御する制御手段と、前記印刷アプリケーションから受信した印刷データの一覧を表示するための表示制御手段と、前記表示制御手段により表示された印刷データの一覧から、ユーザの入力に基づき、少なくとも1つ以上の印刷データを選択し、印刷データに対して課金コードを設定する設定手段とを備えることを特徴とする。

40

【発明の効果】

【0009】

上記構成によれば、ユーザの指示により生成された印刷ジョブに対して、後に印刷ログの一部として有効な追加情報を設定することをユーザに促せ得る印刷制御装置の提供が可能になる。

50

**【発明を実施するための最良の形態】****【0010】**

以下、図面を参照しながら本発明の実施の形態に係る印刷システムについて説明する。

**【0011】**

図1は、本発明の実施の形態にかかる印刷システムの構成を示すブロック図である。101～104は情報処理装置であり、イーサネット（登録商標）などのネットワークケーブルによって、ネットワーク106に接続されている。さらに、アプリケーションプログラム等の各種のプログラムを実行可能であり、印刷データをプリンタに対応するプリンタ言語に変換する機能を有するプリンタドライバを搭載している。また、ネットワーク106に接続されている他の機器と双方向に通信可能な通信装置を備えている。

10

**【0012】**

プリンタドライバは複数のプリンタをサポートするものである。また、プリンタ（入出力装置）は、電子写真方式を採用したレーザービームプリンタ、インクジェット方式を採用したインクジェットプリンタ、熱転写方式を利用したプリンタ等の様々な方式のもののうち何れであってもよい。

**【0013】**

本実施例における101、102はパーソナルコンピュータなどの情報処理装置であり、101はログ収集サーバ、102はプリントサーバとする。ログ収集サーバ101およびプリントサーバ102はネットワークケーブルによってネットワーク106に接続されており、ネットワーク106で使用されるファイルを蓄積したり、ネットワーク106の使用状態を監視したりする。また、ログ収集サーバ101およびプリントサーバ102は、ネットワーク106に接続されている他の機器と双方向に通信可能な通信装置を備えており、複数のプリンタの管理も行っている。

20

**【0014】**

クライアント103～104、ログ取得サーバ101、及びプリントサーバ102は一般的な情報処理装置である。ログ収集サーバ101、およびプリントサーバ102は、クライアント103～104としての機能を併せて持つこともできる。クライアント、ログ収集サーバ、及びプリントサーバのHDDやROMなどの記憶媒体には、各種制御を実現するための制御プログラムが実行可能に格納されている。

**【0015】**

プリントサーバ102は、さらにクライアント103、104から印刷要求が出された印字データを含む印刷ジョブを格納し、画像処理装置に送信する。または、クライアント103、104から印字データを含まないジョブ情報のみを受け取ってクライアント103、104の印刷順序を管理し、印刷の順番が来たクライアントに対して印字データを含む印刷ジョブの送信許可を通知する。さらに、ネットワークプリンタ105のステータスや印刷ジョブの各種情報を取得してクライアント103、104に通知したりする機能を備えている。

30

**【0016】**

105は画像処理装置（プリンタ、複写機、複合機などを含む）であり、ネットワークプリンタは、図示省略したネットワークインタフェースを介してネットワーク106に接続されている。さらに、クライアント102、103、104から送信される印字データを含む印刷ジョブを受信して印刷を行う。

40

**【0017】**

ネットワーク106にはクライアント102、103、104、サーバ101、ネットワークプリンタ105等が接続されている。

**【0018】**

図2において、200はクライアント102、103、104の制御手段であるCPUである。CPU200は、ハードディスク（HDD）205に格納されているアプリケーションプログラム、プリンタドライバプログラム、OSや本発明のネットワークプリンタ制御プログラム等を実行する。さらに、RAM202にプログラムの実行に必要な情報、フ

50

ファイル等を一時的に格納する制御を行う。

【0019】

201は記憶手段であるROMであり、内部には基本I/Oプログラム等のプログラム、文書処理の際に使用するフォントデータ、テンプレート用データ等の各種データが記憶される。202は一時記憶手段であるRAMであり、CPU200の主メモリ、ワークエリア等として機能する。

【0020】

203は記憶媒体読み込み手段としてのフロッピー（登録商標）ディスク（FD）ドライブである。後述する図5に示すように、このFDドライブ203を介して、記憶媒体としてのフロッピー（登録商標）ディスク（FD）204に記憶されたプログラム等をクライアントである本コンピュータシステムにロードすることができる。なお、記憶媒体はFDに限らず、CD-ROM、CD-R、CD-RW、PCカード、DVD、ICメモリカード、MO、メモリスティック等、任意である。

10

【0021】

FD204には、コンピュータによって読み取り可能なプログラムが格納されている。

【0022】

205は外部記憶手段の一つであり、大容量メモリとして機能するハードディスク（HD）であり、アプリケーションプログラム、プリンタドライバプログラム、OS、制御プログラム、関連プログラム等が格納されている。さらにスプーラがここに確保される。スプーラは、クライアントにおいてはクライアントスプーラであり、プリントサーバではサーバスプーラである。また、プリントサーバでは、クライアントから受けたジョブ情報を格納し、順序制御を行うためのテーブルもこの外部記憶手段に生成されて格納される。

20

【0023】

206はキーボード（指示入力手段）であり、デバイスの制御コマンドの命令等を入力するものである。

【0024】

207はディスプレイ（表示手段）であり、キーボード206から入力されたコマンドや、プリンタの状態等を表示するものである。

【0025】

208はシステムバスであり、クライアントであるコンピュータ内のデータの流れを司るものである。

30

【0026】

209はインタフェース（入出力手段）であり、該インタフェース209を介してクライアントは外部装置とのデータのやり取りを行う。

【0027】

本実施の形態では、FD204から制御プログラム及び関連データを直接にRAM202にロードして実行する例を示す。これ以外にも、FD204から制御プログラムを動作させる度に、既にHD205にインストールされている制御プログラムをRAM202にロードするようにしてもよい。

【0028】

なお、制御プログラムを記憶する媒体は、FD以外にCD-ROM、CD-R、PCカード、DVD、ICメモリカードであってもよい。さらに、制御プログラムをROM201に記憶しておき、これをメモリマップの一部となすように構成し、直接にCPU200で実行することも可能である。

40

【0029】

次に画像処理装置の詳細構成を説明する。画像処理装置2は、詳細を後述する接続部226、データ制御部（プリンタエンジン）225、操作部224、外部メモリ223、PCPU220、PRAM222、及びPROM221を主要な構成要素とする。これらの構成要素がシステムバス227を介して互いに接続されている。

【0030】

50

プリンタCPU (PCPU) 220は、画像処理装置2の全体制御を司るものである。PCPU 220は、後述するPROM 221 或いは外部メモリ 223に記憶された制御プログラムに基づいて、接続部 226で受信したプリンタ制御コマンド(送信データ)より、画像信号をデータ制御部 225に送信する。

【0031】

PRAM 222は、PCPU 220の主メモリとして、PCPU 220による制御実行時、ワークデータエリアとして使用される各種データの一時記憶領域を備えている。外部メモリ 223は、オプションとしても接続され、フォントデータ、エミュレーションプログラム、フォームデータ等を記憶する。PROM 221はプリンタ内部メモリであり、外部メモリと同様に、各種データや本画像処理装置を制御する制御プログラム等を記憶している。

10

【0032】

データ制御部 225はプリンタエンジンであり、PCPU 220により制御され、PROM 221、或いは外部メモリ 223に記憶された制御プログラムよりシステムバス 227を介して出力された画像信号を受け、実際の印刷処理を行う。操作部 224は、操作パネルや操作スイッチ等の入力部、及びLED、液晶パネル等の表示部等から構成され、オペレータの操作を受けたり結果を表示するものである。オペレータは操作部 224を介して画像処理装置2の設定を指示するなど、確認することが可能である。

【0033】

226は接続部であり、前述の双方向インタフェース 209を介して情報処理装置1の接続部 210に接続し、印刷制御コマンド(送信データ)を受信する、画像処理装置内部の状態などを通知する事も可能である。

20

【0034】

また、以上の各装置と同等の機能を実現するソフトウェアをもって、ハードウェア装置の代替として構成することもできる。

【0035】

印刷ジョブを作成するクライアントコンピュータと、印刷ジョブを送信するプリントサーバ、印刷ログを収集管理するログ収集サーバは、同一の情報処理装置であっても、別の情報処理装置であっても本発明は適用される。

【0036】

図3は、本実施の形態にかかる情報入力UI(画面)の一例である。このダイアログは、後述の図4記載のスプーラ監視モジュールによって、クライアントコンピュータの表示部 207に表示され、ユーザからキーボード 206等の指示装置による入力を受け付ける。課金コード入力領域 301、入力確定ボタン 302、ジョブ操作ボタン(一時停止 303、再開 304、削除 305)、表示の更新ボタン 306、ジョブリスト表示部 307を持つ。ジョブリスト表示部 307では、ジョブリストととして、1行毎にジョブID、ドキュメント名、ジョブオーナー、課金コード、状態を一覧表示することができる。ユーザの指示により、ジョブを複数選択可能である。選択されたジョブはUI上で識別可能に反転表示される。ジョブが選択されると、課金コード入力領域 301が有効になり、任意の文字列を入力すること可能になる。複数選択時には同時に課金コードの入力が可能になる。

30

40

【0037】

図4は、実施の形態にかかる印刷システムを構成するクライアントコンピュータ上で動作するモジュール、画像処理装置の関係および、ジョブの流れの一例を示す図である。クライアントコンピュータでは、例えば一般的に普及しているパーソナルコンピュータ(本発明の情報処理装置に相当する)にMicrosoft社のWindows(登録商標)をオペレーティングシステム(以下、OSと呼ぶ)として使用する。さらに、印刷処理機能を有するMicrosoft社製Word(登録商標)等のアプリケーション 401(印刷アプリケーション)を起動する。

【0038】

50

ここで本発明の印刷制御装置（印刷制御部）として、少なくともドライバ、スプーラ監視モジュール、UI制御部、ポートモニタを備えるものとする。また、印刷制御装置は、その処理で用いる各種情報を格納するための記憶媒体やその他、印刷アプリケーションを除く図4に図示されるモジュール全てを備えていてもよい。さらに、後述するように、印刷制御装置は、画像処理装置（印刷装置（プリンタ）、複写機など）において実装されてもよい。

#### 【0039】

ユーザは、キーボード206やポインティングデバイス209等の指示装置を用いて、ドキュメントを作成し、アプリケーションに用意された印刷指示手段によって、印刷指示をする。アプリケーション401は、印刷データをプリンタドライバ402に渡し、印字ジョブを作成する。プリンタドライバ402は作成した印字ジョブをスプーラ403に渡す。スプーラ403は複数のドライバから渡された印刷ジョブをスプールして、ユーザ指定のプリンタ毎にスケジューリングし、プリンタが接続されているポートに対応するポートモニタ404にジョブを渡す。ポートモニタ404は、ポート（例えばUSBやネットワーク）に対応するデバイスドライバ405を適宜呼び出し、ジョブを渡す。デバイスドライバ405は、渡されたデータを、所定の方法に基づき、接続部210を介し、画像処理装置406に渡す。

10

#### 【0040】

画像処理装置406は、クライアントコンピュータから渡された印刷ジョブを順次印刷実行する。

20

#### 【0041】

スプーラ監視モジュール407は、クライアントコンピュータ起動時に自動的に動作開始し、OSに予め備えているスプーラ機能を用い、スプーラ403を監視する。プリンタドライバの追加、削除、変更ならびに、プリンタドライバ毎のジョブの生成、状態を監視する。ジョブがスプールされた場合は、予め用意されているジョブリスト408記憶領域に、ジョブを状態やサイズ、プリンタドライバ名、オーナー、ドキュメント名、日時等の付属情報とともに格納する。同時にUI処理部409（表示制御部）は、図3に図示する情報入力ダイアログ（情報入力UI）を、クライアントコンピュータの表示部207に表示する。UI処理部409はユーザから情報入力UIを介して入力を受け取ったら、ジョブリスト408に入力情報を反映する。さらに、スプーラ監視モジュール407は、ログをサーバ通知モジュール410に渡し、ログ収集サーバに通知する。

30

#### 【0042】

図5aは、本実施の形態にかかるログを示す一例である。クライアントコンピュータで動作するスプーラ監視モジュールがRAM202やHD205等の記憶装置に作成し、サーバ通知モジュールによってログ収集サーバに通知される。図5bは各項目の説明であり、項目には値、数値または文字列が入る。

#### 【0043】

図17aは、本実施の形態にかかるジョブリストを示す一例である。図4のジョブリスト408、後述の図14のジョブリスト1408から確認可能なジョブ情報を示すレコードである。図17bには各項目の説明を示す。クライアントコンピュータで動作するスプーラ監視モジュールが、RAM202やHD205等の記憶装置にレコード数分確保した領域に保存する。

40

#### 【0044】

以下に実施例1について図6、図7、図8、図9、図10を用いて説明する。

#### 【0045】

ジョブに関する情報に対して追加情報として課金コードを加えたジョブ情報をログとして扱う。ここで課金コードは、予め課金先が決まってもよい。また、課金コードの利用法としては、特に課金先を決めておかず、コードとしてそれぞれクライアント（ユーザ）が一意にジョブを処理する際に費用管理に用いることもできる。また、本実施例で後述するようにジョブ生成後に、後から課金コードを設定してもよい。課金コードは、“0”

50

1文字以外の任意の文字列コードからなり、独立した3つのコードで構成される。コードの3つ全部を指定してもいいし、どれか2つ、または、1つだけ指定してもいいものとする。ログ集計サーバは、ログに含まれる課金コードを、課金の分配先や課金額の決定に利用する。

【0046】

本実施例において、1台のクライアントコンピュータにより、印刷ジョブの生成、画像処理装置への送信を行うものとする。

【0047】

図6は、実施の形態にかかるスプーラ監視モジュールの基本動作を示すフローの1例である。

10

【0048】

ステップ601で、ジョブ一覧を取得する。例えば、Microsoft社のWindows（登録商標）に用意されているWin32関数EnumJob（）を呼び出す。そして、クライアントコンピュータのスプール領域にスプールされている印刷ジョブの一覧を取得し、予めRAM202、HD205等の記憶装置に確保されているワーク領域に保存する。

【0049】

次にステップ602で、前記ステップ601で取得したジョブ一覧の中に、新規ジョブが発生したか判断する。すなわち、CPU200が、後述のジョブリスト408と比較し、新しいジョブIDのジョブが追加されている場合は、新規のジョブが発生したと判断する。

20

【0050】

前記ステップ602で新規ジョブが発生したと判断した場合はステップ603で、前記ステップ601で新たに取得したジョブとジョブに関する属性情報をジョブリスト408に追加する。ジョブリスト408は、ジョブ1件毎に、ジョブIDおよび、ドキュメント名等のジョブに関する属性情報を持ち、さらに、ジョブ1件毎に、課金コード等の印刷システムが付加する追加情報を合わせて持つ。スプーラ監視モジュールが起動時にRAM202、HD205等の記憶媒体に格納される。ジョブに関する属性情報は、Microsoft社のWindows（登録商標）に用意されているWin32関数GetJob（）等や、印刷システム独自の手段などで取得する。

30

【0051】

次に、ステップ604で、前記ステップ603で作成したジョブリスト408にジョブが1件以上あるか判断する。すなわち、CPU200が、RAM202、HD205等の記憶媒体に格納されたジョブリスト408を検索し、ジョブのレコード数を計算する。ジョブが1件以上ないと判断したときは、ステップ601に進む。

【0052】

ステップ604でジョブが1件以上あると判断したときはステップ605で、UI処理部409により情報入力UIが表示済か判断する。例えば、Microsoft社のWindows（登録商標）に用意されているWin32関数FindWindow（）を呼び出し、UI処理部409の処理に基づくウィンドウが存在する場合は、既に情報入力UIが表示済と判断する。情報入力UIが表示済の場合は、ステップ601以降を繰り返す。

40

【0053】

ステップ605で情報入力UIが表示していないと判断した場合はステップ606で、所定の手続きに基づき、UI処理部409により情報入力UIをCRT207等の表示装置に表示する。

【0054】

ステップ602で新規ジョブが発生しないと判断したときはステップ607で、ジョブが終了したか判断する。すなわち、CPU200が、前記ステップ601で取得したジョブ一覧とRAM202、HD205等の記憶媒体に格納されたジョブリスト408を比較

50

する。ジョブ一覧からジョブがなくなっていれば、ジョブリスト408に保存していたジョブが終了したと判断する。

【0055】

ステップ607でジョブ終了したと判断したときはステップ608で、ログをログ収集サーバに通知する。すなわち、CPU200が、RAM202、HD205等の記憶媒体に格納されたジョブリスト408から終了ジョブの記録を取り出し、図5に示すログを作成し、サーバ通知モジュール401にログを渡す。ジョブリスト408からは該当レコードを削除し、RAM202、HD205等の記憶媒体に格納する。サーバ通知モジュール401は、ネットワークなどを介して、ログをログ収集サーバに通知する。

【0056】

図7は、実施の形態にかかるポートモニタの基本動作を示すフローの一例である。ポートモニタは、スプーラプロセスにより1ジョブ毎に呼び出され、ジョブのIDと印刷先を指示され、必要であれば分割されたジョブデータを渡され、画像処理装置に順次送信する。

【0057】

ステップ701で、スプーラから渡されたジョブのIDを取得する。

【0058】

次にステップ702で、CPU200が、前記ステップ701で取得したジョブIDを、前述のスプーラ監視モジュールにより作成され、RAM202やHD205等の記憶媒体に格納されているジョブリスト408から検索する。

【0059】

ジョブリスト408から、該当ジョブIDをジョブが見つかった場合は、ステップ703で、CPU200が、見つかったジョブの付属情報を検索し、有効な課金コードが設定されているか判断する。すなわち、3つの課金コード領域のうち1つでも、“0”1文字以外の数値が設定されている場合は、有効な課金コードが設定されていると判断する。

【0060】

ステップ703で有効な課金コードが入力されていると判断したときはステップ704で、ネットワークなどを介して、ジョブを画像処理装置に送信する。

【0061】

ステップ702でリストに登録済でないと判断したとき、および、前記ステップ703で有効な課金コードが入力されていないと判断したときはステップ705で、該当ジョブの一時停止処理を行う。例えば、Microsoft社のWindows(登録商標)に用意されているWin32関数SetJob()で該当ジョブを指定し、ジョブの処理に一時停止(JOB\_\_CONTROL\_\_RESUME)を指定する。

この処理により、該当ジョブは一時停止処理をされ、スプーラのキュー中で処理順番が来るまで、処理は停止される。

【0062】

次に、ステップ708で、該当ジョブの再実行を行う。例えば、Microsoft社のWindows(登録商標)に用意されているWin32関数SetJob()で該当ジョブを指定し、ジョブの処理に再実行(JOB\_\_CONTROL\_\_RESTART)を指定する。この処理により、該当ジョブは次回実行にジョブの先頭から実行される。よって、スプーラはキューの中から先頭ジョブを取り出そうとするが、先頭ジョブは前記ステップ705で一時停止されているため、当該ジョブを飛ばし、次の順番のジョブの処理を開始する。

【0063】

図8は、実施の形態にかかるUI制御部における、処理の基本動作を示すフローの一例である。

【0064】

UI制御部は、前述のスプーラ監視モジュールにより起動される他、ユーザの指示により任意のタイミングで起動し、情報入力UIを表示することができる。初期表示状態は、

10

20

30

40

50

図3 ジョブリスト表示部は、ジョブリスト408のレコードを付属情報と共に全件表示している。表示領域に収まりきれない場合は、スクロールバーを表示する。また、リスト上のジョブはすべて非選択状態であり、課金コード入力領域301は空欄になっている。

【0065】

ステップ801で、ボタンクリック等のイベントが発生したか判断する。

【0066】

例えば、Microsoft社のMSDN Library（登録商標）で開示されている方法では、マウスによるクリックやキーボード入力等の操作と、ウィンドウ内のボタンやリストボックス等の各リソースを一意に決定するIDが、イベントとして通知される。

10

イベントが発生するまでステップ801を繰り返す。

【0067】

ステップ801でイベント発生したと判断した場合はステップ802で、イベントがジョブ選択か判断する。すなわち、ステップ801で取得したイベントに、図3のジョブリスト表示部307におけるレコードの示すIDのマウスボタンによる選択操作が含まれるときに、ジョブ選択と判断する。

【0068】

ステップ802でジョブ選択と判断した場合はステップ803で、該当レコードを表示部307上で選択されたことを認識可能にするため、表示部307の表示色などを選択前後で反転表示する。

20

【0069】

ステップ802でジョブ選択と判断しなかった場合はステップ804で、「入力」ボタンがクリックされたか判断する。すなわち、ステップ801で取得したイベントに、図3の「入力」ボタンを示すIDのマウスボタンによる選択操作が含まれるときに、「入力」ボタンクリックと判断する。

【0070】

ステップ804で「入力」ボタンクリックと判断した場合はステップ805で、ジョブリスト408に課金コードを設定する。CPU200が、表示部307上で図3ジョブリストのうち反転表示されているレコードを検索し、見つかったレコードからジョブIDを取得する。RAM202やHD205等の記憶媒体に格納されているジョブリスト408のうち、該当IDを持つレコードの課金コード領域に、図3の第一コード、第二コード、第三コードのそれぞれの値を設定する。

30

【0071】

ステップ804で「入力」ボタンクリックと判断しなかった場合はステップ806で、「一時停止」ボタンがクリックされた判断する。すなわち、ステップ801で取得したイベントに、図3の「一時停止」ボタンを示すIDと、マウスボタンによる選択操作が含まれるときに、「一時停止」ボタンクリックと判断する。

【0072】

ステップ806で「一時停止」ボタンクリックと判断した場合はステップ807で、CPU200が、表示部307上で図3ジョブリストのうち反転表示されているレコードを検索し、見つかったレコードからジョブIDを取得する。次に、例えば、Microsoft社のWindows（登録商標）に用意されているWin32関数SetJob（）で該当ジョブを指定し、ジョブの処理に一時停止（JOB\_\_CONTROL\_\_PAUSE）を指定する。

40

【0073】

ステップ806で「一時停止」ボタンクリックと判断しなかった場合はステップ808で、「再開」ボタンがクリックされた判断する。すなわち、ステップ801で取得したイベントに、図3の「再開」ボタンを示すIDと、マウスボタンによる選択操作が含まれるときに、「再開」ボタンクリックと判断する。

【0074】

50

ステップ 808 で「再開」ボタンクリックと判断した場合はステップ 809 で、CPU 200 が、表示部 307 で図 3 ジョブリストのうち反転表示されているレコードを検索し、見つかったレコードからジョブ ID を取得する。次に、Microsoft 社の Windows (登録商標) に用意されている Win32 関数 SetJob () で該当ジョブを指定し、ジョブの処理に再開 (JOB\_\_CONTROL\_\_RESUME) を指定する。この処理により該当ジョブが一時停止状態の場合は、一時停止が解除される。既にジョブデータの一部が送信済の場合は、続きから送信を開始する。該当ジョブが一時停止状態でない場合は、状態に変化はない。

#### 【0075】

ステップ 808 で「再開」ボタンクリックと判断しなかった場合はステップ 810 で、「削除」ボタンがクリックされたか判断する。すなわち、ステップ 801 で取得したイベントに、図 3 の「削除」ボタンを示す ID と、マウスボタンによる選択操作が含まれるときに、「削除」ボタンクリックと判断する。

10

#### 【0076】

ステップ 810 で「削除」ボタンクリックと判断した場合はステップ 811 で、CPU 200 が、表示部 307 で図 3 ジョブリストのうち反転表示されているレコードを検索し、見つかったレコードからジョブ ID を取得する。次に、例えば Microsoft 社の Windows (登録商標) に用意されている Win32 関数 SetJob () で該当ジョブを指定し、ジョブの処理に削除 (JOB\_\_CONTROL\_\_DELETE) を指定する。

20

#### 【0077】

ステップ 810 で「削除」ボタンクリックと判断しなかった場合はステップ 812 で、「更新」ボタンがクリックされたか判断する。すなわち、ステップ 801 で取得したイベントに、図 3 の「更新」ボタンを示す ID と、マウスボタンによる選択操作が含まれるときに、「更新」ボタンクリックと判断する。

#### 【0078】

ステップ 812 で「更新」ボタンクリックと判断した場合はステップ 813 で、ジョブ一覧を再取得する。例えば、Microsoft 社の Windows (登録商標) に用意されている Win32 関数 EnumJob ()、Win32 関数 GetJob () 等や、印刷システム独自の手段を呼び出し、印刷ジョブおよび付属情報の一覧を取得する。前記取得したジョブ情報で、予め RAM 202、HD 205 等の記憶装置に確保されているジョブリスト 408 の情報を更新する。

30

#### 【0079】

次に、ステップ 814 で、前記ステップ 813 で更新したジョブリスト 408 情報を元に、図 3 のジョブリスト表示部 307 の表示を更新する。

#### 【0080】

以上説明した方法によれば、第一の実施の形態では、複数ジョブに対し、ジョブリストから操作、追加情報入力が一度にでき、利便性が向上する。また、追加情報入力待ちで、スプーラでジョブ処理が止まることがない。そして追加情報が未入力のジョブを削除することなく、後続ジョブの処理を継続実行できる。

40

#### 【0081】

〔第 2 の実施の形態〕

情報入力 UI から情報入力したジョブから優先的に印刷する改良が考えられる。

#### 【0082】

例えば、Microsoft 社の Windows (登録商標) のスプールシステムでは、キューの中の印刷ジョブに優先順位が設定されている。Windows (登録商標) に用意されている Win32 関数 GetJob ()、SetJob () を用い、ジョブを指定して、該当ジョブの優先順位を、取得、設定可能である。優先順位は、最低位の 1 から最高位 99 の範囲で設定可能である。優先順位はジョブがスプールされたときに決まり、一般的には最低位の 1 に設定される。

50

## 【0083】

本実施の形態においては、第1の実施の形態と異なる点のみ説明する。図8のUI処理部における処理の基本フロー、ステップ805「課金コード設定」処理の改良に関して、図9を使って説明する。ここでは、全ジョブの優先度の初期値は1に設定されているものとする。

## 【0084】

ステップ901で、ジョブリスト408に課金コードを設定する。CPU200が、予め用意されている方法により、図3ジョブリスト表示部307の反転表示されているレコードを検索し、見つかったレコードからジョブIDを取得する。次に、RAM202やHD205等の記憶媒体に格納されているジョブリスト408の該当リストの課金コード領域に、図3の第一コード、第二コード、第三コードのそれぞれの値を設定する。

10

## 【0085】

次にステップ902で、前記ステップ901で課金コードを設定したジョブの優先順を上げる。すなわち、例えばMicrosoft社のWindows（登録商標）に用意されているWin32関数GetJob（）で該当ジョブの優先度を取得する。そして、Win32関数SetJob（）により、該当ジョブ情報の優先順位を前記取得した優先度に1を加算した値として優先度設定する。

## 【0086】

この処理により、スプーラにおけるキューのジョブのうち課金コードが設定されたジョブは、優先度が+1されている。よって、キューから優先順位の高いジョブ、すなわち課金コードが設定されているジョブを優先的に処理を進める。そして、課金コードが未入力のジョブがキューの前順位に位置していたとしても、キューの後ろに位置する課金コード設定済のジョブが先に処理されることになる。

20

## 【0087】

第2の実施の形態では、追加情報が追加されたジョブから優先的に処理が実行される用に印刷ジョブに優先度設定する。具体的には、スプーラにおけるキューにおいてスプールされた順番ではなく、課金コードが設定された印刷ジョブが、設定されていない印刷ジョブを追い越して処理されることで、追加情報入力待ちにより処理が停滞しなくなる。

## 【0088】

## 〔第3の実施の形態〕

ユーザが間違えて印刷指示してしまった印刷ジョブに対して課金コードなどの追加情報入力せず放置する場合、そのジョブが処理されずにスプーラのキューから消えないということが考えられる。そこで、一定期間、追加情報が未入力のジョブをキューから自動で削除する改良が考えられる。

30

## 【0089】

本実施例においては、第1、2の実施の形態と異なる点のみ説明する。図6のスプーラ監視モジュールにおける処理の基本フロー、ステップ603「ジョブリスト作成」処理の改良に関して図10を使って説明する。

## 【0090】

ステップ1001で、図6記載のステップ601で取得したジョブとジョブに関する属性情報をジョブリスト408に追加する。ジョブリスト408は、ジョブ1件毎に、ジョブIDおよび、ドキュメント名等のジョブに関する属性情報を持つ。さらに、ジョブ1件毎に、課金コード等の印刷システムが付加する追加情報、並びに「削除タイマー開始時刻」を合わせて持つ。スプーラ監視モジュールが起動時にRAM202、HD205等の記憶媒体に格納される。ジョブに関する属性情報は、例えばMicrosoft社のWindows（登録商標）に用意されているWin32関数GetJob（）等や、印刷システム独自の手段で取得する。初期状態の「削除タイマー開始時刻」は空欄とする。

40

## 【0091】

次にステップ1002で、CPU200が、前記ステップ1001でRAM202、HD205等の記憶媒体に格納されたジョブリスト408からレコード数を調べ、0の場合

50

は「ジョブリスト作成処理」を終了する。レコード数が0以外の場合は、ステップ1003以降を繰り返し、全レコードを順次読み込む。全レコード読み込んだと判断した場合は、「ジョブリスト作成処理」を終了する。

【0092】

次にステップ1003で、CPU200が、前記ステップ1001でRAM202、HD205等の記憶媒体に格納されたジョブリスト408から1レコード読み出し、RAM202、HD205等の記憶媒体で確保したワークエリアに保持する。

【0093】

次にステップ1004で、前記ステップ1003でワークエリアに保持した1レコードのデータを調べ、新規ジョブか判断する。すなわち、レコード中の「削除タイマー開始時刻」が空欄のときは新規ジョブと判断する。

10

【0094】

ステップ1004で新規ジョブと判断したときはステップ1009で、CPU200が、RAM202、HD205等の記憶媒体に格納されたジョブリスト408の該当レコードの「削除タイマー開始時刻」に現在時刻を設定し、保存する。

【0095】

ステップ1004で新規ジョブと判断しなかったときはステップ1005で、前記ステップ1003でワークエリアに保持した1レコードのデータを調べ、タイムアウト発生したか判断する。すなわち、CPU200が、ワークエリアに保持した1レコードのデータ中に「削除タイマー開始時刻」と現在時刻を比較し、差分が予め決められたタイムアウト値を超えていた場合は、タイムアウト発生したと判断する。

20

【0096】

ステップ1005タイムアウト発生しなかったと判断したときは、ステップ1002以降を繰り返す。

【0097】

ステップ1005でタイムアウト発生と判断したときは、ステップ1006で、CPU200が、RAM202、HD205等の記憶媒体に格納されたジョブリスト408から現在のレコードを削除し、保存する。

【0098】

次にステップ1006で、スプーラのキューからジョブを削除する。CPU200が、例えばMicrosoft社のWindows（登録商標）に用意されているWin32関数SetJob（）で該当ジョブを指定し、ジョブの処理に削除（JOB\_\_CONTROL\_\_DELETE）を指定する。次に、ステップ1002以降を繰り返す。

30

【0099】

以上のように、第3の実施の形態では、追加情報が未入力のジョブがスプーラのキューにずっと放置されることがなくなる。

【0100】

〔第4の実施の形態〕

ユーザが間違っって課金コードなどの追加情報入力するなどした際に、印刷指示終了後でも追加情報を変更できるようにする改良が考えられる。

40

【0101】

本実施形態においては、1～3の実施の形態と異なる点のみ説明する。図6のスプーラ監視モジュールにおける処理の基本フロー、ステップ608「ログ送付」処理の改良に関して図11を使って説明する。

【0102】

RAM202、HD205等の記憶媒体に格納されたジョブリスト408には、ジョブリスト408は、ジョブ1件毎に、ジョブIDおよび、ドキュメント名等のジョブに関する属性情報を持つ。さらに、ジョブ1件毎に、課金コード等の印刷システムが付加する追加情報、並びに「送付タイマー開始時刻」を合わせて持つ。スプーラ監視モジュールが起動時にRAM202、HD205等の記憶媒体に格納される。ジョブに関する属性情報は

50

、例えば、Microsoft社のWindows（登録商標）に用意されているWin 32関数GetJob（）や、印刷システム独自の手段などで取得する。初期状態の「送付タイマー開始時刻」は空欄とする。

#### 【0103】

ステップ1101で、送付タイマー開始時刻が設定されているか判断する。すなわち、CPU200が、図6記載のステップ607で終了と判断したジョブに該当するレコードを、RAM202、HD205等の記憶媒体に格納されたジョブリスト408から取り出す。取り出したレコードから、「送付タイマー開始時刻」を読み出し、ワークエリアに保持する。ワークエリアに保持した「送付タイマー開始時刻」が空欄になっていないと判断した場合は、送付タイマー開始時刻が設定されていると判断する。

10

#### 【0104】

前記ステップ1101で送付タイマー開始時刻が設定されていると判断したときはステップ1102で、タイムアウトが発生したか判断する。すなわち、前記ステップ1101でワークエリアに保持した「送付タイマー開始時刻」と現在時刻を比較し、差分が予め決められたタイムアウトの閾値を超えていた場合は、タイムアウトが発生したと判断する。

#### 【0105】

前記ステップ1102でタイムアウトが発生したと判断したときはステップ1103で、ログをログ収集サーバに通知する。すなわち、CPU200が、RAM202、HD205等の記憶媒体に格納されたジョブリスト408から終了ジョブのレコードを取り出す。そして、図5に示すログを作成し、サーバ通知モジュール401にログを渡す。ジョブリスト408からは該当レコードを削除し、RAM202、HD205等の記憶媒体に格納する。サーバ通知モジュール401は、ネットワークなどを介して、ログをログ収集サーバに通知する。

20

#### 【0106】

前記ステップ1102でタイムアウトが発生しないと判断したときは「ログ送付」処理を終了する。

#### 【0107】

前記ステップ1101で送付タイマー開始時刻が設定されていないと判断したときはステップ1104に進む。CPU200が、RAM202、HD205等の記憶媒体に格納されたジョブリスト408の該当レコードの「送付タイマー開始時刻」に現在時刻を設定し、保存する。

30

#### 【0108】

第4の実施の形態では、ログ送付のタイミングを予め設定しておくことにより、印刷指示後の一定時間（印刷ジョブの終了後）、課金コードなどの追加情報の変更が可能となる。

#### 【0109】

##### 〔第5の実施の形態〕

スプーラのキューに多数のジョブが溜まっていない場合、ユーザが追加情報を入力するより早く、スプーラのジョブ処理の順番がやってきて、該当ジョブの追加情報未入力と判断されることがある。その場合、第1の実施の形態においては、ジョブが一時停止されてしまい、ユーザが解除処理を行う必要がある。また、第3の実施の形態では、ジョブが自動で削除されてしまう可能性がある。

40

#### 【0110】

本実施の形態においては、第1～4の実施の形態と異なる点のみ説明する。図8のUI処理部における処理の基本動作を示すフローのステップ803「ジョブ選択」処理、および、ステップ805「課金コード設定」処理の改良を、それぞれ図12、図13を使って説明する。

#### 【0111】

「ジョブ選択」処理では、ステップ1201で、ジョブの一時停止をする。図8のステップ802で取得したIDのジョブに対し、例えば、Microsoft社のWindo

50

ws（登録商標）に用意されているWin32関数SetJob（）で、ジョブの処理に一時停止（JOB\_\_CONTROL\_\_PAUSE）を指定する。

【0112】

次に、ステップ1202で、該当レコードを予め用意された方法で反転表示し、「ジョブ選択」処理を終了する。

【0113】

「課金コード設定」処理では、ステップ1301で、ジョブリスト408に課金コードを設定する。CPU200が、表示部307でジョブリストの反転表示されているレコードを検索し、見つかったレコードからジョブIDを取得する。次に、RAM202やHD205等の記憶媒体に格納されているジョブリスト408の該当リストの課金コード領域に、図3の第1コード、第2コード、第3コードのそれぞれの値を設定する。

10

【0114】

次にステップ1302で、ジョブの再開をする。前記ステップ1301で取得したジョブIDのジョブに対し、Microsoft社のWindows（登録商標）に用意されているWin32関数SetJob（）で、ジョブの処理に再開（JOB\_\_CONTROL\_\_RESUME）を指定する。この処理により該当ジョブが一時停止状態の場合は、一時停止が解除される。既にジョブデータの一部が送信済の場合は、続きから送信を開始する。該当ジョブが一時停止状態でない場合は、状態に変化はない。

【0115】

第5の実施の形態では、課金コードなどの追加情報の入力中に選択ジョブが削除されてしまうことを防ぐことができ、さらに、情報入力したジョブの再開を自動で行うため、ユーザによるジョブ再開操作の手間が省ける。

20

【0116】

〔第6の実施の形態〕

クライアントコンピュータはポートモニタを使わない印刷、例えばNovell社のNetWare（登録商標）のプリンタサーバによる印刷では、ポートモニタによるジョブに追加情報の設定がされたかの確認ができない。よって、確実に追加情報設定済のジョブだけ印刷を許すことができない。そこで、追加情報の確認を画像処理装置（プリンタ、複写機など）が行う改良が考えられる。

【0117】

30

以下に、図3、図5、図14、図15、図16を用いて説明する。本実施の形態は、上述してきた他の実施の形態を適宜、適用可能であり、以下本実施の形態の説明においては、1～5の実施の形態と異なる点のみ説明する。

【0118】

図14は、実施の形態にかかる印刷システムを構成するクライアントコンピュータ上で動作するモジュール、画像処理装置の関係および、ジョブの流れの一例を示す図である。クライアントコンピュータでは、例えば一般的に普及しているパーソナルコンピュータ（本発明の情報処理装置に相当する）にMicrosoft社のWindows（登録商標）をオペレーティングシステム（以下、OSと呼ぶ）として使用する。印刷処理機能を有するMicrosoft社製Word（登録商標）等のアプリケーション1401を起動する。ユーザは、キーボード206やポインティングデバイス209等の指示装置を用いて、ドキュメントを作成し、アプリケーションに用意された印刷指示手段によって、印刷指示をする。アプリケーション401は、印刷データをプリンタドライバ1402に渡し、印字ジョブを作成する。プリンタドライバ1402は作成した印字ジョブをスプーラ1403に渡す。スプーラ1403は複数のドライバから渡された印刷ジョブを、ユーザ指定のプリンタ毎にジョブをスケジューリングし、プリンタが接続されているポートに対応するポートモニタ1404にジョブを渡す。ポートモニタ1404は、ポート（例えばUSBやネットワーク）に対応するデバイスドライバ1405を適宜呼び出し、ジョブを渡す。デバイスドライバ1405はその印刷ジョブを、接続部210、ネットワークなどを介して、画像処理装置1406に渡す。

40

50

## 【0119】

画像処理装置1406は、クライアントコンピュータから渡された印刷ジョブを印刷キューに蓄積し、クライアントコンピュータから印刷再開の指示があるまで一時停止する。

## 【0120】

デバイス監視モジュール1407は、クライアントコンピュータ起動時に自動的に動作開始し、予め備えられているデバイス監視機能を用い、画像処理装置1406を監視する。画像処理装置内のジョブの状態を監視する。印刷キュー内にジョブが蓄積されたときは、ジョブリスト1408に、ジョブを状態やサイズ、オーナー、ドキュメント名、日時等の付属情報とともに格納する。同時にUI処理部1409により情報入力UIをCRT207等の表示装置に表示する。UI処理部1409は、図3に図示する情報入力ダイアログなどを、クライアントコンピュータの表示部207に表示する。UI処理部1409はユーザからの入力を受け取ったら、ジョブリスト1406に入力情報を反映する。さらに、デバイス監視モジュール1407は、画像処理装置に対しジョブの再開を指示し、印刷が終わったジョブのログをサーバ通知モジュール1410に渡し、ログ収集サーバに通知する。

10

## 【0121】

図15を用いて、画像処理装置の基本フローを説明する。

## 【0122】

ここでは、画像処理装置はクライアントコンピュータからのジョブ操作コマンドを受け付けることができる機能を有し、ジョブの状態に変化があったとき、自らクライアントコンピュータにジョブ変化を通知する機能を有するものとする。状態変化を通知するイベントには、画像処理装置内でジョブを一意に決めるジョブIDと、ジョブ状態がわかる情報が含まれるものとする。また、PRAM222等の記憶媒体に確保した印刷キューに、印刷データをジョブ単位で蓄積することができ、任意の印刷ジョブを取り出し、一時停止、再開、削除、印刷実行等の処理が可能である。

20

## 【0123】

ステップ1501で、データを受信したか判断する。双方向インタフェース209を通り、接続部226を介し、ネットワークからデータを受信したときは、データをPRAM等の記憶装置に確保したワークエリアに保存する。データを受信するまで処理を繰り返す。

30

## 【0124】

ステップ1501でデータを受信したと判断したときはステップ1502で、受信データがジョブか判断する。すなわち、PCPU220を用い、PRAM等の記憶装置に確保したワークエリアに保存した受信データを調べ、予め決められたジョブフォーマットのときは、ジョブと判断する。

## 【0125】

ステップ1502で受信データがジョブと判断したときはステップ1503で、予め用意された手続きに基づきジョブの一時停止処理を行い、PRAM等の記憶装置に保存する。

## 【0126】

次にステップ1504で、PCPU220を用い、ジョブ状態として「一時停止」を持つジョブの状態変化イベントを作成し、作成した状態変化イベントデータをPRAM等の記憶装置に確保したワークエリアに保持し、ステップ1507へ進む。

40

## 【0127】

ステップ1502で受信データがジョブでないとは判断したときはステップ1505で、受信データが再開コマンドか判断する。すなわち、PCPU220を用い、PRAM等の記憶装置に確保したワークエリアに保存した受信データを調べ、予め決められたコマンドフォーマットであり、コマンド種がジョブの再開のときは、「再開コマンド」と判断する。

## 【0128】

50

次にステップ106で、ステップ1505で取得したコマンドに含まれるジョブIDのジョブをPRAM222等の記憶装置に確保したキューから取り出し、印刷を実行する。例えば印刷ジョブを所定の画像処理を施した画像データ(ビットマップデータ)に展開し、プリンタエンジンにおいて印刷出力を行う。

【0129】

次にステップ1507で、ステップ1506で印刷処理を行ったジョブに対する終了イベントを作成する。すなわち、PCPU220を用い、ジョブIDとして前記ステップ1506で印刷処理を行ったジョブのジョブIDを、ジョブ状態として「印刷終了」を意味する情報を持つ状態変化イベントデータをPRAM等の記憶装置に確保したワークエリアに保持する。

【0130】

ステップ1508では、前記ステップ1504および前記ステップ1507でPRAM222等の記憶装置に確保したワークエリアに保持したイベントデータをPCPU220により取り出す。取り出したイベントデータを接続部226、双方向インタフェース209を介し、クライアントコンピュータに送付する。

【0131】

図16を用いて、デバイス監視モジュールの基本フローを説明する。

【0132】

ステップ1601で、ジョブ一覧を取得する。PCPU200が、双方向インタフェース209を通り、接続部210を介し、画像処理装置から通知されるジョブの状態変化イベントを受信する。受信したイベントを予めRAM202、HD205等の記憶装置に確保されているワーク領域に保存する。

【0133】

次にステップ1602で、ジョブが発生したか判断する。すなわち、PCPU200が、前記ステップ1601でRAM202、HD205等の記憶装置に確保されているワーク領域に保存したイベントを調べ、状態が「一時停止」となっている場合は、新規のジョブが発生したと判断する。

【0134】

前記ステップ1602で新規ジョブが発生しないと判断したときは、ステップ1607でジョブが終了したか判断する。すなわち、CPU200が、ステップ1601で記憶媒体に確保されているワーク領域に保存したイベントを調べ、状態が「印刷終了」と判断した場合は、そのジョブは印刷が終了したと判断する。

【0135】

前記ステップ1607でジョブが終了したと判断したときは、ステップ1608で、ログをログ収集サーバに通知する。すなわち、CPU200が、RAM202、HD205等の記憶媒体に格納されたジョブリスト1408から終了ジョブの記録を取り出し、図5に示すログを作成し、サーバ通知モジュール401にログを渡す。ジョブリスト1408からは該当記録を削除し、RAM202、HD205等の記憶媒体に格納する。サーバ通知モジュール401は、ネットワークなどを介してログをログ収集サーバに通知する。

【0136】

前記ステップ1602で新規ジョブが発生したと判断した場合、および、前記ステップ1607でジョブが終了していないと判断した場合は、ステップ1603に進む。ステップ1603では、前記ステップ1601で取得したイベントに含まれるジョブID、およびジョブ状態ならびに、オーナーやドキュメント名等の付随する情報をジョブリスト1408に追加する。ジョブリスト1408は、ジョブ1件毎に、ジョブIDおよび、ドキュメント名等のジョブに関する属性情報を持ち、さらに、ジョブ1件毎に、課金コード等の印刷システムが付加する追加情報を併せ持つ。デバイス監視モジュールが起動時にRAM202、HD205等の記憶媒体の所定領域に保存される。

【0137】

10

20

30

40

50

次に、ステップ1604で、前記ステップ1603で作成したジョブリスト1408にジョブが1件以上あるか判断する。すなわち、CPU200が、RAM202、HD205等の記憶媒体に格納されたジョブリスト1408を検索し、ジョブのレコード数を計算する。ジョブが1件以上ないと判断したときは、ステップ1601に進む。

【0138】

前記ステップ1604で、ジョブが1件以上あると判断したときはステップ1605で、UI処理部1409により情報入力UIが表示済か判断する。例えば、Microsoft社のWindows(登録商標)に用意されているWin32関数FindWindow()を呼び出し、UI処理部1409に基づくウィンドウが存在する場合は、既に情報入力UIが表示済と判断する。情報入力UIが表示済の場合は、ステップ1601以降を繰り返す。

10

【0139】

ステップ1605で情報入力UIが表示していないと判断した場合はステップ606で、所定の手続きに基づき、UI処理部1409が情報入力UIを表示制御する。

【0140】

画像処理装置のジョブの状態をクライアントコンピュータに通知する手段として、画像処理装置が任意のタイミングでイベント通知する(トラップ)を例に取り説明した。しかしながら、クライアントコンピュータがジョブ情報の取得タイミングを指定しても、本発明は適用可能である。

【0141】

第6の実施の形態では、クライアントコンピュータでジョブ送信前に追加情報を入力できなくても、画像処理装置による印刷実行前に追加情報を入力が可能になる。

20

【0142】

また、追加情報は画像処理装置において入力するとしてもよい。

【0143】

[他の実施の形態]

以上、様々な実施形態を詳述したが、本発明は、複数の機器から構成されるシステムに適用してもよいし、また、一つの機器からなる装置に適用してもよい。例えば、プリンタ、ファクシミリ、PC、サーバとクライアントとを含むコンピュータシステムなどの如くである。

30

【0144】

また上述の実施例について、印刷装置内での処理であってもよい。この場合、ユーザは印刷装置の備える入力パネルなどから、課金コードなどの設定が可能となる。また、この場合は、出力先の印刷装置として、画像エンジンに出力することとする。尚、この場合の監視モジュールが監視する対象としては、印刷装置内のキュー(記憶領域)を監視し、印刷データの状態を表示したりすることになる。

【0145】

本発明は、前述した実施形態の各機能を実現するソフトウェアプログラムを、システム若しくは装置に対して直接または遠隔から供給し、そのシステム等に含まれるコンピュータが該供給されたプログラムコードを読み出して実行することによっても達成される。

40

【0146】

従って、本発明の機能・処理をコンピュータで実現するために、情報処理装置にインストールされるプログラムコード自体も本発明を実現するものである。つまり、上記機能・処理を実現するためのコンピュータプログラム自体も本発明の一つである。

【0147】

その場合、プログラムの機能を有していれば、オブジェクトコード、インタプリタにより実行されるプログラム、OSに供給するスクリプトデータ等、プログラムの形態を問わない。

【0148】

プログラムを供給するための記録媒体としては、例えば、フレキシブルディスク、ハー

50

ドディスク、光ディスク、光磁気ディスク、MO、CD-ROM、CD-R、CD-RWなどがある。また、記録媒体としては、磁気テープ、不揮発性のメモ리카ード、ROM、DVD(DVD-ROM, DVD-R)などもある。

【0149】

また、プログラムは、クライアントの情報処理装置のブラウザを用いてインターネットのホームページからダウンロードしてもよい。すなわち、該ホームページから本発明のコンピュータプログラムそのもの、もしくは圧縮され自動インストール機能を含むファイルをハードディスク等の記録媒体にダウンロードしてもよいのである。また、本発明のプログラムを構成するプログラムコードを複数のファイルに分割し、それぞれのファイルを異なるホームページからダウンロードすることによっても実現可能である。つまり、本発明の機能処理をコンピュータで実現するためのプログラムファイルを複数のユーザに対してダウンロードさせるWWWサーバも、本発明の構成要件となる場合がある。

10

【0150】

また、本発明のプログラムを暗号化してCD-ROM等の記憶媒体に格納してユーザに配布してもよい。この場合、所定条件をクリアしたユーザにのみ、インターネットを介してホームページから暗号化を解く鍵情報をダウンロードさせ、その鍵情報で暗号化されたプログラムを復号して実行し、プログラムを情報処理装置にインストールしてもよい。

【0151】

また、コンピュータが、読み出したプログラムを実行することによって、前述した実施形態の機能が実現されてもよい。なお、そのプログラムの指示に基づき、コンピュータ上で稼動しているOSなどが、実際の処理の一部または全部を行ってもよい。もちろん、この場合も、前述した実施形態の機能が実現され得る。

20

【0152】

さらに、記録媒体から読み出されたプログラムが、情報処理装置に挿入された機能拡張ボードや情報処理装置に接続された機能拡張ユニットに備わるメモリに書き込まれてもよい。そのプログラムの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行ってもよい。このようにして、前述した実施形態の機能が実現されることもある。

【図面の簡単な説明】

【0153】

30

【図1】本発明の実施の形態にかかる印刷システムの構成の一例を示すブロック図である。

【図2】本発明の実施の形態にかかる印刷システムの詳細な構成の一例を示すブロック図である。

【図3】本発明における、情報入力画面の一例を示す図である。

【図4】本発明における、クライアントコンピュータで動作するモジュールとデバイスの関係の一例を示すブロック図である。

【図5】aは、本発明における、ログの一例示す図である。bは、本発明における、ログの各項目に対応する情報を説明するための図である。

【図6】本発明における、スプーラ監視モジュールにおける処理の基本動作を示すフローの一例である。

40

【図7】本発明における、ポートモニタにおける処理の基本動作を示すフローの一例である。

【図8】本発明における、UI処理部における処理の基本動作を示すフローの一例である。

【図9】本発明における、第2の実施の形態におけるUI処理部における処理のフローの一例である。

【図10】本発明における、第3の実施の形態におけるスプーラ監視モジュールにおける処理のフローの一例である。

【図11】本発明における、第4の実施の形態におけるスプーラ監視モジュールにおける

50

処理のフローの一例である。

【図12】本発明における、第5の実施の形態におけるUI処理部における処理のフローの一例である。

【図13】本発明における、第5の実施の形態におけるUI処理部における処理のフローの一例である。

【図14】本発明における、第6の実施の形態におけるクライアントコンピュータで動作するモジュールとデバイスの関係の一例を示すブロック図である。

【図15】本発明における、第6の実施の形態における画像処理装置における処理の基本フローの一例である。

【図16】本発明における、第6の実施の形態におけるデバイス監視モジュールにおける処理の基本フローの一例である。

【図17】 aは、本発明における、ジョブリストの一例を示す図である。 bは、本発明における、ジョブリストの各項目に対応する情報を説明するための図である。

【符号の説明】

【0154】

101 サーバ

102, 103, 104 クライアントコンピュータ(PC)

105 画像処理装置(プリンタ)

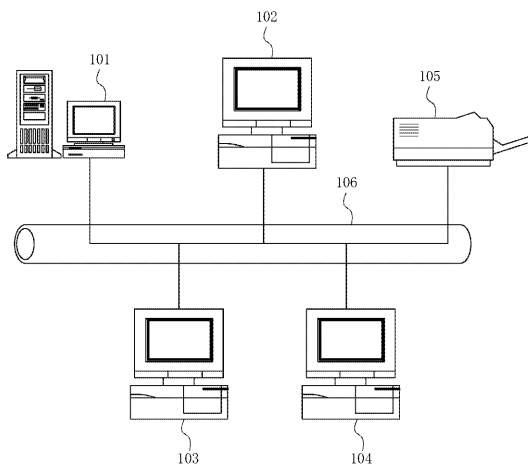
106 ネットワーク

200 CPU

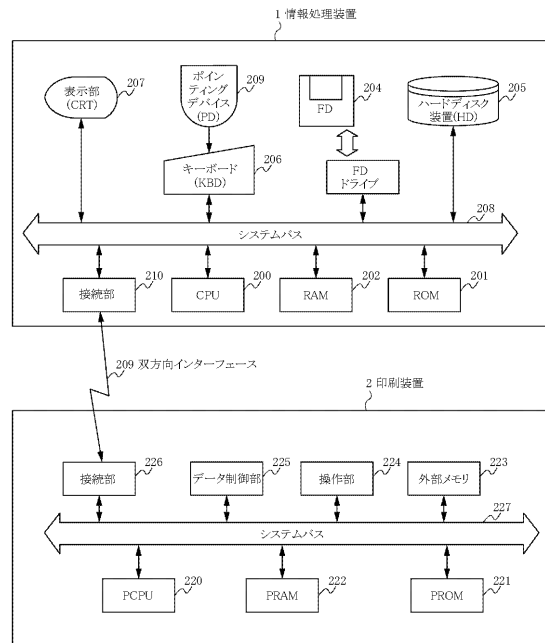
10

20

【図1】



【図2】



【 図 3 】

課金コード入力

第1コード:  301

第2コード:  302

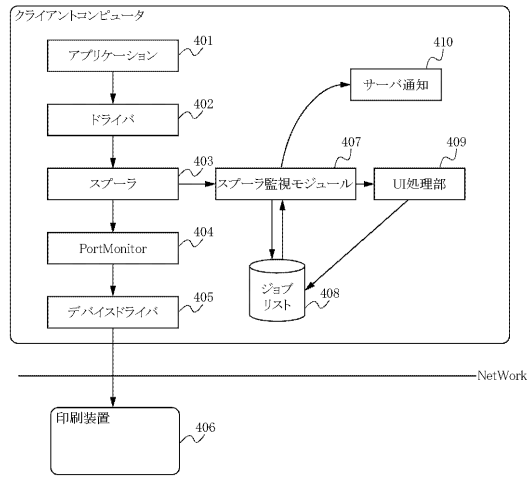
第3コード:  304 305 306

一時停止 再開 削除 更新

Job ID	ドキュメント名	オーナー	課金コード	状態
11	Readme.txt	guest	111.222.333	完了
12	schedule.doc	bob	111.222----	送信中
13	information.ppt	bob	-----	送信待ち
14	info.rtf	tom	-----	送信待ち
15	info.rtf	john	111.222.333	送信待ち
16	patent.doc	bob	-----	スプール済
17	project.xls	bob	-----	スプール中
18	announce.txt	jack	-----	スプール中

307

【 図 4 】



【 図 5 】

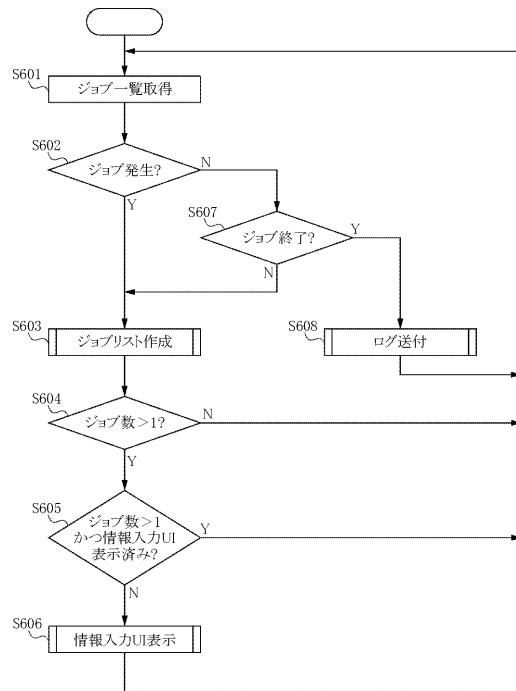
(a)

StartDate Time	"2006/12/01	12:01"
FinishDate Time	"2006/12/01	12:04"
JobName	"Document.doc"	
JobService/JobOwner	"PRINT"	
CompleteState	"Completed"	
DuplexMediumCount	"0"	
Nup	"0"	
TotalAmount	"150"	
TotalMediumCount	"15"	
TotalImpressionCount	"15"	
ColorImpressionCount	"0"	
BWImpressionCount	"15"	
FirstBillingCodeID	"12345"	
SecondBillingCodeID	"111"	
ThirdBillingCodeID	"--"	
DeviceMacAddress	"00-16-35-7A-EB-CC"	
DeviceHostName	"4P-OA.LBP"	
DeviceIPAddress	"150.86.11.22"	
ColorType	"1"	
PaperSize	"A4"	
ImpressionCount	"15"	
MaterialType	"1"	
PaperSize	"1"	
ImpressionCount	"15"	
MediumCount	"15"	

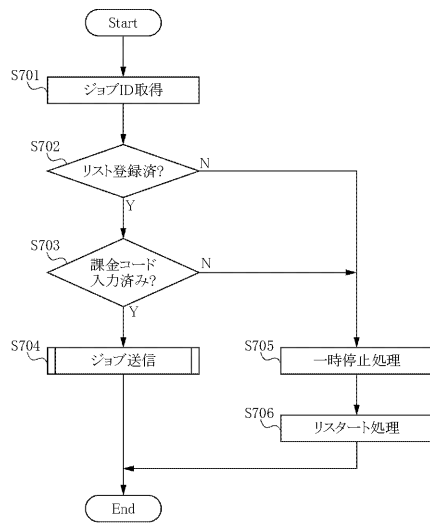
(b)

ログの各項目と項目の意味	
StartDataTime	ジョブの開始時刻
FinishDataTime	ジョブの終了時刻
JobName	ジョブ(ドキュメント)名
JobService/JobOwner	プリント/コピー種別
CompleteState	ジョブ完了状態
DuplexMediumCount	両面枚数
Nup	N-up数
TotalAmount	総印刷金額
TotalMediumCount	総印刷枚数
TotalImpressionCount	総印刷枚数
ColorImpressionCount	カラー面数
BWImpressionCount	白黒面数
FirstBillingCodeID	第一課金コード
SecondBillingCodeID	第二課金コード
ThirdBillingCodeID	第三課金コード
DeviceMacAddress	画像処理装置のMACアドレス
DeviceHostName	画像処理装置のホスト名
DeviceIPAddress	画像処理装置のIPアドレス
カラー・サイズ別情報(※サイズ毎に繰り返し)	
ColorType	白黒/カラー種別
PaperSize	用紙サイズ
ImpressionCount	カラー・サイズごとのジョブ完了面数(印刷実績値)
用紙タイプ別情報(※タイプ毎に繰り返し)	
MaterialType	用紙タイプ
PaperSize	用紙サイズ
ImpressionCount	用紙タイプごとのジョブ完了面数
MediumCount	用紙タイプごとのジョブ完了枚数

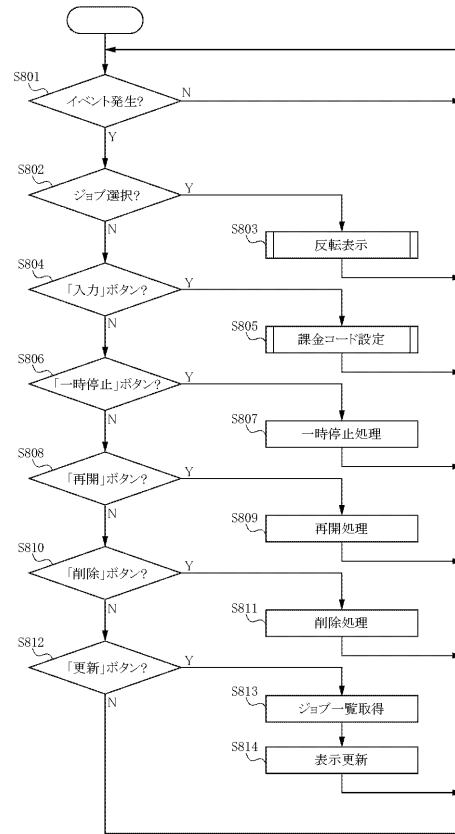
【 図 6 】



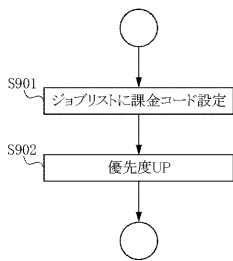
【 図 7 】



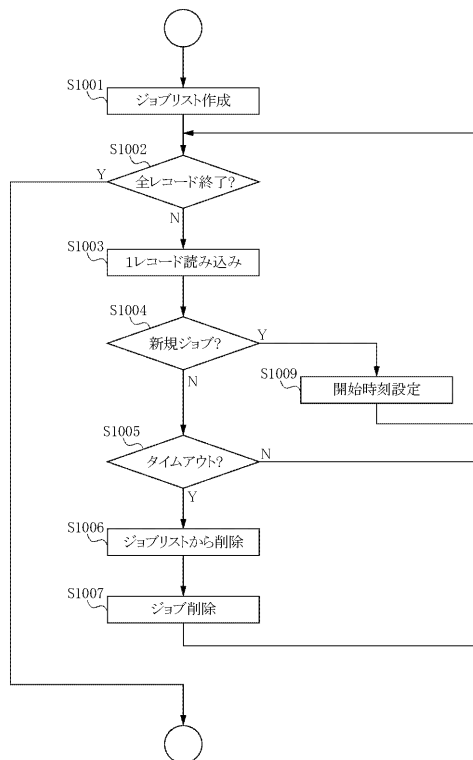
【 図 8 】



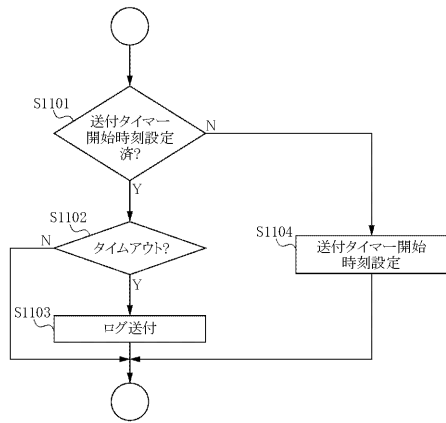
【 図 9 】



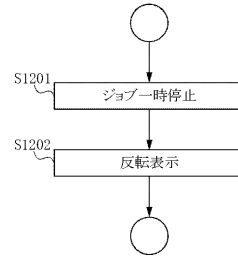
【 図 10 】



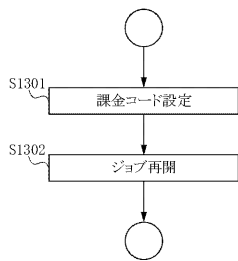
【 図 1 1 】



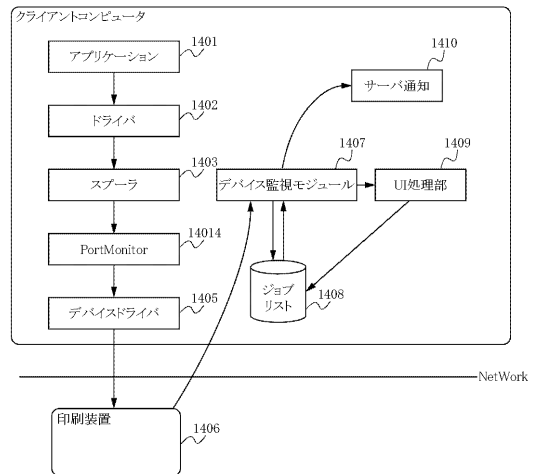
【 図 1 2 】



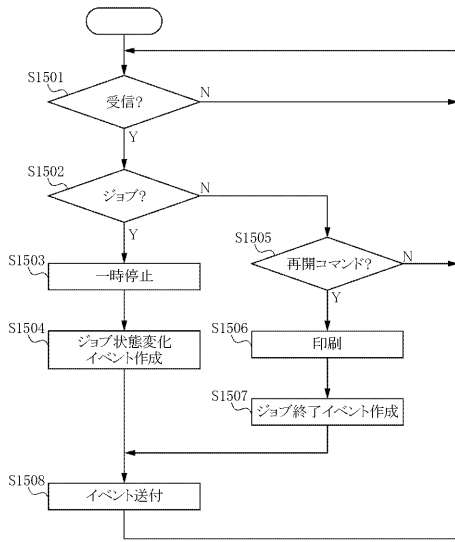
【 図 1 3 】



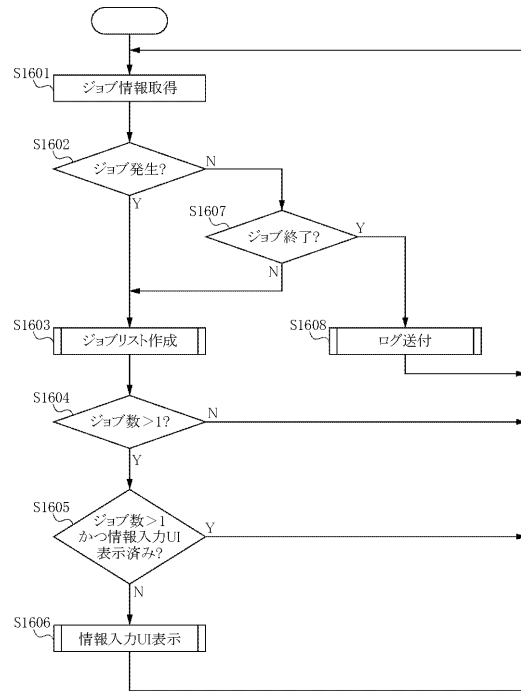
【 図 1 4 】



【 図 1 5 】



【 図 1 6 】



【 図 1 7 】

(a)

Job ID	"11"
Document Name	"Readme.txt"
Owner	"guest"
1st code	"111"
2nd code	"222"
3rd code	"333"
Status	"complete"

(b)

ジョブ情報の各レコードの例	
Job ID	ジョブを一意に決めるID
Document Name	ドキュメント名
Owner	ジョブオーナー
1st code	第1課金コード
2nd code	第2課金コード
3rd code	第3課金コード
Status	ジョブのステータス