

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6890938号
(P6890938)

(45) 発行日 令和3年6月18日 (2021.6.18)

(24) 登録日 令和3年5月28日 (2021.5.28)

(51) Int.Cl.

F I

G 0 6 F 3/12 (2006.01)
B 4 1 J 29/38 (2006.01)G O 6 F 3/12 3 2 5
B 4 1 J 29/38 2 0 1
B 4 1 J 29/38 2 0 2
B 4 1 J 29/38 2 0 4
B 4 1 J 29/38 3 0 1

請求項の数 6 (全 15 頁) 最終頁に続く

(21) 出願番号 特願2016-158735 (P2016-158735)
(22) 出願日 平成28年8月12日 (2016.8.12)
(65) 公開番号 特開2018-26049 (P2018-26049A)
(43) 公開日 平成30年2月15日 (2018.2.15)
審査請求日 令和1年7月31日 (2019.7.31)

前置審査

(73) 特許権者 000001007
キヤノン株式会社
東京都大田区下丸子3丁目30番2号
(74) 代理人 110001243
特許業務法人 谷・阿部特許事務所
(72) 発明者 横山 達之
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 豊田 真弓

最終頁に続く

(54) 【発明の名称】 情報処理装置

(57) 【特許請求の範囲】

【請求項 1】

ネットワークを介して画像処理装置に接続された情報処理装置であって、
前記画像処理装置の状態に関する情報を取得して表示するためのステータス管理プログラムを起動するための制御手段を備え、
前記情報処理装置において印刷ジョブを生成するプリンタドライバは、印刷ジョブの生成指示に従って前記ステータス管理プログラムを起動することができず、
前記制御手段は、前記ステータス管理プログラムをインストールするためのインストーラが備える登録手段によって登録され、前記画像処理装置に対する印刷ジョブが生成された場合、前記ステータス管理プログラムを起動することを特徴とする情報処理装置。

10

【請求項 2】

ネットワークを介して画像処理装置に接続された情報処理装置であって、
前記画像処理装置の状態に関する情報を取得して表示するためのステータス管理プログラムを起動するための制御手段を備え、
前記情報処理装置において印刷ジョブを生成するプリンタドライバは、印刷ジョブの生成指示に従って前記ステータス管理プログラムを起動することができず、
前記制御手段は、前記ステータス管理プログラムに付随する登録手段によって登録され、前記画像処理装置に対する印刷ジョブが生成された場合、前記ステータス管理プログラムを起動することを特徴とする情報処理装置。

【請求項 3】

20

ネットワークを介して画像処理装置に接続された情報処理装置であって、

前記画像処理装置の状態に関する情報を取得して表示するためのステータス管理プログラムを起動するための制御手段を備え、

前記情報処理装置において印刷ジョブを生成するプリンタドライバは、印刷ジョブの生成指示に従って前記ステータス管理プログラムを起動することができず、

前記制御手段は、前記画像処理装置に対する印刷が実行された場合に登録され、前記画像処理装置に対する印刷ジョブが生成された場合、前記ステータス管理プログラムを起動することを特徴とする情報処理装置。

【請求項 4】

前記画像処理装置または前記画像処理装置に対応するプリントキューを監視対象として登録する登録手段と、

前記画像処理装置または前記プリントキューにおけるイベントの発生を検出するイベント判定手段と、

を更に備え、

前記制御手段は、前記イベント判定手段がイベントを検出した際に、前記ステータス管理プログラムを起動することを特徴とする請求項 1 ないし 3 のいずれか 1 項に記載の情報処理装置。

【請求項 5】

スプーラーサービスの再起動を検知する検知手段を更に備え、

前記検知手段が前記スプーラーサービスの再起動を検知した場合、前記登録手段は、前記画像処理装置または前記画像処理装置に対応するプリントキューを監視対象のデバイスとして再登録することを特徴とする請求項 4 に記載の情報処理装置。

【請求項 6】

前記画像処理装置は、前記情報処理装置から提供された印刷ジョブに基づいてプリント動作を実行するプリンタであることを特徴とする請求項 1 ないし 5 のいずれか 1 項に記載の情報処理装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ネットワーク環境下において画像処理装置のようなデバイスに接続する情報処理装置、その制御方法およびプログラムに関する。

【背景技術】

【0002】

パーソナルコンピュータなどの情報処理装置とプリンタなどの画像処理装置とがネットワーク接続される情報処理システムにおいて、情報処理装置は、画像処理装置の状態（ステータス）を監視するための監視機能（ステータス管理プログラム）を備える。ステータス管理プログラムは、ステータスを最新情報に更新するため、定期的に画像処理装置からステータス情報を取得することが好ましい。しかし、定期取得を行うと、ネットワークトラフィックが増加したり、画像処理装置の処理負荷が増大したりして、画像処理装置または情報処理装置の処理速度が低下してしまう場合がある。

【0003】

マイクロソフト社の OS である Windows（登録商標）が提供するプリンタドライバ（v3 プリンタドライバ）アーキテクチャーにおいては、情報処理装置にベンダー独自のユーザインターフェースモジュールを登録する仕組みがある。そして、特許文献 1 には、プリント開始時に、OS からベンダー独自のユーザインターフェースモジュールの `DrvDocumentEvent` 関数がコールされ、当該関数内をベンダーが独自にカスタマイズする構成が開示されている。

【0004】

すなわち、従来の Windows プリンタドライバ（v3 プリンタドライバ）環境下で特許文献 1 を利用すれば、ステータス管理プログラムは、その動作期間を最小限に抑える

10

20

30

40

50

ことができる。具体的には、プリント開始のコマンドとともに起動され、ベンダー独自のユーザインターフェースモジュールからステータス管理プログラムを起動することができる。そして、情報処理装置は、画像処理装置との通信によって、プリントの進捗や、画像処理装置で発生しているエラーに関する情報を取得し、これをユーザに報知することができる。また、画像処理装置でプリント動作が完了したことをステータス管理プログラムが検知すると、ステータス管理プログラムは動作を終了することができる。

【0005】

このように、従来のWindowsプリンタドライバ(v3プリンタドライバ)アーキテクチャーでは、ステータス管理プログラムを、画像処理装置のプリント動作中などに限定して起動することが可能であった。

【先行技術文献】

【特許文献】

【0006】

【特許文献1】特開2010-140281号公報

【発明の概要】

【発明が解決しようとする課題】

【0007】

しかしながら、次世代のWindowsプリンタドライバ(v4プリンタドライバ)アーキテクチャーでは、上述したベンダー独自のユーザインターフェースモジュールの登録を行う仕組みがサポートされていない。そのため、次世代のWindowsプリンタドライバ(v4プリンタドライバ)では、プリント開始時にユーザインターフェースモジュールからステータス管理プログラムを起動することができない。

【0008】

v4プリンタドライバでは、Microsoft社のWindows(登録商標)OSの通知機能であるトースト通知機能を利用して、ステータス管理プログラムを起動する仕組みはサポートされている。しかし、OSによるトースト通知は、デフォルトで5秒程の表示で行われ、この通知期間にユーザがトーストをクリックしないとステータス管理プログラムを起動することができない。

【0009】

つまり、v4プリンタドライバのアーキテクチャーにおいては、プリント開始時にv4プリンタドライバからステータス管理プログラムを起動することができない。その結果、ユーザに対して、報知すべき情報が報知されないおそれがあった。

【0010】

この課題を回避するために、インストーラからv4プリンタドライバのインストールを行う際に、ステータス管理プログラムをスタートアップ登録やサービス登録をすることで常駐起動させる手法がある。しかし、インストーラを利用できず、プリンタドライバのみを単体でインストールせざるを得ない場合には、この手法が適用できない。例えば、マイクロソフト社が運営するWindows Updateの配信サーバーからプリンタドライバがインストールされる環境や、マイクロソフト社のPoint&Print(登録商標)の仕組みで構成されるプリントサーバシステム環境がこれに該当する。本発明は上記課題を解決するためになされたものである。よってその目的とするところは、v4プリンタドライバのアーキテクチャーにおいても、インストーラの利用可否に係らず、画像処理装置の動作中のステータスを監視することが可能な状況を実現することである。

【課題を解決するための手段】

【0011】

そのために本発明は、ネットワークを介して画像処理装置に接続された情報処理装置であって、前記画像処理装置の状態に関する情報を取得して表示するためのステータス管理プログラムを起動するための制御手段を備え、前記情報処理装置において印刷ジョブを生成するプリンタドライバは、印刷ジョブの生成指示に従って前記ステータス管理プログラムを起動することができず、前記制御手段は、前記ステータス管理プログラムをインスト

10

20

30

40

50

ールするためのインストーラが備える登録手段によって登録され、前記画像処理装置に対する印刷ジョブが生成された場合、前記ステータス管理プログラムを起動することを特徴とする。

【発明の効果】

【0012】

本発明によれば、v4プリンタドライバのアーキテクチャーにおいても、インストーラの利用可否に係らず、画像処理装置の動作中のステータスを監視することが可能となる。

【図面の簡単な説明】

【0013】

【図1】情報処理装置と画像処理装置のハードウェア構成図である。

10

【図2】情報処理装置と画像処理装置を接続するネットワーク環境を示す図である。

【図3】ステータス管理プログラムによるUI表示の一例である。

【図4】インストーラの起動に伴う処理を示すフローチャートである。

【図5】インストーラの起動を伴わない場合の処理の一例を示すフローチャートである。

【図6】インストーラの起動を伴わない場合の処理の別例を示すフローチャートである。

【図7】プリンタの監視処理を説明するためのフローチャートである。

【図8】第2の実施形態におけるネットワーク環境を示す図である。

【図9】プリンタの監視処理を説明するためのフローチャートである。

【図10】スプーラー監視処理を説明するためのフローチャートである。

【発明を実施するための形態】

20

【0014】

(第1の実施形態)

図1は、本実施形態で使用する情報処理装置100と画像処理装置150のハードウェア構成図である。情報処理装置100は、入力インタフェース102、CPU103、ROM104、RAM105、外部記憶装置106、出力インタフェース108、表示部107、キーボード101、マウス109、ネットワークインタフェース110を有する。

【0015】

ネットワークインタフェース110は、ネットワークケーブル111を介して、情報処理装置100とネットワーク220を接続し、両者の間で情報の入出力を行う。ROM104には初期化プログラムが保存されており、外部記憶装置106にはアプリケーションプログラム群、OS(Operating System)、プリンタドライバ、その他各種のデータが保存されている。CPU103は、外部記憶装置106に格納される各種プログラムに従って、RAM105をワークメモリとして使用しながら、各種処理を実行する。

30

【0016】

画像処理装置150は、ネットワークインタフェース151、RAM152、プリントエンジン153、ROM154、CPU156を有する。ネットワークインタフェース151は、ネットワークケーブル157を介して、画像処理装置150とネットワーク220を接続し、両者の間で情報の入出力を行う。CPU156は、ROM154に記憶されている各種プログラムに従って、RAM152をワークエリアとして使用しながら、各種処理を実行する。例えば、CPU156は、ネットワーク220より受信した画像データを一度RAM152に保存し、プリントエンジン153を起動して当該画像データに従ったプリント動作を行わせる。

40

【0017】

ROM154に記憶されている各種プログラムのうち、ステータス管理モジュール155は、画像処理装置150の内部に備えられた各種センサの情報をもとに、画像処理装置の状態を監視するためのプログラムである。CPU156は、ステータス管理モジュール155に従って、各種センサの情報をもとに画像処理装置のステータス情報を作成し、RAM152に記憶する。

【0018】

50

図2は、情報処理装置100と画像処理装置150を接続するネットワーク環境を示す図である。ここでは、ネットワーク220に対し、情報処理装置100としてのPC210と、画像処理装置150としてのプリンタ260と、サーバーPC250が接続された状態を示し、互いに関係するソフトウェアの構成をブロック図で示している。ただし、ネットワーク220には、更に多くの制御装置やデバイスが接続されていても良い。なお、各装置のCPUが図2に記載されているソフトウェアを実行することで、各ソフトウェアの処理が実行される。

【0019】

PC210には、インストーラ211、プリンタドライバ221、監視アプリケーション241が用意されている。インストーラ211は、ユーザによって起動されることにより、プリンタドライバ221や監視アプリケーションをPC210にインストールするためのソフトウェアである。プリンタドライバ221は、プリンタ260に印刷ジョブを送信してこれを動作させたり、プリンタ260の状態(ステータス)を取得してユーザに情報を提供したりするためのソフトウェアである。監視アプリケーション241は、PC210のプリントキューを監視し、プリントキューに変化が発生した場合に、その内容を解析する。変化の要因が印刷ジョブの投入である場合、プリンタドライバ221に印刷ジョブが投入された旨を通知し、ステータス管理プログラム231を起動させるためのソフトウェアである。一方、プリンタ260には、プリント動作を制御するためのプリント部261と、プリンタ260のステータスを保持する管理情報保持部262が用意されている。

【0020】

本実施形態では、インストーラ211が起動されることによってプリンタドライバ221をインストールすることも出来る。また他の方法として、Point & Print環境下でクライアントとみなされたPC210に対し、サーバーPC250からプリンタドライバ221をインストールすることもできる。また、Windows Updateなどの配信サーバーを利用することで、PC210は、プリンタドライバ221をインストールすることも出来る。

【0021】

プリンタドライバ221がインストーラ211の起動によってインストールされる場合、監視アプリケーション241はインストーラ211内の監視アプリケーション登録モジュールA212によってスタートアップ登録される。なお、スタートアップ登録とは、OSが管理するスタートアップフォルダに、アプリケーション等のソフトウェアまたはソフトウェア起動用のリンクファイルを保存することである。そして、スタートアップ登録されたソフトウェアは、ユーザーがログインした場合、自動的に呼び出されて常駐ソフトとして起動される。一方、プリンタドライバ221がインストーラ211を介さずにインストールされる場合、監視アプリケーション241はプリンタドライバ221内の監視アプリケーション登録モジュールB222によってスタートアップ登録される。例えば、サーバーPC250がPoint & PrintによってPC210をクライアントとみなしてプリンタドライバ221をインストールする場合が後者に相当する。また、Windows Updateなどの配信サーバーからプリンタドライバ221をインストールする場合も後者に相当する。

【0022】

プリンタドライバ221のステータス管理プログラム231は、管理情報取得モジュール232と、管理情報表示モジュール233で構成される。管理情報取得モジュール232はプリンタ260にアクセスし、プリンタ260のステータス情報を管理情報保持部262から取得する。管理情報表示モジュール233は、管理情報取得モジュール232が取得したステータス情報に基づいて、ユーザに必要な情報をUI表示する。

【0023】

なお、上述したようにPC210で動作するOSは、プリンタドライバが印刷ジョブの生成指示を受けることに従って、プリンタドライバ自らステータス管理プログラム231

10

20

30

40

50

を起動することができない（または制限された）環境を提供する。

【 0 0 2 4 】

図 3 は、ステータス管理プログラム 2 3 1 による UI 表示の一例である。タイトルバー 3 0 1 はプリンタの名称等を表示する。メッセージ表示部 3 0 2 は、管理情報取得モジュール 2 3 2 で取得した情報に基づいて、プリンタ 2 6 0 の状況をユーザに報知する。ジョブ情報表示部 3 0 3 は、管理情報取得モジュール 2 3 2 で取得した情報に基づいて、プリント動作の進捗状況を表示する。インク情報表示部 3 0 4 は、管理情報取得モジュール 2 3 2 で取得した情報に基づいて、プリンタ 2 6 0 で使用するインクの消費状況を表示する。このような情報を確認することにより、ユーザは、プリントジョブの進捗状況を把握したり、インクを交換する時期や種類を確認したり、エラー発生の有無を認識したりすることができる。そして、このような情報は特にプリンタの動作中において有用である。

10

【 0 0 2 5 】

図 2 に戻る。監視アプリケーション 2 4 1 は、イベント登録モジュール 2 4 2 と、イベント判定モジュール 2 4 3 と、ステータス管理プログラム起動モジュール 2 4 4 で構成される。イベント登録モジュール 2 4 2 は、P C 2 1 0 における監視対象となるプリントキューに対応するイベントの登録および管理を行う。本実施形態では、P C 2 1 0 のプリントキューが監視対象の 1 つとして登録されている。当該プリントキューに印刷ジョブが投入されると、プリンタ 2 6 0 に対して、印刷ジョブが送信される。イベント判定モジュール 2 4 3 は、イベント登録モジュール 2 4 2 によって登録されているプリントキューにおけるイベントの発生を検出し、その内容が印刷ジョブの投入に伴うものであるか解析し、判定を行う。具体的には、新規の印刷ジョブが投入されたか否かを解析することで、この判定処理が実現される。ステータス管理プログラム起動モジュール 2 4 4 は、イベント判定モジュール 2 4 3 によるイベントの判定結果を受けて、印刷ジョブの投入に伴うイベントであった場合に、ステータス管理プログラム 2 3 1 を起動する。

20

【 0 0 2 6 】

このように、P C 2 1 0 のプリントキューの監視や図 3 に示す情報表示は、ステータス管理プログラム起動モジュール 2 4 4 が、イベント発生時に、プリンタドライバ 2 2 1 のステータス管理プログラム 2 3 1 を起動させることにより実現される。このため、監視アプリケーション 2 4 1 は、プリンタドライバ 2 2 1 のインストール方法によらず、スタートアップ登録されていることが望まれる。

30

【 0 0 2 7 】

図 4 は、ユーザによってインストーラ 2 1 1 の起動コマンドが入力された場合に、P C 2 1 0 の制御部（C P U 1 0 3）がインストーラ 2 1 1 を実行することで実現される処理を示すフローチャートである。本処理が開始され、ステップ S 4 0 1 でインストーラ 2 1 1 が起動されると、インストーラ 2 1 1 は、ステップ S 4 0 2 にて、プリンタドライバ 2 2 1 とともにステータス管理プログラム 2 3 1 をインストールする。

【 0 0 2 8 】

ステップ S 4 0 3 において、インストーラ 2 1 1 は、監視アプリケーション 2 4 1 が既に P C 2 1 0 に登録されているか否かを確認する。具体的には、O S が管理するログインユーザのスタートアップフォルダに監視アプリケーション 2 4 1 が存在するか否かを確認する。未だ登録されていないと判断した場合は、ステップ S 4 0 4 に進み、インストーラ 2 1 1 は、監視アプリケーション登録モジュール A 2 1 2 を用いて、監視アプリケーション 2 4 1 をスタートアップ登録し、ステップ S 4 0 5 に進む。一方、監視アプリケーション 2 4 1 が既に登録されていると判断した場合は、そのままステップ S 4 0 5 にジャンプする。この S 4 0 4 の処理が実行されることで、O S が管理するログインユーザのスタートアップフォルダに監視アプリケーション 2 4 1 が保存される。

40

【 0 0 2 9 】

続くステップ S 4 0 5 において、インストーラ 2 1 1 は、監視アプリケーション 2 4 1 を起動させ、さらにステップ S 4 0 6 で監視処理を開始させる。以上で本処理は終了する。

50

【 0 0 3 0 】

図5は、インストーラ211を用いることなくプリンタドライバ221をインストールする場合の一例を説明する図である。具体的には、PC210は、インストーラを用いることなく、プリンタドライバ221をインストールし、インストール完了後、ユーザーからのプリントコマンドの入力に伴い、制御部(CPU103)が実行するフローチャートである。なお、プリントコマンドとは、例えば、ユーザがアプリケーションのUIを用いてプリンタドライバを選択して、印刷実行を指示することに相当する。

【 0 0 3 1 】

本処理が開始されるとCPU103は、まずステップS501で、プリンタドライバ221をインストールする。具体的には、PC210は、Windows Updateなどの配信サーバーや、Point & Print環境下でサーバーPC250にて管理されているプリンタを検索して、検索結果から選択されたプリンタのプリンタドライバ221をインストールする。

10

【 0 0 3 2 】

続いて、CPU103はステップS502に進み、ステータス管理プログラム231をインストールする。ステップS503に進み、ユーザによってプリントコマンドが入力されると、プリンタドライバ221は、ステップS501でインストールしたプリンタドライバのフィルターモジュールを用いて受信したプリントコマンドに従い、プリント動作を実行する。具体的には、CPU103は、プリンタドライバ221を用いて、アプリケーションから発行されたアプリケーションデータから印刷ジョブを生成し、プリントキューを介してプリンタ260へ送信する。

20

【 0 0 3 3 】

ステップS504では、CPU103は、監視アプリケーション241が既にPC210に登録されているか否かを確認する。具体的には、OSが管理するログインユーザのスタートアップフォルダに監視アプリケーション241が存在するか否かを確認する。未だ登録されていないと判断した場合は、ステップS505に進み、CPU103は、監視アプリケーション登録モジュールB222を用いて、監視アプリケーション241を登録する。具体的には、CPU103は、OSのログインユーザのスタートアップフォルダに、プリンタドライバ221の監視アプリケーション登録モジュールB222をコピーして保存する。この処理により、監視アプリケーション241が、スタートアップ登録される。一方、ステップS504で監視アプリケーション241が既に登録されていると判断した場合は、そのままステップS506にジャンプする。

30

【 0 0 3 4 】

ステップS506では、CPU103は、PC210を再起動し、ステップS507ではステップS505で登録された監視アプリケーションを起動させる。更に、ステップS508で監視処理を開始させ、本処理は終了する。

【 0 0 3 5 】

図6は、インストーラ211を用いることなくプリンタドライバ221をインストールする場合の別例を説明する図である。具体的には、ユーザがプリンタドライバ221とともに監視アプリケーションをインストールし、スタートアップ登録させる場合に、制御部(CPU103)が実行するフローチャートである。

40

【 0 0 3 6 】

本処理が開始されるとCPU103は、まずステップS601で、プリンタドライバ221をインストールする。詳細な説明は、S501と同じ処理であるため省略する。更に、ステップS602にて、CPU103は、ステータス管理プログラム231をインストールし、続くステップS603において、インストールしたステータス管理プログラム231を起動する。

【 0 0 3 7 】

続くステップS604において、CPU103は、監視アプリケーション241が既にスタートアップ登録されているか否かを確認する。詳細な説明は、S504と同じ処理で

50

あるため省略する。未だ登録されていないと判断した場合は、ステップS605に進み、CPU103は、ステータス管理プログラム231に付随されている監視アプリケーション登録モジュールB222を用いて、監視アプリケーション241をスタートアップ登録する。具体的には、OSのログインユーザのスタートアップフォルダに、監視アプリケーション241をコピーする。一方、監視アプリケーション241が既に登録されていると判断した場合は、そのままステップS606にジャンプする。

【0038】

その後、ステップS606で、CPU103は、監視アプリケーションを起動させ、ステップS607で監視処理を開始させ、本処理は終了する。以上図4～図6で説明したように、本実施形態では様々な方法で、プリンタドライバ221および監視アプリケーション241をインストールおよび登録することができる。

10

【0039】

図7は、PC210の制御部(CPU103)が、監視アプリケーション241を実行することで実現されるPC210のプリントキューの監視処理を説明するためのフローチャートである。なお、図7の処理は、上述したS406、S508、S607により開始される。本処理が開始されると、監視アプリケーション241は、まずステップS701にて、イベント登録モジュール242を用い、デバイスフォルダに記憶されているデバイスアイコン(プリントキュー)に対応するプリンタドライバ名やポート名等の情報を取得する。ステップS702では、監視アプリケーション241は、ステップS701で取得したプリンタドライバ名やポート名の情報に基づき、監視対象となるデバイスアイコン(プリントキュー)が存在するか判定を行う。つまり、監視アプリケーション241は、監視対象とすべきプリンタに関する情報を保持しており、その保持している情報と、S701で取得された情報を比較することで、S702の判定を行う。監視対象となるデバイスアイコン(プリントキュー)が存在しない場合は本処理を終了する。監視対象となるデバイスアイコン(プリントキュー)が存在する場合は、ステップS703に進む。そして、S703において、監視アプリケーション241は、イベント登録モジュール242を用いて、監視対象と判定されたデバイスアイコン(プリントキュー)に対してイベントの登録処理を行う。

20

【0040】

ステップS703の登録処理においては、例えばOSが提供するFindFirstPrinterChangeNotification関数を利用することができる。監視アプリケーション241は、上述した関数を用いて、監視対象と判定されたデバイスアイコンにイベントが発生した場合にイベント発生を報知するようにスプーラーに登録する。この処理により、例えば、監視対象のプリンタ260に対応するプリントキューに印刷ジョブが投入された等のイベントをスプーラーが検知した場合、スプーラーが監視アプリケーション241にイベント発生を通知する。その結果、監視アプリケーション241は、監視対象のプリンタ260に対応するプリントキューに印刷ジョブが投入されたことを認識して後述するS704の判定処理を実行することが可能となる。

30

【0041】

ステップS704において、監視アプリケーション241は、イベント判定モジュール243を用いてイベントの発生を監視する状態に入る。監視下においてイベント判定モジュール243がイベントを検出すると、監視アプリケーション241はステップS705に進み、イベント判定モジュール243を用いてイベントの発生情報を解析する。

40

【0042】

ステップS706において、監視アプリケーション241は、ステップS705で解析した結果に基づき、検出したイベントがプリントに関するイベントであるか否かを判断する。プリントに関するイベントであると判断した場合は、ステップS707に進む。S707において、監視アプリケーション241は、ステータス管理プログラム起動モジュール244を用いて、ステータス管理プログラム231を起動する。そして、新たなイベントの発生を検出するために、再びステップ704のイベント監視状態に戻る。

50

【 0 0 4 3 】

一方、ステップ S 7 0 6 で、検出したイベントがプリントに関するイベントでは無いと判断した場合は、ステップ S 7 0 8 にジャンプし、監視アプリケーション 2 4 1 は、検出したイベントがデバイスアイコンの増減に伴うイベントであるか否かを判断する。デバイスアイコンの増減に伴うイベントである場合は、再度ステップ S 7 0 1 に戻り、デバイスアイコンに対応するプリンタドライバ名やポート名等の最新情報を取得する。デバイスアイコンの増減に伴うイベントである場合は、再度、S 7 0 3 の処理を行う必要があるか否かを決める必要があるため、S 7 0 1 へと処理が戻る。デバイスアイコンの増減に伴うイベントでない場合は、ステップ S 7 0 4 のイベント監視状態に戻る。

【 0 0 4 4 】

10

以上説明したように、本実施形態においては、プリンタドライバ 2 2 1 とは別に監視アプリケーション 2 4 1 を用意する。その結果、P C 2 1 0 のプリントキューにおけるイベントの発生に伴って、監視対象のプリンタ 2 6 0 のプリントキューに対応するステータス管理プログラムを起動させることができる。このため、ステータス管理プログラムを常駐させる必要がなくなり、必要以上にネットワークトラフィックを増加させることなく、ユーザは必要なデバイス（プリンタ）のステータスを必要なタイミングで把握することができる。

【 0 0 4 5 】

更に、本実施形態によれば、インストーラ 2 1 1 だけでなくプリンタドライバ 2 2 1 においても、監視アプリケーションを情報処理装置にインストールするためのモジュールを備えている。よって、プリンタドライバ 2 2 1 自体が、どのような過程でインストールされたとしても、情報処理装置に監視アプリケーション 2 4 1 をインストールすることができる。すなわち、O S のバージョンやインストーラの利用可否に係らず、画像処理装置のステータスを適切なタイミングで監視することが可能となる。

20

【 0 0 4 6 】

（第 2 の実施形態）

図 8 は、本実施形態におけるネットワーク環境を示す図である。第 1 の実施形態と異なる点は、監視アプリケーション 8 4 1 にスプーラー再起動検知モジュール 8 4 5 が備えられている点である。なお、本実施形態においても、図 1 で説明した情報処理装置 1 0 0 と画像処理装置 1 5 0 を使用するものとする。また、監視アプリケーション 8 4 1 の登録においても、図 4 ～ 図 6 で説明したフローチャートに従って行うことができるものとする。

30

【 0 0 4 7 】

第 1 の実施形態では、図 7 のステップ S 7 0 3 において、O S が提供している `Find First Printer Change Notification` 関数を利用することにより、イベント登録処理を行った。但し、この方法を採用した場合には、スプーラーが再起動されるとイベント登録も無効になってしまう。つまり、スプーラーが再起動されると、監視アプリケーション 2 4 1 が監視対象のプリンタ 2 6 0 のプリントキューに対応するステータス管理プログラムを起動できなくなるおそれがある。なお、スプーラーの再起動は、例えばユーザの指示により発生する。その他、新規にプリンタドライバがインストールされた場合に、スプーラーが再起動されることもある。このため、本実施形態では、スプーラー再起動検知モジュール 8 4 5 を用意し、スプーラーの再起動が検知されたタイミングで、イベント情報を再登録する。

40

【 0 0 4 8 】

図 9 は、本実施形態において、P C 8 1 0 の C P U 1 0 3 が、監視アプリケーション 8 4 1 として実行する監視処理を示すフローチャートである。本処理が開始されると、監視アプリケーション 8 4 1 は、まずステップ S 9 0 0 において、スプーラー再起動検知モジュール 8 4 5 を用いて、スプーラー監視処理を開始する。スプーラー監視処理の詳細については、後に詳しく説明する。

【 0 0 4 9 】

スプーラー監視処理が開始されると、監視アプリケーション 8 4 1 はステップ S 9 0 1

50

- S 9 0 2 の処理を実行するが、これらの処理は S 7 0 1 - S 7 0 2 と同じであるため詳細な説明は省略する。監視対象となるデバイスアイコン（プリントキュー）が存在しない場合は、ステップ S 9 1 0 に進み、スプーラー監視フラグを O F F に設定し、本処理を終了する。スプーラー監視フラグを O F F に設定することにより、本実施形態では、スプーラー再起動検知モジュール 8 4 5 がこれを確認し、ステップ S 9 0 0 で開始させたスプーラー監視処理が終了されるようになっている。

【 0 0 5 0 】

一方、ステップ S 9 0 2 で監視対象となるデバイスアイコン（プリントキュー）が存在すると判断した場合は、ステップ S 9 0 3 に進む。そして、イベント登録モジュールを用いて監視対象と判定されたデバイスアイコン（プリントキュー）に対してイベントの登録処理が行われる。この際、O S が提供する F i n d F i r s t P r i n t e r C h a n g e N o t i f i c a t i o n 関数が利用される。

10

【 0 0 5 1 】

ステップ S 9 0 4 において、監視アプリケーション 8 4 1 は、スプーラー再起動検知モジュール 8 4 5 を用いてスプーラーの再起動を検知したか否かを判断する。検知したと判断した場合は、デバイスアイコン（プリントキュー）に対応するプリンタドライバ名やポート名等の情報を再度取得するために、ステップ S 9 0 1 に戻る。一方、スプーラーの再起動を検知していない場合は、ステップ S 9 0 5 に進む。

【 0 0 5 2 】

ステップ S 9 0 5 では、監視アプリケーション 8 4 1 は、イベント判定モジュール 8 4 3 を用いてイベントの発生を検出したか否かを判断する。イベント判定モジュール 2 4 3 がイベントを検出したと判断すると、監視アプリケーション 8 4 1 はステップ S 9 0 6 に進み、イベント判定モジュール 8 4 3 を用いてイベントの内容を解析する。一方、イベント判定モジュール 2 4 3 がイベントを検出していない場合は、ステップ S 9 0 4 に戻る。ステップ S 9 0 4 および S 9 0 5 は、スプーラーの再起動またはイベント発生 of 少なくとも一方が検出されるまで繰り返され、監視アプリケーション 8 4 1 による監視状態が継続される。

20

【 0 0 5 3 】

図 9 におけるステップ S 9 0 6 - S 9 0 9 は、図 7 の S 7 0 5 - S 7 0 8 と同じであるため詳細な説明は省略する。

30

図 1 0 は、C P U 1 0 3 が、スプーラー再起動検知モジュール 8 4 5 を用いて実行するスプーラー監視処理を説明するためのフローチャートである。本処理は、図 9 のステップ S 9 0 0 で開始され、図 9 で説明した監視処理とは別スレッドで並行して行われる。

【 0 0 5 4 】

本処理が開始されると、スプーラー再起動検知モジュール 8 4 5 は、まずステップ S 1 0 0 1 においてスプーラー監視フラグを O N に設定する。続くステップ S 1 0 0 2 において、スプーラー再起動検知モジュール 8 4 5 は、スプーラー監視フラグの O N / O F F を確認する。O F F である場合は、本処理を終了する。一方、O N である場合は、ステップ S 1 0 0 3 に進み、スプーラーサービスの状態を取得する。

【 0 0 5 5 】

40

ステップ S 1 0 0 4 において、スプーラー再起動検知モジュール 8 4 5 は、ステップ S 1 0 0 3 で取得した情報に基づき、スプーラーサービスが開始状態であるか否かを判断する。開始状態である場合は、ステップ S 1 0 0 5 に進み、スプーラー停止イベントを待ち受け、これを検知すると、その後ステップ S 1 0 0 6 に進む。なお、S 1 0 0 5 は、スプーラー停止イベントを検知するまで継続される。一方、ステップ S 1 0 0 4 でスプーラーサービスが開始状態ではないと判断した場合は、そのままステップ S 1 0 0 6 にジャンプする。

【 0 0 5 6 】

ステップ S 1 0 0 6 では、スプーラー開始イベントを待ち受け、これを検知すると、その後ステップ S 1 0 0 7 に進む。ステップ S 1 0 0 7 において、スプーラー再起動検知モ

50

ジュール 8 4 5 は、スプーラーの再起動を検知した旨をイベント判定モジュール 8 4 3 に通知し、監視状態を継続するために、ステップ S 1 0 0 2 に戻る。そして、以上の監視処理は、図 9 のステップ S 9 1 0 において、スプーラー監視フラグが O F F に設定されるまで継続する。

【 0 0 5 7 】

以上説明した本実施形態によれば、監視アプリケーション 8 4 1 の監視中にスプーラーが再起動されたとしても、これを検知し即座に監視対象のデバイスアイコン（プリントキュー）に対してイベントの再登録することができる。具体的には、監視アプリケーション 8 4 1 は、S 1 0 0 7 の通知を受けることで S 9 0 4 にて Y e s と判定し、図 9 の処理を S 9 0 1 に戻すことで、監視対象のプリンタ 2 6 0 のデバイスアイコン（プリントキュー）に対してイベントを再登録できる。その結果、デバイスアイコン（プリントキュー）におけるイベントの発生を逃すことなく、そのステータスを必要なタイミングで把握することができる。

10

【 0 0 5 8 】

（その他の実施形態）

以上の実施形態では、監視アプリケーションのインストールおよび登録を、図 4 ～ 図 6 で説明した 3 つの方法で行う場合を例示したが、本発明はこれら方法に限定されるものではない。例えば、サービス登録その他のモジュールが、監視アプリケーション 2 4 1 の常時起動登録を行っても良い。

【 0 0 5 9 】

20

また、監視アプリケーション登録モジュール 2 1 2 （ 2 2 2 ）は、ログインしていないユーザのスタートアップフォルダや D e f a u l t のスタートアップフォルダに対しても、監視アプリケーション 2 4 1 のスタートアップ登録を行っても良い。これにより、ログインしていないユーザや、新規にアカウントを作成したユーザの環境でも、監視アプリケーション 2 4 1 が常駐起動することができる。

【 0 0 6 0 】

また、監視アプリケーションの登録および起動は、必ずしもステータス管理プログラム 2 3 1 の起動に伴って行われなくても良い。例えば、プリンタドライバ 2 2 1 がプリント設定画面を表示したタイミングや、トーストをクリックして監視アプリケーション 2 4 1 を起動する際に、登録および起動される形態とすることもできる。

30

【 0 0 6 1 】

また、実施形態 1 および 2 では、P C 2 1 0 上のプリントキューを監視し、当該プリントキューに発生するイベントを検出する方法をとったが、プリンタ 2 6 0 に対して、イベント登録する方法をとっても良い。その場合、プリンタ 2 6 0 でプリントが実行されたり、エラーが発生したりすると、プリンタ 2 6 0 から監視アプリケーションに対してイベントが通知され、それを受けて監視アプリケーションはステータス管理プログラム 2 3 1 の起動を行う。

【 0 0 6 2 】

さらに、以上では、情報処理装置にステータス情報を提供する画像処理装置の一例としてプリンタを用いたが、無論本発明はこのような形態に限定されるものではない。例えば、原稿をスキャンして読み取った画像データを情報処理装置に提供するスキャナや、読み取った画像を更にコピーする機能を備えた複写機なども、本発明の画像処理装置 1 5 0 として利用することができる。このような場合でも、原稿の読み取り処理の進捗状態などがステータス情報として情報処理装置に提供されれば、上記実施形態と同様の効果を得ることが出来る。

40

【 0 0 6 3 】

さらにまた、本発明は、以下の処理を実行することによっても実現される。即ち、上述した実施形態の機能を実現するソフトウェア（プログラム）を、ネットワーク又は各種記憶媒体を介してシステム或いは装置に供給し、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 等）がプログラムを読み出して実行する処理である。また、プログ

50

ラムは、１つのコンピュータで実行させても、複数のコンピュータを連動させて実行させるようにしてもよい。また、上記した処理の全てをソフトウェアで実現する必要はなく、処理の一部または全部をＡＳＩＣ等のハードウェアで実現するようにしてもよい。また、ＣＰＵも１つのＣＰＵで全ての処理を行うものに限らず、複数のＣＰＵが適宜連携をしながら処理を行うものとしてもよい。

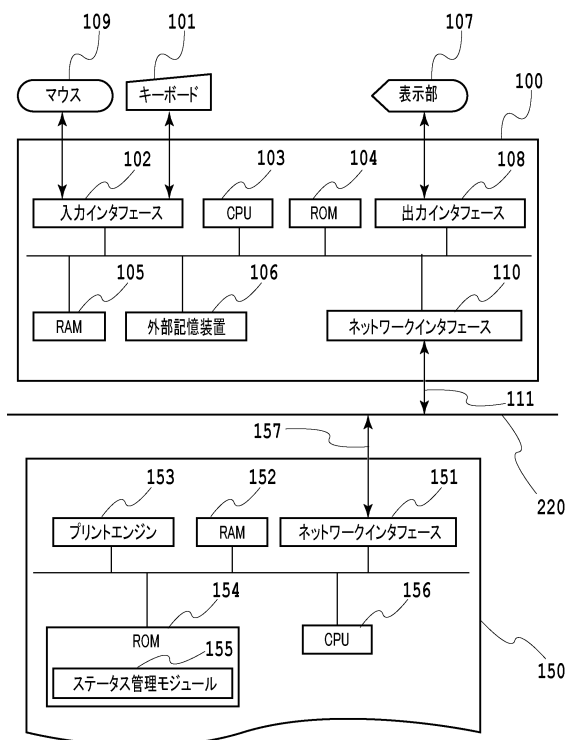
【符号の説明】

【 ０ ０ ６ ４ 】

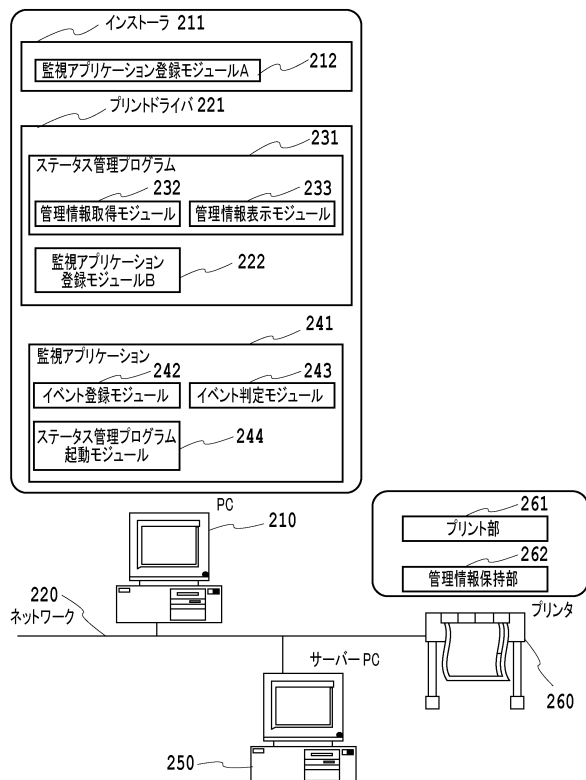
１００	情報処理装置
１０３	ＣＰＵ
１５０	画像処理装置
２３１	ステータス管理プログラム
２４１	監視アプリケーション
２６０	プリンタ

10

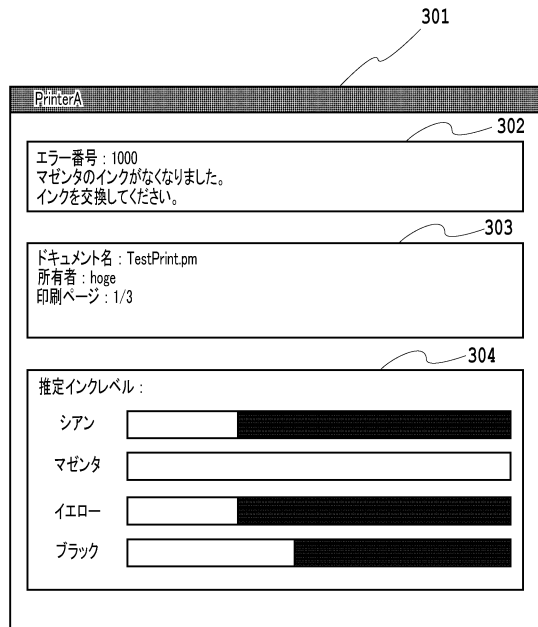
【図１】



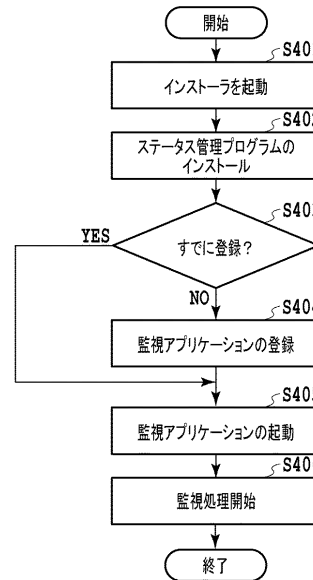
【図２】



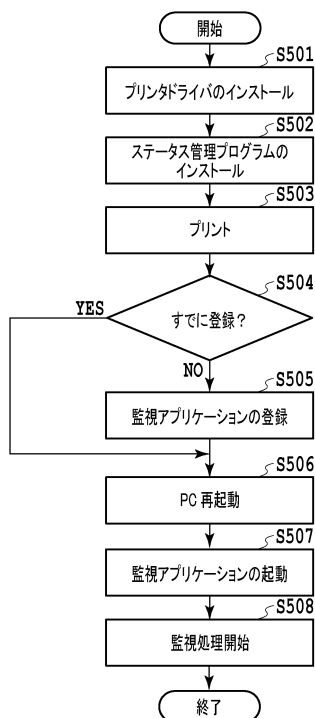
【図 3】



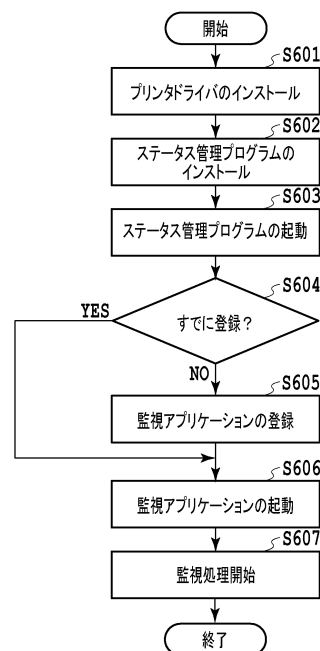
【図 4】



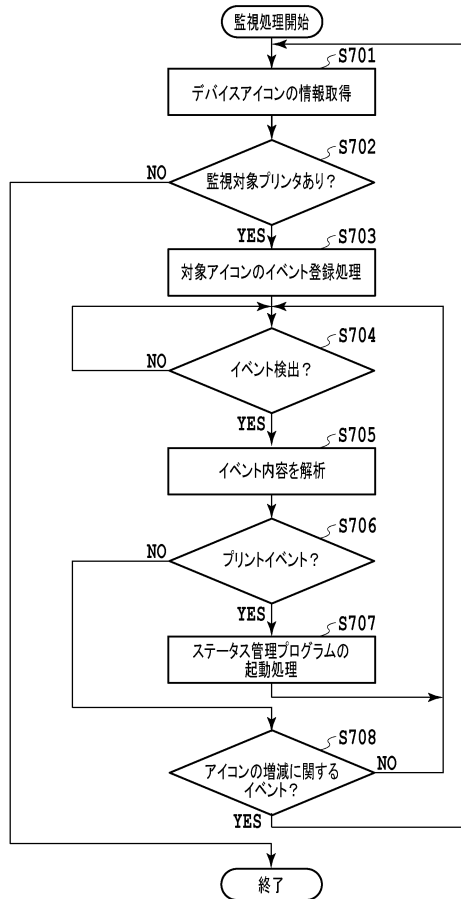
【図 5】



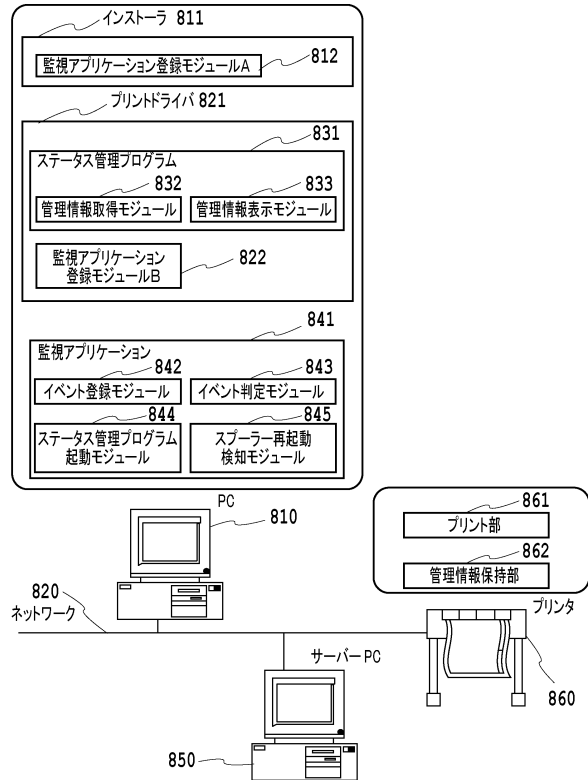
【図 6】



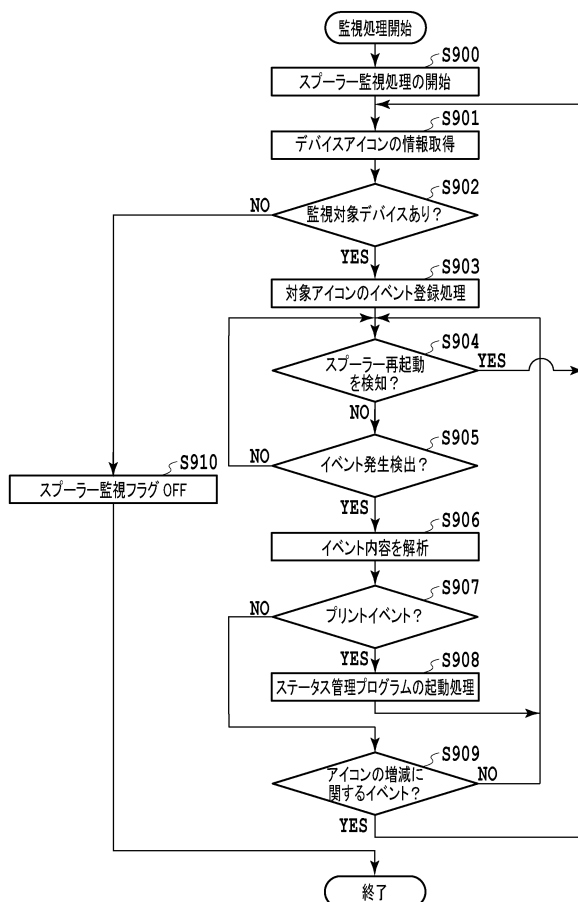
【図 7】



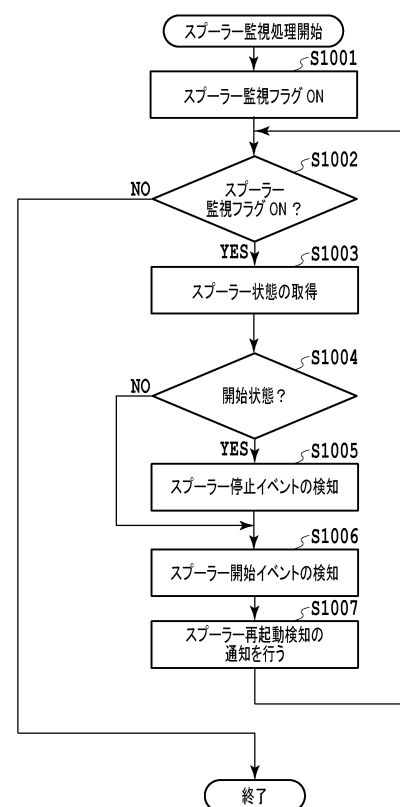
【図 8】



【図 9】



【図 10】



フロントページの続き

(51)Int.Cl.	F I		
	G 0 6 F	3/12	3 0 3
	G 0 6 F	3/12	3 2 9
	G 0 6 F	3/12	3 5 9
	G 0 6 F	3/12	3 8 5

(56)参考文献 特開平 1 1 - 0 5 3 1 3 0 (J P , A)
特開 2 0 0 7 - 0 7 2 7 1 1 (J P , A)
特開 2 0 0 8 - 1 0 2 8 3 8 (J P , A)
特開 2 0 0 0 - 3 2 2 2 2 7 (J P , A)
特開 2 0 0 7 - 1 4 8 6 7 9 (J P , A)
特開 2 0 1 7 - 0 1 6 4 8 0 (J P , A)

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