

(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2006-85529
(P2006-85529A)

(43) 公開日 平成18年3月30日(2006.3.30)

(51) Int.Cl.
G06F 9/445 (2006.01)

F I
G O 6 F 9/06 6 5 O C

テーマコード (参考)
5 B O 7 6
5 B 1 7 6

審査請求 未請求 請求項の数 11 O L (全 14 頁)

(21) 出願番号	特願2004-271008 (P2004-271008)	(71) 出願人	000001007
(22) 出願日	平成16年9月17日 (2004. 9. 17)		キヤノン株式会社
			東京都大田区下丸子3丁目30番2号
		(74) 代理人	100076428
			弁理士 大塚 康德
		(74) 代理人	100112508
			弁理士 高柳 司郎
		(74) 代理人	100115071
			弁理士 大塚 康弘
		(74) 代理人	100116894
			弁理士 木村 秀二
		(72) 発明者	望月 泰志
			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)	5B076 AB17
			5B176 AB17

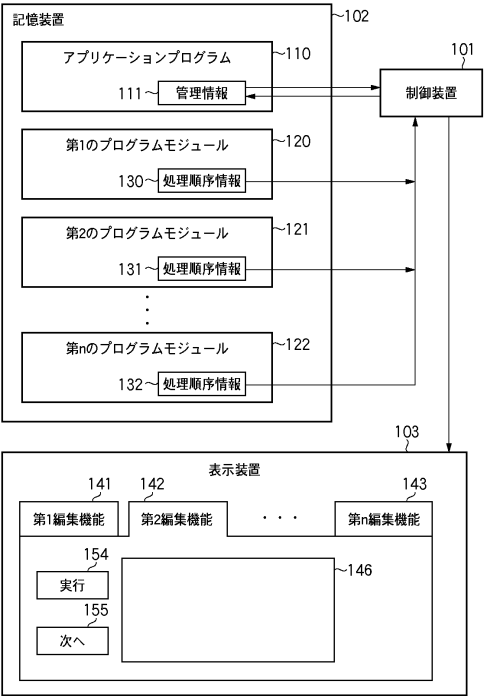
(54) 【発明の名称】 アプリケーションプログラム、情報処理装置および情報処理方法

(57) 【要約】 (修正有)

【課題】複数のプログラムモジュールを利用するアプリケーションプログラムにおいて、プログラムモジュールを追加したり、削除したりする場合に、アプリケーションプログラム本体を改変することなく、好適な順序で複数のプログラムモジュールを実行する。

【解決手段】アプリケーションプログラム110が起動されると、制御装置101は、記憶装置102に記憶されているプログラムモジュール120～122のそれぞれについて、プログラムモジュールの処理順序に関する処理順序情報130～132を読み出す。制御装置101は、読み出された処理順序情報に従って、複数のプログラムモジュールの処理順序を決定する。

【選択図】図1



【特許請求の範囲】

【請求項 1】

記憶装置および制御装置を有する情報処理装置において実行され、複数のデータ編集機能のそれぞれに対応する複数のプログラムモジュールを利用するアプリケーションプログラムであって、該アプリケーションプログラムが起動されると、前記情報処理装置に、

前記記憶装置に記憶されている前記プログラムモジュールのそれぞれについて、該プログラムモジュールの処理順序に関する処理順序情報を前記制御装置によって前記記憶装置から読み出すステップと、

複数の前記プログラムモジュールのそれぞれについて読み出された前記処理順序情報に従って、複数の該プログラムモジュールの処理順序を決定するステップと
を実行させるアプリケーションプログラム。

10

【請求項 2】

前記処理順序に従って、前記プログラムモジュールのそれぞれに対応する前記データ編集機能の各操作インタフェースが配置されたユーザインタフェースを表示するステップをさらに含む請求項 1 に記載のアプリケーションプログラム。

【請求項 3】

前記表示のステップには、

前記プログラムモジュールのそれぞれについて読み出された前記処理順序情報を、前記記憶装置に記憶される前記アプリケーションプログラムの管理情報へと、前記制御装置によって登録するステップと、

20

前記管理情報に登録された前記処理順序情報を読み出し、読み出された該処理順序情報に従って複数の前記データ編集機能の各操作インタフェースが配置された前記ユーザインタフェースを作成するステップと
が含まれる、請求項 2 に記載のアプリケーションプログラム。

【請求項 4】

前記アプリケーションプログラムの管理情報は、該アプリケーションプログラム内の内部テーブル、OS のレジストリデータ、または該アプリケーションプログラムから分離された設定ファイルの何れかに含まれる、請求項 3 に記載のアプリケーションプログラム。

【請求項 5】

編集対象となるデータに対して、決定された前記処理順序に従って、複数の前記データ編集機能のそれぞれに対応するデータ編集処理を前記制御装置によって実行するステップをさらに含む請求項 1 に記載のアプリケーションプログラム。

30

【請求項 6】

前記アプリケーションプログラムに、他のデータ編集機能に対応する他のプログラムモジュールを追加する際に、追加される該プログラムモジュールが、複数ある前記プログラムモジュールのうち第 1 のプログラムモジュールと第 2 のプログラムモジュールとの間の順番で実行されるべきときは、該第 1 のプログラムモジュールの次であって該第 2 のプログラムモジュールの前となる実行順番を表す値が、追加される該プログラムモジュールの処理順序情報に設定される、請求項 1 に記載のアプリケーションプログラム。

【請求項 7】

40

前記データ編集機能は、画像データの補正機能である、請求項 1 に記載のアプリケーションプログラム。

【請求項 8】

前記プログラムモジュールの処理順序情報は、該プログラムモジュール内のリソースデータ、OS のレジストリデータ、または前記プログラムモジュールから分離された設定ファイルの何れかに含まれる、請求項 1 に記載のアプリケーションプログラム。

【請求項 9】

請求項 1 ないし請求項 8 の何れかに記載のアプリケーションプログラムを記憶したコンピュータ可読記憶媒体。

【請求項 10】

50

複数のデータ編集機能のそれぞれに対応する複数のプログラムモジュールを利用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であって、

前記アプリケーションプログラムと前記複数のプログラムモジュールとを記憶する記憶手段と、

前記アプリケーションプログラムが起動されると、前記記憶手段に記憶されている前記プログラムモジュールのそれぞれについて、該プログラムモジュールの処理順序に関する処理順序情報を読み出す読み出し手段と、

複数の前記プログラムモジュールのそれぞれについて読み出された前記処理順序情報に従って、複数の該プログラムモジュールの処理順序を決定する決定手段とを含む情報処理装置。

10

【請求項 11】

複数のデータ編集機能のそれぞれに対応する複数のプログラムモジュールを利用するアプリケーションプログラムを実行する情報処理方法であって、

前記アプリケーションプログラムと前記複数のプログラムモジュールとを記憶装置に記憶するステップと、

前記アプリケーションプログラムが起動されると、前記記憶装置に記憶されている前記プログラムモジュールのそれぞれについて、該プログラムモジュールの処理順序に関する処理順序情報を読み出すステップと、

複数の前記プログラムモジュールのそれぞれについて読み出された前記処理順序情報に従って、複数の該プログラムモジュールの処理順序を決定するステップとを含む情報処理方法。

20

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、好適な処理順序が概ね決まっている複数のプログラムモジュールを利用するアプリケーションプログラムに対して、好適に他のプログラムモジュールを追加する技術に関する。

【背景技術】

【0002】

データを編集するための複数のプログラムモジュールを利用するアプリケーションプログラムにおいて、各プログラムモジュールの処理順序については、好ましい順番が決まっている場合がある。とりわけ、画像処理の分野では、画像の特性を補正する複数の機能の処理順序には、重要な意味が存在する。

30

【0003】

たとえば、画像データに予め格納されている撮影時の情報を利用した補正処理を「補正処理 1」とし、任意に明るさを補正する補正処理を「補正処理 2」とする。また、撮影時「フラッシュなし」のデータについては、自動的に補正処理 1 を適用して、明るさをより明るくする補正が実行されるように設定されているものとする。

【0004】

この前提において、「フラッシュなし」で撮影された画像データについて、補正処理 2 によってユーザの好みに応じて、明るさがより暗くなるように画像データを補正した後で、自動的に補正処理 1 を実行してしまうと、折角、明るさを暗めに補正したにもかかわらず、再び明るさが元に戻されてしまうといった不都合が生じる。

40

【0005】

この場合は、撮影情報に従って補正処理 1 を実行した後で、ユーザの好みに応じて補正処理 2 を実行すれば、上述の問題を生じない。このように、各補正処理については予め好ましい処理順序が存在するのである。

【0006】

従来、このような補正順序や各補正処理の機能は、アプリケーションプログラム内でのみ規定されていた。そのため、新規の補正機能をアプリケーションプログラムに追加しよ

50

うとすると、その補正処理の順序を考慮した上で、アプリケーションプログラムを改変し
なければならなかった。

【 0 0 0 7 】

ところで、補正機能に関するモジュールをプラグイン形式で追加したり削除したりする
アプリケーションプログラムが提案されている（特許文献１）。

【特許文献１】特開２００３－１２５２２２号公報。

【発明の開示】

【発明が解決しようとする課題】

【 0 0 0 8 】

特許文献１に記載の発明によって、新規の補正機能を容易に追加できるようになるが、
依然として、その補正処理の順序を考慮した上で、アプリケーションプログラムを改変し
なければならないといった課題が存在する。 10

【 0 0 0 9 】

そこで、本発明は、このような課題および他の課題の少なくとも１つを解決することを
目的とする。なお、他の課題については明細書の全体を通して理解できよう。

【課題を解決するための手段】

【 0 0 1 0 】

上記課題を解決すべく、本発明は、記憶装置および制御装置を有する情報処理装置にお
いて実行され、複数のデータ編集機能のそれぞれに対応する複数のプログラムモジュール
を利用するアプリケーションプログラムであって、該アプリケーションプログラムが起動
されると、前記情報処理装置に、前記記憶装置に記憶されている前記プログラムモジュール
のそれぞれについて、該プログラムモジュールの処理順序に関する処理順序情報を前記
制御装置によって前記記憶装置から読み出すステップと、複数の前記プログラムモジュール
のそれぞれについて読み出された前記処理順序情報に従って、複数の該プログラムモジ
ュールの処理順序を決定するステップとを実行させるアプリケーションプログラムを提供
する。 20

【発明の効果】

【 0 0 1 1 】

本発明によれば、各プログラムモジュールの処理順序を処理順序情報によって管理する
ようにするので、とくに、新規のプログラムモジュールが追加されるときは、当該モジ
ュールの処理順序情報を好適に設定することで、アプリケーションプログラムを改変するこ
となく、好適な順序で複数のプログラムモジュールを実行することが可能になる。 30

【発明を実施するための最良の形態】

【 0 0 1 2 】

以下に本発明の上位概念、中位概念および下位概念の理解に役立つ一実施形態を示す。
なお、以下の実施形態に含まれる概念について、そのすべてが特許請求の範囲に記載され
ているとは限らない。ただし、これは特許発明の技術的範囲から意識的に除外したのでは
なく、特許発明と均等の関係にあるため特許請求の範囲には記載していない場合があるこ
とを理解していただきたい。

【 0 0 1 3 】

40

〔第１の実施形態〕

図１は、実施形態に係る情報処理装置の例示的なブロック図である。図において、制御
装置１０１は、アプリケーションプログラム１１０など従って各機能ブロックを制御する
ＣＰＵなどの制御手段である。記憶装置１０２は、ＲＡＭやハードディスクドライブなど
から構成される記憶手段である。記憶装置１０２は、複数のデータ編集機能のそれぞれに
対応する複数のプログラムモジュール１２０、１２１および１２２や、これらのプログラ
ムモジュールを利用するアプリケーションプログラム１１０が記憶されている。

【 0 0 1 4 】

記憶装置１０２に記憶されているプログラムモジュール１２０～１２２のそれぞれにつ
いては、当該プログラムモジュールの処理順序に関する処理順序情報１３０、１３１、お
 50

よび 1 3 2 が記憶されている。処理順序情報 1 3 0 ~ 1 3 1 は、プログラムモジュール内のリソースデータ、OS（オペレーティングシステム）のレジストリデータ、またはプログラムモジュールから分離された設定ファイルなど、何らかの保存手法によって記憶装置 1 0 2 に記憶されている。

【 0 0 1 5 】

管理情報 1 1 1 は、アプリケーションプログラム 1 1 0 の内部テーブル、OS のレジストリデータ、またはプログラムモジュールから分離された設定ファイルなど、何らかの保存手法によって記憶装置 1 0 2 に記憶されている。この管理情報 1 1 0 には、プログラムモジュール 1 2 0 ~ 1 2 2 のそれぞれについて読み出された処理順序情報 1 3 0 ~ 1 3 2 が登録されている。とりわけ本実施形態では、アプリケーションプログラム 1 1 0 が起動されるたびに制御装置 1 0 1 によって管理情報 1 1 1 の内容が更新されることが好ましい。

10

【 0 0 1 6 】

制御装置 1 0 1 は、複数のプログラムモジュール 1 2 0 ~ 1 2 2 のそれぞれについて読み出された処理順序情報 1 3 0 ~ 1 3 2 に従って、各プログラムモジュールの処理順序を決定する。管理情報 1 1 1 には、処理順序情報 1 3 0 ~ 1 3 2 がそのまま登録されてもよいし、決定された処理順序が登録されてもよい。

【 0 0 1 7 】

表示装置 1 0 3 は、複数の前記データ編集機能の各操作インタフェースが配置されたユーザインタフェースを表示する表示手段である。この例のユーザインタフェースによれば、第 1 のプログラムモジュール 1 2 0 に対応する第 1 の編集機能についての操作インタフェース 1 4 1 が示されている。同様に、第 2 のプログラムモジュール 1 2 1 に対応する操作インタフェース 1 4 2、第 n のプログラムモジュール 1 2 2 に対応する操作インタフェース 1 4 3 がそれぞれ示されている（n は 2 以上の自然数）。実行ボタン 1 5 4 は、各操作インタフェース上に設けられており、これが操作されると対応する編集処理が、編集対象データに対して実行される。次へボタン 1 5 5 は、第 n 編集機能以外の操作インタフェースに設けられており、これが操作されると、次の処理順序の操作インタフェース（すなわち、ユーザインタフェース上で次に配置されている操作インタフェース）がアクティブとなる。表示ボックス 1 4 6 には、編集対象のデータが表示される。

20

【 0 0 1 8 】

図 2 は、実施形態に係る情報処理方法の一例を示すフローチャートである。このフローチャートは上述の情報処理装置により実行されるアプリケーションプログラム 1 1 0 の処理を示している。

30

【 0 0 1 9 】

アプリケーションプログラム 1 1 0 が起動されると、ステップ S 2 0 1 において、制御装置 1 0 1 は、記憶装置 1 0 2 に記憶されているプログラムモジュール 1 2 0 ~ 1 2 2 のそれぞれについて、当該プログラムモジュールの処理順序に関する処理順序情報 1 3 0 ~ 1 3 2 を記憶装置 1 0 2 から読み出す。

【 0 0 2 0 】

ステップ S 2 0 2 において、制御装置 1 0 1 は、複数のプログラムモジュール 1 2 0 ~ 1 2 2 のそれぞれについて読み出された処理順序情報 1 3 0 ~ 1 3 2 に従って、各プログラムモジュールの処理順序を決定する。そして、プログラムモジュール 1 2 0 ~ 1 2 2 のそれぞれについて読み出された処理順序情報 1 3 0 ~ 1 3 2 に従って、複数のデータ編集機能の各操作インタフェース 1 4 1 ~ 1 4 3 が配置されたユーザインタフェースを表示装置 1 0 3 に表示する。

40

【 0 0 2 1 】

なお、ステップ S 2 0 2 には、たとえば、次のステップに細分化することができる。ステップ S 2 1 0 において、制御装置 1 0 1 は、プログラムモジュール 1 2 0 ~ 1 2 2 のそれぞれについて読み出された処理順序情報 1 3 0 ~ 1 3 2 を、記憶装置 1 0 2 に記憶されているアプリケーションプログラム 1 1 0 の管理情報 1 1 1 へと登録する。この際に、制

50

御装置 101 は、各プログラムモジュール 120 ~ 122 のそれぞれの処理順序情報 130 ~ 132 に従って、各プログラムモジュール 120 ~ 122 の処理順序を決定し、決定された処理順序を表す情報を管理情報 111 に登録してもよい。

【0022】

ステップ S211 において、制御装置 101 は、管理情報 111 に登録された処理順序情報を読み出し、読み出された処理順序情報に従って複数のデータ編集機能の各操作インタフェース 141 ~ 143 を配置し、ユーザインタフェースを作成する。

【0023】

そして、ステップ S203 において、制御装置 101 は、編集対象となるデータに対して、ユーザインタフェース上の配置順序または処理順序情報に規定された処理順序に従って、複数のデータ編集機能のそれぞれに対応するデータ編集処理を実行する。 10

【0024】

本実施形態によれば、各プログラムモジュール 120 ~ 122 の処理順序を処理順序情報 130 ~ 132 によって管理し、この処理順序情報に従ってユーザインタフェースを表示するようにしたので、とくに、新規のプログラムモジュールが追加されるときは、当該モジュールの処理順序情報を好適に設定することで、アプリケーションプログラムを改変することなく、好適な順序で複数のプログラムモジュールを実行することが可能になる。

【0025】

図 3 は、実施形態において、他のプログラムモジュールが追加されたときの動作を説明する図である。この例では、アプリケーションプログラム 110 に、他のデータ編集機能 20 に対応する他の第 3 のプログラムモジュール 123 を追加することを考えてみる。また、追加される第 3 のプログラムモジュール 123 が、第 1 のプログラムモジュール 120 と第 2 のプログラムモジュール 121 との間の順番で実行されるべきものとする。この場合、第 1 のプログラムモジュール 120 の次であって、第 2 のプログラムモジュール 121 の前となる実行順番を表す値が、追加される第 3 のプログラムモジュールの処理順序情報 133 に設定される。

【0026】

制御装置 101 は、アプリケーションプログラム 110 が起動されると、第 3 のプログラムモジュール 123 の処理順序情報 133 を読み出し (S201)、処理順序情報 130 ~ 133 に従って、各プログラムモジュールの処理順序を決定し、続いて、第 3 のプログラムモジュール 123 の操作インタフェース 144 を、第 1 の操作インタフェース 141 と、第 2 の操作インタフェース 142 との間に配置する (S202)。また、編集処理の順番も管理情報 111 に反映されるので、第 1 のプログラムモジュール 120 に対応する第 1 編集機能 141 が実行された後に、第 3 のプログラムモジュール 120 に対応する第 3 編集機能 144 が実行される (S203)。 30

【0027】

このように、本実施形態では、アプリケーションプログラムを改変することなく、好適な順序で複数のプログラムモジュールを実行することが可能になる。

【0028】

また、処理順序に従って、各プログラムモジュールに対応する操作インタフェースをユーザインタフェース上で再配置するようにしたので、プログラムモジュールを追加したり削除したりした結果が適確にユーザインタフェースに反映される。そのため、ユーザは、視覚的にどの編集機能を使用できるかを把握できる。しかも操作インタフェースは処理順序に従って並んでいるので、ユーザは、視覚的に処理順序を認識しやすくなる。 40

【0029】

また、アプリケーションプログラムの管理情報は、アプリケーションプログラム内の内部テーブルだけでなく、OS のレジストリデータ、または該アプリケーションプログラムから分離された設定ファイルの何れかによっても管理できる。これにより、アプリケーションプログラムを改変することなく、プログラムモジュールを追加または削除でき、しかも複数のプログラムモジュールの実行順序を、アプリケーションプログラムを改変するこ 50

となく変更できる。

【0030】

編集対象となるデータに対して、決定された処理順序に従って、複数のデータ編集機能のそれぞれに対応するデータ編集処理を実行することで、ユーザが不適切な順序でデータ編集処理を実行してしまう恐れを低減できる。

【0031】

プログラムモジュールの処理順序情報は、プログラムモジュール内のリソースデータ、OSのレジストリデータ、またはプログラムモジュールから分離された設定ファイルの何れかに含まれるようにしたので、処理順序情報を開発者等が設定しやすくなるといったメリットがある。

10

【0032】

[第2の実施形態]

本実施形態では、アプリケーションプログラムの一例として画像補正プログラムに本発明を適用する例を説明する。

【0033】

図4は、実施形態に係る画像形成システムの一例を示す図である。本実施形態における画像形成システムでは、印刷データ（たとえば、制御コマンドなど）を生成する情報処理装置400と、その印刷データに基づいて画像を形成するプリンタ470とが含まれる。プリンタ470は、画像形成装置の一例であり、印刷機能を内蔵した複合機などであってもよい。また、情報処理装置400は、画像データの補正を行なうためのアプリケーションプログラム411がインストールされており、画像処理装置として機能する。

20

【0034】

情報処理装置400は、メモリ410、CPU430、ハードディスク・コントローラ(HDC)420、ハードディスク(HD)421、フロッピー(登録商標)ディスク・コントローラ(FDC)425、フロッピー(登録商標)ディスクドライブ(FD)426、プリンタ・コントローラ(PRTC)440、キーボード・コントローラ(KBC)450、キーボード(KB)451、CRTコントローラ(CRTC)460、CRT461などを備えている。

【0035】

メモリ410は、たとえば、HD421からロードされた画像補正処理を司るアプリケーションプログラム411、プリンタ470に対応する印刷データを生成するためのソフトウェアであるプリンタドライバ412、プリンタ470に供給すべき印刷データをスプールするスプーラ領域413、その他不図示のOS(オペレーティングシステム)やワーク領域等を有する。

30

【0036】

CPU430は、メモリ410内のアプリケーションプログラム411、プリンタドライバ412、OS等に基づいて動作する。なお、CPU430は、電源が投入されると、不図示のROMに記憶されたブートプログラムにより起動し、HD421からメモリ410にOSをロードした後、アプリケーションプログラム411も同様にメモリ411にロードする。これによって、情報処理装置400およびプリンタ470は、画像形成システムとして機能する。また、当然のことながら、CPU430は、HDC420を介してHD421にアクセスし、データを読み出したり、書き込んだりすることができる。

40

【0037】

PRTC440は、スプーラ領域413に蓄積された印刷データを順次プリンタ470に送信する。KBC450は、KB451を制御し、ユーザから指示されたデータを装置内に取り込む。CRTC460は、表示装置として機能するCRT461を制御するコントローラである。これらのKBC450、KB451、CRTC460、およびCRT461等は、CPU430の制御に応じてユーザインタフェースを構成する。なお、情報処理装置400は、ポインティング・デバイス等の他の機能ブロックを備えてもよい。

【0038】

50

また、プリンタ 470 には、情報処理装置 400 から印刷データを受信したり、各種ステータスを情報処理装置 400 に送信したりするためのインタフェース 471 と、主として受信した印刷データを解釈し、ビットマップイメージデータを生成するプリンタ制御部 472 と、プリンタ制御部 472 から出力されてきたビットマップイメージデータを受信し、実際に画像を形成するプリンタエンジン 473 が含まれている。なお、図示してはいないが操作パネル等が含まれていてもよい。

【0039】

図 5 は、実施形態に係る画像補正モジュールのリソースデータの構成例を示す図である。本実施形態では、固定のデータであるリソースデータ 500 が各画像補正用の DLL (ダイナミック・リンク・ライブラリ) モジュール内に含まれている。リソースデータ 500 には、当該 DLL モジュールが画像補正モジュールであることを示すための画像補正モジュール識別フラグ 501 と、アプリケーションプログラム 411 が画像補正モジュールを特定するための画像補正 ID 502 と、アプリケーションプログラム 411 がユーザインタフェースのタブ上に表示するためのタブ名称 503 と、補正処理の相対的な順序を示す処理順序情報 504 と、その他の情報を格納するその他情報 505 とを含んでいる。

10

【0040】

図 6 は、実施形態に係るアプリケーションプログラムの内部に作成される内部テーブルの構成例を示す図である。図に示すように、内部テーブル 600 には、3 つの画像補正モジュールに関する制御情報 610 ~ 612 が登録されている。各制御情報には、補正処理の実行順序を示す処理順序情報 601、画像補正モジュールを識別するための画像補正 ID 602、ユーザインタフェースのタブ上に表示されるタブ名称 603、およびその他の情報を示すその他情報 604 から構成される。このように、内部テーブル 600 では、アプリケーションプログラム 411 によって利用される複数の画像補正モジュールから読み出されたリソースデータ 500 の内容が制御情報として反映される。

20

【0041】

図 7 は、実施形態に係る画像補正用のアプリケーションプログラムに関する例示的なフローチャートである。なお、このフローチャートに関する処理は、アプリケーションプログラム 411 に従って、CPU 430 により実行されるものとする。

【0042】

画像補正用のアプリケーションプログラム 411 を起動すると、ステップ S701 において、CPU 430 は、アプリケーションプログラム 411 のインストール先フォルダから DLL モジュールを読み出す。インストール先フォルダが、HD 421 に記憶されていることはいうまでもない。

30

【0043】

ステップ S702 において、CPU 430 は、読み出された DLL モジュールからリソースデータ 500 を読み出す。

【0044】

ステップ S703 において、CPU 430 は、読み出されたリソースデータ 500 に画像補正モジュール識別フラグ 501 が存在するかどうかを判定する。存在しなければステップ S706 に進む。一方、存在した場合は、ステップ S704 に進む。

40

【0045】

ステップ S704 において、CPU 430 は、リソースデータ 500 内の処理順序情報に従って画像補正モジュールの処理順序を決定する。さらに、当該画像補正モジュールに対応するタブの、ユーザインタフェース上における表示位置を決定する。たとえば、画像補正モジュール A の処理順序情報に数値「100」が設定されており、画像補正モジュール B の処理順序情報に数値「200」が設定されており、かつ、画像補正モジュール C の処理順序情報に数値「300」が設定されていたとする。この場合 CPU 430 は、各処理順序情報の大小を判定し、画像補正モジュール A の配置順序を 1 番目とし、画像補正モジュール B の配置順序を 2 番目とし、画像補正モジュール C の配置順序を 3 番目と決定する。

50

【 0 0 4 6 】

ステップ S 7 0 5 において、C P U 4 3 0 は、決定された表示位置や処理順序に関する情報を保持するために内部テーブル 6 0 0 に登録する。なお、C P U 4 3 0 は、単純に、リソースデータ 5 0 0 の内容を内部テーブル 6 0 0 に登録するだけで、ステップ S 7 0 4 を省略してもよい。リソースデータ 5 0 0 の処理順序情報には予め相対的な処理順序の値が設定されているからである。

【 0 0 4 7 】

ステップ S 7 0 6 において、C P U 4 3 0 は、インストール先フォルダに次の D L L モジュールが存在するかどうかを判定する。判定の結果、次の D L L モジュールが存在した場合は、ステップ S 7 0 1 に戻り、上述の処理を繰り返す。一方、次の D L L モジュールが存在しなければ、ステップ S 7 0 7 に進む。

【 0 0 4 8 】

ステップ S 7 0 7 において、C P U 4 3 0 は、内部テーブル 6 0 0 を参照しながら、処理順序情報に従って各画像補正モジュール用の補正タブをユーザインタフェース上に配置し、補正タブの名称として、内部テーブルのタブ名称 6 0 3 を読み出して補正タブに設定するなどして、ユーザインタフェースを作成する。

【 0 0 4 9 】

ステップ S 7 0 8 において、C P U 4 3 0 は、C P T C 4 6 0 を通じて、作成したユーザインタフェースを C R T 4 6 1 に表示する。

【 0 0 5 0 】

ステップ S 7 0 9 において、C P U 4 3 0 は、画像補正の対象となる画像データを読み込む。すなわち、H D 4 2 1 に記憶されている画像データを指定して、メモリ 4 1 0 にロードする。

【 0 0 5 1 】

ステップ S 7 1 0 において、C P U 4 3 0 は、補正タブの配置順序（リソースデータ 5 0 0 の処理順序情報 5 0 4、または内部テーブル 6 0 0 内の処理順序情報 6 0 1）に従って、順番に画像補正処理を、画像データに適用してゆく。

【 0 0 5 2 】

本実施形態によれば、画像補正用のアプリケーションプログラムにおいて、補正機能ごと個別の画像補正モジュールを作成し、かつ、各画像補正モジュール内に補正処理の処理順序情報を備えるようにする。これにより、各画像補正モジュールの処理順序情報を適宜設定するだけで、アプリケーションプログラム本体を変更することなく、補正機能を追加したり削除したりすることが容易に可能となる。

【 0 0 5 3 】

図 8 は、実施形態に係る画像補正モジュールの具体例を示す図である。この例では、画像補正機能 A を実現する画像補正モジュールである「補正 A . D L L」と、画像補正機能 B を実現する画像補正モジュールである「補正 B . D L L」と、画像補正機能 C を実現する画像補正モジュールである「補正 C . D L L」とが H D 4 2 1 にインストールされている。また、それぞれ、処理順序情報は、1 0 0、2 0 0、3 0 0 に設定されている。

【 0 0 5 4 】

図 9 は、実施形態に係る画像補正アプリケーションプログラムのユーザインタフェースの一例を示す図である。図 9 は、とりわけ図 8 の環境を反映させたものである。すなわち、アプリケーションプログラム 4 1 1 起動されると、上述のフローチャートに従って、アプリケーション用のウィンドウ 9 0 0 が作成され、表示される。

【 0 0 5 5 】

たとえば、処理順序情報によれば、画像補正機能 A（補正 A . D L L）が 1 番目に実行されるべきであるから、画像補正機能 A 用のタグ 9 4 1 が最も左に配置される。また、処理順序情報によれば、画像補正機能 B（補正 B . D L L）が 2 番目に実行されるべきであるから、画像補正機能 B 用のタグ 9 4 2 が左から 2 番目に配置される。画像補正機能 C（補正 C . D L L）が 3 番目に実行されるべきであるから、画像補正機能 C 用のタグ 9 4 3

10

20

30

40

50

が左から3番目に配置される。

【0056】

図9の例では、表示ボックス946に表示されている処理対象の画像データに対してすでに画像補正機能Aが実行されており、画像補正機能Bの実行順番となっている。そのため画像補正機能B用のタグ942がアクティブになっている。ここで補正実行ボタン954が操作されると、画像補正機能Bが画像データに適用される。次へボタン955が操作されると画像補正機能Cのタグ943がアクティブとなる。

【0057】

図10は、実施形態に係る画像補正モジュールの他の具体例を示す図である。この例では、新規の画像補正機能として、画像補正モジュールD(補正D・DLL)と、画像補正モジュールE(補正E・DLL)とがインストール先フォルダに追加されている。なお、開発者は、画像補正モジュールDを画像補正モジュールAとBとの間に追加し、画像補正モジュールEを、画像補正モジュールBとCとの間に追加したいと希望しているものとする。この場合は、画像補正モジュールDについては、画像補正モジュールAの処理順序情報「100」と、画像補正モジュールBの処理順序情報「200」との間にくるような処理順序情報を画像補正モジュールDに設定すればよいので、たとえば「150」などを設定する。画像補正モジュールEについては、画像補正モジュールBの処理順序情報「200」と、画像補正モジュールCの処理順序情報「300」との間にくるような処理順序情報を画像補正モジュールEに設定すればよいので、たとえば「250」などを設定する。

10

【0058】

この状態で、アプリケーションプログラム411が起動されると、画像補正モジュールAないしEについてリソースデータが読み出され(S701、S702)、その内容が内部テーブルに反映される(S703~S706)。そして、新しい内部テーブルに登録されている各画像補正モジュールの処理順序情報に従って、それぞれのタブが順番に配置され、ユーザインタフェースが作成されて(S707)、表示される(S708)。

20

【0059】

図11は、実施形態に係る画像補正アプリケーションプログラムの他のユーザインタフェースの一例を示す図である。ユーザインタフェース1100には、処理順序情報に従って、各画像補正モジュールのタグ1141ないし1145が配置される。すなわち、画像補正機能A(補正A・DLL)が1番目に実行されるべきであるから、画像補正機能A用のタグ1141が最も左に配置される。また、処理順序情報によれば、画像補正機能D(補正D・DLL)が2番目に実行されるべきであるから、画像補正機能D用のタグ1144が左から2番目に配置される。また、処理順序情報によれば、画像補正機能B(補正B・DLL)が3番目に実行されるべきであるから、画像補正機能B用のタグ1142が左から3番目に配置される。また、処理順序情報によれば、画像補正機能E(補正E・DLL)が4番目に実行されるべきであるから、画像補正機能E用のタグ1145が左から4番目に配置される。そして、画像補正機能C(補正C・DLL)が5番目に実行されるべきであるから、画像補正機能C用のタグ1143が左から5番目に配置される。

30

【0060】

以上説明したように、本実施形態によれば、アプリケーションプログラムを改変することなく、画像補正モジュールを容易に追加できる。しかも、複数の画像補正モジュールが利用される場合に、その処理順序に意味がある場合は、各画像補正モジュールの処理順序情報を適宜設定することで、アプリケーションプログラムを改変することなく、ユーザインタフェースを動的に変更することができる。また、各画像補正モジュールの処理順序についても、アプリケーションプログラムを改変することなく、処理順序情報によって好適に制御できるようになる。

40

【0061】

また、必ずしも補正モジュールの動作順を厳密に規定させ無くても良い。つまり使用者は、処理順序情報に従った順序で処理をしなくても可能に設計しておいても構わない。しかし、各種の画像処理の経験が高い人のお薦めの処理順序や、プログラム設計上のお薦め

50

の処理順序を、使用者に明示出来るので、使用者にとっても、詳しい知識を必要とせず、好ましい処理順序で処理が可能になる。

【 0 0 6 2 】

[他の実施形態]

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。すなわち、単体に P C にアプリケーションプログラムやプログラムモジュールをインストールしてもよいし、アプリケーションプログラムやプログラムモジュールをアプリケーションサーバに記憶しておき、クライアント P C にダウンロードして本発明を実現してもよい。

【 0 0 6 3 】

なお、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（本実施形態では図 2 または 7 に示すフローチャートに対応したプログラム）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、そのシステム若しくは装置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。

【 0 0 6 4 】

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、該コンピュータ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。

【 0 0 6 5 】

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、O S 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。

【 0 0 6 6 】

プログラムを供給するための記録媒体としては、たとえば、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、M O、C D - R O M、C D - R、C D - R W、磁気テープ、不揮発性のメモリカード、R O M、D V D（D V D - R O M、D V D - R）などがある。

【 0 0 6 7 】

その他、プログラムの供給方法としては、クライアントコンピュータのブラウザを用いてインターネットのホームページに接続し、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することによっても供給できる。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 W W W サーバも、本発明の構成要件となる場合がある。

【 0 0 6 8 】

また、本発明のプログラムを暗号化して C D - R O M 等の記憶媒体に格納してユーザに配布し、所定の条件をクリアしたユーザに対し、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、その鍵情報を使用することにより暗号化されたプログラムを実行してコンピュータにインストールさせて実現することも可能である。

【 0 0 6 9 】

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る他、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 O S などが、実際の処理の一部または全部を行い、その処理によっても前述した実施形態の機能が実現され得る。

【 0 0 7 0 】

さらに、記録媒体から読み出されたプログラムが、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き込まれた後、

10

20

30

40

50

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い、その処理によっても前述した実施形態の機能が実現される。

【図面の簡単な説明】

【0071】

【図1】図1は、実施形態に係る情報処理装置の例示的なブロック図である。

【図2】図2は、実施形態に係る情報処理方法の一例を示すフローチャートである。

【図3】図3は、実施形態において、他のプログラムモジュールが追加されたときの動作を説明する図である。

【図4】図4は、実施形態に係る画像形成システムの一例を示す図である。

10

【図5】図5は、実施形態に係る画像補正モジュールのリソースデータの構成例を示す図である。

【図6】図6は、実施形態に係るアプリケーションプログラムの内部に作成される内部テーブルの構成例を示す図である。

【図7】図7は、実施形態に係る画像補正用のアプリケーションプログラムに関する例示的なフローチャートである。

【図8】図8は、実施形態に係る画像補正モジュールの具体例を示す図である。

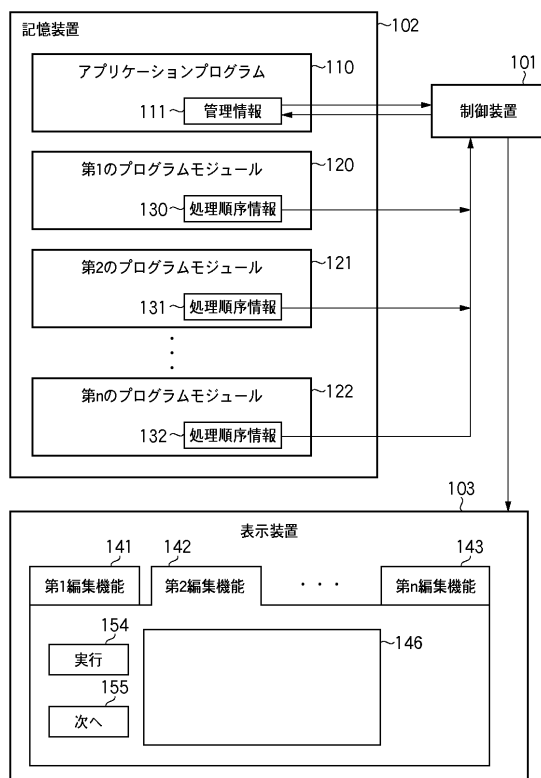
【図9】図9は、実施形態に係る画像補正アプリケーションプログラムのユーザインタフェースの一例を示す図である。

【図10】図10は、実施形態に係る画像補正モジュールの他の具体例を示す図である。

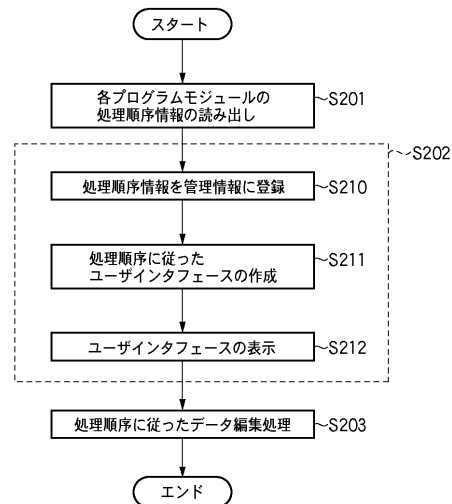
20

【図11】図11は、実施形態に係る画像補正アプリケーションプログラムの他のユーザインタフェースの一例を示す図である。

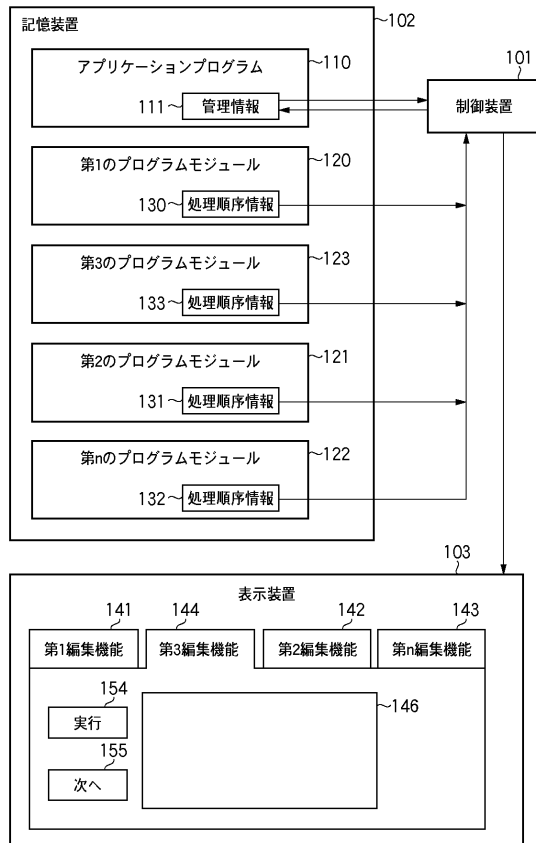
【図1】



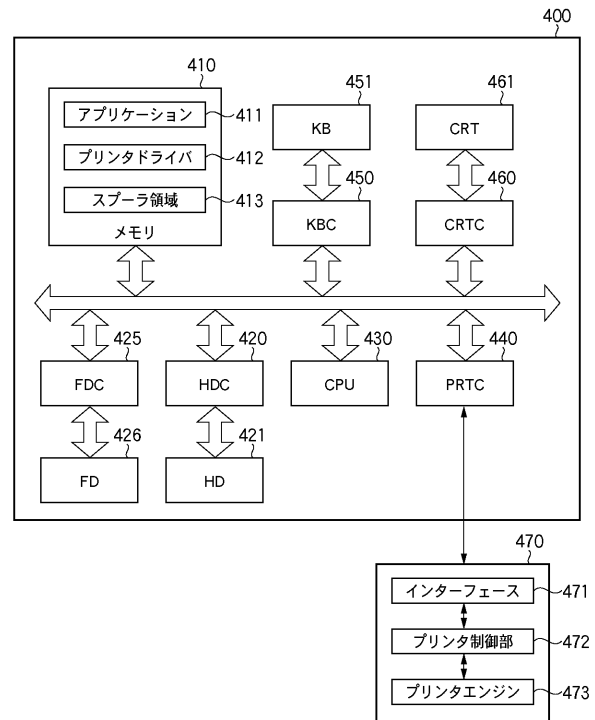
【図2】



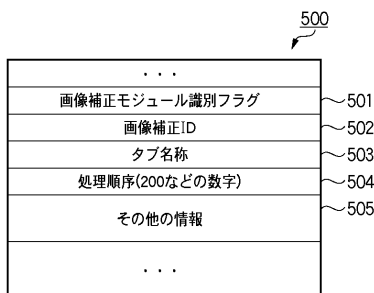
【図 3】



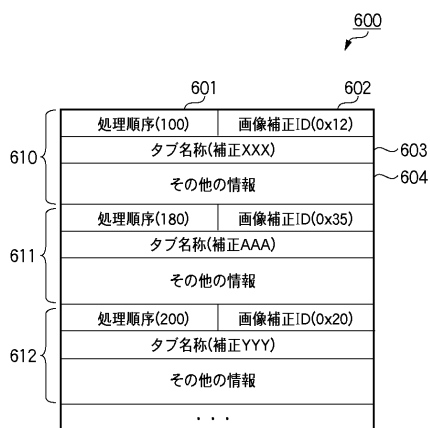
【図 4】



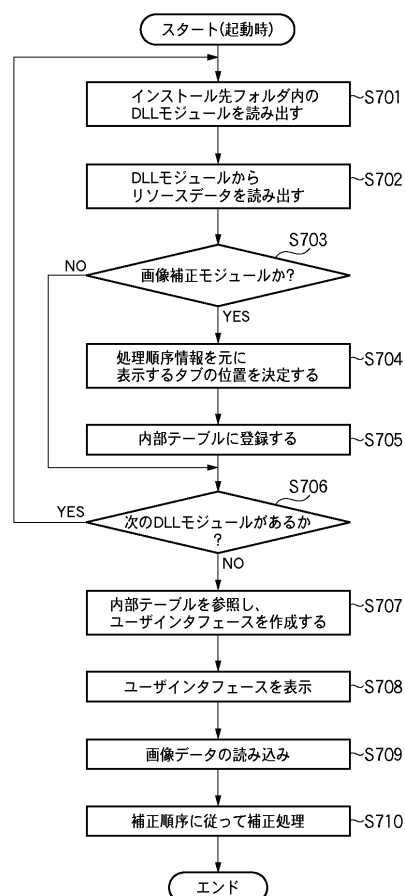
【図 5】



【図 6】



【図 7】



【図 8】

インストール先フォルダ

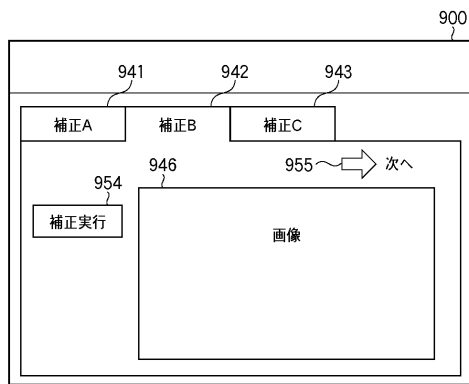


【図 10】

インストール先フォルダ



【図 9】



【図 11】

