

(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2018-22370  
(P2018-22370A)

(43) 公開日 平成30年2月8日(2018.2.8)

(51) Int.Cl.	F I	テーマコード (参考)
<b>G06F 9/44 (2018.01)</b>	G06F 9/06 610A	5B069
<b>G06F 3/0481 (2013.01)</b>	G06F 3/0481	5B376
<b>G06F 3/14 (2006.01)</b>	G06F 3/14 350A	5E555
	G06F 3/14 360A	

審査請求 未請求 請求項の数 8 O L (全 12 頁)

(21) 出願番号 特願2016-153652 (P2016-153652)  
(22) 出願日 平成28年8月4日 (2016.8.4)

(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 100125254  
弁理士 別役 重尚  
(72) 発明者 阿久澤 政男  
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) 5B069 AA01 AA20 CA01 CA15 DD19  
5B376 AA11 AA38 FA16 GA01  
5E555 AA21 AA79 BA02 BA27 BA52  
BB02 BB27 BC07 BC08 CB72  
DB02 DC05 DC63 DC84 EA11  
FA00

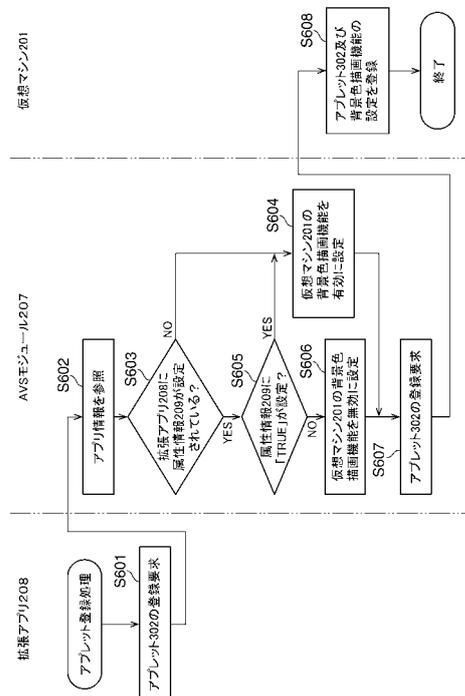
(54) 【発明の名称】 アプリケーション実行装置及びその制御方法、並びにプログラム

(57) 【要約】

【課題】 不完全な画像の表示を防止することができるアプリケーション実行装置を提供する。

【解決手段】 MF P 1 0 0 は、拡張アプリ 2 0 8 の仕様が背景色描画を実行する仕様である旨を示す「FALSE」が属性情報 2 0 9 に設定されていない場合、仮想マシン 2 0 1 による背景色描画処理を実行する旨を設定する。

【選択図】 図 6



**【特許請求の範囲】****【請求項 1】**

インストールされたアプリケーションを制御する仮想マシンを備えるアプリケーション実行装置であって、

前記アプリケーションによって画像を描画する際に当該画像の背景を前記アプリケーションによって描画するか否かを示す属性情報を参照する参照手段と、

前記属性情報に前記画像の背景を前記アプリケーションによって描画する旨が設定されていない場合、前記仮想マシンによって前記画像の背景を描画する旨を設定する設定手段とを備えることを特徴とするアプリケーション実行装置。

**【請求項 2】**

前記属性情報に前記画像の背景を前記アプリケーションによって描画する旨が設定されていない場合は、前記属性情報に前記画像の背景を前記アプリケーションによって描画しない旨が設定されている場合、又は前記アプリケーションに前記属性情報が設定されていない場合であることを特徴とする請求項 1 記載のアプリケーション実行装置。

**【請求項 3】**

前記設定手段は、前記属性情報に前記画像の背景を前記アプリケーションによって描画する旨が設定されている場合、前記仮想マシンによって前記画像の背景を描画しない旨を設定することを特徴とする請求項 1 又は 2 記載のアプリケーション実行装置。

**【請求項 4】**

前記画像を描画するための描画情報を登録する登録手段を更に備え、

前記登録手段は、前記描画情報を前記設定手段による設定に対応付けて登録することを特徴とする請求項 1 乃至 3 のいずれか 1 項に記載のアプリケーション実行装置。

**【請求項 5】**

前記属性情報をユーザに設定させる属性情報設定手段を更に備えることを特徴とする請求項 1 乃至 4 のいずれか 1 項に記載のアプリケーション実行装置。

**【請求項 6】**

インストールされたアプリケーションを制御する仮想マシンを備えるアプリケーション実行装置の制御方法であって、

前記アプリケーションによって画像を描画する際に当該画像の背景を前記アプリケーションによって描画するか否かを示す属性情報を参照する参照ステップと、

前記属性情報に前記画像の背景を前記アプリケーションによって描画する旨が設定されていない場合、前記仮想マシンによって前記画像の背景を描画する旨を設定する設定ステップとを有することを特徴とするアプリケーション実行装置の制御方法。

**【請求項 7】**

インストールされたアプリケーションを制御する仮想マシンを備えるアプリケーション実行装置の制御方法をコンピュータに実行させるプログラムであって、

前記アプリケーション実行装置の制御方法は、

前記アプリケーションによって画像を描画する際に当該画像の背景を前記アプリケーションによって描画するか否かを示す属性情報を参照する参照ステップと、

前記属性情報に前記画像の背景を前記アプリケーションによって描画する旨が設定されていない場合、前記仮想マシンによって前記画像の背景を描画する旨を設定する設定ステップとを有することを特徴とするプログラム。

**【請求項 8】**

アプリケーション実行装置にインストールされたアプリケーションを実行するプログラムであって、

前記プログラムは、前記アプリケーションによって画像を描画する際に当該画像の背景を前記アプリケーションによって描画するか否かを示す属性情報を含むことを特徴とするプログラム。

**【発明の詳細な説明】****【技術分野】**

10

20

30

40

50

## 【0001】

本発明は、アプリケーション実行装置及びその制御方法、並びにプログラムに関する。

## 【背景技術】

## 【0002】

Java（登録商標）VM（Virtual Machine）等の仮想マシンモジュール（以下、単に「仮想マシン」という。）が組み込まれたアプリケーション実行装置としてのMFPが知られている。MFPは、機能を拡張する複数の拡張アプリケーションモジュール（以下、単に「拡張アプリ」という。）をインストール可能であり、インストールされた各拡張アプリを仮想マシンが制御する。MFPでは、拡張アプリがインストールされると、例えば、該拡張アプリを示すアイコンがMFPの操作画面に表示される。ユーザによって該アイコンが選択されると、MFPは選択されたアイコンの拡張アプリのGUI（Graphical User Interface）の画像（以下、「拡張アプリGUI」という。）を表示する表示制御処理を実行する。表示制御処理では、拡張アプリによって予め設定された背景色で拡張アプリGUIの背景を描画する背景色描画を仮想マシンが行い、その後、背景以外の残りの拡張アプリGUIの描画処理を拡張アプリが行う。

10

## 【0003】

MFPにインストールされる拡張アプリには、拡張アプリGUIを表示する際に背景色描画を含む描画処理を実行する拡張アプリも存在する。このような拡張アプリの拡張アプリGUIを表示する場合、拡張アプリが背景を含む全ての拡張アプリGUIの描画処理を行うが、システムの仕様上仮想マシンも拡張アプリGUIの背景色描画を行う場合がある。この場合、拡張アプリGUIの背景色描画が複数回実行され、拡張アプリGUIを表示する際のちらつきが顕著になってしまう。これに対して、拡張アプリGUIの描画処理において、仮想マシンによる背景色描画を実行しない技術が提案されている（例えば、特許文献1参照）。

20

## 【先行技術文献】

## 【特許文献】

## 【0004】

【特許文献1】特開2006-171274号公報

## 【発明の概要】

## 【発明が解決しようとする課題】

30

## 【0005】

しかしながら、上述した特許文献1の技術では、拡張アプリGUIの背景色描画を仮想マシンに実行させる仕様の拡張アプリにおいて、背景が描画されない不完全な拡張アプリGUIが表示されてしまうという問題が生じる。

## 【0006】

本発明の目的は、不完全な画像の表示を防止することができるアプリケーション実行装置及びその制御方法、並びにプログラムを提供することにある。

## 【課題を解決するための手段】

## 【0007】

上記目的を達成するために、本発明のアプリケーション実行装置は、インストールされたアプリケーションを制御する仮想マシンを備えるアプリケーション実行装置であって、前記アプリケーションによって画像を描画する際に当該画像の背景を前記アプリケーションによって描画するか否かを示す属性情報を参照する参照手段と、前記属性情報に前記画像の背景を前記アプリケーションによって描画する旨が設定されていない場合、前記仮想マシンによって前記画像の背景を描画する旨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】

## 【0008】

本発明によれば、不完全な画像の表示を防止することができる。

## 【図面の簡単な説明】

50

【 0 0 0 9 】

【 図 1 】 本発明の実施の形態に係るアプリケーション実行装置としてのMFPの構成を概略的に示すブロック図である。

【 図 2 】 図 1 のMFPのソフトウェアモジュールの構成を概略的に示すブロック図である。

【 図 3 】 図 1 のMFPで登録されるアプレットを説明するための図である。

【 図 4 】 図 1 のMFPで実行される起動制御処理の手順を示すフローチャートである。

【 図 5 】 図 4 のステップS403のアプリ起動処理の手順を示すフローチャートである。

【 図 6 】 図 1 のMFPで実行されるアプレット登録処理の手順を示すフローチャートである。

10

【 図 7 】 図 1 のMFPにインストールされた拡張アプリのアプリ情報を管理するための管理データの一例を示す図である。

【 図 8 】 図 1 のMFPで実行される背景描画制御処理の手順を示すフローチャートである。

【 発明を実施するための形態 】

【 0 0 1 0 】

以下、本発明の実施の形態を図面を参照しながら詳述する。

【 0 0 1 1 】

本実施の形態では、アプリケーション実行装置としてのMFPに本発明を適用した場合について説明するが、本発明の適用先はMFPに限られず、拡張アプリGUIの描画処理を行う拡張アプリをインストール可能な装置であれば本発明を適用することができる。

20

【 0 0 1 2 】

図 1 は、本発明の実施の形態に係るアプリケーション実行装置としてのMFP100の構成を概略的に示すブロック図である。

【 0 0 1 3 】

図 1 において、MFP100は、制御部101、タッチパネル110（属性情報設定手段）、ディスプレイ111、及び外部メモリ112を備え、制御部101はタッチパネル110、ディスプレイ111、及び外部メモリ112のそれぞれと接続されている。制御部101は、CPU102、RAM103、ROM104、入力制御部105、表示制御部106、外部メモリI/F107、及び通信I/F108を備える。CPU102、RAM103、ROM104、入力制御部105、表示制御部106、外部メモリI/F107、及び通信I/F108はシステムバス113を介して互いに接続されている。

30

【 0 0 1 4 】

MFP100は機能を拡張する複数の拡張アプリをインストール可能である。制御部101はMFP100全体を統括的に制御する。CPU102はROM104に格納されたプログラムを実行して後述する図2のソフトウェアモジュール200の各処理を行う。RAM103は揮発性メモリである。RAM103はCPU102の作業領域として用いられ、また、RAM103は各データの一時格納領域として用いられる。ROM104は不揮発性メモリであり、CPU102が実行するプログラムや各データを格納する。入力制御部105は図示しないキーボード及びマウス等のユーザの操作を受け付ける入力デバイスや、タッチパネル110等とデータ通信を行う。表示制御部106はディスプレイ111における画像の表示を制御する。外部メモリI/F108は外部メモリ112から各データを読み出し、また、外部メモリI/F108は外部メモリ112に各データを書き込む。通信I/F108はネットワーク109等に接続された図示しない外部装置との通信を制御する。

40

【 0 0 1 5 】

タッチパネル110は、例えば、抵抗膜方式、静電容量方式、表面弾性波方式、赤外線方式、電磁誘導方式、画像認識方式、及び光センサ方式のいずれかの方式によってユーザによるタッチ操作を検知する。タッチパネル110は、ディスプレイ111の表示を妨げない程度の透明度であり、ディスプレイ111の表示面の上層に一体的に取り付けられ、

50

タッチパネル 110 の位置座標及びディスプレイ 111 の表示座標は対応付けされている。これにより、ユーザはあたかもディスプレイ 111 に表示された画像を直接操作しているかのような操作感を得ることができる。また、タッチパネル 110 は該タッチパネル 110 においてユーザにタッチされた位置を示す座標情報を制御部 101 に送信する。ディスプレイ 111 は MFP 100 の各設定を行う GUI の画像等を表示する。外部メモリ 112 はハードディスク、フロッピーディスク、CD、DVD、及びメモリカード等の MFP 100 に接続可能な記憶装置である。

#### 【0016】

図 2 は、図 1 の MFP 100 のソフトウェアモジュール 200 の構成を概略的に示すブロック図である。

#### 【0017】

図 2 において、ソフトウェアモジュール 200 は、仮想マシン 201、ネイティブ制御モジュール 202、アプリフレームワークモジュール 204、及びアプリケーションモジュール（以下、単に「アプリ」という。）205 を備える。ネイティブ制御モジュール 202 は画面関連制御モジュール 203 を備え、アプリ 205 はシステムサービスモジュール 206 及び拡張アプリ 208 を備える。システムサービスモジュール 206 は AVS (Applet viewer Service) モジュール 207 を備える。ソフトウェアモジュール 200 の各処理は、CPU 102 が ROM 104 に格納されたプログラムを実行することによって行われる。なお、本実施の形態では、説明を容易にするため、アプリ 205 が 1 つの拡張アプリ 208 を備える場合を前提とするが、アプリ 205 はインストールされた複数の拡張アプリを備えても良い。

#### 【0018】

仮想マシン 201 はアプリ 205 の実行環境を提供する。仮想マシン 201 は、コード解釈機能、コマンド制御機能、リソース管理機能、グラフィックス処理制御機能、及びイベント制御機能等の各機能を備える。コード解釈機能はコンパイルされたバイトコードを解釈するための機能であり、コマンド制御機能は仮想マシン 201 によって解釈されたネイティブコマンドを制御するための機能である。リソース管理機能はアプリ 205 による各処理の実行に必要となるリソースや、ネイティブ制御モジュール 202 へ処理を要求する際に必要となるリソースを管理する機能である。グラフィックス処理制御機能はアプリ 205 からの描画命令をネイティブ制御モジュール 202 に要求するための機能である。イベント制御機能はタッチパネル 110 における入力操作の検知や、通信 I/F 108 におけるネットワーク接続の検知等の MFP 100 の状態変化をアプリ 205 に通知するための機能である。

#### 【0019】

ネイティブ制御モジュール 202 は入力制御部 105 及び表示制御部 106 の制御を行う。画面関連制御モジュール 203 は、ディスプレイ 111 に各画像、例えば、インストールされた複数の拡張アプリの中から所望する拡張アプリの実行をユーザに選択させるための図示しない設定メニューを表示させる。また、画面関連制御モジュール 203 は、タッチパネル 110 において、タッチダウン、タッチオン、タッチアップ、ムーブ、タッチオフ、及びフリック等の入力操作を検知すると、入力操作が行われた位置を座標情報に変換する。さらに、画面関連制御モジュール 203 は、タッチパネル 110 の入力操作を検知した旨を示す入力操作検知通知を仮想マシン 201 に出力する。入力操作検知通知に応じて仮想マシン 201 から描画要求が送信されると、画面関連制御モジュール 203 は当該描画要求に応じた画像をディスプレイ 111 に表示させる。例えば、図示しない設定メニューにおいて拡張アプリ 208 が選択されると、画面関連制御モジュール 203 はディスプレイ 111 の表示を上記設定メニューから拡張アプリ GUI に切り替える。アプリフレームワークモジュール 204 はアプリ 205 の各モジュールの起動等を制御する。

#### 【0020】

アプリ 205 は、計算処理、文字列処理、通信処理、画面制御処理、及び暗号処理等の MFP 100 の標準的な機能や、拡張アプリによる拡張機能を実現する。システムサービ

10

20

30

40

50

スモジュール 206 は予め MFP 100 に組み込まれているモジュールであり、拡張アプリが MFP 100 にインストールされた際に当該拡張アプリを実行可能に設定する。

【0021】

A V S モジュール 207 は該 A V S モジュール 207 の起動時に拡張アプリ G U I の枠を表示するためのプログラムである図 3 ( a ) のフレーム 301 の生成を仮想マシン 201 に要求する。フレーム 301 は、例えば、J a v a の java.awt.frame クラスで表される。また、A V S モジュール 207 は、拡張アプリ 208 から受信したアプレットの登録要求に基づいてディスプレイ 111 に画像を表示するための図 3 ( b ) のアプレット 302 ( 描画情報 ) や図 3 ( c ) のアプレット 303 をフレーム 301 に登録する。アプレット 302 , 303 は表示される画像の仕様、例えば、画像のサイズ、テキストフィールドやボタンを含めるか否かを示す情報、及び背景色の情報を含む。アプレット 302 は、例えば、拡張アプリ G U I を表示するためのプログラムである。アプレット 303 は、例えば、エラー画像を表示するためのプログラムである。本実施の形態では、フレーム 301 に対して複数のアプレットを登録可能である。フレーム 301 に登録された時点では、アプレット 302 , 303 の表示属性は不可視状態に設定され、アプレット 302 , 303 の各々に対して所定の表示条件が設定される。所定の表示条件を満たすと、登録されたアプレット 302 , 303 のうち所定の表示条件を満たしたアプレット、例えば、アプレット 302 のみが可視状態に設定され、MFP 100 はディスプレイ 111 に当該アプレット 302 に対応する拡張アプリ G U I を表示する。

【0022】

さらに、A V S モジュール 207 は、拡張アプリ 208 等から後述する図 2 の属性情報 209 等の拡張アプリの仕様を示すアプリ情報を取得し、取得されたアプリ情報を設定する旨を仮想マシン 201 に要求する。A V S モジュール 207 は、拡張アプリ 208 から拡張アプリ G U I の表示要求を受信した場合、又はネイティブ制御モジュール 202 によってユーザの入力操作を検知した場合等に、ディスプレイ 111 に拡張アプリ G U I を表示させる表示制御を行う。

【0023】

拡張アプリ 208 は、MFP 100 の機能を拡張するためのアプリであり、当該拡張アプリ 208 の起動時に A V S モジュール 207 にアプレット 302 の登録要求を行う。拡張アプリ 208 には、拡張アプリ 208 自身が拡張アプリ G U I の背景色描画を実行する仕様であるか否かを示す属性情報 209 が予め設定されている。例えば、拡張アプリ 208 自身が拡張アプリ G U I の背景色描画を実行しない仕様である場合、属性情報 209 には「T R U E」が設定され、拡張アプリ 208 自身が背景色描画を実行する仕様である場合、属性情報 209 には「F A L S E」が設定される。拡張アプリ 208 は、仮想マシン 201 によって処理可能にコンパイルされたバイトコードで構成される図示しない標準ライブラリを A P I 210 を介して参照する。標準ライブラリは、計算処理、文字列処理、通信処理、画面制御処理、及び暗号処理等の MFP 100 の標準的な機能を実現するためのライブラリである。

【0024】

図 4 は、図 1 の MFP 100 で実行される起動制御処理の手順を示すフローチャートである。

【0025】

図 4 の処理は、C P U 102 が R O M 104 に格納されたプログラムを実行することによって行われ、アプリフレームワークモジュール 204 及びアプリ 205 によって実行される。

【0026】

図 4 において、まず、アプリフレームワークモジュール 204 は、仮想マシン 201 からアプリ 205 の起動要求を受信すると、アプリ 205 における全てのモジュールが起動しているか否かを判別する ( ステップ S 401 ) 。

【0027】

ステップS401の判別の結果、アプリ205における全てのモジュールが起動しているとき、アプリフレームワークモジュール204は本処理を終了する。一方、ステップS401の判別の結果、アプリ205におけるいずれかのモジュールが起動していないとき、アプリフレームワークモジュール204は、アプリ205に対して起動していないモジュールの起動要求を行う(ステップS402)。次いで、アプリ205は、アプリフレームワークモジュール204から起動要求を受信すると、後述する図5のアプリ起動処理を行って(ステップS403)、起動していないモジュールを起動させ、ステップS401の処理に戻る。

【0028】

図5は、図4のステップS403のアプリ起動処理の手順を示すフローチャートである。

10

【0029】

図5の処理は、アプリフレームワークモジュール204からアプリ205にAVSモジュール207の起動要求が行われた場合を前提とし、AVSモジュール207、仮想マシン201、及び画面関連制御モジュール203によって実行される。

【0030】

図5において、まず、AVSモジュール207は、アプリフレームワークモジュール204からの起動要求に応じて起動処理を行い、起動処理が完了すると、仮想マシン201にフレーム301の生成要求を行う(ステップS501)。次いで、仮想マシン201は、当該生成要求に応じてフレーム301に対応する画面バッファの確保要求を画面関連制御モジュール203に行う(ステップS502)。次いで、画面関連制御モジュール203は当該画面バッファの確保要求に応じてフレーム301に対応する画面バッファをRAM103に確保する(ステップS503)。その後、画面関連制御モジュール203は画面バッファを確保した旨を示す確保通知を仮想マシン201を介してAVSモジュール207に出力する。次いで、AVSモジュール207は、確保通知を受信すると、MFP100にインストールされた各拡張アプリのアプリ情報を管理するための領域をROM104等に確保し(ステップS504)、図4のステップS401の処理を行う。図4の処理の実行が完了すると、アプリ205は拡張アプリ208からアプレットの登録要求が行われるまで待機する。

20

【0031】

図6は、図1のMFP100で実行されるアプレット登録処理の手順を示すフローチャートである。

30

【0032】

図6の処理は、CPU102がROM104に格納されたプログラムを実行することによって行われる。図6の処理は、図5の処理の実行が完了した後に実行され、アプリ205の拡張アプリ208、AVSモジュール207、及び仮想マシン201によって実行される。

【0033】

図6において、まず、拡張アプリ208はAVSモジュール207にアプレット302の登録要求を行う(ステップS601)。次いで、AVSモジュール207は該登録要求を行った拡張アプリ208のアプリ情報を参照する(ステップS602)。本実施の形態では、拡張アプリがMFP100にインストールされると、当該拡張アプリのアプリ情報がROM104に格納された図7の管理データ701に設定される。管理データ701はインストールされた全ての拡張アプリのアプリ情報を含む。管理データ701は、識別子702、アプリ名703、背景色描画属性704、及び画面サイズ705を備える。識別子702は拡張アプリを識別するために一意に割り当てられた番号、例えば、UIDである。アプリ名703は拡張アプリの種別を示す情報である。背景色描画属性704は属性情報209に対応する情報である。画面サイズ705は拡張アプリGUI等の各画像のサイズである。なお、本実施の形態では、アプリ情報の管理方法として、インストールされた全ての拡張アプリのアプリ情報を管理データ701に設定して各拡張アプリのアプリ

40

50

情報を管理する場合を説明したが、アプリ情報の管理方法はこれに限られない。各拡張アプリのアプリ情報はA V Sモジュール207が読み出し可能に管理されていれば良く、例えば、アプリ情報がアプリ毎にROM104等に直接格納されていても良い。

**【0034】**

次いで、A V Sモジュール207は拡張アプリ208に属性情報209が設定されているか否かを判別する(ステップS603)。ステップS603では、例えば、管理データ701における拡張アプリ208の背景色描画属性704に「TRUE」及び「FALSE」のいずれかが設定されている場合、拡張アプリ208に属性情報209が設定されていると判別する。一方、管理データ701における拡張アプリ208の背景色描画属性704に属性情報209が設定されていない旨を示す「未設定」や「-」が設定されている場合、拡張アプリ208に属性情報209が設定されていないと判別する。

10

**【0035】**

ステップS603の判別の結果、拡張アプリ208に属性情報209が設定されていないとき、A V Sモジュール207は仮想マシン201の背景色描画機能を有効、つまり、仮想マシン201による背景色描画処理を実行する旨を設定する(ステップS604)。次いで、A V Sモジュール207は後述するステップS607以降の処理を行う。

**【0036】**

ステップS603の判別の結果、拡張アプリ208に属性情報209が設定されているとき、A V Sモジュール207は属性情報209に「TRUE」が設定されているか否かを判別する(ステップS605)。

20

**【0037】**

ステップS605の判別の結果、属性情報209に「TRUE」が設定されているとき、A V Sモジュール207はステップS604以降の処理を行う。すなわち、本実施の形態では、拡張アプリ208自身が背景色描画を実行する仕様である旨を示す「FALSE」が属性情報209に設定されていない場合に、仮想マシン201による背景色描画処理を実行する旨が設定される。一方、ステップS605の判別の結果、属性情報209に「FALSE」が設定されているとき、A V Sモジュール207は仮想マシン201の背景色描画機能を無効、つまり、仮想マシン201による背景色描画処理を実行しない旨を設定する(ステップS606)。次いで、A V Sモジュール207はアプレット302をフレーム301に登録するためのアプレット302の登録要求を仮想マシン201に行う(ステップS607)。次いで、仮想マシン201はA V Sモジュール207から受信したアプレット登録要求に応じてアプレット302及び背景色描画機能の設定を対応付けして登録し(ステップS608)(登録手段)、本処理を終了する。

30

**【0038】**

図8は、図1のMFP100で実行される背景描画制御処理の手順を示すフローチャートである。

**【0039】**

図8の処理は、CPU102がROM104に格納されたプログラムを実行することによって行われる。図8の処理は、仮想マシン201が拡張アプリGUIの表示要求を受信した場合を前提とし、仮想マシン201及び画面関連制御モジュール203によって実行される。

40

**【0040】**

図8において、まず、拡張アプリGUIの表示要求を受信すると、仮想マシン201は、登録された全てのアプレットに対する処理を完了したか否かを判別する(ステップS801)。

**【0041】**

ステップS801の判別の結果、登録された全てのアプレットに対する処理を完了したとき、仮想マシン201は本処理を終了する。一方、ステップS801の判別の結果、登録されたいずれかのアプレットに対する処理を完了しないとき、仮想マシン201は、処理を完了していないアプレットから一のアプレットを選択する。その後、仮想マシン20

50

1は一のアプレットの表示属性が可視状態に設定されているか否かを判別する(ステップS802)。

【0042】

ステップS802の判別の結果、一のアプレットの表示属性が不可視状態に設定されているとき、仮想マシン201はステップS801の処理に戻る。一方、ステップS802の判別の結果、一のアプレットの表示属性が可視状態に設定されているとき、仮想マシン201は一のアプレットに対応付けされた背景色描画機能の設定が「有効」に設定されているか否かを判別する(ステップS803)。

【0043】

ステップS803の判別の結果、背景色描画機能の設定が「無効」に設定されているとき、仮想マシン201はステップS801の処理に戻る。一方、ステップS803の判別の結果、背景色描画機能の設定が「有効」に設定されているとき、仮想マシン201は、一のアプレット、具体的に、アプレット302に設定された背景色の描画要求を画面関連制御モジュール203に行う(ステップS804)。次いで、画面関連制御モジュール203は、当該描画要求の背景色を画面バッファに設定し(ステップS805)、本処理を終了する。その後、拡張アプリ208は該拡張アプリ208の仕様に依じて拡張アプリGUIの描画処理を行う。例えば、拡張アプリ208自身が背景色描画を実行する仕様である場合、拡張アプリ208は背景を含む全ての拡張アプリGUIの描画処理を行う。一方、拡張アプリ208自身が背景色描画を実行しない仕様である場合、拡張アプリ208は背景以外の拡張アプリGUIの描画処理を行う。これにより、ディスプレイ111には拡張アプリGUIが表示される。

10

20

【0044】

上述した本実施の形態によれば、拡張アプリ208の仕様が背景色描画を実行する仕様である旨を示す「FALSE」が属性情報209に設定されていない場合、仮想マシン201による背景色描画処理を実行する旨が設定される。これにより、拡張アプリGUIを表示する際に当該拡張アプリGUIの背景が描画されないといった事態が発生するのを回避することができ、もって、不完全な拡張アプリGUIの表示を防止することができる。

【0045】

また、上述した本実施の形態では、拡張アプリ208の仕様が背景色描画を実行する仕様である旨を示す「FALSE」が属性情報209に設定されていない場合は、属性情報209に「TRUE」が設定されている場合、又は拡張アプリ208に属性情報209が設定されていない場合である。これにより、拡張アプリ208によって拡張アプリGUIの背景色描画が実行されない可能性がある場合に対して拡張アプリGUIの背景を仮想マシン201によって確実に描画することができる。

30

【0046】

さらに、上述した本実施の形態では、属性情報209に「FALSE」が設定されている場合、仮想マシン201による背景色描画処理を実行しない旨が設定される。これにより、拡張アプリ208が背景を含む全ての拡張アプリGUIの描画処理を行う場合に仮想マシン201も拡張アプリGUIの背景色描画を行う事態を無くすることができる。その結果、背景色描画が複数回実行されることに起因する拡張アプリGUIの表示のちらつきを抑制することができる。

40

【0047】

上述した本実施の形態では、アプレット302が背景色描画機能の設定に対応付けて登録される。これにより、登録されたアプレット302を用いて拡張アプリGUIを表示する際に仮想マシン201に背景色描画を実行させる必要があるか否かを容易に特定することができる。

【0048】

また、上述した本実施の形態では、拡張アプリは属性情報209を含むので、当該拡張アプリの仕様を確実に取得することができる。

【0049】

50

上述した本実施の形態では、拡張アプリの属性情報209をタッチパネル110によってユーザに設定させても良い。これにより、属性情報209の誤設定に気付いた際に属性情報209を変更するために、属性情報209を設定し直した拡張アプリを再インストールするといった処理を行う必要をなくすことができ、もって、属性情報209の設定を容易に変更することができる。

【0050】

また、上述した本実施の形態では、タブレット302等の描画処理を行うための各データを解析し、解析結果に基づいて拡張アプリ208自身が背景色描画を実行しない仕様であるか否かを判別しても良い。これにより、拡張アプリ208に属性情報209が無くても、仮想マシン201に背景色描画を実行させる必要があるか否かを特定することができる。

10

【0051】

本発明は、上述の実施の形態の1以上の機能を実現するプログラムをネットワーク又は記憶媒体を介してシステム又は装置に供給し、該システム又は装置のコンピュータにおける1つ以上のプロセッサがプログラムを読み出して実行する処理でも実現可能である。また、本発明は、1以上の機能を実現する回路(例えば、ASIC)によっても実現可能である。

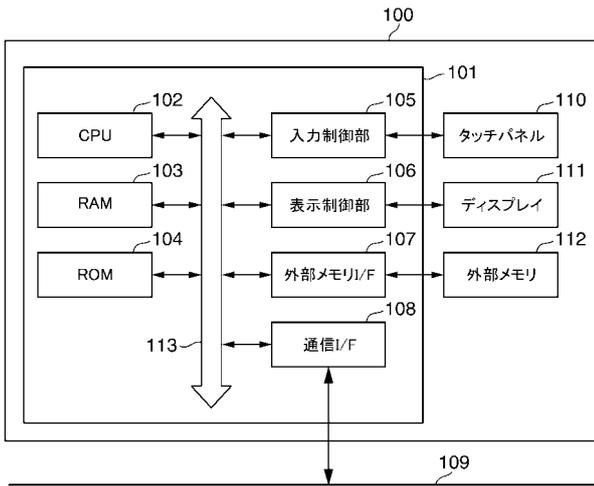
【符号の説明】

【0052】

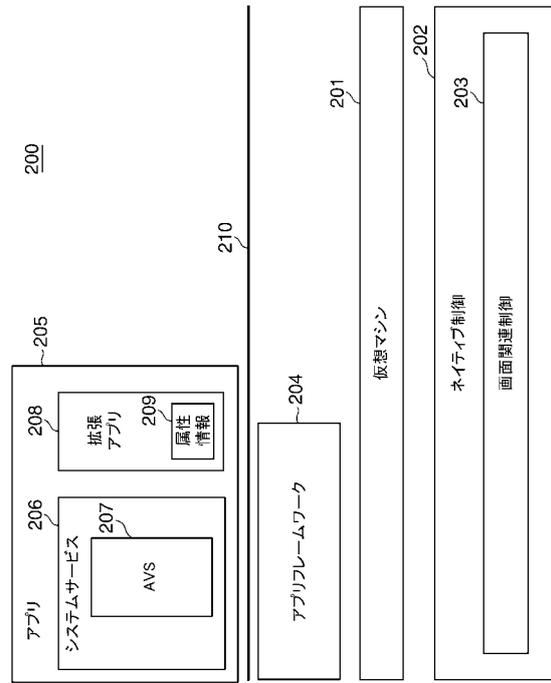
100 MFP  
102 CPU  
106 表示制御部  
110 タッチパネル  
201 仮想マシン  
208 拡張アプリ  
302 タブレット

20

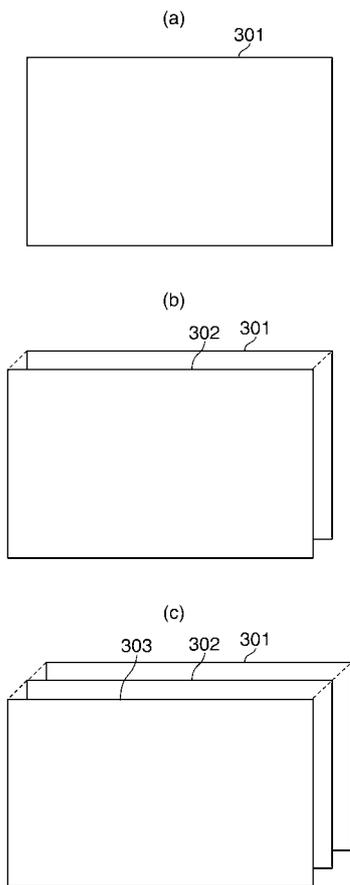
【図1】



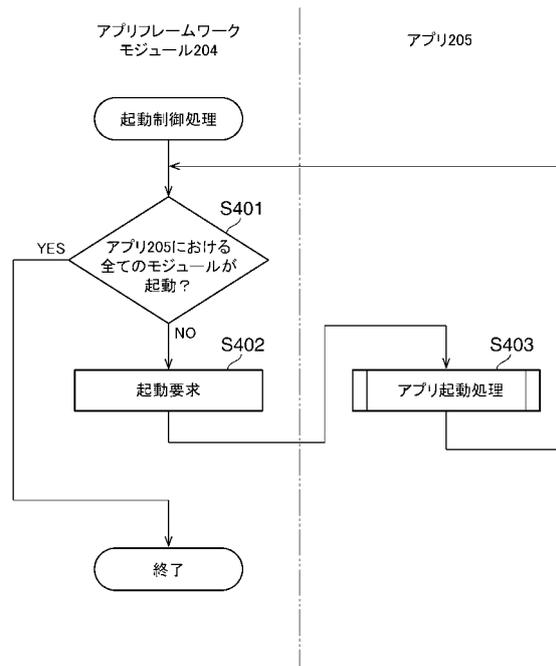
【図2】



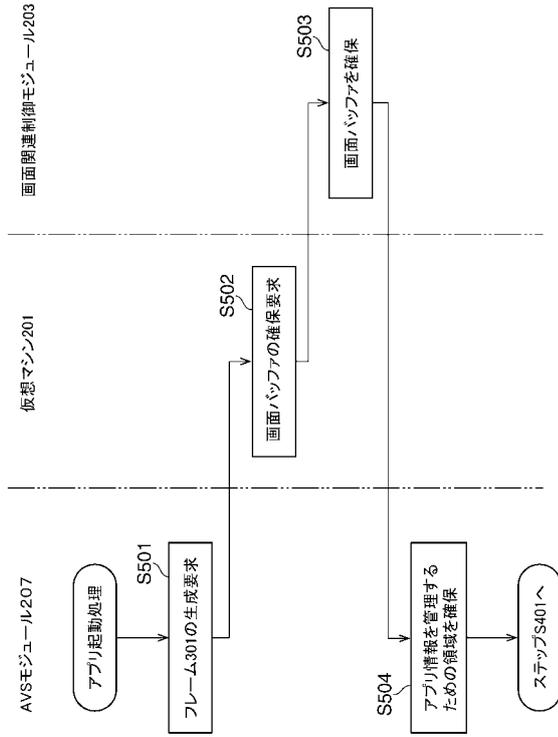
【図3】



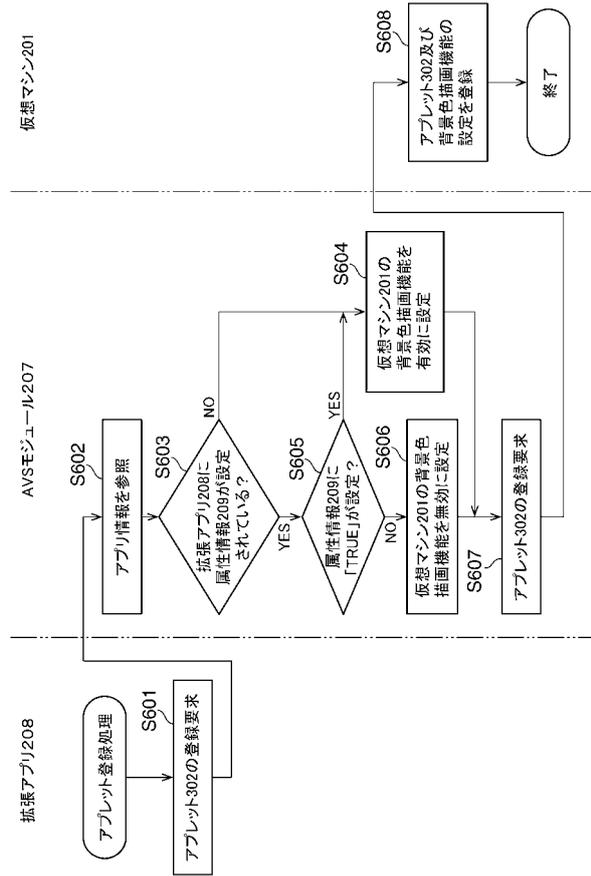
【図4】



【図5】



【図6】



【図7】

701	識別子	拡張アプリ208	アプリX	アプリY
702	アプリ名	メニュー	エラーリカバリ	ブラウザ
703	背景色描画属性	TRUE	FALSE	未設定
704	画面サイズ	800×400	640×400	800×600

【図8】

