

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4625343号
(P4625343)

(45) 発行日 平成23年2月2日(2011.2.2)

(24) 登録日 平成22年11月12日(2010.11.12)

(51) Int. Cl.		F I			
HO4N	1/00	(2006.01)	HO4N	1/00	C
GO3G	21/00	(2006.01)	GO3G	21/00	386
GO6F	9/445	(2006.01)	GO3G	21/00	396
			GO6F	9/06	610A

請求項の数 27 (全 33 頁)

(21) 出願番号	特願2005-33318 (P2005-33318)	(73) 特許権者	000006747
(22) 出願日	平成17年2月9日(2005.2.9)		株式会社リコー
(65) 公開番号	特開2005-269619 (P2005-269619A)		東京都大田区中馬込1丁目3番6号
(43) 公開日	平成17年9月29日(2005.9.29)	(74) 代理人	100070150
審査請求日	平成19年11月21日(2007.11.21)		弁理士 伊東 忠彦
(31) 優先権主張番号	特願2004-40471 (P2004-40471)	(72) 発明者	安藤 光男
(32) 優先日	平成16年2月17日(2004.2.17)		東京都大田区中馬込1丁目3番6号 株式
(33) 優先権主張国	日本国(JP)	(72) 発明者	秋吉 邦洋
前置審査			東京都大田区中馬込1丁目3番6号 株式
			会社リコー内
		審査官	松尾 淳一

最終頁に続く

(54) 【発明の名称】 画像形成装置、端末装置、情報処理方法、情報処理プログラム、及び記録媒体

(57) 【特許請求の範囲】

【請求項1】

アプリケーションを管理するプラットフォームを備える画像形成装置であって、
前記プラットフォームによって管理されるアプリケーションとして、
前記プラットフォームによって管理される他のアプリケーションを操作対象とする操作画面を表示する操作画面表示アプリケーションと、
前記他のアプリケーションにエラーが発生した場合に、エラーが発生した旨の情報を出力するエラー発生出力アプリケーションと、
前記他のアプリケーションから、前記プラットフォームによって管理されるアプリケーションのインタフェースに対する処理要求に、予め決めておいた動作条件の許可範囲を超えるルール違反があった場合に、ルール違反があった旨の情報を出力するルール違反出力アプリケーションと、を備え、

前記他のアプリケーション、前記操作画面表示アプリケーション、前記エラー発生出力アプリケーション、及び前記ルール違反出力アプリケーションは、同一のプロセス上で実行されることを特徴とする画像形成装置。

【請求項2】

前記操作画面表示アプリケーションは、前記操作画面として、画面を占有させるアプリケーションの切替操作を行うための切替操作画面を表示することを特徴とする請求項1に記載の画像形成装置。

【請求項3】

前記操作画面表示アプリケーションは、前記操作画面として、アプリケーションの起動操作を行うための起動操作画面を表示することを特徴とする請求項 1 乃至 2 のいずれか 1 項に記載の画像形成装置。

【請求項 4】

前記操作画面表示アプリケーションは、前記操作画面として、アプリケーションの終了操作を行うための終了操作画面を表示することを特徴とする請求項 1 乃至 3 のいずれか 1 項に記載の画像形成装置。

【請求項 5】

前記操作画面表示アプリケーションは、前記操作画面として、アプリケーションのインストール操作を行うためのインストール操作画面を表示することを特徴とする請求項 1 乃至 4 のいずれか 1 項に記載の画像形成装置。

10

【請求項 6】

前記操作画面表示アプリケーションは、前記操作画面として、アプリケーションのアンインストール操作を行うためのアンインストール操作画面を表示することを特徴とする請求項 1 乃至 5 のいずれか 1 項に記載の画像形成装置。

【請求項 7】

前記プラットフォームによって管理されるアプリケーションとして、前記プラットフォームによって管理される他のアプリケーションを操作対象とするタスクバーを表示するタスクバー表示アプリケーションを、さらに備え、

前記他のアプリケーションと前記タスクバー表示アプリケーションとは、同一のプロセス上で実行されることを特徴とする請求項 1 乃至 6 のいずれか 1 項に記載の画像形成装置。

20

【請求項 8】

前記タスクバー表示アプリケーションは、前記タスクバーとして、画面を占有させるアプリケーションの切替操作を行うための切替タスクバーを表示することを特徴とする請求項 7 に記載の画像形成装置。

【請求項 9】

前記プラットフォームは、アプリケーションの実行ファイルの自動更新処理を、1 対 1 で当該アプリケーションの情報を管理する管理ファイルの内容を元にして実行することを特徴とする請求項 1 乃至 8 のいずれか 1 項に記載の画像形成装置。

30

【請求項 10】

前記プラットフォームは、1 対 1 でアプリケーションを管理する個別管理手段と、複数の前記個別管理手段を管理する一括管理手段と、前記一括管理手段を管理する管理手段を備えることを特徴とする請求項 1 乃至 9 のいずれか 1 項に記載の画像形成装置。

【請求項 11】

前記プラットフォームは、アプリケーションの初期化状態、アプリケーションの活性化状態、アプリケーションの停止状態、アプリケーションの終了状態を含むライフサイクルを管理することを特徴とする請求項 1 乃至 10 のいずれか 1 項に記載の画像形成装置。

【請求項 12】

前記アプリケーションは、コンパイラによりソースコードからバイトコードに一括翻訳されており、仮想マシンにより逐次実行されることを特徴とする請求項 1 乃至 11 のいずれか 1 項に記載の画像形成装置。

40

【請求項 13】

前記アプリケーションは、それぞれスレッドとして実行されることを特徴とする請求項 1 乃至 12 のいずれか 1 項に記載の画像形成装置。

【請求項 14】

前記アプリケーションは、X11 であることを特徴とする請求項 1 乃至 13 のいずれか 1 項に記載の画像形成装置。

【請求項 15】

前記アプリケーションと前記プラットフォームは、同一のプロセス上で実行されること

50

を特徴とする請求項 1 乃至 14 のいずれか 1 項に記載の画像形成装置。

【請求項 16】

アプリケーションを管理するプラットフォームを備える画像形成装置内で実行される情報処理方法であって、

前記プラットフォームによって管理されるアプリケーションとして備えられた、前記プラットフォームによって管理される他のアプリケーションを操作対象とする操作画面を表示する操作画面表示アプリケーションと、前記他のアプリケーションにエラーが発生した場合に、エラーが発生した旨の情報を出力するエラー発生出力アプリケーションと、前記他のアプリケーションから、前記プラットフォームによって管理されるアプリケーションのインタフェースに対する処理要求に、予め決めておいた動作条件の許可範囲を超えるル 10
ール違反があった場合に、ルール違反があった旨の情報を出力するルール違反出力アプ
リケーションと、を、前記他のアプリケーションと同一のプロセス上で実行するステップ
を備えることを特徴とする情報処理方法。

【請求項 17】

前記プラットフォームによって管理されるアプリケーションとして備えられた、前記プラットフォームによって管理される他のアプリケーションを操作対象とするタスクバーを表示するタスクバー表示アプリケーションを、前記他のアプリケーションと同一のプロセス上で実行するステップを備えることを特徴とする請求項 16 に記載の情報処理方法。

【請求項 18】

アプリケーションを管理するプラットフォームを備える画像形成装置内で実行するため 20
の情報処理プログラムであって、

前記プラットフォームによって管理されるアプリケーションとして備えられた、前記プラットフォームによって管理される他のアプリケーションを操作対象とする操作画面を表示する操作画面表示アプリケーションと、前記他のアプリケーションにエラーが発生した場合に、エラーが発生した旨の情報を出力するエラー発生出力アプリケーションと、前記他のアプリケーションから、前記プラットフォームによって管理されるアプリケーション 30
のインタフェースに対する処理要求に、予め決めておいた動作条件の許可範囲を超えるル
ール違反があった場合に、ルール違反があった旨の情報を出力するルール違反出力アプ
リケーションと、を、前記他のアプリケーションと同一のプロセス上で実行させることを特
徴とする情報処理プログラム。

【請求項 19】

前記プラットフォームによって管理されるアプリケーションとして備えられた、前記プラットフォームによって管理される他のアプリケーションを操作対象とするタスクバーを表示するタスクバー表示アプリケーションを、前記他のアプリケーションと同一のプロセス上で実行させることを特徴とする請求項 18 に記載の情報処理プログラム。

【請求項 20】

画像形成装置のエミュレータとして機能する端末装置であって、

当該端末装置のプラットフォームにより管理されるアプリケーションとして、

当該端末装置又は当該画像形成装置のプラットフォームにより管理される他のアプリー 40
ケーションを操作対象とする操作画面を表示する操作画面表示アプリケーションと、

前記他のアプリケーションにエラーが発生した場合に、エラーが発生した旨の情報を出力するエラー発生出力アプリケーションと、

前記他のアプリケーションから、前記プラットフォームによって管理されるアプリー 50
ーションのインタフェースに対する処理要求に、予め決めておいた動作条件の許可範囲を超
えるルール違反があった場合に、ルール違反があった旨の情報を出力するルール違反出力
アプリケーションと、を備え、

前記他のアプリケーション、前記操作画面表示アプリケーション、前記エラー発生出力アプリケーション、及び前記ルール違反出力アプリケーションとは、同一のプロセス上で実行されることを特徴とする端末装置。

【請求項 21】

10

20

30

40

50

前記プラットフォームにより管理されるアプリケーションとして、当該端末装置又は当該画像形成装置のプラットフォームにより管理される他のアプリケーションを操作対象とするタスクバーを表示するタスクバー表示アプリケーションを、さらに備え、

前記他のアプリケーションと前記タスクバー表示アプリケーションとは、同一のプロセス上で実行されることを特徴とする請求項 20 に記載の端末装置。

【請求項 22】

画像形成装置のエミュレータとして機能する端末装置内で実行される情報処理方法であって、

当該端末装置のプラットフォームにより管理されるアプリケーションとして備えられた、当該端末装置又は当該画像形成装置のプラットフォームにより管理される他のアプリケーションを操作対象とする操作画面を表示する操作画面表示アプリケーションと、前記他のアプリケーションにエラーが発生した場合に、エラーが発生した旨の情報を出力するエラー発生出力アプリケーションと、前記他のアプリケーションから、前記プラットフォームによって管理されるアプリケーションのインタフェースに対する処理要求に、予め決めておいた動作条件の許可範囲を超えるルール違反があった場合に、ルール違反があった旨の情報を出力するルール違反出力アプリケーションと、を、前記他のアプリケーションと同一のプロセス上で実行するステップを備えることを特徴とする情報処理方法。

10

【請求項 23】

前記プラットフォームにより管理されるアプリケーションとして備えられた、当該端末装置又は当該画像形成装置のプラットフォームにより管理される他のアプリケーションを操作対象とするタスクバーを表示するタスクバー表示アプリケーションを、前記他のアプリケーションと同一のプロセス上で実行するステップを備えることを特徴とする請求項 22 に記載の情報処理方法。

20

【請求項 24】

画像形成装置のエミュレータとして機能する端末装置内で実行するための情報処理プログラムであって、

当該端末装置のプラットフォームにより管理されるアプリケーションとして備えられた、当該端末装置又は当該画像形成装置のプラットフォームにより管理される他のアプリケーションを操作対象とする操作画面を表示する操作画面表示アプリケーションと、前記他のアプリケーションにエラーが発生した場合に、エラーが発生した旨の情報を出力するエラー発生出力アプリケーションと、前記他のアプリケーションから、前記プラットフォームによって管理されるアプリケーションのインタフェースに対する処理要求に、予め決めておいた動作条件の許可範囲を超えるルール違反があった場合に、ルール違反があった旨の情報を出力するルール違反出力アプリケーションと、を、前記他のアプリケーションと同一のプロセス上で実行させることを特徴とする情報処理プログラム。

30

【請求項 25】

前記プラットフォームにより管理されるアプリケーションとして備えられた、当該端末装置又は当該画像形成装置のプラットフォームにより管理される他のアプリケーションを操作対象とするタスクバーを表示するタスクバー表示アプリケーションを、前記他のアプリケーションと同一のプロセス上で実行させることを特徴とする請求項 24 に記載の情報処理プログラム。

40

【請求項 26】

請求項 18 又は 19 に記載の情報処理プログラムを記録した記録媒体。

【請求項 27】

請求項 24 又は 25 に記載の情報処理プログラムを記録した記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、コピー/プリンタ/スキャナ/ファクシミリ/複合機/融合機等の画像形成装置、パーソナルコンピュータ等の端末装置、情報処理方法、情報処理プログラム、SD

50

card / CD - ROM等の記録媒体に関する。

【背景技術】

【0002】

近年、コピーとプリンタとスキャナとファクシミリとしての機能を備える複合機や融合機が市販されるようになった。複合機や融合機は、撮像部や印刷部や通信部や操作表示部等のハードウェアを備えると共に、コピーとプリンタとスキャナとファクシミリに対応する4種類のソフトウェアを備え、これらのソフトウェアが切り替わることにより、コピーやプリンタやスキャナやファクシミリとして機能する。複合機や融合機は、コピーやプリンタとして機能する場合、画像を印刷用紙等に印刷することになり、コピーやスキャナやとして機能する場合、画像を読取原稿から読み取ることになり、ファクシミリとして機能する場合、画像を電話回線等を介して他の機器と授受することになる。

10

【特許文献1】特開2002-84383号公報

【発明の開示】

【発明が解決しようとする課題】

【0003】

複合機や融合機内では、様々なアプリケーションが実行される。複合機や融合機内でのアプリケーションの実行態様としては、1プロセス上で1アプリケーションが実行される「1プロセス1アプリ」が主流であるが、1プロセス上で複数のアプリケーションが実行される「1プロセス複数アプリ」も登場しつつある。しかし、1プロセス1アプリを前提とするシステムで1プロセス複数アプリの情報処理を実行するような場合、1プロセス1

20

アプリ向けのアプリケーション管理機構では1プロセス複数アプリの各アプリケーションを管理することができないので、1プロセス複数アプリ向けのアプリケーション管理機構が必要となる。この例からも理解されるように、複合機や融合機には、アプリケーションの実行態様に応じたアプリケーション管理機構が必要である。

【0004】

したがって、本発明は、新たなアプリケーション管理機構を備える画像形成装置を提案することを課題とする。

【課題を解決するための手段】

【0005】

請求項1に記載の発明は、アプリケーションを管理するプラットフォームを備える画像形成装置であって、前記プラットフォームによって管理されるアプリケーションとして、前記プラットフォームによって管理される他のアプリケーションを操作対象とする操作画面を表示する操作画面表示アプリケーションと、前記他のアプリケーションにエラーが発生した場合に、エラーが発生した旨の情報を出力するエラー発生出力アプリケーションと、前記他のアプリケーションから、前記プラットフォームによって管理されるアプリケーションのインタフェースに対する処理要求に、予め決めておいた動作条件の許可範囲を超えるルール違反があった場合に、ルール違反があった旨の情報を出力するルール違反出力アプリケーションと、を備え、前記他のアプリケーション、前記操作画面表示アプリケーション、前記エラー発生出力アプリケーション、及び前記ルール違反出力アプリケーションは、同一のプロセス上で実行されることを特徴とする画像形成装置である。

30

40

【0006】

請求項2に記載の発明は、前記操作画面表示アプリケーションは、前記操作画面として、画面を占有させるアプリケーションの切替操作を行うための切替操作画面を表示することを特徴とする請求項1に記載の画像形成装置である。

【0007】

請求項3に記載の発明は、前記操作画面表示アプリケーションは、前記操作画面として、アプリケーションの起動操作を行うための起動操作画面を表示することを特徴とする請求項1乃至2のいずれか1項に記載の画像形成装置である。

【0008】

請求項4に記載の発明は、前記操作画面表示アプリケーションは、前記操作画面として

50

、アプリケーションの終了操作を行うための終了操作画面を表示することを特徴とする請求項 1 乃至 3 のいずれか 1 項に記載の画像形成装置である。

【 0 0 0 9 】

請求項 5 に記載の発明は、前記操作画面表示アプリケーションは、前記操作画面として、アプリケーションのインストール操作を行うためのインストール操作画面を表示することを特徴とする請求項 1 乃至 4 のいずれか 1 項に記載の画像形成装置である。

【 0 0 1 0 】

請求項 6 に記載の発明は、前記操作画面表示アプリケーションは、前記操作画面として、アプリケーションのアンインストール操作を行うためのアンインストール操作画面を表示することを特徴とする請求項 1 乃至 5 のいずれか 1 項に記載の画像形成装置である。

10

【 0 0 1 1 】

請求項 7 に記載の発明は、前記プラットフォームによって管理されるアプリケーションとして、前記プラットフォームによって管理される他のアプリケーションを操作対象とするタスクバーを表示するタスクバー表示アプリケーションを、さらに備え、前記他のアプリケーションと前記タスクバー表示アプリケーションとは、同一のプロセス上で実行されることを特徴とする請求項 1 乃至 6 のいずれか 1 項に記載の画像形成装置である。

【 0 0 1 2 】

請求項 8 に記載の発明は、前記タスクバー表示アプリケーションは、前記タスクバーとして、画面を占有させるアプリケーションの切替操作を行うための切替タスクバーを表示することを特徴とする請求項 7 に記載の画像形成装置である。

20

【 0 0 1 5 】

請求項 9 に記載の発明は、前記プラットフォームは、アプリケーションの実行ファイルの自動更新処理を、1 対 1 で当該アプリケーションの情報を管理する管理ファイルの内容を元にして実行することを特徴とする請求項 1 乃至 8 のいずれか 1 項に記載の画像形成装置である。

【 0 0 1 6 】

請求項 1 0 に記載の発明は、前記プラットフォームは、1 対 1 でアプリケーションを管理する個別管理手段と、複数の前記個別管理手段を管理する一括管理手段と、前記一括管理手段を管理する管理手段を備えることを特徴とする請求項 1 乃至 9 のいずれか 1 項に記載の画像形成装置である。

30

【 0 0 1 7 】

請求項 1 1 に記載の発明は、前記プラットフォームは、アプリケーションの初期化状態、アプリケーションの活性化状態、アプリケーションの停止状態、アプリケーションの終了状態を含むライフサイクルを管理することを特徴とする請求項 1 乃至 1 0 のいずれか 1 項に記載の画像形成装置である。

【 0 0 1 9 】

請求項 1 2 に記載の発明は、前記アプリケーションは、コンパイラによりソースコードからバイトコードに一括翻訳されており、仮想マシンにより逐次実行されることを特徴とする請求項 1 乃至 1 1 のいずれか 1 項に記載の画像形成装置である。

【 0 0 2 0 】

請求項 1 3 に記載の発明は、前記アプリケーションは、それぞれスレッドとして実行されることを特徴とする請求項 1 乃至 1 2 のいずれか 1 項に記載の画像形成装置である。

40

【 0 0 2 1 】

請求項 1 4 に記載の発明は、前記アプリケーションは、X 1 e t であることを特徴とする請求項 1 乃至 1 3 のいずれか 1 項に記載の画像形成装置である。

【 0 0 2 2 】

請求項 1 5 に記載の発明は、前記アプリケーションと前記プラットフォームは、同一のプロセス上で実行されることを特徴とする請求項 1 乃至 1 4 のいずれか 1 項に記載の画像形成装置である。

【 0 0 2 3 】

50

請求項 16 に記載の発明は、アプリケーションを管理するプラットフォームを備える画像形成装置内で実行される情報処理方法であって、前記プラットフォームによって管理されるアプリケーションとして備えられた、前記プラットフォームによって管理される他のアプリケーションを操作対象とする操作画面を表示する操作画面表示アプリケーションと、前記他のアプリケーションにエラーが発生した場合に、エラーが発生した旨の情報を出力するエラー発生出力アプリケーションと、前記他のアプリケーションから、前記プラットフォームによって管理されるアプリケーションのインタフェースに対する処理要求に、予め決めておいた動作条件の許可範囲を超えるルール違反があった場合に、ルール違反があった旨の情報を出力するルール違反出力アプリケーションと、を、前記他のアプリケーションと同一のプロセス上で実行するステップを備えることを特徴とする情報処理方法である。

10

【 0 0 2 4 】

請求項 17 に記載の発明は、前記プラットフォームによって管理されるアプリケーションとして備えられた、前記プラットフォームによって管理される他のアプリケーションを操作対象とするタスクバーを表示するタスクバー表示アプリケーションを、前記他のアプリケーションと同一のプロセス上で実行するステップを備えることを特徴とする請求項 16 に記載の情報処理方法である。

【 0 0 2 7 】

請求項 18 に記載の発明は、アプリケーションを管理するプラットフォームを備える画像形成装置内で実行するための情報処理プログラムであって、前記プラットフォームによって管理されるアプリケーションとして備えられた、前記プラットフォームによって管理される他のアプリケーションを操作対象とする操作画面を表示する操作画面表示アプリケーションと、前記他のアプリケーションにエラーが発生した場合に、エラーが発生した旨の情報を出力するエラー発生出力アプリケーションと、前記他のアプリケーションから、前記プラットフォームによって管理されるアプリケーションのインタフェースに対する処理要求に、予め決めておいた動作条件の許可範囲を超えるルール違反があった場合に、ルール違反があった旨の情報を出力するルール違反出力アプリケーションと、を、前記他のアプリケーションと同一のプロセス上で実行させることを特徴とする情報処理プログラムである。

20

【 0 0 2 8 】

請求項 19 に記載の発明は、前記プラットフォームによって管理されるアプリケーションとして備えられた、前記プラットフォームによって管理される他のアプリケーションを操作対象とするタスクバーを表示するタスクバー表示アプリケーションを、前記他のアプリケーションと同一のプロセス上で実行させることを特徴とする請求項 18 に記載の情報処理プログラムである。

30

【 0 0 3 1 】

請求項 20 に記載の発明は、画像形成装置のエミュレータとして機能する端末装置であって、当該端末装置のプラットフォームにより管理されるアプリケーションとして、当該端末装置又は当該画像形成装置のプラットフォームにより管理される他のアプリケーションを操作対象とする操作画面を表示する操作画面表示アプリケーションと、前記他のアプリケーションにエラーが発生した場合に、エラーが発生した旨の情報を出力するエラー発生出力アプリケーションと、前記他のアプリケーションから、前記プラットフォームによって管理されるアプリケーションのインタフェースに対する処理要求に、予め決めておいた動作条件の許可範囲を超えるルール違反があった場合に、ルール違反があった旨の情報を出力するルール違反出力アプリケーションと、を備え、前記他のアプリケーション、前記操作画面表示アプリケーション、前記エラー発生出力アプリケーション、及び前記ルール違反出力アプリケーションとは、同一のプロセス上で実行されることを特徴とする端末装置である。

40

【 0 0 3 2 】

請求項 21 に記載の発明は、前記プラットフォームにより管理されるアプリケーション

50

として、当該端末装置又は当該画像形成装置のプラットフォームにより管理される他のアプリケーションを操作対象とするタスクバーを表示するタスクバー表示アプリケーションを、さらに備え、前記他のアプリケーションと前記タスクバー表示アプリケーションとは、同一のプロセス上で実行されることを特徴とする請求項 2 0 に記載の端末装置である。

【 0 0 3 5 】

請求項 2 2 に記載の発明は、画像形成装置のエミュレータとして機能する端末装置内で実行される情報処理方法であって、当該端末装置のプラットフォームにより管理されるアプリケーションとして備えられた、当該端末装置又は当該画像形成装置のプラットフォームにより管理される他のアプリケーションを操作対象とする操作画面を表示する操作画面表示アプリケーションと、前記他のアプリケーションにエラーが発生した場合に、エラーが発生した旨の情報を出力するエラー発生出力アプリケーションと、前記他のアプリケーションから、前記プラットフォームによって管理されるアプリケーションのインタフェースに対する処理要求に、予め決めておいた動作条件の許可範囲を超えるルール違反があった場合に、ルール違反があった旨の情報を出力するルール違反出力アプリケーションと、を、前記他のアプリケーションと同一のプロセス上で実行するステップを備えることを特徴とする情報処理方法である。

10

【 0 0 3 6 】

請求項 2 3 に記載の発明は、前記プラットフォームにより管理されるアプリケーションとして備えられた、当該端末装置又は当該画像形成装置のプラットフォームにより管理される他のアプリケーションを操作対象とするタスクバーを表示するタスクバー表示アプリケーションを、前記他のアプリケーションと同一のプロセス上で実行するステップを備えることを特徴とする請求項 2 2 に記載の情報処理方法である。

20

【 0 0 3 9 】

請求項 2 4 に記載の発明は、画像形成装置のエミュレータとして機能する端末装置内で実行するための情報処理プログラムであって、当該端末装置のプラットフォームにより管理されるアプリケーションとして備えられた、当該端末装置又は当該画像形成装置のプラットフォームにより管理される他のアプリケーションを操作対象とする操作画面を表示する操作画面表示アプリケーションと、前記他のアプリケーションにエラーが発生した場合に、エラーが発生した旨の情報を出力するエラー発生出力アプリケーションと、前記他のアプリケーションから、前記プラットフォームによって管理されるアプリケーションのインタフェースに対する処理要求に、予め決めておいた動作条件の許可範囲を超えるルール違反があった場合に、ルール違反があった旨の情報を出力するルール違反出力アプリケーションと、を、前記他のアプリケーションと同一のプロセス上で実行させることを特徴とする情報処理プログラムである。

30

【 0 0 4 0 】

請求項 2 5 に記載の発明は、前記プラットフォームにより管理されるアプリケーションとして備えられた、当該端末装置又は当該画像形成装置のプラットフォームにより管理される他のアプリケーションを操作対象とするタスクバーを表示するタスクバー表示アプリケーションを、前記他のアプリケーションと同一のプロセス上で実行させることを特徴とする請求項 2 4 に記載の情報処理プログラムである。

40

【 0 0 4 3 】

請求項 2 6 に記載の発明は、請求項 1 8 又は 1 9 に記載の情報処理プログラムを記録した記録媒体である。

【 0 0 4 4 】

請求項 2 7 に記載の発明は、請求項 2 4 又は 2 5 に記載の情報処理プログラムを記録した記録媒体である。

【 0 0 4 5 】

請求項 1 に記載の発明は、アプリケーション管理機構として、アプリケーションのライフサイクル等を管理する「プラットフォーム」と、このプラットフォームによってライフサイクル等を管理されるアプリケーションを操作対象とする操作画面を表示する「操作画

50

面表示アプリケーション」とを備える。アプリケーションのライフサイクル等を「プラットフォーム」が管理し、このプラットフォームによってライフサイクル等を管理されるアプリケーションを操作対象とする操作画面を「操作画面表示アプリケーション」が提供するのである。そのため、アプリケーションの実装等が容易になる。操作画面表示アプリケーションも、このプラットフォームによってライフサイクル等を管理されるアプリケーションである。そのため、この操作画面表示アプリケーションの実装、入れ替え等も容易である。請求項7に記載の発明（タスクバー表示アプリケーション）や、請求項9に記載の発明（エラー発生出力アプリケーション）や、請求項10に記載の発明（ルール違反出力アプリケーション）に関しても同様の事が言える。

【0046】

請求項20に記載の発明は、アプリケーション管理機構として、端末装置又は画像形成装置のプラットフォームによりライフサイクル等を管理されるアプリケーションを操作対象とする操作画面を表示する「操作画面表示アプリケーション」を備える。端末装置又は画像形成装置のプラットフォームによりライフサイクル等を管理されるアプリケーションを操作対象とする操作画面を「操作画面表示アプリケーション」が提供するのである。そのため、アプリケーションの実装等が容易になる。操作画面表示アプリケーションも、端末装置のプラットフォームによりライフサイクル等を管理されるアプリケーションである。そのため、この操作画面表示アプリケーションの実装、入れ替え等も容易である。タスクバー表示アプリケーションや、エラー発生出力アプリケーションや、ルール違反出力アプリケーションに関しても同様の事が言える。

【発明の効果】

【0047】

このように、本発明は、新たなアプリケーション管理機構を備える画像形成装置を提案するものである。

【発明を実施するための最良の形態】

【0048】

図1は、本発明の実施例に該当する融合機101を表す。図1の融合機101は、種々のハードウェア111と、種々のソフトウェア112と、融合機起動部113により構成される。

【0049】

融合機101のハードウェア111としては、撮像部121と、印刷部122と、その他のハードウェア123が存在する。撮像部121は、読取原稿から画像（画像データ）を読み取るためのハードウェアである。印刷部122は、画像（画像データ）を印刷用紙に印刷するためのハードウェアである。

【0050】

融合機101のソフトウェア112としては、種々のアプリケーション131と、種々のプラットフォーム132が存在する。これらのプログラムは、UNIX（登録商標）等のOS（オペレーティングシステム）によりプロセス単位で並列的に実行される。

【0051】

アプリケーション131としては、コピー用のアプリケーションであるコピーアプリ141、プリンタ用のアプリケーションであるプリンタアプリ142、スキャナ用のアプリケーションであるスキャナアプリ143、ファクシミリ用のアプリケーションであるファクシミリアプリ144、ネットワークファイル用のアプリケーションであるネットワークファイルアプリ145が存在する。

【0052】

アプリケーション131は、専用のSDK（ソフトウェア開発キット）を使用して開発することができる。SDKを使用して開発したアプリケーション131をSDKアプリと呼ぶ。専用のSDKとしては、C言語でアプリケーション131を開発するための「CSDK」や、Java（登録商標）言語でアプリケーション131を開発するための「JS

10

20

30

40

50

DK」が提供される。CSDKを使用して開発したアプリケーション131を「CSDKアプリ」と呼び、JSDKを使用して開発したアプリケーション131を「JSDKアプリ」と呼ぶ。図1の融合機101にも、CSDKアプリ146と、JSDKアプリ147が存在する。図1の融合機101にはさらに、Java（登録商標）言語で記述されたJSDKアプリ147とC言語で記述された他のソフトウェア112との仲介を行うソフトウェア112として、JSDKプラットフォーム148が存在する。

【0053】

プラットフォーム132としては、種々のコントロールサービス151、システムリソースマネージャ152、種々のハンドラ153が存在する。コントロールサービス151としては、ネットワークコントロールサービス（NCS）161、ファクシミリコントロールサービス（FCS）162、デリバリコントロールサービス（DCS）163、エンジンコントロールサービス（ECS）164、メモリコントロールサービス（MCS）165、オペレーションパネルコントロールサービス（OCS）166、サートファイケーションコントロールサービス（CCS）167、ユーザディレクトリコントロールサービス（UCS）168、システムコントロールサービス（SCS）169が存在する。ハンドラ153としては、ファクシミリコントロールユニットハンドラ（FCUH）171、イメージメモリハンドラ（IMH）172が存在する。

【0054】

NCS161のプロセスは、ネットワーク通信の仲介を行う。FCS162のプロセスは、ファクシミリのAPIを提供する。DCS163のプロセスは、蓄積文書の配信処理に関する制御を行う。ECS164のプロセスは、撮像部121や印刷部122に関する制御を行う。MCS165のプロセスは、メモリやハードディスクドライブに関する制御を行う。OCS166のプロセスは、オペレーションパネルに関する制御を行う。CCS167のプロセスは、認証処理や課金処理に関する制御を行う。UCS168のプロセスは、ユーザ情報の管理に関する制御を行う。SCS169のプロセスは、システムの管理に関する制御を行う。

【0055】

アプリケーション131とプラットフォーム132の仲介を行うソフトウェア112として、仮想アプリケーションサービス（VAS）135が存在する。VAS135は、アプリケーション131をクライアントとするサーバプロセスとして動作すると共に、プラットフォーム132をサーバとするクライアントプロセスとして動作する。VAS135は、アプリケーション131から見てプラットフォーム132を隠蔽するラッピング機能を備え、プラットフォーム132のバージョンアップに伴うバージョン差を吸収する役割等を担う。

【0056】

融合機起動部113は、融合機101の電源投入時に最初に実行される。これにより、UNIX（登録商標）等のOSが起動され、アプリケーション131やプラットフォーム132が起動される。これらのプログラムは、ハードディスクドライブやメモリカードに蓄積されており、ハードディスクドライブやメモリカードから再生されて、メモリに起動されることになる。

【0057】

図2は、図1の融合機101に係るハードウェア構成図である。融合機101のハードウェア111としては、コントローラ201と、オペレーションパネル202と、ファクシミリコントロールユニット（FCU）203と、撮像部121と、印刷部122が存在する。

【0058】

コントローラ201は、CPU211、ASIC212、NB221、SB222、MEM-P231、MEM-C232、HDD（ハードディスクドライブ）233、メモリカードスロット234、NIC（ネットワークインタフェースコントローラ）241、USBデバイス242、IEEE1394デバイス243、センストロニクスデバイス244

10

20

30

40

50

により構成される。

【0059】

CPU211は、種々の情報処理用のICである。ASIC212は、種々の画像処理用のICである。NB221は、コントローラ201のノースブリッジである。SB222は、コントローラ201のサウスブリッジである。MEM-P231は、融合機101のシステムメモリである。MEM-C232は、融合機101のローカルメモリである。HDD233は、融合機101のストレージである。メモリカードスロット234は、メモリカード235をセットするためのスロットである。NIC241は、MACアドレスによるネットワーク通信のコントローラである。USBデバイス242は、USB規格の接続端子を提供するためのデバイスである。IEEE1394デバイス243は、IEEE1394規格の接続端子を提供するためのデバイスである。セントロニクスデバイス244は、セントロニクス仕様の接続端子を提供するためのデバイスである。

10

【0060】

オペレーションパネル202は、オペレータが融合機101に入力を行うためのハードウェア（操作部）であると共に、オペレータが融合機101から出力を得るためのハードウェア（表示部）である。

【0061】

図3は、図1の融合機101に係る外観図である。図3には、撮像部121の位置と、印刷部122の位置と、オペレーションパネル202の位置が図示されている。図3には更に、読取原稿のセット先となる原稿セット部301と、印刷用紙の給紙先となる給紙部302と、印刷用紙の排紙先となる排紙部303が図示されている。

20

【0062】

オペレーションパネル202は、図4のように、タッチパネル311と、テンキー312と、スタートボタン313と、リセットボタン314と、機能キー315と、初期設定ボタン316により構成される。タッチパネル311は、タッチ操作で入力を行うためのハードウェア（タッチ操作部）であると共に、画面表示で出力を得るためのハードウェア（画面表示部）である。テンキー312は、キー（ボタン）操作で数字入力を行うためのハードウェアである。スタートボタン313は、ボタン操作でスタート操作を行うためのハードウェアである。リセットボタン314は、ボタン操作でリセット操作を行うためのハードウェアである。機能キー315は、キー（ボタン）操作でCSDKアプリ146やJSDKアプリ147による操作画面を表示させるためのハードウェアである。初期設定ボタン316は、ボタン操作で初期設定画面を表示させるためのハードウェアである。

30

【0063】

原稿セット部301は、ADF（自動原稿搬送装置）321と、フラットベッド322と、フラットベッドカバー323により構成される。給紙部302は、4個の給紙トレイにより構成される。排紙部303は、1個の排紙トレイにより構成される。

【0064】

（JSDK）

図5は、図1のJSDKアプリ147とJSDKプラットフォーム148のクラス図である。JSDKアプリ147とJSDKプラットフォーム148は、全体で1プロセスとして、同一プロセス上で実行される。JSDKアプリ147とJSDKプラットフォーム148中の各ブロックは、それぞれこの1プロセス上のスレッドとして、スレッド単位で並列的に実行（マルチスレッド）される。JSDKアプリ147とJSDKプラットフォーム148は、Java（登録商標）コンパイラによりソースコードからバイトコードに一括翻訳されており、Java（登録商標）仮想マシンにより逐次実行される。JSDKアプリ147とJSDKプラットフォーム148は、Java（登録商標）2 Micro EditionのPersonal Basis Profileをベースとする実装となっている。

40

【0065】

JSDKアプリ147としては、ユーザアプリ501と、JSDK GUI Mana

50

ger511と、Task Bar Manager512と、エラーアプリ513と、ルールアプリ514等が存在する。

【0066】

ユーザアプリ501は、融合機101のユーザ（例えばベンダ）がJSDKを使用して開発したJSDKアプリである。JSDK GUI Manager511は、他のJSDKアプリ（ユーザアプリ501等）を操作対象とする操作画面の表示等を行うJSDKアプリである。Task Bar Manager512は、他のJSDKアプリ（ユーザアプリ501等）を操作対象とするタスクバーの表示等を行うJSDKアプリである。エラーアプリ513は、他のJSDKアプリ（ユーザアプリ501等）にエラーが発生した場合に、エラーが発生した旨の情報の表示や通知等を行うJSDKアプリである。ルールアプリ514は、他のJSDKアプリ（ユーザアプリ501等）の処理要求にルール違反があった場合に、ルール違反があった旨の情報の表示や通知等を行うJSDKアプリである。

10

【0067】

ユーザアプリ501はここでは、スタンドアロンアプリケーションやアプレットと並ぶJava（登録商標）アプリケーションであるXletである。JSDK GUI Manager511とTask Bar Manager512とエラーアプリ513とルールアプリ514はここでは、独自の拡張を施したXlet（XletEx）である。

【0068】

JSDKプラットフォーム148には、JSDK Main521と、JSDK Environment522と、Locale Manager523と、Xlet Manager531と、Multi Xlet Manager532と、JSDK Manager533と、Send Manager541と、Event Manager542と、System Event Manager543と、Panel Manager544と、Install Manager545と、Server/Client Manager546等のクラスが存在する。

20

【0069】

JSDK Main521は、JSDKシステムの起動設定を行うクラスである。JSDK Environment522は、JSDKシステムの起動環境設定を行うクラスである。Locale Manager523は、国際化対応（言語指定）を行うクラスである。

30

【0070】

Xlet Manager531は、1対1でXletを管理するクラスである。ここでは、7個のXletの状態に係る情報やライフサイクル等が1対1で7個のXlet Manager531によって管理される。Xletの状態に係る情報は、Xlet Manager531によって管理され、Xlet Manager531からXletExに通知され、XletExによって表示処理や通知処理に利用される。Multi Xlet Manager532は、全てのXlet Manager531を管理するクラスである。ここでは、7個のXlet Manager531のライフサイクル等が全て1個のMulti Xlet Manager532によって管理される。JSDK Manager533は、JSDKシステム全体を管理するクラスである。例えば、Multi Xlet Manager532, Send Manager541, Event Manager542, System Event Manager543, Panel Manager544, Install Manager545, Server/Client Manager546のライフサイクル等が、JSDK Manager533によって管理される。

40

【0071】

System Event Manager543は、図1のプラットフォーム132からのシステムイベント（電力モード等）の管理を行うクラスである。Panel Manager544は、1個のXletがオペレーションパネル202の画面を占有する際

50

の調停等を行うクラスである。Install Manager 545は、SD card やWebからのインストールやアンインストールの管理を行うクラスである。

【0072】

図5のJSDKシステムでは、APIとして、JSDK API 551とJSDK API 552が利用される。なお、XletとXletExの差異として、オブジェクトにアクセスするのにJSDK API 551の利用とJSDKプラットフォーム148へのアクセスが可能である点が挙げられる。図1の融合機101にはさらに、図5のJSDKシステムに係る要素として、C言語とJava（登録商標）言語のインタフェースとなるJSDK Session 553とNative JSDK Session 554や、JSDKアプリ147とJSDKプラットフォーム148を実行するためJava（登録商標）仮想マシンであるCVM 555等が存在する。

10

【0073】

図6は、JSDKシステムの起動手順について説明するための図である。JSDKシステムではまず、JSDKアプリ147を起動するmain（）関数を有するクラスであるJSDK Main 521が、JSDK Environment 522を生成（S1）する。続いて、JSDK Environment 522が、Native層の構築（S2）と言語環境の構築（S3）をもって、JSDKシステムの実行環境の構築を行う。続いて、JSDK Environment 522が、JSDK Manager 533を生成（S4）する。JSDKシステムではそして、JSDK Manager 533が、システム層のManagerを生成（S5-12）し、Multi Xlet Manager 532とXlet Manager 531を通じて、アプリ層のManagerを生成（S13-16）する。

20

【0074】

図7は、JSDKシステムのGUI（Graphical User Interface）について説明するための図である。融合機101の機能キー315（図4参照）を押すと、図6のS1-16が実行されて、図7のGUIが融合機101のタッチパネル311（図4参照）に表示される。図7のGUIは、ユーザアプリ501を操作対象とする操作画面であり、図6のS13で起動されたJSDK GUI Manager 511によって表示される。

【0075】

図7Aは、画面を占有させるユーザアプリ501の切替操作を行うための切替操作画面である。図7Bは、ユーザアプリ501の起動操作を行うための起動操作画面である。図7Cは、ユーザアプリ501の終了操作を行うための終了操作画面である。図7Dは、ユーザアプリ501のインストール操作を行うためのインストール操作画面である。図7Eは、ユーザアプリ501のアンインストール操作を行うためのアンインストール操作画面である。図7Aが、機能キー315を押した直後にタッチパネル311に表示される初期画面であり、図7A、B、C、D、Eはそれぞれ切り替え、起動、終了、インストール、アンインストールのボタンをタッチすると表示されることになる。

30

【0076】

Simple Print, Simple Copy, Simple Ocr, Simple Scan等はそれぞれユーザアプリ501である。例えば、Simple Printは、電子文書のファイル名を入力することで当該電子文書のプリントを実行できるようにするためのアプリケーションである。なお、アプリケーションのタイプとは、GUIを有するアプリケーションであるか否かを意味する。

40

【0077】

図7Aの切替操作画面について説明する。図7Aの切替操作画面には、切替操作対象としてインストール済みで実行中のユーザアプリ501の一覧が表示される。

【0078】

図7Aの切替操作画面で「Simple Print」にタッチすると、Simple Printにオペレーションパネル202のオーナー権が付与されることになり、Simple

50

lePrintが画面を占有することになる。これに伴って、図7Aの画面から図8Aの画面に画面が移る。図8Aは、画面を占有するSimplePrintにより表示されるGUIである。

【0079】

図7Aの切替操作画面で「SimpleCopy」にタッチすると、SimpleCopyにオペレーションパネル202のオーナー権が付与されることになり、SimpleCopyが画面を占有することになる。これに伴って、図7Aの画面から図8Bの画面に画面が移る。図7Aに示すように、SimpleCopyの状態は「異常」なので、図8Bのように、画面上にはSimpleCopyに係るエラーメッセージが表示される。

【0080】

図7Aの切替操作画面で「SimpleOcr」にタッチすると、SimpleOcrにオペレーションパネル202のオーナー権が付与されることになり、SimpleOcrが画面を占有することになる。これに伴って、図7Aの画面から図8Cの画面に画面が移る。図7Aに示すように、SimpleOcrのタイプは「GUI無」なので、図8Cのように、画面上にはSimpleOcrの状態が表示される。

【0081】

図8A, B, Cの画面の左端(楕円形で示す領域内)には、図6のS14で起動されたTask Bar Manager 512により、ユーザアプリ501を操作対象とするタスクバーが表示される。これらのタスクバーは、画面を占有させるユーザアプリ501の切替操作を行うための切替タスクバーである。

【0082】

図8Aの画面で切替タスクバー中の「SC」や「SO」にタッチすると、SimpleCopyやSimpleOcrにオペレーションパネル202のオーナー権が付与されることになり、SimpleCopyやSimpleOcrが画面を占有することになる。これに伴い、図8Aの画面から図8Bや図8Cの画面に画面が移る。図8Aの画面で切替タスクバー中の「JSDK」にタッチすると、図7Aの画面に画面が戻る。

【0083】

図8Bの画面で切替タスクバー中の「SP」や「SO」にタッチすると、SimplePrintやSimpleOcrにオペレーションパネル202のオーナー権が付与されることになり、SimplePrintやSimpleOcrが画面を占有することになる。これに伴い、図8Bの画面から図8Aや図8Cの画面に画面が移る。図8Bの画面で切替タスクバー中の「JSDK」にタッチすると、図7Aの画面に画面が戻る。

【0084】

図8Cの画面で切替タスクバー中の「SP」や「SC」にタッチすると、SimplePrintやSimpleCopyにオペレーションパネル202のオーナー権が付与されることになり、SimplePrintやSimpleCopyが画面を占有することになる。これに伴い、図8Cの画面から図8Aや図8Bの画面に画面が移る。図8Cの画面で切替タスクバー中の「JSDK」にタッチすると、図7Aの画面に画面が戻る。

【0085】

図7Bの起動操作画面について説明する。図7Bの起動操作画面には、起動操作対象としてインストール済みで実行されていないユーザアプリ501の一覧が表示される。

【0086】

図7Bの起動操作画面で「SimplePrint」にタッチすると、SimplePrintの起動処理を実行するか否かを確認する確認画面(図9)が表示される。図9の確認画面でプロダクトコードを入力し「実行」にタッチすると、SimplePrintが起動されることになる。

【0087】

図7Cの終了操作画面について説明する。図7Cの終了操作画面には、終了操作対象としてインストール済みで実行中のユーザアプリ501の一覧が表示される。

【0088】

10

20

30

40

50

図7Cの終了操作画面で「SimplePrint」にタッチすると、SimplePrintの強制終了処理を実行するか否かを確認する確認画面(図10)が表示される。図10の確認画面で「実行」にタッチすると、SimplePrintが強制終了されることになる。

【0089】

図7Dのインストール操作画面について説明する。ユーザアプリ501のインストール処理により、起動可能ユーザアプリとして未登録のユーザアプリ501が起動可能ユーザアプリとして登録されることになる。図7Dのインストール操作画面には、インストール操作対象としてユーザアプリ501の一覧がインストール状況と共に表示される。

【0090】

図7Dのインストール操作画面で「SimplePrint」にタッチすると、SDcardを格納元とするSimplePrintのインストール先を確認する確認画面(図11)が表示される。図11の確認画面で「SDcard」にタッチすると、SimplePrintがSDcardをインストール先としてインストールされることになり、図11の確認画面で「HDD」にタッチすると、SimplePrintがHDDをインストール先としてインストールされることになる。

【0091】

図7Eのアンインストール操作画面について説明する。ユーザアプリ501のアンインストール処理により、起動可能ユーザアプリとして登録済みのユーザアプリ501が起動可能ユーザアプリとしての登録を抹消されることになる。図7Eのアンインストール操作画面には、アンインストール操作対象として登録済みのユーザアプリ501の一覧が表示される。

【0092】

図7Eのアンインストール操作画面で「SimplePrint」にタッチすると、SimplePrintのアンインストール処理を実行するか否かを確認する確認画面(図12)が表示される。図12の確認画面で「実行」にタッチすると、SimplePrintがアンインストールされることになる。

【0093】

図13は、起動操作画面の表示手順について説明するための図である。切替操作画面で起動のボタンがタッチされると、JSDK GUI Manager 511からInstall Manager 545に、インストール済みのユーザアプリ501の一覧要求が送信(S1)される。これに応じて、Install Manager 545は、インストール済みのユーザアプリ501の格納位置に係る情報(融合機101内のNVRAMに格納されている)を元に、インストール済みのユーザアプリ501のJNLPファイルを取得(S2)する。融合機101にセットされたSDcardからは、当該SDcard内のユーザアプリ501のJNLPファイルが取得される。融合機101内のHDDからは、当該HDD内のユーザアプリ501のJNLPファイルが取得される。JNLPファイルは、ユーザアプリ501と1対1で対応しており、ユーザアプリ501の定義付けに係る情報を含んでいる。そして、インストール済みのユーザアプリ501のJNLPファイルの内容を元に、Install Manager 545からJSDK GUI Manager 511に、インストール済みのユーザアプリ501の一覧応答が送信(S3)される。これにより、切替操作画面から起動操作画面に画面が移ることになる。

【0094】

図14は、起動処理の実行手順について説明するための図である。起動操作画面で選択操作が実行されて確認画面で確認操作が実行されると、JSDK GUI Manager 511からJSDK Manager 533に、ユーザアプリ501の起動要求が送信(S1)される。これに応じて、JSDK Manager 533からMulti Xlet Manager 532に、Authentication Manager 547による認証処理を経由してから、ユーザアプリ501の起動要求が送信(S2)される。これに応じて、Multi Xlet Manager 532がXlet Manager

10

20

30

40

50

r 5 3 1を起動 (S 3) し、X l e t M a n a g e r 5 3 1がP a n e l M a n a g e r 5 4 4からルートウィンドウを取得 (S 4) し、X l e t M a n a g e r 5 3 1がユーザアプリ5 0 1を起動 (S 5) する。そして、M u l t i X l e t M a n a g e r 5 3 2からJ S D K M a n a g e r 5 3 3に、ユーザアプリ5 0 1の起動応答が送信 (S 6) される。そして、J S D K M a n a g e r 5 3 3からJ S D K G U I M a n a g e r 5 1 1に、ユーザアプリ5 0 1の起動応答が送信 (S 7) される。

【 0 0 9 5 】

図 1 5 は、インストール操作画面の表示手順について説明するための図である。切替操作画面上でインストールのボタンがタッチされると、J S D K G U I M a n a g e r 5 1 1からI n s t a l l M a n a g e r 5 4 5に、ユーザアプリ5 0 1の一覧要求が送信 (S 1) される。これに応じて、I n s t a l l M a n a g e r 5 4 5は、ユーザアプリ5 0 1のJ N L Pファイルを取得 (S 2) する。融合機 1 0 1にセットされたS D c a r dからは、当該S D c a r d内のユーザアプリ5 0 1のJ N L Pファイルが取得される。融合機 1 0 1とネットワークで接続されたWebサーバからは、当該Webサーバ内のユーザアプリ5 0 1のJ N L Pファイルが取得される。J N L Pファイルは、ユーザアプリ5 0 1と1対1で対応しており、ユーザアプリ5 0 1の定義付けに係る情報を含んでいる。そして、ユーザアプリ5 0 1のインストール状況に関する情報 (融合機 1 0 1内のN V R A Mに格納されている) や、ユーザアプリ5 0 1のJ N L Pファイルの内容を元にして、I n s t a l l M a n a g e r 5 4 5からJ S D K G U I M a n a g e r 5 1 1に、ユーザアプリ5 0 1の一覧応答が送信 (S 3) される。これにより、切替操作画面からインストール操作画面に画面が移ることになる。

【 0 0 9 6 】

図 1 6 は、インストール処理の実行手順について説明するための図である。インストール操作画面で選択操作が実行されて確認画面で確認操作が実行されると、J S D K G U I M a n a g e r 5 1 1からI n s t a l l M a n a g e r 5 4 5に、ユーザアプリ5 0 1のインストール要求が送信 (S 1) される。これに応じて、I n s t a l l M a n a g e r 5 4 5は、A u t h e n t i c a t i o n M a n a g e r 5 4 7による認証処理を経由してから、ユーザアプリ5 0 1をインストール (S 2) する。そして、I n s t a l l M a n a g e r 5 4 5からJ S D K G U I M a n a g e r 5 1 1に、ユーザアプリ5 0 1のインストール応答が送信 (S 3) される。

【 0 0 9 7 】

図 1 7 は、J N L Pファイルの構文の例を表す。J N L P (J a v a (登録商標) N e t w o r k L a u n c h i n g P r o t o c o l) ファイルはX M L (e X t e n s i b l e M a r k u p L a n g u a g e) ファイルであり、J N L Pファイル書式はJ N L P規格に準拠している。ただし、J S D K向けに独自の拡張を施している部分があるため、以下その部分について説明する。

【 0 0 9 8 】

記述 1 は、i n f o r m a t i o n要素であり、アプリケーション名を示すt i t l e要素 (記述 1 A) や、ベンダ名を示すv e n d e r要素 (記述 1 B) や、ベンダの電話番号を示すt e l e p h o n e要素 (記述 1 C) や、ベンダのファクス電話番号を示すf a x要素 (記述 1 D) や、アプリケーションの製品IDを示すp r o d u c t I D要素 (記述 1 E) を含んでいる。

【 0 0 9 9 】

記述 2 は、s e c u r i t y要素である。

【 0 1 0 0 】

記述 3 は、r e s o u r c e要素であり、J S D Kのバージョンを指定するj s d k要素 (記述 3 A) や、J A Rファイル (アプリケーションの実行ファイル) とそのバージョンを指定するj a r要素 (記述 3 B) や、S U B - J N L Pファイルを指定するs u b - j n l p要素 (記述 3 C) を含んでいる。

【 0 1 0 1 】

10

20

30

40

50

記述4は、updateエレメントであり、更新処理の実行方法を設定するエレメントである。autoであればアプリケーションの更新処理は自動更新で実行され、manualであればアプリケーションの更新処理は手動更新で実行され、mailであればアプリケーションの更新処理が実行可能である場合にその旨を通知する更新通知メールが配信される。

【0102】

記述5は、installエレメントであり、インストール処理の実行方法を設定するエレメントである。autoであればアプリケーションのインストール先は自動選択にて選択され、manualであればアプリケーションのインストール先は手動選択にて選択される。

10

【0103】

記述6は、アプリケーションのタイプが「GUI有」であるか「GUI無」であるかを示す。

【0104】

図17のupdateエレメントで触れたように、インストール済みのJSDKアプリ147は更新処理の対象となる。以下、インストール済みのJSDKアプリ147を自動更新する自動更新処理について説明する。

【0105】

図18は、SDcard又はHDD内のJSDKアプリ147（更新対象アプリ）を、SDcardからのJSDKアプリ147（更新用アプリ）に自動更新する自動更新処理に係るフローチャートである。

20

【0106】

JSDKプラットフォーム148は、融合機101に接続された更新用SDcardスロットに更新用SDcardがセットされた場合、更新用SDcard内の更新用アプリのJNLPファイルを取得（S101）する。JSDKプラットフォーム148は、そのJNLPファイルのupdateエレメントが「auto」で、そのJNLPファイルに係る更新用アプリと同一の更新対象アプリが存在する場合（S102・S103）には、その更新用アプリのバージョンとその更新対象アプリのバージョンとを比較（S104）する。JSDKプラットフォーム148は、更新用アプリのバージョンが更新対象アプリのバージョンより新しい場合（S105）には、更新用アプリをもって更新対象アプリを更新（S106）する。

30

【0107】

図19は、SDcard又はHDD内のJSDKアプリ147（更新対象アプリ）を、WebからのJSDKアプリ147（更新用アプリ）に自動更新する自動更新処理に係るフローチャートである。

【0108】

JSDKプラットフォーム148は、更新対象アプリのJNLPファイルを取得（S201）する。JSDKプラットフォーム148は、そのJNLPファイルのupdateエレメントが「auto」で、そのJNLPファイルに係る更新対象アプリと同一の更新用アプリが、融合機101とネットワーク接続されたWebサーバ内に存在する場合（S202・S203）には、その更新対象アプリのバージョンとその更新用アプリのバージョンとを比較（S204）する。JSDKプラットフォーム148は、更新対象アプリのバージョンより更新用アプリのバージョンが新しい場合（S205）には、更新対象アプリを更新用アプリをもって更新（S206）する。

40

【0109】

最後に、図5の説明中で登場したXlet等のライフサイクルについて説明することにする。

【0110】

図20は、Xletの状態遷移図である。Xletの状態としては、初期化状態（Loaded）と、停止状態（Paused）と、活性化状態（Active）と、終了状態

50

(Destroyed)が存在する。Xletの状態は、起動(initXlet)や、開始(startXlet)や、停止(pauseXlet)や、終了(destroyXlet)等のメソッドコールによって遷移する。Xletのライフサイクルとは、このようなXletの状態遷移にほかならない。ただし、どのような状態が存在するか、どのようにして状態が遷移するか、については様々なバリエーションがあり得るだろう。このことは、XletManagerやMultiXletManagerその他の各Managerについても同様である。

【0111】

初期化状態は、Xletインスタンスの生成直後の状態である。初期化状態への遷移は1度限りである。停止状態は、サービス提供停止中の状態であり、活性化状態は、サービス提供中の状態である。停止状態と活性化状態との間の遷移は何度でも可能である。終了状態は、Xletインスタンスの消滅直後の状態である。終了状態への遷移は1度限りである。

10

【0112】

図5の説明中で記載したように、JSDKアプリ147のライフサイクルは、JSDKプラットフォーム148によって管理される。例えば、JSDKアプリ147は、自己の開始・停止・終了の際には自己の開始・停止・終了をJSDKプラットフォーム148に依頼するのである。JSDKアプリ147のライフサイクルをJSDKプラットフォーム148によって管理するメリットとしては例えば、メモリの節約が可能になることが挙げられる。特に、多数のJSDKアプリ147がそれぞれスレッドとして実行されるマルチスレッドにおいては、このメリットの存在価値は大きいと言える。JSDKアプリ147をXletとするメリットとしては例えば、JSDKアプリ147のライフサイクル管理が容易になることが挙げられる。

20

【0113】

1) エラーアプリ

図5のエラーアプリ513について説明する。

【0114】

エラーアプリ513は、他のJSDKアプリにエラーが発生した場合に、エラーが発生した旨の情報の表示を行うJSDKアプリである。エラーが発生した旨の情報は、図21Aのように、エラー発生表示ダイアログ601Aが融合機101のオペレーションパネル202の最前面に表示される形で表示される。エラー発生表示ダイアログ601Aには、エラー発生に係る情報として、エラーメッセージ611、エラーの種類612、エラーの発生箇所613等が表示される。OKボタン614を押下すると、エラー発生表示ダイアログ601Aが閉じる。Detailボタン615を押下すると、図21Bのエラー発生表示ダイアログ601Bに画面が移る。エラー発生表示ダイアログ601Bには、エラー発生に係る情報として、エラーの詳細616等が表示される。Closeボタン617を押下すると、エラー発生表示ダイアログ601Bが閉じる。これらの情報はネットワークを経由して外部機器に通知されるようにしてもよい。例えば、サービスセンタのサポートが可能になるように、サービスセンタに通知されるようにしてもよいだろう。

30

【0115】

エラーとしてここでは、Java(登録商標)のエクセプション(例外)を取り扱う。エラーアプリ513はここでは、他のJSDKアプリからスローされたエクセプションがキャッチされなかった場合に、スローされたエクセプションがキャッチされなかった旨の情報を表示する。エクセプションの授受にはここでは、Java(登録商標)のスレッドグループが利用される。スレッドグループには、スレッドグループに属するスレッドからスローされたエクセプションがキャッチされなかった場合に、Java(登録商標)仮想マシンによってThreadGroup.uncaughtExceptionメソッドがコールされると言う機能がある。スレッドグループには更に、各スレッドが各スレッドグループに属する他のスレッドに係る情報にアクセスできると言う機能がある。スレッドグループには更に、スレッドグループ間に親子関係(木構造)を設定できると言う機能が

40

50

ある。スレッドグループには更に、各スレッドグループの名称を登録できると言う機能がある。

【0116】

図22は、スレッドグループの設定処理に係るクラス図である。

【0117】

JSDK Manager 533は、Multi Xlet Manager 532等の各Managerを生成する際に、Multi Xlet Manager 532等の各Managerにスレッドグループを付与(S1)する。Multi Xlet Manager 532等の各Managerは、当該スレッドグループを使用してスレッドを生成する。これを利用することで、Multi Xlet Manager 532は、Xlet Manager 531のライフサイクル等を管理する。

10

【0118】

Multi Xlet Manager 532は、各Xlet Manager 531を生成する際に、各Xlet Manager 531にスレッドグループを付与(S2)する。各Xlet Manager 531は、当該スレッドグループを使用してスレッドを生成する。これを利用することで、Xlet Manager 531は、ユーザアプリ501等のXletの状態に係る情報やライフサイクル等を管理する。

【0119】

Xlet Manager 531は、ユーザアプリ501等のXletがサブスレッドを生成する場合に、ユーザアプリ501等のXletにスレッドグループを付与(S3)する。ユーザアプリ501等のXletは、当該スレッドグループを使用してスレッドを生成する。

20

【0120】

このようにして、スレッドグループの木構造が形成される。スレッドグループの木構造の頂点は、JSDK Manager 533により管理される。

【0121】

図23は、エラー発生表示処理に係るシーケンス図である。

【0122】

先ず、エラーアプリ513が起動した際(S11)には、エラーが発生した際にエラーが発生した旨をJSDK Manager 533からエラーアプリ513に通知する旨の通知登録が、エラーアプリ513からJSDK Manager 533に要求(S12)される。

30

【0123】

次に、エラーが発生した際(S21)には、CVM 555によってJSDK Manager 533のThreadGroup.uncaughtExceptionメソッドがコール(S22)される。次に、JSDK Manager 533の当該メソッドからエラーアプリ513に、エラーが発生した旨が通知登録に従って通知(S23)される。次に、エラーアプリ513からMulti Xlet Manager 532に、エラー発生表示ダイアログ601を最前面に表示する旨が要求(S24)される。次に、Multi Xlet Manager 532によって当該エラー発生表示ダイアログ601が最前面に表示(S25)される。

40

【0124】

2) ルールアプリ

図5のルールアプリ514について説明する。

【0125】

ルールアプリ514は、他のJSDKアプリの処理要求にルール違反があった場合に、ルール違反があった旨の情報の表示を行うJSDKアプリである。ルール違反があった旨の情報は、図24のように、ルール違反表示ダイアログ602が融合機101のオペレーションパネル202の最前面に表示される形式で表示される。ルール違反表示ダイアログ602には、ルール違反に係る情報として、ルール違反メッセージ621、ルール違反の

50

種類 6 2 2、ルール違反の箇所 6 2 3 等が表示される。OK ボタン 6 2 4 を押下すると、ルール違反表示ダイアログ 6 0 2 が閉じる。これらの情報はネットワークを經由して外部機器に通知されるようにしてもよい。

【 0 1 2 6 】

ルール違反としてここでは、J S D K アプリから J S D K A P I 5 5 1 へのサービス要求のルール違反を取り扱う。サービス要求のルール違反の具体例としては、スレッド数の制限に係るルール違反、ファイルオープン数の制限に係るルール違反、プリント処理に係るカラー/モノクロ設定や用紙サイズや用紙枚数等のルール違反、スキャン処理に係るカラー/モノクロ設定や用紙サイズや用紙枚数等のルール違反などが挙げられる。ルールアプリ 5 1 4 はここでは、他の J S D K アプリから J S D K A P I 5 5 1 へのサービス要求にルール違反があった場合に、ルール違反があった旨の情報を表示する。ルール違反の監視にはここでは、J a v a (登録商標) のスレッドグループが利用される。

10

【 0 1 2 7 】

図 2 5 は、ルール違反表示処理に係るシーケンス図である。

【 0 1 2 8 】

先ず、ルールアプリ 5 1 4 が起動した際 (S 3 1) には、ルール違反があった際にルール違反があった旨を M u l t i X l e t M a n a g e r 5 3 2 からルールアプリ 5 1 4 に通知する旨の通知登録が、ルールアプリ 5 1 4 から M u l t i X l e t M a n a g e r 5 3 2 に要求 (S 3 2) される。

20

【 0 1 2 9 】

次に、J S D K アプリから J S D K A P I 5 5 1 へのサービス要求があった際 (S 4 1) には、J S D K A P I 5 5 1 によってサービス要求元のスレッドグループの情報が取得 (S 4 2) される。次に、J S D K A P I 5 5 1 から M u l t i X l e t M a n a g e r 5 3 2 に、サービス要求元のスレッドにサービスを受ける権限があるかどうかの情報が要求 (S 4 3) される。次に、M u l t i X l e t M a n a g e r 5 3 2 によってサービス要求元のスレッドがスレッドグループに基づいて特定 (S 4 4) される。次に、M u l t i X l e t M a n a g e r 5 3 2 によってサービス要求元のスレッドにサービスを受ける権限があるかどうかの調査 (S 4 5) される。次に、サービス要求元のスレッドにサービスを受ける権限がない場合は、J S D K アプリから J S D K A P I 5 5 1 へのサービス要求にルール違反があった旨が、M u l t i X l e t M a n a g e r 5 3 2 からルールアプリ 5 1 4 に通知登録に従って通知 (S 4 6) される。次に、ルールアプリ 5 1 4 から M u l t i X l e t M a n a g e r 5 3 2 に、ルール違反表示ダイアログ 6 0 2 を最前面に表示する旨が要求 (S 4 7) される。次に、M u l t i X l e t M a n a g e r 5 3 2 によって当該ルール違反表示ダイアログ 6 0 2 が最前面に表示 (S 4 8) される。

30

【 0 1 3 0 】

3) 画像形成処理

図 2 6 は、融合機 1 0 1 で実行可能な画像形成処理について説明するための図である。融合機 1 0 1 では、画像形成処理用の画像データは、記憶領域が連続的になるような格納形式のファイルに格納されて H D D 内に記憶される。このようなファイルを「R A W ファイル」と呼ぶ。U N I X (登録商標) のファイルシステムでは、情報処理用のデータは、記憶領域が離散的になるような格納形式のファイルに格納されて H D D 内に記憶される。このようなファイルを「F F S ファイル」と呼ぶ。高速読み出しや高速書き込みの観点等では R A W ファイルの方が優れているが、融合機 1 0 1 から P C 等にファイルを提供する観点等では F F S ファイルの方が優れている。

40

【 0 1 3 1 】

図 2 6 A は、スキャン画像のプリント処理について説明するための図である。スキャン画像は、R A W ファイル 6 3 1 として H D D 内に蓄積される。プリント処理では、R A W ファイル 6 3 1 として H D D 内に蓄積された画像データがプリントされる。

【 0 1 3 2 】

50

図26Bは、スキャン画像の保存処理について説明するための図である。スキャン画像は、RAWファイル631としてHDD内に蓄積され、RAWファイル631からFFSファイル632に変換され、FFSファイル632としてHDD内に保存される。

【0133】

図26Cは、保存画像のプリント処理について説明するための図である。保存画像は、FFSファイル632としてHDD内に保存されている。プリント処理では、FFSファイル632としてHDD内に保存されている画像データがプリントされる。

【0134】

(エミュレータ)

図27は、本発明の実施例に該当するPC(パーソナルコンピュータ)701を表す。図27のPC701は、図1の融合機101とネットワーク801で接続されており、図1の融合機101のエミュレータとして機能する。

10

【0135】

図27のPC701は、PC本体711と、キーボード712と、マウス713と、ディスプレイ714等により構成される。PC本体711には、CPU、ROM、RAM、NVRAM、HDD、MODEM、NIC等が存在する。キーボード712とマウス713は、融合機101のオペレーションパネル202に代わって、オペレータが入力を行うためのハードウェア(操作部)となる。ディスプレイ714は、融合機101のオペレーションパネル202に代わって、オペレータが出力を得るためのハードウェア(表示部)となる。

20

【0136】

図28は、図27のPC701内のJSDKアプリ147とJSDKプラットフォーム148のクラス図である。図1の融合機101と同様のJSDKアプリ147とJSDKプラットフォーム148が、図27のPC701内にも存在するのである。図1の融合機101内のJSDKアプリ147とJSDKプラットフォーム148のクラス図については、図5を参照されたい。以下、図5と図28の差異について説明する。

【0137】

第1に、図5のPanel Manager 544は、図28のPanel Manager Emulator 744に置換されている。第2に、図5のJSDK Session 553は、図28のEmulator JSDK Session 753に置換されている。第3に、図5のNative JSDK Session 554は、図28のEvent Emulator 754に置換されている。第4には、図5のCVM 555は、図28のJVM 755に置換されている。

30

【0138】

Panel Manager Emulator 744は、Panel Manager 544をエミュレートして、オペレーションパネル202の操作をキーボード712やマウス713の操作に変換するクラスである。Emulator JSDK Session 753は、通信経路の確立処理を実行するクラスである。Event Emulator 754は、融合機101の動作をエミュレートするクラスである。JVM 755は、JSDKアプリ147とJSDKプラットフォーム148を実行するためJava(登録商標)仮想マシンである。

40

【0139】

なお、PC701内のJSDKシステムの起動手順は、図6と同様である。また、PC701内のJSDKシステムのGUIは、図7-12と同様である。当該GUIは、融合機101又はPC701内のユーザアプリ501を操作対象とする操作画面乃至タスクバーであり、PC701内のJSDK GUI Manager 511乃至Task Bar Manager 512により表示される。操作部はキーボード712とマウス713であり、表示部はディスプレイ714である。

【0140】

図29は、図1の融合機101内で実行される遠隔操作の準備処理の実行手順について

50

説明するための図である。

【0141】

最初に、融合機101内のServer/Client Manager 546とPC701内のServer/Client Manager 546との間で、セッションが確立(S1)される。

【0142】

次に、融合機101内のServer/Client Manager 546は、遠隔操作に移行できるか否かを融合機101内のJSDK Manager 533に問い合わせ、遠隔操作に移行できるか否かを判断(S2)する。

【0143】

次に、融合機101内のServer/Client Manager 546は、遠隔操作に移行できる場合には、遠隔操作に必要なメッセージに係る要求を、融合機101内のSend Manager 541を介してPC701に送信(S3)して、遠隔操作に必要なメッセージに係る応答を、融合機101内のEvent Manager 542を介してPC701から受信(S4)する。

【0144】

次に、融合機101内のServer/Client Manager 546は、遠隔操作の情報をPC701に転送する旨の要求を、Panel Manager 544に送信(S5)する。

【0145】

最後に、融合機101内のServer/Client Manager 546とPC701内のServer/Client Manager 546との間で、遠隔操作準備処理が完了した旨の通知が授受(S6)される。これによって、PC701からの融合機101の遠隔操作が可能になる。

【0146】

(変形例)

図1の融合機101は、本発明(画像形成装置)の実施例に該当し、図1の融合機101内で実行される情報処理は、本発明(情報処理方法)の実施例に該当する。図5のJSDK GUI Manager 511やTask Bar Manager 512やエラーアプリ513やルールアプリ514はそれぞれ、本発明(情報処理プログラム)の実施例に該当し、図5のJSDK GUI Manager 511やTask Bar Manager 512やエラーアプリ513やルールアプリ514を記録したSD cardやCD-ROMはそれぞれ、本発明(記録媒体)の実施例に該当する。これらの利用態様としては、これら情報処理プログラムのダウンロードやこれら記録媒体のオプション販売等が考えられる。

【0147】

図27のPC701は、本発明(端末装置)の実施例に該当し、図27のPC701内で実行される情報処理は、本発明(情報処理方法)の実施例に該当する。図28のJSDK GUI Manager 511やTask Bar Manager 512やエラーアプリ513やルールアプリ514はそれぞれ、本発明(情報処理プログラム)の実施例に該当し、図28のJSDK GUI Manager 511やTask Bar Manager 512やエラーアプリ513やルールアプリ514を記録したSD cardやCD-ROMはそれぞれ、本発明(記録媒体)の実施例に該当する。これらの利用態様としては、これら情報処理プログラムのダウンロードやこれら記録媒体のオプション販売等が考えられる。

【図面の簡単な説明】

【0148】

【図1】本発明の実施例に該当する融合機を表す。

【図2】図1の融合機に係るハードウェア構成図である。

【図3】図1の融合機に係る外観図である。

10

20

30

40

50

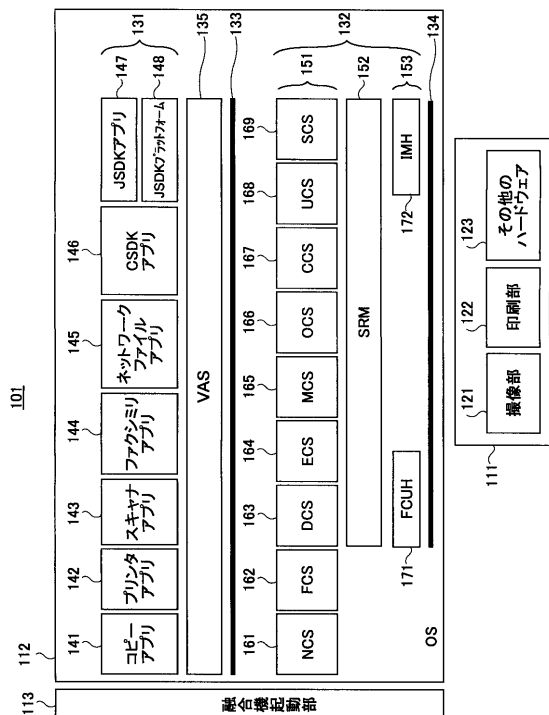
- 【図4】オペレーションパネルを表す。
- 【図5】図1のJSDKアプリとJSDKプラットフォームのクラス図である。
- 【図6】JSDKシステムの起動手順について説明するための図である。
- 【図7】JSDKシステムのGUIについて説明するための図である。
- 【図8】切替操作画面について説明するための図である。
- 【図9】起動操作画面について説明するための図である。
- 【図10】終了操作画面について説明するための図である。
- 【図11】インストール操作画面について説明するための図である。
- 【図12】アンインストール操作画面について説明するための図である。
- 【図13】起動操作画面の表示手順について説明するための図である。 10
- 【図14】起動処理の実行手順について説明するための図である。
- 【図15】インストール操作画面の表示手順について説明するための図である。
- 【図16】インストール処理の実行手順について説明するための図である。
- 【図17】JNLPファイルの構文の例を表す。
- 【図18】JSDKアプリをSDcardからのJSDKアプリに自動更新する自動更新処理に係るフローチャートである。
- 【図19】JSDKアプリをWebからのJSDKアプリに自動更新する自動更新処理に係るフローチャートである。
- 【図20】Xletの状態遷移図である。 20
- 【図21】エラー発生表示ダイアログの画面図である。
- 【図22】スレッドグループの設定処理に係るクラス図である。
- 【図23】エラー発生表示処理に係るシーケンス図である。
- 【図24】ルール違反表示ダイアログの画面図である。
- 【図25】ルール違反表示処理に係るシーケンス図である。
- 【図26】融合機で実行可能な画像形成処理について説明するための図である。
- 【図27】本発明の実施例に該当するPCを表す。
- 【図28】PC内のJSDKアプリとJSDKプラットフォームのクラス図である。
- 【図29】融合機内で実行される遠隔操作の準備処理の実行手順について説明するための図である。
- 【符号の説明】 30
- 【0149】
- 101 融合機
 - 111 ハードウェア
 - 112 ソフトウェア
 - 113 融合機起動部
 - 121 撮像部
 - 122 印刷部
 - 123 その他のハードウェア
 - 131 アプリケーション
 - 132 プラットフォーム 40
 - 133 アプリケーションプログラムインタフェース
 - 134 エンジンインタフェース
 - 135 仮想アプリケーションサービス
 - 141 コピーアプリ
 - 142 プリンタアプリ
 - 143 スキャナアプリ
 - 144 ファクシミリアプリ
 - 145 ネットワークファイルアプリ
 - 146 CSDKアプリ
 - 147 JSDKアプリ 50

1 4 8	J S D K プラットフォーム	
1 5 1	コントロールサービス	
1 5 2	システムリソースマネージャ	
1 5 3	ハンドラ	
1 6 1	ネットワークコントロールサービス	
1 6 2	ファクシミリコントロールサービス	
1 6 3	デリバリコントロールサービス	
1 6 4	エンジンコントロールサービス	
1 6 5	メモリコントロールサービス	
1 6 6	オペレーションパネルコントロールサービス	10
1 6 7	サーティフィケーションコントロールサービス	
1 6 8	ユーザディレトリコントロールサービス	
1 6 9	システムコントロールサービス	
1 7 1	ファクシミリコントロールユニットハンドラ	
1 7 2	イメージメモリハンドラ	
2 0 1	コントローラ	
2 0 2	オペレーションパネル	
2 0 3	ファクシミリコントロールユニット	
2 1 1	C P U	
2 1 2	A S I C	20
2 2 1	N B	
2 2 2	S B	
2 3 1	M E M - P	
2 3 2	M E M - C	
2 3 3	H D D	
2 3 4	メモリカードスロット	
2 3 5	メモリカード	
2 4 1	N I C	
2 4 2	U S B デバイス	
2 4 3	I E E E 1 3 9 4 デバイス	30
2 4 4	セントロニクスデバイス	
3 0 1	原稿セット部	
3 0 2	給紙部	
3 0 3	排紙部	
3 1 1	タッチパネル	
3 1 2	テンキー	
3 1 3	スタートボタン	
3 1 4	リセットボタン	
3 1 5	機能キー	
3 1 6	初期設定ボタン	40
3 2 1	A D F	
3 2 2	フラットベッド	
3 2 3	フラットベッドカバー	
5 0 1	ユーザアプリ	
5 1 1	J S D K G U I M a n a g e r	
5 1 2	T a s k B a r M a n a g e r	
5 1 3	エラーアプリ	
5 1 4	ルールアプリ	
5 2 1	J S D K M a i n	
5 2 2	J S D K E n v i r o n m e n t	50

5 2 3	Local e Manager	
5 3 1	Xlet Manager	
5 3 2	Multi Xlet Manager	
5 3 3	JSDK Manager	
5 4 1	Send Manager	
5 4 2	Event Manager	
5 4 3	System Event Manager	
5 4 4	Panel Manager	
5 4 5	Install Manager	
5 4 6	Server/Client Manager	10
5 4 7	Authentication Manager	
5 5 1	JSDK API	
5 5 2	JSDK API	
5 5 3	JSDK Session	
5 5 4	Native JSDK Session	
5 5 5	CVM	
6 0 1	エラー発生表示ダイアログ	
6 0 2	ルール違反表示ダイアログ	
6 1 1	エラーメッセージ	
6 1 2	エラーの種類	20
6 1 3	エラーの発生箇所	
6 1 4	OKボタン	
6 1 5	Detailボタン	
6 1 6	エラーの詳細	
6 1 7	Closeボタン	
6 2 1	ルール違反メッセージ	
6 2 2	ルール違反の種類	
6 2 3	ルール違反の箇所	
6 2 4	OKボタン	
6 3 1	RAWファイル	30
6 3 2	FFSファイル	
7 0 1	PC	
7 1 1	PC本体	
7 1 2	キーボード	
7 1 3	マウス	
7 1 4	ディスプレイ	
7 4 4	Panel Manager Emulator	
7 5 3	Emulator JSDK Session	
7 5 4	Event Emulator	
7 5 5	JVM	40
8 0 1	ネットワーク	

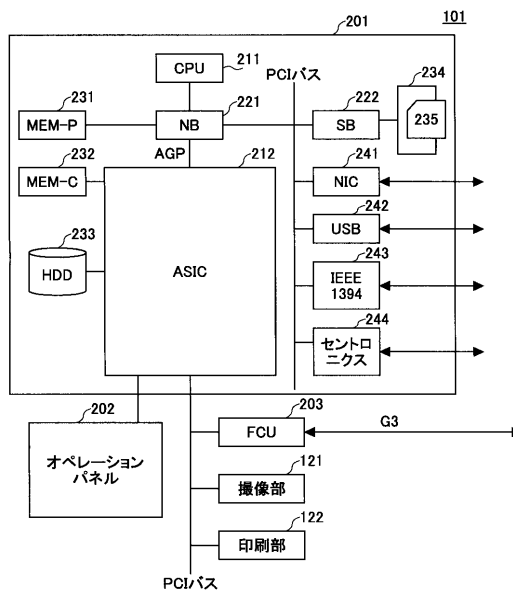
【図1】

本発明の実施例に該当する融合機を表す図



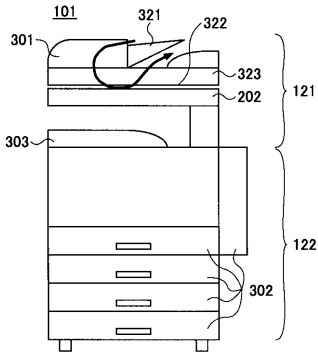
【図2】

図1の融合機に係るハードウェア構成図



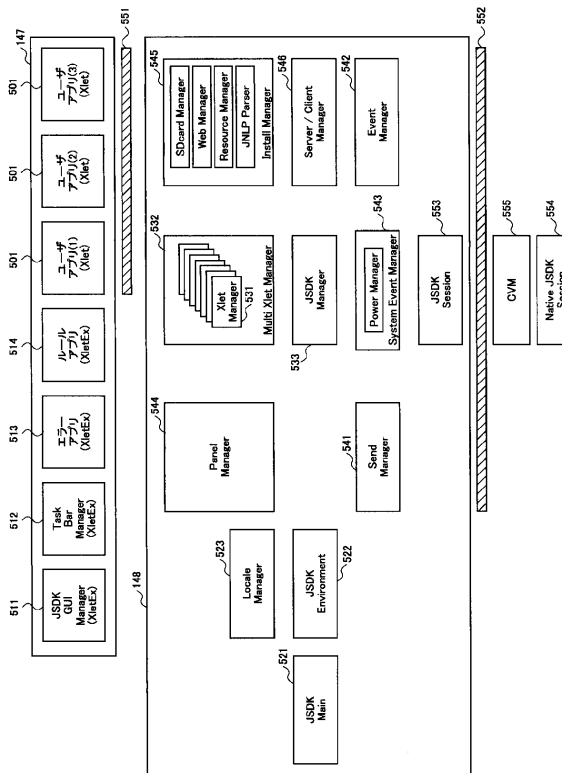
【図3】

図1の融合機に係る外観図



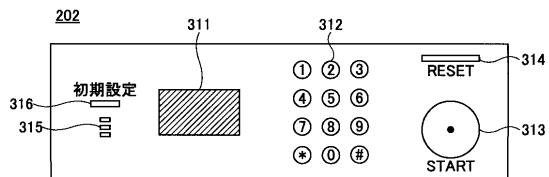
【図5】

図1のJSDKアプリとJSDKプラットフォームのクラス図



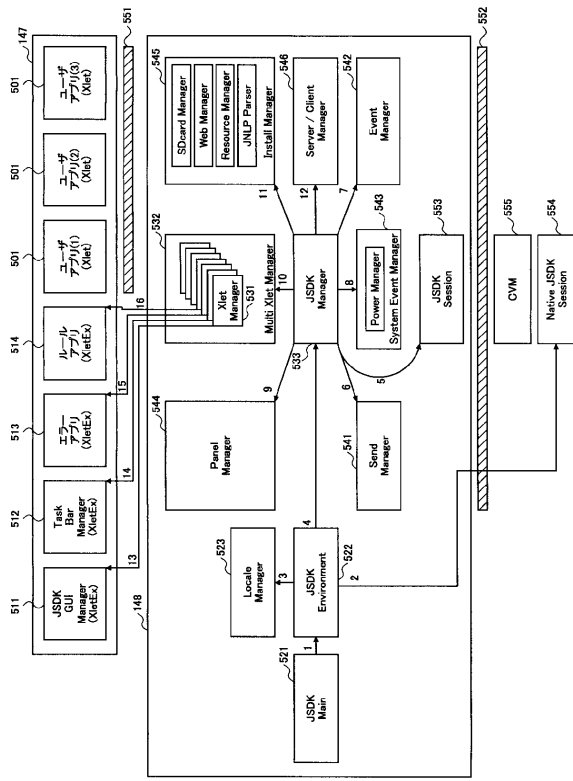
【図4】

オペレーションパネルを表す図



【図 6】

JSDKシステムの起動手順について説明するための図



【図 7】

JSDKシステムのGUIについて説明するための図

A

JSDKマネージャ

切り替え 起動 終了 設定 インストール アンインストール オプション

アプリケーション名	説明	タイプ	バージョン	格納位置	状態
SimplePrint	簡易プリント	GUI有	1.2.3	HDD	正常
SimpleCopy	簡易コピー	GUI有	1.0.0	HDD	正常
SimpleOcr	簡易OCR	GUI無	1.0.2	SDcard	正常

B

JSDKマネージャ

切り替え 起動 終了 設定 インストール アンインストール オプション

アプリケーション名	説明	タイプ	バージョン	格納位置	状態
SimplePrint	簡易プリント	GUI有	1.2.3	HDD	正常
SimpleCopy	簡易コピー	GUI有	1.0.0	HDD	正常
SimpleOcr	簡易OCR	GUI無	1.0.2	SDcard	正常

C

JSDKマネージャ

強制終了したいアプリケーションを選択してください。

切り替え 起動 終了 設定 インストール アンインストール オプション

状態	アプリケーション名	説明	タイプ	バージョン	格納位置	状態
正常	SimplePrint	簡易プリント	GUI有	1.2.3	HDD	
異常	SimpleCopy	簡易コピー	GUI有	1.0.0	HDD	
正常	SimpleOcr	簡易OCR	GUI無	1.0.2	SDcard	

D

JSDKマネージャ

インストールしたいアプリケーションを選択してください。

切り替え 起動 終了 設定 インストール アンインストール オプション

アプリケーション名	説明	タイプ	バージョン	格納位置	インストール状況
SimplePrint	簡易プリント	GUI有	1.2.3	SDcard	未
SimpleCopy	簡易コピー	GUI有	1.0.0	Web	済
SimpleScan	簡易スキャン	GUI無	1.0.2	SDcard	未

E

JSDKマネージャ

アンインストールしたいアプリケーションを選択してください。

切り替え 起動 終了 設定 インストール アンインストール オプション

アプリケーション名	説明	タイプ	バージョン	格納位置	状態
SimplePrint	簡易プリント	GUI有	1.2.3	HDD	
SimpleCopy	簡易コピー	GUI有	1.0.0	HDD	
SimpleOcr	簡易OCR	GUI無	1.0.2	SDcard	

【図 8】

切替操作画面について説明するための図

A

JSDK SimplePrint

SP

PRINT STOP

B

JSDK SimpleCopy

メモリ確保に失敗しました。

アプリケーション名 SimpleCopy
バージョン 1.0.0
格納位置 HDD

再実行 終了

C

JSDK SimpleOcr

アプリケーション名 SimpleOcr
バージョン 1.0.2
インストール先 SDcard
ベンダ名 RICOH
格納位置 TEL:03-xxxx-xxxx
FAX:03-xxxx-xxxx
EMAIL:xxx@xxx.co.jp

【図 9】

起動操作画面について説明するための図

JSDKマネージャ

強制終了したいアプリケーションを選択してください。

切り替え 起動 終了 設定 インストール アンインストール オプション

SimplePrintを起動します。

プロダクトコードを入力してください。

格納位置

アプリケーション名	説明	タイプ	バージョン	格納位置	状態
SimplePrint	簡易プリント	GUI有	1.2.3	HDD	
SimpleCopy	簡易コピー	GUI有	1.0.0	HDD	
SimpleOcr	簡易OCR	GUI無	1.0.2	SDcard	

クリア 取消 実行

【図 10】

終了操作画面について説明するための図

JSDKマネージャ

強制終了したいアプリケーションを選択してください。

切り替え 起動 終了 設定 インストール アンインストール オプション

SimplePrintを強制終了します。

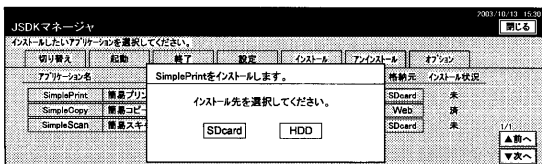
格納位置

アプリケーション名	説明	タイプ	バージョン	格納位置	状態
SimplePrint	簡易プリント	GUI有	1.2.3	HDD	
SimpleCopy	簡易コピー	GUI有	1.0.0	HDD	
SimpleOcr	簡易OCR	GUI無	1.0.2	SDcard	

取消 実行

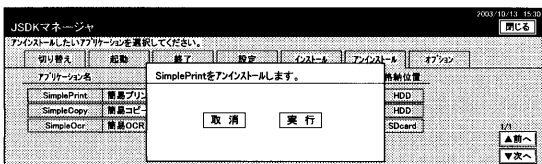
【図11】

インストール操作画面について説明するための図



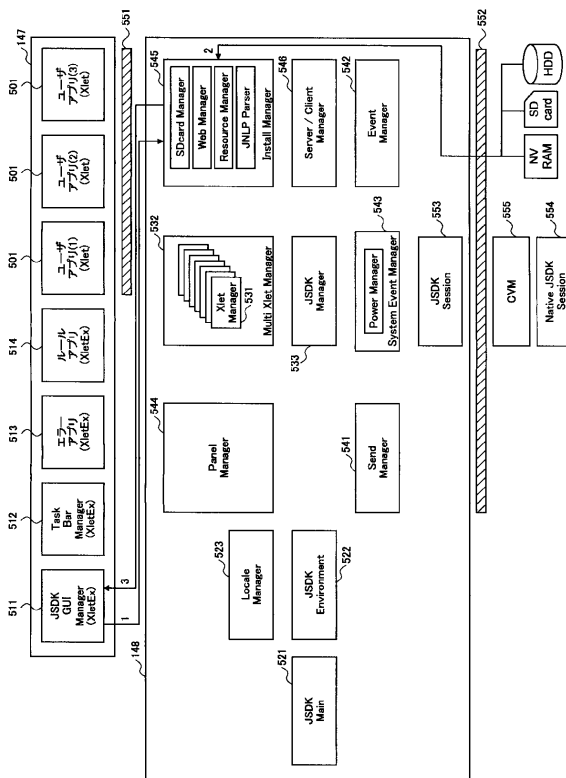
【図12】

アンインストール操作画面について説明するための図



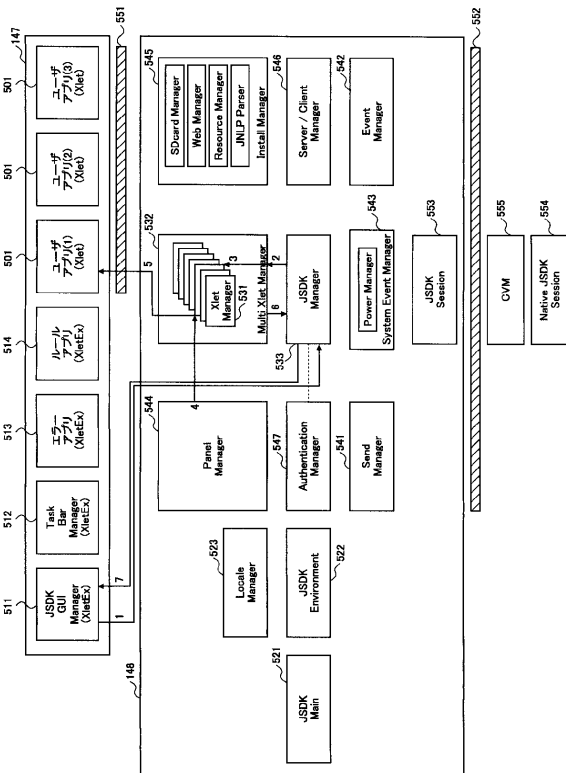
【図13】

起動操作画面の表示手順について説明するための図



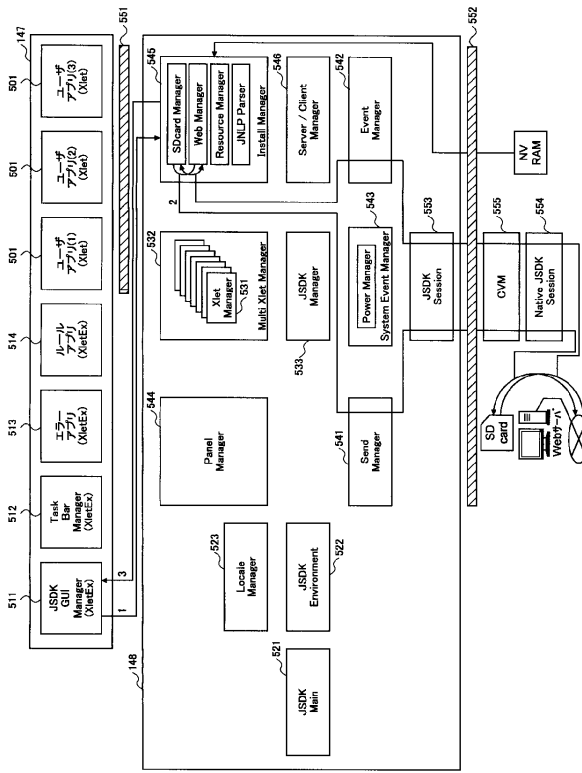
【図14】

起動処理の実行手順について説明するための図



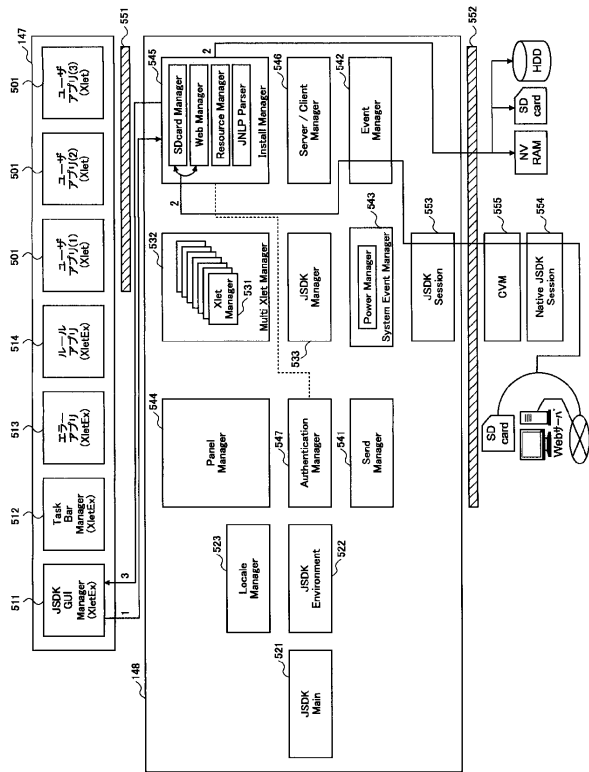
【図15】

インストール操作画面の表示手順について説明するための図



【図16】

インストール処理の実行手順について説明するための図



【図17】

JNLPファイルの構文の例を表す図

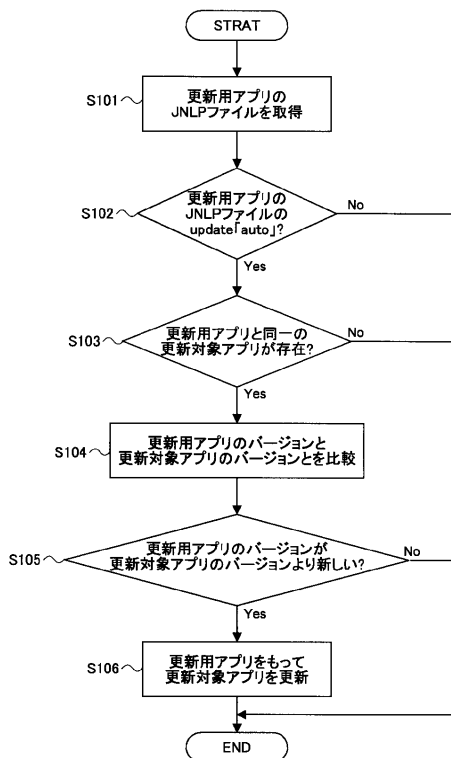
```

<?xml version="1.0" encoding="utf-8"?>
<!-- JNLP File for JSDK Demo Application -->
<jnlp
spec="1.0+"
codebase="sdcard://jsdk/0123"
href="jsdk_demo.jnlp">
  <information>
    <title>Jsdk Demo Application</title>
    <vendor>JSDK Providers Inc.</vendor>
    <telephone>092-441-####</telephone>
    <fax>092-411-####</fax>
    <product-id>0123456</product-id>
    <homepage href="docs/help.html"/>
    <description>JSDK Demo Application</description>
    <icon href="images/JSDK_demo.jpg"/>
    <offline-allowed/>
  </information>
  <security>
    <all-permissions/>
  </security>
  <resources>
    <jsdk version="1.0"/>
    <jar href="JSDKdemo.jar" version="1.0.1"/>
    <sub-jnlp href="http://www.ricoh.com/jsdk/ocr.jnlp">
  </resources>
  <update mode="mail">
    <mail address="xxx@yyy.zzz.co.jp"/>
  </update>
  <install mode="auto">
    <destination path="sdcard://jsdk/0123"/>
  </install>
  <xlet-desc class="JSDKdemo" type="noGUI"/>
</jnlp>

```

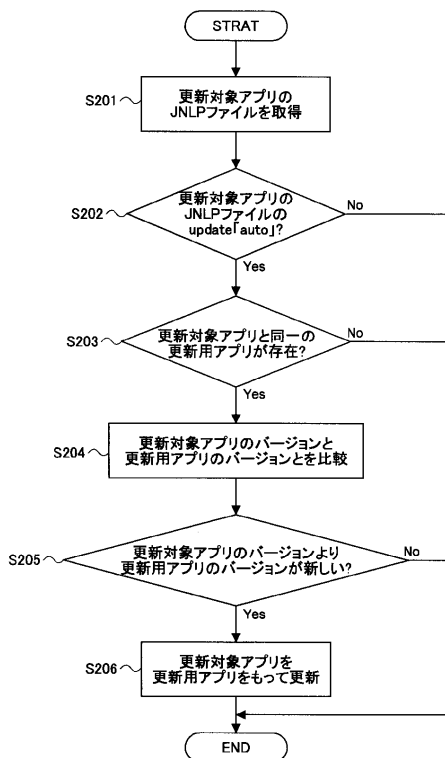
【図18】

JSDKアプリをSDcardからのJSDKアプリに自動更新する自動更新処理に係るフローチャート



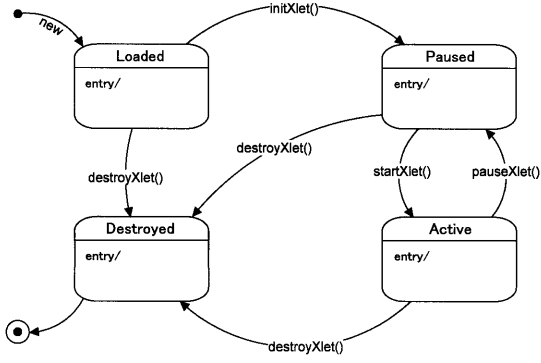
【図19】

JSDKアプリをWebからのJSDKアプリに自動更新する自動更新処理に係るフローチャート



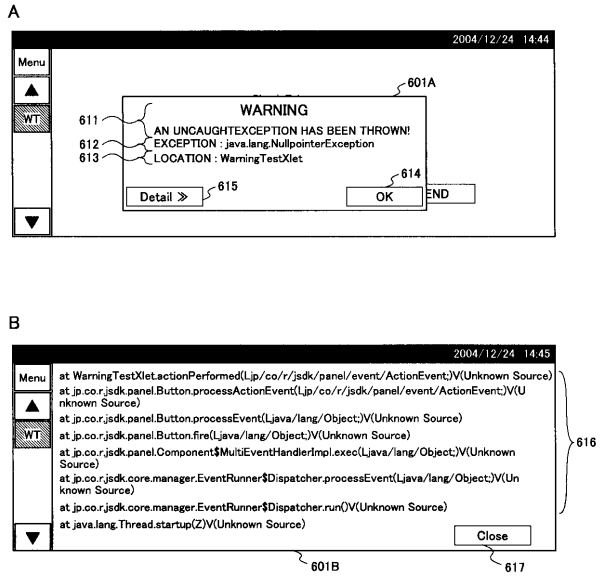
【図20】

Xletの状態遷移図



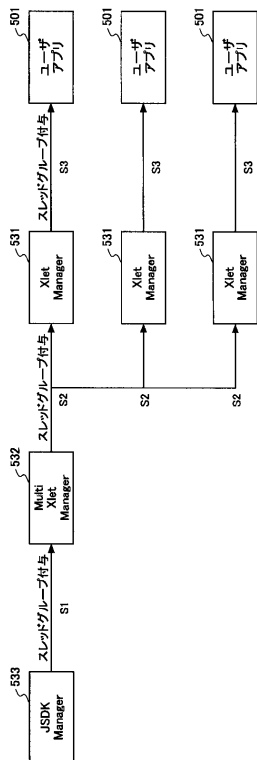
【図21】

エラー発生表示ダイアログの画面図



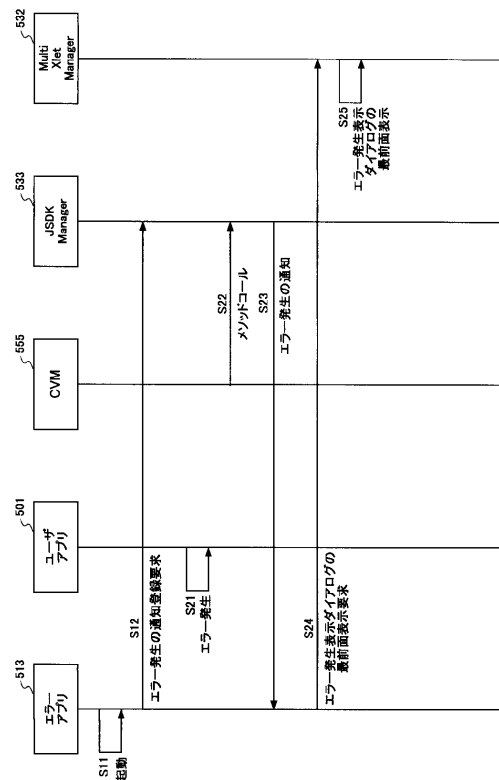
【図22】

スレッドグループの設定処理に係るクラス図



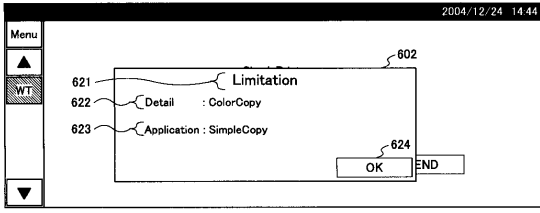
【図23】

エラー発生表示処理に係るシーケンス図



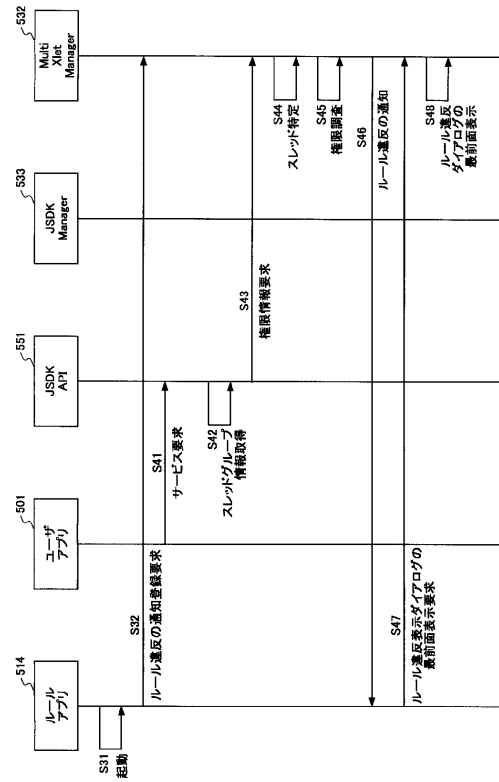
【図24】

ルール違反表示ダイアログの画面図



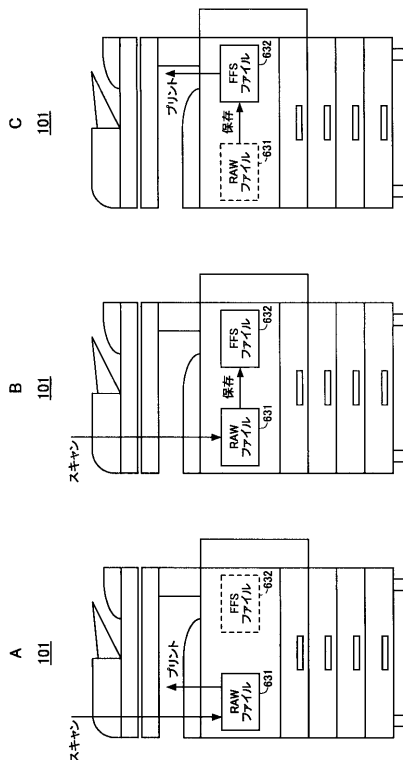
【図25】

ルール違反表示処理に係るシーケンス図



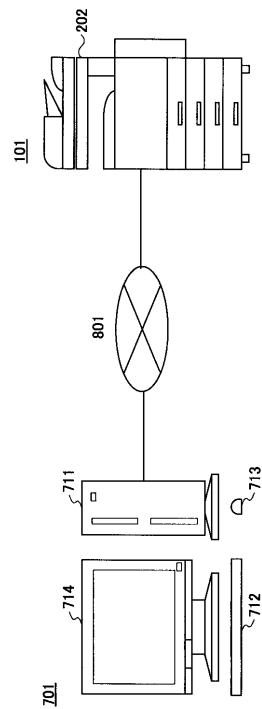
【図26】

融合機で実行可能な画像形成処理について説明するための図



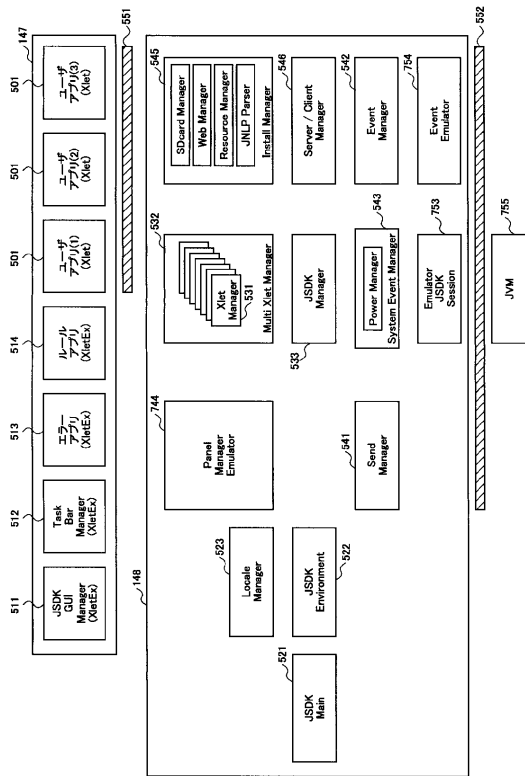
【図27】

本発明の実施例に該当するPCを表す図



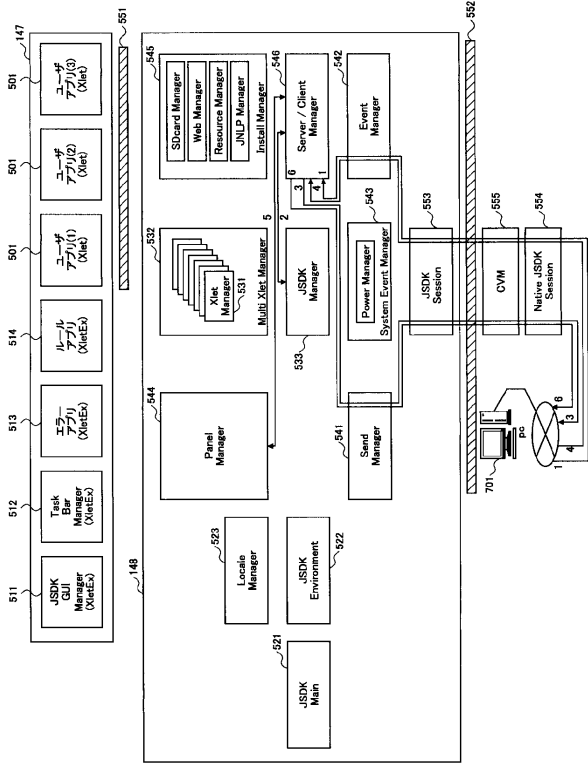
【 図 28 】

PC内のJSDKアプリとJSDKプラットフォームのクラス図



【 図 29 】

融合機内で実行される
連携操作の準備処理の実行手順について説明するための図



フロントページの続き

(56)参考文献 特開平06-037962(JP,A)
特開平10-021060(JP,A)
特開平11-282684(JP,A)
特開2002-082806(JP,A)
特開2000-211217(JP,A)
特開2002-084383(JP,A)
特開2002-215420(JP,A)
特開2002-244754(JP,A)
特開2003-330671(JP,A)
特開2003-182191(JP,A)
特開2004-312711(JP,A)
特開2008-079340(JP,A)
特開2008-278535(JP,A)
特開2009-189047(JP,A)
国際公開第03/101112(WO,A1)

(58)調査した分野(Int.Cl., DB名)

G03G 15/00
G03G 15/36
G03G 21/00 - 21/04
G03G 21/14
G03G 21/20
G06F 9/06
G06F 9/44 - 9/445
G06F 9/48 - 9/50
G06F 9/54
G06F 11/00
G06F 11/36
G06F 21/22
H04N 1/00