

(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2011-154472

(P2011-154472A)

(43) 公開日 平成23年8月11日(2011.8.11)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 9/54 (2006.01)	G06F 9/46 480Z	5E501
G06F 3/048 (2006.01)	G06F 3/048 651A	

審査請求 未請求 請求項の数 7 O L (全 18 頁)

(21) 出願番号 特願2010-14654 (P2010-14654)
 (22) 出願日 平成22年1月26日 (2010.1.26)

(特許庁注：以下のものは登録商標)

1. FLASH

(71) 出願人 000006747
 株式会社リコー
 東京都大田区中馬込1丁目3番6号
 (74) 代理人 100070150
 弁理士 伊東 忠彦
 (72) 発明者 櫻田 羊奈
 東京都大田区中馬込1丁目3番6号 株式
 会社リコー内
 Fターム(参考) 5E501 AC10 BA20 FB34

(54) 【発明の名称】 情報処理装置、プラグイン連携方法、プラグイン連携プログラム、及びそのプログラムを記録した記録媒体

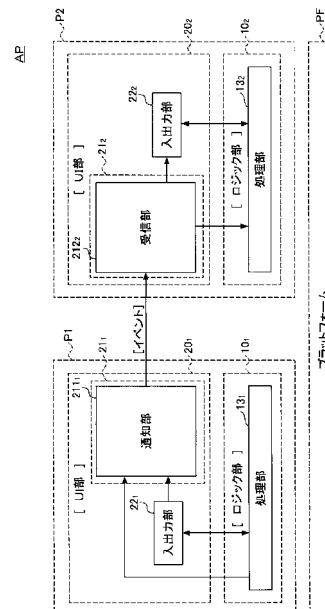
(57) 【要約】

【課題】プラグインが有するUI部とロジック部との独立性を保ち、プラグイン間の連携動作が可能な情報処理装置、プラグイン連携方法、プラグイン連携プログラム、及びそのプログラムを記録した記録媒体を提供する。

【解決手段】情報処理装置100は、複数のプラグインPを有するアプリケーションAPが動作する装置であって、プラグインPが、プラグインPの表示画面を生成するUI (User Interface) 部20と、プラグインPの機能を実現するための処理を行うロジック部10と、を有し、プラグイン間において、連携元のプラグインP1が、UI部20₁により、連携先のプラグインP2のUI部20₂に対して、ロジック部10₁による処理結果を通知し、連携先のプラグインP2が、UI部20₂により、通知手段211₁が通知した処理結果を受信し、受信した処理結果に基づき、同プラグインの表示画面を生成することを特徴とする。

【選択図】 図7

本発明の第1の実施形態に係る情報処理装置が有する機能構成例を示す図



【特許請求の範囲】**【請求項 1】**

複数のプラグインを有するアプリケーションが動作する情報処理装置であって、
前記プラグインが、プラグインの表示画面を生成する U I (User Interface) 部と、プラグインの機能を実現するための処理を行うロジック部と、を有し、
前記 U I 部が、同プラグインのロジック部による処理結果を他のプラグインに通知する通知手段、及び / 又は前記処理結果を他のプラグインから受信する受信手段、を有し、
プラグイン間において、
連携元のプラグインが、
前記通知手段により、連携先のプラグインの U I 部に対して、同プラグインのロジック部による処理結果を通知し、
前記連携先のプラグインが、
前記受信手段により、前記通知手段が通知した処理結果を受信し、受信した処理結果に基づき、同プラグインの U I 部により表示画面を生成することを特徴とする情報処理装置

10

【請求項 2】

前記通知手段は、
受信先として登録された連携先のプラグインが前記処理結果を取得可能な通知イベントを生成し、前記連携先のプラグインに対してブロードキャストすることを特徴とする請求項 1 に記載の情報処理装置。

20

【請求項 3】

前記受信手段は、
前記通知手段によるブロードキャストにより受信した通知イベントから、連携元のプラグインのロジック部による処理結果を取得することを特徴とする請求項 2 に記載の情報処理装置。

【請求項 4】

前記連携先のプラグインは、
前記通知手段により、前記連携元のプラグインに対して、同プラグインのロジック部による処理結果を通知し、
前記連携元のプラグインは、
前記受信手段により、前記通知手段が通知した処理結果を受信し、受信した処理結果に基づき、同プラグインの U I 部により表示画面を生成することを特徴とする請求項 1 ないし 3 のいずれか一項に記載の情報処理装置。

30

【請求項 5】

プラグインの表示画面を生成する U I 部と、プラグインの機能を実現するための処理を行うロジック部と、を有する複数のプラグインで構成されるアプリケーションが動作する情報処理装置におけるプラグイン連携方法であって、
プラグイン間において、
連携元のプラグインが、
同プラグインの U I 部により、連携先のプラグインの U I 部に対して、同プラグインのロジック部による処理結果を通知し、
前記連携先のプラグインが、
同プラグインの U I 部により、前記連携元のプラグインが通知した処理結果を受信し、受信した処理結果に基づき、同プラグインの表示画面を生成することを特徴とするプラグイン連携方法。

40

【請求項 6】

プラグインの表示画面を生成する U I 部と、プラグインの機能を実現するための処理を行うロジック部と、を有する複数のプラグインで構成されるアプリケーションが動作する情報処理装置におけるプラグイン連携プログラムであって、
コンピュータを、

50

プラグイン間において、
連携元のプラグインが、
同プラグインのUI部により、連携先のプラグインのUI部に対して、同プラグインのロジック部による処理結果を通知し、
前記連携先のプラグインが、
同プラグインのUI部により、前記連携元のプラグインが通知した処理結果を受信し、受信した処理結果に基づき、同プラグインの表示画面を生成するように動作させるためのプラグイン連携プログラム。

【請求項7】

請求項6に記載のプログラムを記憶した、コンピュータが読み取り可能な記録媒体。

10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、情報処理装置に関し、特に、複数のプラグインを有するアプリケーションにおいて、プラグインが画面遷移し、連携動作する技術に関するものである。

【背景技術】

【0002】

例えば、特許文献1には、プラグインごとの操作画面のカスタマイズ効率を高めることを目的とした表示画面カスタマイズプログラムに関する技術が開示されている。その中で、表示画面カスタマイズプログラムは、各プラグイン共通の画面構成情報と画面カスタマイズ情報を有し、これらの情報に基づき、各プラグインの表示画面を生成する。つまり、特許文献1に開示される技術では、表示画面の生成機能を複数のプラグインに対して共通化することで、プラグインごとの表示画面を統一し、カスタマイズ効率の向上を実現している。しかし、本来であれば、このようなプラグインの表示画面は、提供機能の利便性や利用者の操作性を考慮し、プラグインごとでカスタマイズしたい。

20

【0003】

また、近年のプラグインは、迅速な機能提供や優れた拡張性を実現するために、他の開発部署や他のソフトウェアベンダーと言った複数の異なる組織により開発・提供される場合が多い。このような環境下では、開発を行った組織単位で、プラグインの表示画面をカスタマイズした方が、利便性や操作性を考慮したUI (User Interface) の提供を実現するだけでなく、開発効率の面でも効果的である。

30

【0004】

そこで、プラグインのソフトウェア構成には、プラグインの開発効率や表示画面のカスタマイズ性を考慮し、表示画面を生成するUI部と、機能を実現するための処理を実行するロジック部と、を有する構成が考えられる。

【発明の概要】

【発明が解決しようとする課題】

【0005】

しかしながら、従来のプラグインには、他のプラグインと連携する際に次のような問題がある。従来のプラグインでは、ロジック部によりUI部の画面表示を制御している。そのため、プラグイン連携時の画面遷移が、連携プラグインの各ロジック部により制御される。例えば、プラグインAとプラグインBとが連携動作し、プラグインAの表示画面からプラグインBの表示画面へと遷移する場合を例に説明する。まず、プラグインAのロジック部がプラグインAのUI部から受け付けた利用者からの動作指示に従って所定の処理を実行する。その処理結果は、プラグインAのロジック部からプラグインBのロジック部へと通知され、画面遷移が指示される。プラグインBのロジック部は、通知された処理結果を基にプラグインBのUI部へ画面表示を指示する。これにより、プラグインAの表示画面からプラグインBの表示画面へと遷移する。

40

【0006】

上記動作例からも分かるように、従来のプラグインでは、動作連携時の画面遷移制御を

50

ロジック部が行っているため、ロジック部とUI部との独立性が保たれない構成となってしまう。また、従来のプラグインでは、ロジック部が実現する機能をプラグイン間で共通化したい場合であっても、画面遷移制御までは共通化できないため、異なるロジック部に同一機能が重複して存在することになる。

【0007】

このように、従来のプラグインでは、ソフトウェアの開発・保守を困難なものとし、リソースを無駄に消費してしまう可能性があった。

【0008】

本発明は上記従来技術の問題点を鑑み提案されたものであり、その目的とするところは、プラグインが有するUI部とロジック部との独立性を保ち、プラグイン間の連携動作が可能な情報処理装置、プラグイン連携方法、プラグイン連携プログラム、及びそのプログラムを記録した記録媒体を提供することにある。

10

【課題を解決するための手段】

【0009】

上記目的を達成するため、本発明に係る情報処理装置は、複数のプラグインを有するアプリケーションが動作する情報処理装置であって、前記プラグインが、プラグインの表示画面を生成するUI部と、プラグインの機能を実現するための処理を行うロジック部と、を有し、前記UI部が、同プラグインのロジック部による処理結果を他のプラグインに通知する通知手段、及び/又は前記処理結果を他のプラグインから受信する受信手段、を有し、プラグイン間において、連携元のプラグインが、前記通知手段により、連携先のプラグインのUI部に対して、同プラグインのロジック部による処理結果を通知し、前記連携先のプラグインが、前記受信手段により、前記通知手段が通知した処理結果を受信し、受信した処理結果に基づき、同プラグインのUI部により表示画面を生成することを特徴とする。

20

【0010】

このような構成によって、本発明に係る情報処理装置は、連携元のプラグインのUI部から連携先のプラグインのUI部に対してロジック部の処理結果を通知し、遷移後の画面表示を指示する。

【0011】

これによって、本発明に係る情報処理装置では、ロジック部でなくUI部により、プラグイン連携時の画面遷移が制御され、ロジック部による処理結果がUI部を介して連携先へと伝達される。その結果、プラグインが有するロジック部とUI部との独立性が保たれた構成でプラグイン間の連携動作が行える。

30

【0012】

上記目的を達成するため、本発明に係るプラグイン連携方法は、プラグインの表示画面を生成するUI部と、プラグインの機能を実現するための処理を行うロジック部と、を有する複数のプラグインで構成されるアプリケーションが動作する情報処理装置におけるプラグイン連携方法であって、プラグイン間において、連携元のプラグインが、同プラグインのUI部により、連携先のプラグインのUI部に対して、同プラグインのロジック部による処理結果を通知し、前記連携先のプラグインが、同プラグインのUI部により、前記連携元のプラグインが通知した処理結果を受信し、受信した処理結果に基づき、同プラグインの表示画面を生成することを特徴とする。

40

【0013】

このような手順によって、本発明に係るプラグイン連携方法では、連携元のプラグインのUI部から連携先のプラグインのUI部に対してロジック部の処理結果を通知し、遷移後の画面表示を指示するという動作を実現する。

【0014】

これによって、本発明に係るプラグイン連携方法は、プラグインが有するUI部とロジック部との独立性を保ち、プラグイン間の連携動作が可能な環境を提供できる。

【発明の効果】

50

【 0 0 1 5 】

本発明によれば、プラグインが有するUI部により、プラグイン連携時の画面遷移を制御し、ロジック部による処理結果をUI部が連携先へと伝達することで、ロジック部とUI部との独立性を保ち、プラグイン間の連携動作が可能な情報処理装置、プラグイン連携方法、プラグイン連携プログラム、及びそのプログラムを記録した記録媒体を提供することができる。

【 図面の簡単な説明 】

【 0 0 1 6 】

【 図 1 】 本発明の第 1 の実施形態に係る情報処理装置のハードウェア構成例を示す図である。

10

【 図 2 】 アプリケーションのソフトウェア構成例を示す図である。

【 図 3 】 従来のプラグイン連携の処理手順例を示すシーケンス図である。

【 図 4 】 従来のプラグイン連携を行うソフトウェア構成例を示す図である。

【 図 5 】 本発明の第 1 の実施形態に係るプラグイン連携を行うソフトウェア構成例を示す図である。

【 図 6 】 本発明の第 1 の実施形態に係るプラグイン連携の処理手順例を示すシーケンス図である。

【 図 7 】 本発明の第 1 の実施形態に係る情報処理装置が有する機能構成例を示す図である。

【 図 8 】 本発明の第 1 の実施形態に係るプラグイン連携の詳細な処理手順例を示すシーケンス図である。

20

【 図 9 】 本発明の変形例に係るプラグイン連携の動作例を示す図である。

【 図 1 0 】 本発明の変形例に係る情報処理装置が有する機能構成例を示す図である。

【 図 1 1 】 本発明の変形例に係るプラグイン連携の詳細な処理手順例を示すシーケンス図である。

【 発明を実施するための形態 】

【 0 0 1 7 】

以下、本発明の好適な実施の形態（以下「実施形態」と言う）について、図面を用いて詳細に説明する。

【 0 0 1 8 】

30

[第 1 の実施形態]

< ハードウェア構成 >

本実施形態に係る情報処理装置のハードウェア構成について説明する。

図 1 は、本実施形態に係る情報処理装置 1 0 0 のハードウェア構成例を示す図である。

図 1 に示すように、情報処理装置 1 0 0 は、入力装置 1 0 1、表示装置 1 0 2、ドライブ装置 1 0 3、R A M (Random Access Memory) 1 0 4、R O M (Read Only Memory) 1 0 5、C P U (Central Processing Unit) 1 0 6、インタフェース装置 1 0 7、及びH D D (Hard Disk Drive) 1 0 8などを備え、それぞれがバス B で相互に接続されている。

【 0 0 1 9 】

40

入力装置 1 0 1 は、キーボード及びマウスなどを含み、情報処理装置 1 0 0 に各操作信号を入力するのに用いられる。表示装置 1 0 2 は、ディスプレイなどを含み、情報処理装置 1 0 0 による処理結果を表示する。

【 0 0 2 0 】

インタフェース装置 1 0 7 は、情報処理装置 1 0 0 を所定のデータ伝送路（例えば「L A N : Local Area Network」）に接続するインタフェースである。よって、情報処理装置 1 0 0 は、インタフェース装置 1 0 7 を介して、接続される周辺機器 2 0 0（例えば「プリンタ」や「P C : Personal Computer」）とデータ通信を行うことができる。

【 0 0 2 1 】

H D D 1 0 8 は、各種プログラム及びデータを格納している不揮発性の記憶装置である

50

。格納されるプログラム及びデータには、例えば、情報処理装置 100 全体を制御する情報処理システム（「Windows（登録商標）」や「UNIX（登録商標）」などの基本ソフトウェアである OS : Operating System）、及び情報処理システム上において各種機能を提供するアプリケーションなどがある。また、HDD 108 は、格納している上記プログラム及びデータを、所定のファイルシステム及び / 又は DB（Data Base）により管理している。

【0022】

ドライブ装置 103 は、着脱可能な記録媒体 103 a とのインタフェースである。これにより、情報処理装置 100 は、ドライブ装置 103 を介して、記録媒体 103 a の読み取り及び / 又は書き込みを行うことができる。

10

【0023】

ROM 105 は、電源を切っても内部データを保持することができる不揮発性の半導体メモリ（記憶装置）である。ROM 105 には、情報処理装置 100 が起動されるときに実行される BIOS（Basic Input/Output System）や、情報処理装置 100 のシステム設定及びネットワーク関連設定などのデータが格納されている。

【0024】

RAM 104 は、上記各種記憶装置から読み出されたプログラム及びデータを一時保持する揮発性の半導体メモリ（記憶装置）である。CPU 106 は、ROM 105 や HDD 108 から RAM 104 上に読み出したプログラムを実行することにより、情報処理装置 100 の全体制御及び各種搭載機能の動作を実現する。

20

【0025】

情報処理装置 100 では、上記ハードウェア構成により、情報処理サービス（情報処理機能）を提供することができる。

【0026】

また、情報処理装置 100 は、上記ハードウェア構成からも分かるように PC（Personal Computer）などと略同一の構成をしている。

【0027】

<ソフトウェア構成>

次に、上記情報処理装置 100 に搭載される（インストールされる）アプリケーションのソフトウェア構成について説明する。

30

図 2 は、アプリケーション AP のソフトウェア構成例を示す図である。

図 2 に示すように、アプリケーション AP は、複数のプラグイン P1 ~ Pn（以下総称する場合は「プラグイン P」と言う）と、プラグイン P の動作環境であるプラットフォーム PF とを有する構成となっている。つまり、アプリケーション AP の機能は、各プラグイン P1 ~ Pn がプラットフォーム PF 上で連携動作することで実現・提供される。このようなソフトウェア構成とするメリットには、開発効率の向上や独立した開発部品（ソフトウェア部品）による迅速な機能提供、また優れた拡張性の実現などが挙げられる。

【0028】

さらに、各プラグイン P1 ~ Pn は、各機能を実現するための処理を行うプラグインロジック（以下単に「ロジック部」と言う）10₁ ~ 10_n（以下総称する場合は「ロジック部 10」と言う）と、各表示画面を生成するプラグイン UI（以下単に「UI部」と言う）20₁ ~ 20_n（以下総称する場合は「UI部 20」と言う）とを有する構成となっている。このようなソフトウェア構成とするメリットには、プラグイン P の開発効率や表示画面のカスタマイズ性の向上などが挙げられる。

40

【0029】

また、ロジック部 10 と UI 部 20 との関係は、次の通りである。例えば、ロジック部 10₁ と UI 部 20₁ とは、同じプラグイン（同プラグイン）に属するソフトウェア部品である。一方、ロジック部 10₂ と UI 部 20₁ とは、異なるプラグインに属するソフトウェア部品である。つまり、UI 部 20₁ から見たロジック部 10₂ は、他のプラグインに属するソフトウェア部品と言うことになる。この点については、ロジック部同士であっ

50

てもUI部同士であっても同じことが言える。よって、上記ソフトウェア構成におけるプラグイン連携とは、異なるプラグインPに属するソフトウェア部品同士が連携することを意味する。

【0030】

このようなソフトウェア構成において、従来のアプリケーションAPでは、画面遷移制御部11₁～11_n（以下総称する場合は「画面遷移制御部11」と言う）を、プラグインPのロジック部10が有している。

【0031】

<プラグイン連携機能>

まず、従来のアプリケーションAPにおけるプラグイン連携動作について説明する。

図3は、従来のプラグイン連携の処理手順例を示すシーケンス図である。図3には、4つのプラグインP1～P4を有するアプリケーションAPにおけるプラグイン連携の処理手順例が示されている。また、このアプリケーションAPでは、プラグインP1からプラグインP2への画面遷移、プラグインP3からプラグインP4への画面遷移において、利用者認証が必要となる。このようなアプリケーションAPを搭載した情報処理装置100では、アプリケーションAP内のプラグイン連携が次のように行われる。

【0032】

図3に示すように、まず、情報処理装置100では、プラグインP1のロジック部10₁からUI部20₁に対して、画面表示が指示される（ステップS11）。その結果、表示装置102には、プラグインP1の操作画面（認証画面：画面1）が表示される。情報処理装置100では、利用者入力で受け付けた認証情報（例えば「ユーザID」や「パスワード」）がUI部20₁からロジック部10₁へと渡される。

【0033】

情報処理装置100では、ロジック部10₁により、認証情報に基づく利用者認証処理が行われ（ステップS12）、その認証結果が、連携先であるプラグインP2のロジック部10₂へと通知され、プラグインP1からプラグインP2へと画面遷移（画面1 2）が指示される（ステップS13）。

【0034】

続いて、情報処理装置100では、プラグインP2のロジック部10₂からUI部20₂に対して、画面表示が指示される（ステップS14）。その結果、表示装置102には、認証結果が反映されたプラグインP2の操作画面（画面2）が表示される。情報処理装置100では、利用者入力で受け付けた操作情報（例えば「次処理指定」）がUI部20₂からロジック部10₂へと渡される。

【0035】

情報処理装置100では、ロジック部10₂により、必要に応じて所定の処理が行われ、その処理結果が、連携先であるプラグインP3のロジック部10₃へと通知され、プラグインP2からプラグインP3へと画面遷移（画面2 3）が指示される（ステップS15）。

【0036】

続いて、情報処理装置100では、プラグインP3のロジック部10₃からUI部20₃に対して、画面表示が指示される（ステップS16）。その結果、表示装置102には、プラグインP3の操作画面（認証画面：画面3）が表示される。情報処理装置100では、利用者入力で受け付けた認証情報がUI部20₃からロジック部10₃へと渡される。

【0037】

情報処理装置100では、ロジック部10₃により、認証情報に基づく利用者認証処理が行われ（ステップS17）、その認証結果が、連携先であるプラグインP4のロジック部10₄へと通知され、プラグインP3からプラグインP4へと画面遷移（画面3 4）が指示される（ステップS18）。

【0038】

10

20

30

40

50

続いて、情報処理装置 100 では、プラグイン P 4 のロジック部 10₄ から UI 部 20₄ に対して、画面表示が指示される（ステップ S 19）。その結果、表示装置 102 には、認証結果が反映されたプラグイン P 4 の操作画面（画面 4）が表示される。

【0039】

このように、従来のアプリケーション AP では、プラグイン連携処理を行う場合、図 4 に示すようなソフトウェア構成となってしまう。

【0040】

図 4 は、従来のプラグイン連携を行うソフトウェア構成例を示す図である。

図 4 に示すように、各プラグイン P 1 ~ P 4 が有するロジック部 10₁ ~ 10₄ は、連携先への画面遷移を制御する画面遷移制御部 11₁ ~ 11₄ を個々に有している。そのため、プラグイン P では、UI 部 20 とロジック部 10 との独立性が保たれず、連携先の変更を行う場合、ロジック部担当の開発者に必要以上の技術を要求することになる。つまり、ロジック部担当の開発者には、機能実現処理以外の画面処理に関する技術の理解も必要とされる。これは、開発者にとって煩雑であり、作業分担による高い開発効率を実現する上で弊害となる。

10

【0041】

また、プラグイン P 1 及び P 3 が有するロジック部 10₁ 及び 10₃ は、利用者認証を行う同一の機能（認証部 12₁ 及び 12₃）を有している。これは、ロジック部 10₁ 及び 10₃ が画面遷移制御部 11₁ 及び 11₃ を有しているため、画面遷移制御部 11₁ 及び 11₃ の動作時に利用者認証を行わなければならないからである。そのため、プラグイン P のロジック部 10 では、共通化可能な機能が重複してしまい、無駄なリソースを消費することになる。また、ロジック部担当の開発者には、煩雑なソフトウェア開発・保守を強いることになる。

20

【0042】

このように、従来のプラグイン P では、ソフトウェアの開発・保守を困難なものとし、リソースを無駄に消費してしまう可能性があった。

【0043】

そこで、本実施形態に係る情報処理装置 100 では、連携元のプラグイン P 1 の UI 部 20₁ から連携先のプラグイン P 2 の UI 部 20₂ に対してロジック部 10₁ の処理結果を通知し、遷移後の画面表示を指示する。情報処理装置 100 は、このようなプラグイン連携機能を有している。

30

【0044】

図 5 は、本実施形態に係るプラグイン連携を行うソフトウェア構成例を示す図である。図 5 には、上記に説明を行った図 3 と同じプラグイン連携を行う本実施形態に係るアプリケーション AP のソフトウェア構成例が示されている。

【0045】

本実施形態に係るアプリケーション AP は、図 4 に示した従来のソフトウェア構成と次の点で異なる。本実施形態に係るアプリケーション AP では、プラグイン P 1 ~ P 4 のロジック部 10₁ ~ 10₄ ではなく、UI 部 20₁ ~ 20₄ が、連携時の画面遷移を制御する機能部（画面遷移制御部 21₁ ~ 21₄）を有している点である。これにより、プラグイン P では、ロジック部 10 と UI 部 20 との独立性が保たれ、画面処理に関する技術に精通した UI 部担当の開発者による対応が可能となり、開発・保守作業の分担が明確となる。

40

【0046】

また、プラグイン P 1 及び P 3 のロジック部 10₁ 及び 10₃ が有していた重複機能（認証部 12₁ 及び 12₃）を、1つの機能に共通化（プラグイン化）している点である。この機能の共通化は、上記構成（UI 部が画面遷移機能を有する構成）としたことで、ロジック部 10 の独立性が確保されたことによるものである。

【0047】

以下に、本実施形態に係るアプリケーション AP におけるプラグイン連携の処理手順例

50

を示す。

【0048】

図6は、本実施形態に係るプラグイン連携の処理手順例を示すシーケンス図である。

図6に示すように、情報処理装置100では、プラグインP1のUI部20₁により、表示画面が生成される(ステップS21)。その結果、表示装置102には、プラグインP1の操作画面(認証画面:画面1)が表示される。情報処理装置100では、利用者入力で受け付けた認証情報が、UI部20₁から共通化された認証プラグインPaのロジック部10_aへと渡される(ステップS22)。

【0049】

情報処理装置100では、認証プラグインPaのロジック部10_aにより、認証情報に基づく利用者認証処理が行われ(ステップS23)、その認証結果が、プラグインP1のUI部20₁へと通知される。情報処理装置100では、認証結果が、プラグインP1のUI部20₁から連携先であるプラグインP2のUI部20₂へと渡され、プラグインP1からプラグインP2へと画面遷移(画面1→2)が指示される(ステップS24)。

【0050】

続いて、情報処理装置100では、プラグインP2のUI部20₂により、表示画面が生成される(ステップS25)。その結果、表示装置102には、認証結果が反映されたプラグインP2の操作画面(画面2)が表示される。情報処理装置100では、利用者入力で受け付けた操作情報が、プラグインP2のUI部20₂から連携先であるプラグインP3のUI部20₃へと渡され、プラグインP2からプラグインP3へと画面遷移(画面2→3)が指示される(ステップS26)。

【0051】

続いて、情報処理装置100では、プラグインP3のUI部20₃により、表示画面が生成される(ステップS27)。その結果、表示装置102には、プラグインP3の操作画面(認証画面:画面3)が表示される。情報処理装置100では、利用者入力で受け付けた認証情報が、UI部20₃から再び認証プラグインPaのロジック部10_aへと渡される(ステップS28)。

【0052】

情報処理装置100では、認証プラグインPaのロジック部10_aにより、認証情報に基づく利用者認証処理が行われ(ステップS29)、その認証結果が、プラグインP3のUI部20₃へと通知される。情報処理装置100では、認証結果が、プラグインP3のUI部20₃から連携先であるプラグインP4のUI部20₄へと渡され、プラグインP3からプラグインP4へと画面遷移(画面3→4)が指示される(ステップS30)。

【0053】

続いて、情報処理装置100では、プラグインP4のUI部20₄により、表示画面が生成される(ステップS31)。その結果、表示装置102には、認証結果が反映されたプラグインP4の操作画面(画面4)が表示される。

【0054】

このように、本実施形態に係る情報処理装置100では、プラグインPのロジック部10でなくUI部20により、プラグイン連携時の画面遷移が制御され、ロジック部10による処理結果がUI部20を介して連携先へと伝達される。これによって、本実施形態に係る情報処理装置100では、ロジック部10とUI部20との独立性が保たれた構成でプラグイン間の連携動作が行える。

【0055】

以下に、本実施形態に係るプラグイン連携機能の構成とその動作について説明する。

図7は、本実施形態に係る情報処理装置100が有する機能構成例を示す図である。図7には、情報処理装置100に、プラグインP1及びプラグインP2を有するアプリケーションAPが搭載(インストール)されている場合の機能構成例が示されている。また、図7に示す機能構成例は、アプリケーションAPにおいて、プラグインP1の処理結果をプラグインP2に通知する連携動作を想定している。

10

20

30

40

50

【 0 0 5 6 】

図 7 に示すように、まず、プラグイン P 1 及びプラグイン P 2 は、処理部 1 3₁ , 1 3₂ (以下総称する場合「処理部 1 3」と言う)と入出力部 2 2₁ , 2 2₂ (以下総称する場合「入出力部 2 2」と言う)とを、プラグイン P に共通する機能部として有している。

【 0 0 5 7 】

処理部 1 3 は、ロジック部 1 0 が有し、プラグイン P の提供機能を実現するための処理を行う機能部である。また、入出力部 2 2 は、UI 部 2 0 が有し、利用者からの操作入力受け付けや利用者への情報表示出力を行う機能部である。

【 0 0 5 8 】

ここで、上記機能部の具体的な動作について、プラグイン P が認証機能を提供する場合を例に簡単に説明する。まず、入出力部 2 2 は、利用者に対して認証情報(「ユーザ ID」や「パスワード」)の入力を促す認証画面(ログイン画面)を表示する。続いて、入出力部 2 2 は、利用者により入力された認証情報を受け付け、処理部 1 3 に認証情報を渡す。処理部 1 3 は、認証情報に基づき利用者認証を行い、入出力部 2 2 に認証結果を渡す。入出力部 2 2 は、認証結果を反映した画面を表示する。

【 0 0 5 9 】

一方、プラグイン P 1 は通知部 2 1 1₁ を、プラグイン P 2 は受信部 2 1 2₂ を、プラグイン連携時の画面遷移を制御する機能部(画面遷移制御部)として有している。

【 0 0 6 0 】

通知部 2 1 1₁ は、プラグイン P 1 の UI 部 2 0₁ が有する画面遷移制御部 2 1₁ の一機能部ある。通知部 2 1 1₁ は、入出力部 2 2₁ 及び/又は処理部 1 3₁ からの情報をイベントとしてプラグイン P 2 へと通知し、連携先のプラグイン P 2 に対して画面遷移を指示する。このとき、通知部 2 1 1₁ は、連携先のプラグイン P 2 に渡す情報を基に、プラグイン P 2 (受信部)が参照可能なイベントクラスを生成することで、イベントを通知する。なお、入出力部 2 2₁ からの情報には、例えば、利用者からの処理要求指示や入力値などがある。また、処理部 1 3₁ からの情報には、例えば、処理結果などがある。

【 0 0 6 1 】

受信部 2 1 2₂ は、プラグイン P 2 の UI 部 2 0₂ が有する画面遷移制御部 2 1₂ の一機能部である。受信部 2 1 2₂ は、プラグイン P 1 から通知されたイベントを受信する。このとき、受信部 2 1 2₂ は、通知部 2 2₁ により生成されたイベントクラスを参照することで、イベントを受信する。受信部 2 1 2₂ は、受信イベントから得た情報を、入出力部 2 2₂ 及び/又は処理部 1 3₂ に渡し、連携後の処理を指示する。なお、連携後の処理には、画面遷移処理が含まれる。

【 0 0 6 2 】

ここで、上記機能部の具体的な動作について、プラグイン連携(プラグイン P 1 からプラグイン P 2 への連携)において利用者認証が必要な場合を例に簡単に説明する。まず、プラグイン P 1 の通知部 2 1 1₁ は、利用者により入力された認証情報を基に、プラグイン P 2 が参照可能なイベントクラスを生成し、イベントを通知する。プラグイン P 2 の受信部 2 1 2₂ は、受信イベントから認証情報を得て、プラグイン P 2 の処理部 1 3₂ に認証情報を渡す。処理部 1 3₂ は、認証情報に基づき利用者認証を行い、入出力部 2 2₂ に認証結果を渡す。入出力部 2 2₂ は、認証結果を反映したプラグイン P 2 の画面を表示する。その結果、表示装置 1 0 2 の表示画面が、プラグイン P 1 の表示画面からプラグイン P 2 の表示画面へと遷移する。

【 0 0 6 3 】

このように、本実施形態に係る情報処理装置 1 0 0 では、連携元のプラグイン P 1 の UI 部 2 0₁ から連携先のプラグイン P 2 の UI 部 2 0₂ に対してロジック部 1 0₁ の処理結果を通知し、遷移後の画面表示を指示する。

【 0 0 6 4 】

次に、本実施形態に係るプラグイン連携機能の詳細な動作(機能部群の連係動作)について、処理手順を示すシーケンス図を用いて説明する。

10

20

30

40

50

【 0 0 6 5 】

プラグイン連携機能は、情報処理装置 1 0 0 に搭載（インストール）されるプログラム（プラグイン連携機能を実現するソフトウェア部品）が、CPU 1 0 6 により、格納先（例えば「ROM」や「HDD」）からRAM 1 0 4 上に読み出され、以下の処理が実行されることで実現される。

【 0 0 6 6 】

図 8 は、本実施形態に係るプラグイン連携の詳細な処理手順例を示すシーケンス図である。図 8 には、図 6 で示した利用者認証を伴うプラグイン P 1 からプラグイン P 2 へのプラグイン連携（ステップ S 2 1 ~ S 2 5 ）の詳細な処理手順例が示されている。

【 0 0 6 7 】

図 8 に示すように、情報処理装置 1 0 0 では、プラグイン P 1 のUI部 2 0₁ が有する入出力部 2 2₁ より、表示画面から入力された利用者の認証情報（「ユーザID」や「パスワード」）を付け付ける（ステップ S 1 0 1 ）。入出力部 2 2₁ は、受け付けた認証情報を認証プラグイン P a のロジック部 1 0_a が有する処理部 1 3_a（認証部に該当）へと渡し、利用者の認証許可判定を指示する（ステップ S 1 0 2 ）。

10

【 0 0 6 8 】

情報処理装置 1 0 0 では、認証プラグイン P a のロジック部 1 0_a が有する処理部 1 3_a により、認証情報に基づき、利用者の認証許可判定が行われる（ステップ S 2 0 1 ）。処理部 1 3_a は、認証結果をプラグイン P 1 のUI部 2 0₁ が有する通知部 2 1 1₁ へと渡す（ステップ S 2 0 2 ）。通知部 2 1 1₁ は、dispatchEventメソッドにより、認証結果をイベントとして通知する（ステップ S 2 0 3 ）。

20

【 0 0 6 9 】

通知部 2 1 1₁ は、具体的に次のような処理を行う。通知部 2 1 1₁ は、dispatchEventメソッドにイベントオブジェクト（loginEvent）を指定し呼び出す。その結果、dispatchEventメソッドは、イベントを受信するためにaddEventListenerメソッドを介して登録されたリスナー（イベント受信先）に対して、指定イベントオブジェクトをブロードキャストする。このとき、メモリ上には、イベントを受信するためにaddEventListenerメソッドを介して登録されたプラグイン P 2（連携先）が参照可能なライブラリ形式で、認証結果が定義されたイベントクラス（loginEvent { 認証OK }）が生成される。

30

【 0 0 7 0 】

なお、addEventListenerメソッドでは、ブロードキャスト時の受信イベントに対応するイベントオブジェクト（eventObj）と、イベント受信時の実行処理（action）とを指定することで、リスナーを登録できる。よって、addEventListenerメソッドにより、連携先のプラグイン P 2 を、プラグイン P 1 の指定イベントオブジェクト（loginEvent）と、プラグイン P 1 からのイベント受信時の実行処理（display）とを指定して登録しておけば、プラグイン連携時の画面遷移が可能となる。具体的には、次のように処理され画面遷移する。

【 0 0 7 1 】

情報処理装置 1 0 0 では、プラグイン P 2 のUI部 2 0₂ が有する受信部 2 1 2₂ により、プラグイン P 1 からの認証結果通知を受信する（ステップ S 3 0 1 ）。このとき、受信部 2 1 2₂ は、イベントクラスから定義された認証結果を取得する。

40

【 0 0 7 2 】

続いて、受信部 2 1 2₂ は、イベント受信時の実行処理（display）に従って、同UI部 2 0₂ が有する入出力部 2 2₂ に認証結果を渡し、認証結果を反映した画面表示を指示する（ステップ S 3 0 2 ）。これにより、プラグイン P 1 からプラグイン P 2 へと画面遷移（画面 1 2）が指示される。

【 0 0 7 3 】

情報処理装置 1 0 0 では、プラグイン P 2 のUI部 2 0₂ が有する入出力部 2 2₂ により、表示画面が生成される（ステップ S 3 0 3 ）。その結果、表示装置 1 0 2 には、認証

50

結果が反映されたプラグイン P 2 の操作画面（画面 2）が表示される。

【 0 0 7 4 】

以上のように、本実施形態に係る情報処理装置 1 0 0 では、プラグイン P のロジック部 1 0 でなく U I 部 2 0 により、プラグイン連携時の画面遷移が制御され、ロジック部 1 0 による処理結果が U I 部 2 0 を介して連携先へと伝達されることで、画面遷移を伴うプラグイン連携が実現される。

【 0 0 7 5 】

《 変形例 》

図 7 に示した機能構成では、連携元のプラグイン P 1 が有する U I 部 2 0₁ から連携先のプラグイン P 2 が有する U I 部 2 0₂ に対して情報を通知する（一方向に情報伝達を行う）ことでプラグイン連携を実現している。

10

【 0 0 7 6 】

しかし、プラグイン連携機能には、プラグイン間において双方向に情報伝達を行うことで連携動作する構成も考えられる。つまり、連携先と連携元とが入れ替わる構成も考えられる。

【 0 0 7 7 】

以下には、変形例として、プラグイン間において双方向の情報伝達を行うプラグイン連携機能について説明する。

【 0 0 7 8 】

図 9 は、本変形例に係るプラグイン連携の動作例を示す図である。

20

本変形例に示すプラグイン連携機能では、例えば、図 9 に示すような、プラグイン間において双方向に情報伝達を行う連携動作を実現することができる。

【 0 0 7 9 】

まず、プラグイン P 1 の画面 W 1 には、プラグイン P 1 の U I 部 2 0₁ により、識別子（ I D ）によりグループ化された 3 つの設定値（値 1 ， 値 2 ， 値 3 ）が一覧表示されている。プラグイン P 1 は、利用者による画面 W 1 上の [追加] ボタン押下を受けて、新たな設定値を保存領域（データエリア）に追加する追加イベントを、 U I 部 2 0₁ からプラグイン P 2 の U I 部 2 0₂ に通知する（図中の（ 1 ））。

【 0 0 8 0 】

プラグイン P 2 は、追加イベントの通知を受けて、設定値を入力する画面 W 2 を表示し、プラグイン P 1 からプラグイン P 2 へと画面が遷移する。プラグイン P 2 は、利用者による画面 W 2 上の識別子及び設定値（ I D : 0 0 3 ， 値 1 : G ， 値 2 : H ， 値 3 : I ）入力と、 [保存] ボタン押下とを受けて、 U I 部 2 0₂ から保存処理の指示を受け付けたロジック部 1 0₂ により、設定値を保存領域に追加・保存する（図中の（ 2 ））。その保存結果 [成功 / 失敗] は、ロジック部 1 0₂ から U I 部 2 0₂ へと渡される（図中の（ 3 ））。その結果、 U I 部 2 0₂ は、保存結果 [成功 / 失敗] 及び保存内容 [0 0 3 ， G ， H ， I] を、保存結果イベントとして、プラグイン P 1 の U I 部 2 0₁ に通知する（図中の（ 4 ））。

30

【 0 0 8 1 】

プラグイン P 1 は、保存結果イベントの通知を受けて、追加された設定値を含む一覧画面 W 3 を表示し、プラグイン P 2 からプラグイン P 1 へと画面が遷移する。これにより、プラグイン P 1 の表示画面が更新される（画面 W 1 画面 W 3 ）。

40

【 0 0 8 2 】

図 1 0 は、本変形例に係る情報処理装置 1 0 0 が有する機能構成例を示す図である。

本変形例に係るプラグイン連携機能は、図 7 と図 1 0 との比較からも分かるように、 U I 部 2 0 の画面遷移制御部 2 1 が、通知部 2 1 1 及び受信部 2 1 2 の両方を有している。本変形では、このような機能構成により、プラグイン間において双方向に情報伝達を行うことができる。

【 0 0 8 3 】

本変形例に係るプラグイン連携機能の処理手順は、次の通りである。

50

図 1 1 は、本変形例に係るプラグイン連携の詳細な処理手順例を示すシーケンス図である。なお、図 1 1 には、図 9 に示した連携動作時の処理手順例が示されている。

【 0 0 8 4 】

図 1 1 に示すように、情報処理装置 1 0 0 では、プラグイン P 1 の UI 部 2 0₁ が有する入出力部 2 2₁ より、新たな設定値（データ）の追加命令（[追加] ボタン押下）を受け付ける（ステップ S 4 0 1）。入出力部 2 2₁ は、受け付けたデータ追加の通知を UI 部 2 0₁ が有する通知部 2 1 1₁ に指示する（ステップ S 4 0 2）。通知部 2 1 1₁ は、dispatchEvent メソッドにより、データ追加イベントを通知する（ステップ S 4 0 3）。

【 0 0 8 5 】

通知部 2 1 1₁ は、具体的に次のような処理を行う。通知部 2 1 1₁ は、dispatchEvent メソッドにイベントオブジェクト（addEvent）を指定し呼び出す。その結果、dispatchEvent メソッドは、イベントを受信するために addEventListener メソッドを介して登録されたりスナー（プラグイン 2）に対して、指定イベントオブジェクトをブロードキャストする。このとき、メモリ上には、イベントを受信するために addEventListener メソッドを介して登録されたプラグイン P 2（連携先）が参照可能なライブラリ形式で、データ追加命令が定義されたイベントクラス（addEvent{x}）が生成される。

【 0 0 8 6 】

情報処理装置 1 0 0 では、プラグイン P 2 の UI 部 2 0₂ が有する受信部 2 1 2₂ により、プラグイン P 1 からデータ追加通知を受信する（ステップ S 5 0 1）。このとき、受信部 2 1 2₂ は、イベントクラスから定義されたデータ追加命令を取得する。

【 0 0 8 7 】

続いて、受信部 2 1 2₂ は、イベント受信時の実行処理（input）に従って、同 UI 部 2 0₂ が有する入出力部 2 2₂ に、識別子及び設定値の入力を受け付ける画面表示を指示する（ステップ S 5 0 2）。これにより、プラグイン P 1 からプラグイン P 2 へと画面遷移（画面 1 2）が指示される。

【 0 0 8 8 】

入出力部 2 2₂ は、表示画面を介して入力された識別子及び設定値（入力情報）を受け付け（ステップ S 5 0 3）、受け付けた入力情報 [0 0 3, G, H, I] をプラグイン P 2 のロジック部 1 0₂ が有する処理部 1 3₂ に渡し、保存処理の実行を指示する（ステップ S 5 0 4）。

【 0 0 8 9 】

処理部 1 3₂ は、受け取った入力情報を保存領域に保存し（ステップ S 5 0 5）、保存結果 [成功] 及び保存内容 [ID : 0 0 3, 値 1 : G, 値 2 : H, 値 3 : I] を、UI 部 2 0₂ が有する通知部 2 1 1₂ に渡し、実行結果の通知を指示する（ステップ S 5 0 6）。通知部 2 1 1₂ は、dispatchEvent メソッドにより、実行結果イベントを通知する（ステップ S 5 0 7）。

【 0 0 9 0 】

通知部 2 1 1₂ は、具体的に次のような処理を行う。通知部 2 1 1₂ は、dispatchEvent メソッドにイベントオブジェクト（saveEvent）を指定し呼び出す。その結果、dispatchEvent メソッドは、イベントを受信するために addEventListener メソッドを介して登録されたりスナー（プラグイン 1）に対して、指定イベントオブジェクトをブロードキャストする。このとき、メモリ上には、イベントを受信するために addEventListener メソッドを介して登録されたプラグイン P 1（連携先）が参照可能なライブラリ形式で、実行結果（結果 [成功] の場合は保存内容）が定義されたイベントクラス（saveEvent{003, G, H, I}）が生成される。

【 0 0 9 1 】

情報処理装置 1 0 0 では、プラグイン P 1 の UI 部 2 0₁ が有する受信部 2 1 2₁ によ

10

20

30

40

50

り、プラグイン P 2 からの実行結果通知を受信する（ステップ S 6 0 1）。このとき、受信部 2 1 2₁ は、イベントクラスから定義された実行結果を取得する。

【 0 0 9 2 】

続いて、受信部 2 1 2₁ は、イベント受信時の実行処理（display）に従って、同 UI 部 2 0₁ が有する入出力部 2 2₁ に実行結果を渡し、実行結果を反映した画面表示を指示する（ステップ S 6 0 2）。これにより、プラグイン P 2 からプラグイン P 1 へと画面遷移（画面 2 1）が指示される。

【 0 0 9 3 】

情報処理装置 1 0 0 では、プラグイン P 1 の UI 部 2 0₁ が有する入出力部 2 2₁ により、表示画面（設定値の一覧画面）が生成される（ステップ S 6 0 3）。その結果、表示装置 1 0 2 には、新たに追加された設定値を含む実行結果が反映されたプラグイン P 1 の操作画面（画面 1）が表示される。

10

【 0 0 9 4 】

<まとめ>

以上のように、本実施形態に係る情報処理装置 1 0 0 によれば、連携元のプラグイン P 1 の UI 部 2 0₁ から連携先のプラグイン P 2 の UI 部 2 0₂ に対してロジック部 1 0₂ の処理結果を通知し、遷移後の画面表示を指示する。

【 0 0 9 5 】

これによって、本実施形態に係る情報処理装置 1 0 0 では、ロジック部 1 0 でなく UI 部 2 0 により、プラグイン連携時の画面遷移が制御され、ロジック部 1 0 による処理結果が UI 部 2 0 を介して連携先へと伝達される。また、プラグイン連携時の情報伝達は、プラグイン間において、連携元から連携先への一方向だけでなく双方向でも行われる。

20

【 0 0 9 6 】

このように、本実施形態に係る情報処理装置 1 0 0 では、プラグイン P が有するロジック部 1 0 と UI 部 2 0 との独立性が保たれた構成でプラグイン間の連携動作が行える。

【 0 0 9 7 】

ここまで、上記実施形態の説明を行ってきたが、上記実施形態に係る情報処理装置 1 0 0 が有する「プラグイン連携機能」は、図を用いて説明を行った各処理手順を、動作環境（プラットフォーム）にあったプログラミング言語でコード化したプログラムが、CPU 1 0 6 により実行されることで実現される。

30

【 0 0 9 8 】

上記プログラムは、コンピュータが読み取り可能な記録媒体 1 0 3 a に格納することができる。上記記録媒体 1 0 3 a には、例えば、フロッピー（登録商標）ディスク、CD（Compact Disk）、及び DVD（Digital Versatile Disk）、ならびに SD メモリカード（SD Memory Card）及び USB（Universal Serial Bus）メモリなどがある。

【 0 0 9 9 】

よって、上記プログラムは、上記記録媒体 1 0 3 a に記憶させることで、記録媒体 1 0 3 a を読み取り可能なドライブ装置 1 0 3 などを介して情報処理装置 1 0 0 にインストールすることができる。また、情報処理装置 1 0 0 は、インタフェース装置 1 0 7 を備えていることから、インターネットなどの電気通信回線を用いて上記プログラムをダウンロードし、インストールすることもできる。

40

【 0 1 0 0 】

また、上記実施形態では、プラグイン P が有する UI 部 2 0 の間で行うプラグイン連携時のイベント処理に、ActionScript（Flash に使用されるプログラミング言語）の EventDispatcher クラスに属するメソッドを用いた場合について説明を行ったが、この限りでない。

【 0 1 0 1 】

最後に、上記実施形態に挙げた形状や構成に、その他の要素との組み合わせなど、ここで示した要件に、本発明が限定されるものではない。これらの点に関しては、本発明の主旨をそこなわない範囲で変更することが可能であり、その応用形態に応じて適切に定める

50

ことができる。

【符号の説明】

【0102】

- 10 ロジック部 (プラグインロジック)
- 11 画面遷移制御部 (ロジック部側)
- 12 認証部
- 13 処理部
- 20 UI部 (プラグインUI)
- 21 画面遷移制御部 (UI部側)
- 211 通知部 (₁ : プラグイン1 , ₂ : プラグイン2) 10
- 212 受信部 (₁ : プラグイン1 , ₂ : プラグイン2)
- 22 入出力部 (₁ : プラグイン1 , ₂ : プラグイン2)
- 100 情報処理装置
- 101 入力装置
- 102 表示装置
- 103 ドライブ装置 (a : 記録媒体)
- 104 RAM (揮発性の半導体メモリ)
- 105 ROM (不揮発性の半導体メモリ)
- 106 CPU (中央処理装置)
- 107 インタフェース装置 (NIC : Network I/F Card) 20
- 108 HDD (不揮発性の記憶装置)
- AP アプリケーション
- Pn プラグイン
- PF プラットフォーム

【先行技術文献】

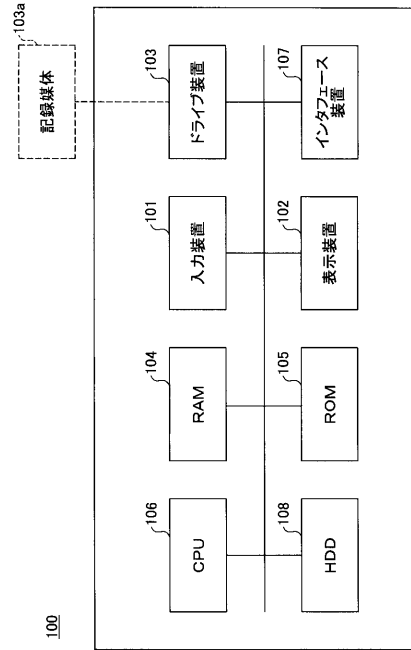
【特許文献】

【0103】

【特許文献1】特開2009-54027号公報

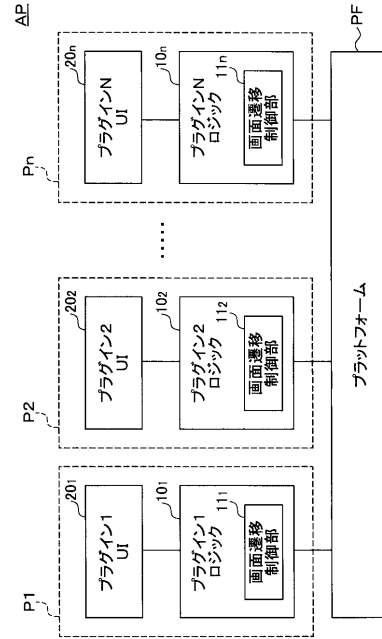
【図1】

本発明の第1の実施形態に係る情報処理装置のハードウェア構成例を示す図



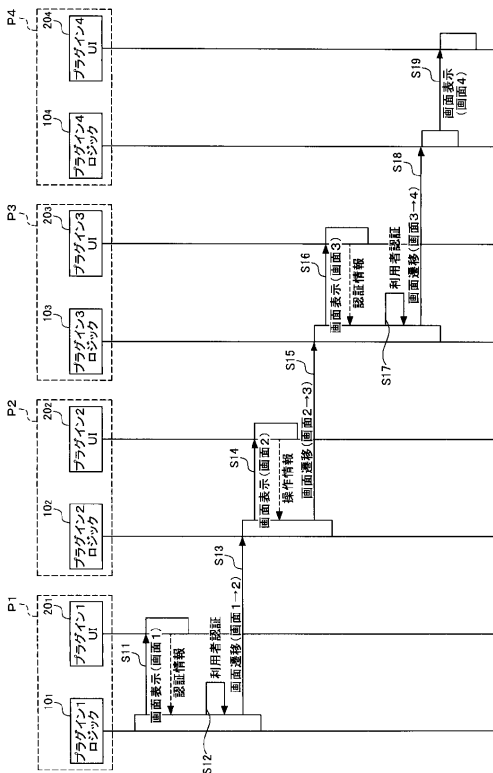
【図2】

アプリケーションのソフトウェア構成例を示す図



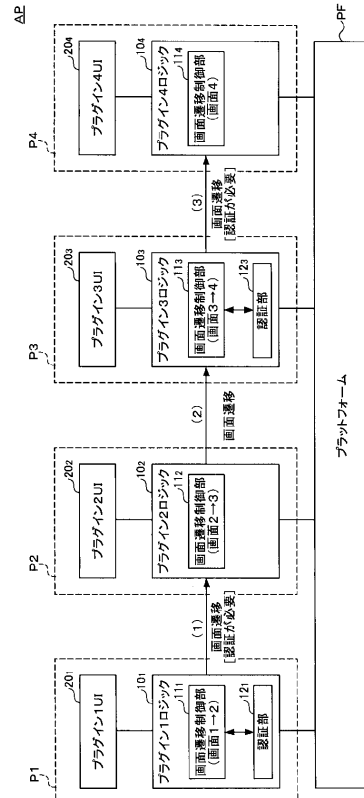
【図3】

従来のプラグイン連携の処理手順例を示すシーケンス図



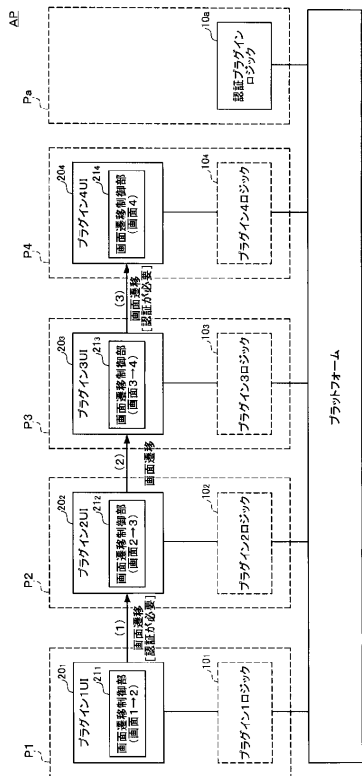
【図4】

従来のプラグイン連携を行うソフトウェア構成例を示す図



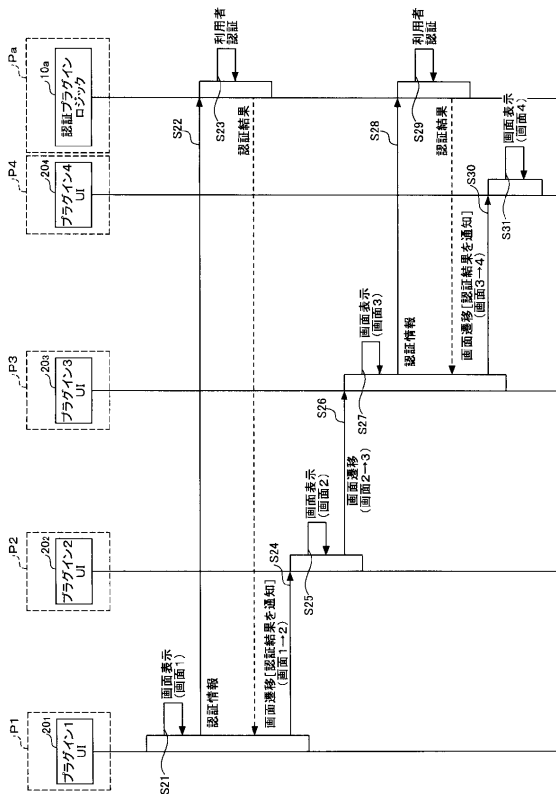
【図5】

本発明の第1の実施形態に係る
プラグイン連携を行うソフトウェア構成例を示す図



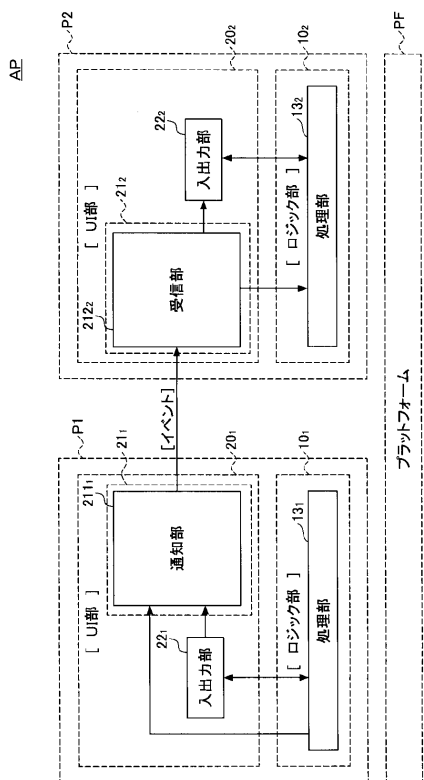
【図6】

本発明の第1の実施形態に係る
プラグイン連携の処理手順例を示すシーケンス図



【図7】

本発明の第1の実施形態に係る情報処理装置が有する機能構成例を示す図



【図8】

本発明の第1の実施形態に係る
プラグイン連携の詳細な処理手順例を示すシーケンス図

