

(19)日本国特許庁(JP)

(12)公開特許公報(A)

(11)公開番号

特開2023-7730  
(P2023-7730A)

(43)公開日 令和5年1月19日(2023.1.19)

(51)国際特許分類 F I テーマコード(参考)  
 G 0 6 F 9/445(2018.01) G 0 6 F 9/445 1 3 0 5 B 3 7 6  
 G 0 6 F 8/60 (2018.01) G 0 6 F 8/60

審査請求 未請求 請求項の数 22 O L (全31頁)

(21)出願番号	特願2021-110760(P2021-110760)	(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
(22)出願日	令和3年7月2日(2021.7.2)	(74)代理人	100126240 弁理士 阿部 琢磨
(特許庁注:以下のものは登録商標)		(74)代理人	100124442 弁理士 黒岩 創吾
1. JAVASCRIPT		(72)発明者	熊谷 空 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
		(72)発明者	木暮 岳史 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
		(72)発明者	池田 篤 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

最終頁に続く

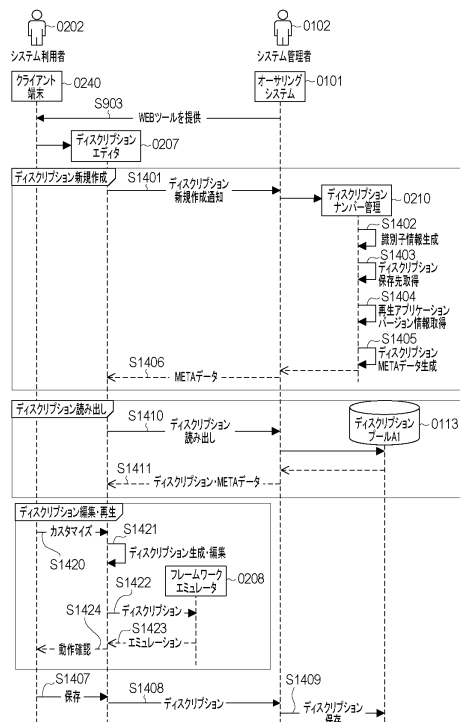
(54)【発明の名称】 情報処理システム、情報処理方法、プログラム

(57)【要約】

【課題】 記述ファイルとアプリケーションデータに基づくカスタムアプリケーションを適切に管理可能なシステムを提供する。

【解決手段】 MFPで実行されるアプリケーションのAppNoを管理するシステムは、再生アプリケーションと第1のディスクリプションの組み合わせを第1のカスタムアプリケーションとして提供し、再生アプリケーションと第1のディスクリプションの組み合わせを第2のカスタムアプリケーションとして提供し、第1のDesNoと第2のDesNoとを管理し、第1のDesNoと第2のDesNoに基づいて第1のカスタムアプリケーションと第2のカスタムアプリケーションを区別可能に管理する。

【選択図】 図14



**【特許請求の範囲】****【請求項 1】**

画像処理装置で実行されるアプリケーションをアプリケーション識別子で管理する情報処理システムであって、

動作手順が定義された記述ファイルを再生する再生アプリケーションと第 1 の記述ファイルの組み合わせ情報を第 1 の組み合わせアプリケーションとして提供する手段と、

前記再生アプリケーションと第 2 の記述ファイルの組み合わせ情報を第 2 の組み合わせアプリケーションとして提供する手段と、

前記第 1 の記述ファイルに対応する第 1 の記述識別子と、前記第 2 の記述ファイルに対応する第 2 の記述識別子と、を管理する手段と、

前記第 1 の記述識別子および前記第 2 の記述識別子に基づいて前記第 1 の組み合わせアプリケーションと前記第 2 の組み合わせアプリケーションを区別可能に管理する手段と、

有することを特徴とする情報処理システム。

10

**【請求項 2】**

前記第 1 の組み合わせアプリケーションに含まれる再生アプリケーションと、前記第 2 の組み合わせアプリケーションに含まれる再生アプリケーションと、は同一のアプリケーション識別子で管理されることを特徴とする請求項 1 に記載の情報処理システム。

**【請求項 3】**

前記第 1 の組み合わせアプリケーションは、前記再生アプリケーションと前記第 1 の記述ファイルを含むパッケージデータに基づいて前記画像処理装置にインストールされることを特徴とする請求項 1 または 2 に記載の情報処理システム。

20

**【請求項 4】**

前記画像処理装置は、前記パッケージデータに基づくインストール処理において、前記再生アプリケーションのアプリケーション識別子の代わりに前記第 1 の記述識別子を用いてアプリケーションの登録をおこなうことを特徴とする請求項 3 に記載の情報処理システム。

**【請求項 5】**

前記第 1 の記述ファイルを作成する作成手段と、

前記作成手段からの通知に基づいて前記第 1 の記述識別子を発行する手段と、を有することを特徴とする請求項 1 乃至 4 のいずれか 1 項に記載の情報処理システム。

30

**【請求項 6】**

前記作成手段は、外部端末のウェブブラウザ上で実行される編集プログラムであって、

前記編集プログラムは、ユーザの編集操作を受け付けるための画面情報を出力することを特徴とする請求項 5 に記載の情報処理システム。

**【請求項 7】**

前記画面情報は、編集中のアプリケーションの動作確認をおこなうための情報を含むことを特徴とする請求項 6 に記載の情報処理システム。

**【請求項 8】**

前記画面情報は、編集中のアプリケーションに追加可能な部品情報を含むことを特徴とする請求項 6 または 7 に記載の情報処理システム。

40

**【請求項 9】**

前記画面情報は、編集中のアプリケーションに用いられる表示部品を編集するための情報を含むことを特徴とする請求項 6 乃至 8 のいずれか 1 項に記載の情報処理システム。

**【請求項 10】**

前記編集プログラムは、前記第 1 の記述ファイルを再生する手段を有することを特徴とする請求項 6 乃至 9 のいずれか 1 項に記載の情報処理システム。

**【請求項 11】**

前記再生する手段は、前記再生アプリケーションと同一の言語であって JavaScript とは異なる言語に基づく再生プログラムであることを特徴とする請求項 10 に記載の情報処理システム。

50

## 【請求項 1 2】

前記再生プログラムは前記再生アプリケーションと同一のプログラムであることを特徴とする請求項 1 1 に記載の情報処理システム。

## 【請求項 1 3】

前記異なる言語は Lua 言語であることを特徴とする請求項 1 1 または 1 2 に記載の情報処理システム。

## 【請求項 1 4】

前記編集プログラムは前記再生プログラムを実行するためのエミュレート手段を備えることを特徴とする請求項 1 1 乃至 1 3 のいずれか 1 項に記載の情報処理システム。

## 【請求項 1 5】

前記第 1 の記述ファイルは JSON 形式で記述されたデータであることを特徴とする請求項 1 乃至 1 4 のいずれか 1 項に記載の情報処理システム。

## 【請求項 1 6】

前記パッケージデータは、前記再生アプリケーションのバージョン情報を含むことを特徴とする請求項 3 に記載の情報処理システム。

## 【請求項 1 7】

前記再生アプリケーションの署名情報と、前記第 1 の記述ファイルの署名情報と、を管理する手段を有することを特徴とする請求項 1 乃至 1 6 のいずれか 1 項に記載の情報処理システム。

## 【請求項 1 8】

前記画像処理装置はシートに画像を形成する画像形成部を備えることを特徴とする請求項 1 乃至 1 7 のいずれか 1 項に記載の情報処理システム。

## 【請求項 1 9】

前記画像処理装置は原稿から画像を読み取る画像読取部を備えることを特徴とする請求項 1 乃至 1 8 のいずれか 1 項に記載の情報処理システム。

## 【請求項 2 0】

前記画像処理装置と通信するサーバ装置と、前記サーバ装置と通信する情報処理装置と、を有することを特徴とする請求項 1 乃至 1 9 のいずれか 1 項に記載の情報処理システム。

## 【請求項 2 1】

画像処理装置で実行されるアプリケーションをアプリケーション識別子で管理する情報処理装置であって、

動作手順が定義された記述ファイルを再生する再生アプリケーションと第 1 の記述ファイルの組み合わせ情報を第 1 の組み合わせアプリケーションとして提供する手段と、

前記再生アプリケーションと第 2 の記述ファイルの組み合わせ情報を第 2 の組み合わせアプリケーションとして提供する手段と、

前記第 1 の記述ファイルに対応する第 1 の記述識別子と、前記第 2 の記述ファイルに対応する第 2 の記述識別子と、を管理する手段と、

前記第 1 の記述識別子および前記第 2 の記述識別子に基づいて前記第 1 の組み合わせアプリケーションと前記第 2 の組み合わせアプリケーションを区別可能に管理する手段と、

有することを特徴とする情報処理装置。

## 【請求項 2 2】

画像処理装置で実行されるアプリケーションをアプリケーション識別子で管理する情報処理装置が実行するプログラムであって、

動作手順が定義された記述ファイルを再生する再生アプリケーションと第 1 の記述ファイルの組み合わせ情報を第 1 の組み合わせアプリケーションとして提供する工程と、

前記再生アプリケーションと第 2 の記述ファイルの組み合わせ情報を第 2 の組み合わせアプリケーションとして提供する工程と、

前記第 1 の記述ファイルに対応する第 1 の記述識別子と、前記第 2 の記述ファイルに対応する第 2 の記述識別子と、を管理する工程と、

10

20

30

40

50

前記第 1 の記述識別子および前記第 2 の記述識別子に基づいて前記第 1 の組み合わせアプリケーションと前記第 2 の組み合わせアプリケーションを区別可能に管理する工程と、有する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】

【0001】

本発明は、情報処理装置にインストールするアプリケーションを編集するシステムに関する。この情報処理装置としては、例えば、プリンタ、スキャナ、FAX、複写機、およびこれらの複合機等の画像形成装置を利用することができる。また、この情報処理装置として、パーソナルコンピュータやモバイル端末等の電子機器を利用してもよい。

10

【背景技術】

【0002】

MFP等の情報処理装置にアプリケーションをインストールして機能を拡張する仕組みが知られている。特許文献1には情報処理装置にインストールしたアプリケーションID等を用いて管理する技術について開示している。

【先行技術文献】

【特許文献】

【0003】

【特許文献1】特開2016-2470号公報

【発明の概要】

20

【発明が解決しようとする課題】

【0004】

情報処理装置を扱う顧客のニーズは様々である。そのため、顧客のニーズにあった機能をアプリケーションの形で提供することが望ましい。また、顧客の要望に迅速に対応できるよう、サービスマンや営業担当者等の関係者がアプリケーションをカスタマイズして顧客に提供できるとよい。

【0005】

そこで、本願発明人は、編集が容易な記述データとしてのディスクリプション情報と汎用の再生アプリケーションを組み合わせることでカスタマイズされたアプリケーション(カスタムアプリケーション)を提供する仕組みについて検討を進めている。このようなカスタムアプリケーションは、顧客の要望の数に比例して種類が増大するため、適切に管理できる環境を構築することが望ましい。

30

【0006】

本発明の他の目的は、記述ファイルとアプリケーションデータに基づくカスタムアプリケーションを適切に管理可能なシステムを提供することである。

【課題を解決するための手段】

【0007】

本発明の情報処理システムは、画像処理装置で実行されるアプリケーションをアプリケーション識別子で管理する情報処理システムであって、動作手順が定義された記述ファイルを再生する再生アプリケーションと第1の記述ファイルの組み合わせ情報を第1の組み合わせアプリケーションとして提供する手段と、前記再生アプリケーションと第2の記述ファイルの組み合わせ情報を第2の組み合わせアプリケーションとして提供する手段と、前記第1の記述ファイルに対応する第1の記述識別子と、前記第2の記述ファイルに対応する第2の記述識別子と、を管理する手段と、前記第1の記述識別子および前記第2の記述識別子に基づいて前記第1の組み合わせアプリケーションと前記第2の組み合わせアプリケーションを区別可能に管理する手段と、有することを特徴とするものである。

40

【発明の効果】

【0008】

本発明によれば、記述ファイルとアプリケーションデータに基づくカスタムアプリケーションを適切に管理可能なシステムを提供できる。

50

## 【図面の簡単な説明】

【0009】

【図1】オーサリングシステムについて説明する図である。

【図2】図2(A)は、オーサリングシステムのソフトウェア構成を示す図である。図2(B)は、オーサリングシステムに用いられるサーバ装置のハードウェア構成を示す図である。

【図3】図3(A)は、MFPのハードウェア構成を示す図である。図3(B)は、MFPのソフトウェア構成を示す図である。

【図4】MFPにおけるソフトウェアの階層構造を示す図である。

【図5】図5(A)はクライアント端末のハードウェア構成を示す図である。図5(B)はクライアント端末が実行するディスクリプションエディタのソフトウェア構成を示す図である。

【図6】ディスクリプションエディタに関するソフトウェアの階層構造を示す図である。

【図7】図7(A)は、ディスクリプションエディタの画面を示す図である。図7(B)は、MFPにおけるカスタムアプリケーションの実行画面を示す図である。

【図8】ディスクリプションエディタのソフトウェアモジュール間のシーケンスを示す図である。

【図9】オーサリングシステムの利用フローを示す図である。

【図10】カスタムアプリケーションのデータ構造を示す図である。

【図11】ディスクリプションを示す図である。

【図12】MFPにインストールしたカスタムアプリケーションの構造を示す図である。

【図13】アプリケーションのインストールフローを示す図である。

【図14】ディスクリプション生成の詳細フローを示す図である。

【図15】図15(A)はアプリケーションパッケージのmetaデータとMFPのアプリケーションデータ管理を説明する図である。図15(B)は各metaデータとMFP内のデータの対応を示す表である。

【図16】ディスクリプションの管理テーブルを示す図である。

## 【発明を実施するための形態】

【0010】

以下、本発明の実施するための形態について、実施例を挙げ図面を用いて具体的に説明する。なお、本発明の範囲は実施例に記載の構成には限定されない。同様の効果が得られる範囲において、構成の一部や処理の一部について均等物へ置き換えたり省略したりといった変更をおこなってもよい。

【0011】

(実施例)

&lt;オーサリングシステム&gt;

図1は、オーサリングシステムについて説明する図である。オーサリングシステム0101は、MFP0203で実行されるカスタムアプリケーションの編集環境を提供する情報処理システムである。本実施例では、オーサリングシステム0101を用いることで、サービスマンや営業担当者等の関係者によるアプリケーションのカスタマイズ可能にする。これにより、顧客の要望にあったカスタムアプリケーションを迅速に提供することが可能となる。また、顧客の要望毎に種類が増えていくカスタムアプリケーションを適切に管理にすることができる。

【0012】

オーサリングシステム0101は汎用クラウドサービス上に構築されるシステムであり、ネットワークを介して様々な装置からアクセスすることができる。オーサリングシステム0101を使用するすべての管理者および利用者は、ユーザ認証の仕組みによってログイン管理される。すなわち、管理者および利用者は、オーサリングシステム0101の利用開始時にユーザIDやパスワード等を用いたユーザ認証手続きをおこなう。本実施例では、ユーザ認証の仕組みとして汎用クラウドサービスが提供する機能を利用するものとし

10

20

30

40

50

、その詳細な説明は割愛する。なお、オーサリングシステム 0101 は、独自のユーザ認証機能を備えていてもよい。

【0013】

オーサリングシステム 0101 はテナント発行サイト 0106 を備えており、用途に応じて複数のテナントを作成することができる。図 1 は、少なくとも、テナント A (0107)、テナント B (0108)、および、テナント C (0109) が発行された状態を示している。例えば、システム管理者 0102 は、クライアント端末 0240 からウェブブラウザ介してテナント発行サイト 0106 にアクセスすることで、各国のグループ販売会社に対してテナントを作成することができる。グループ販売会社ではなくシステムインテグレータに対してテナントを発行してもよい。

10

【0014】

テナントが発行された場合、ひとつのテナントには、それを管理する一人以上の管理者が割り当てられる。図 1 では、少なくとも、テナント A (0107) を管理するテナント A 管理者 0103、テナント B (0108) を管理するテナント B 管理者 0104、および、テナント C (0109) を管理するテナント C 管理者 0105 が存在する例を示している。あるテナントのテナント管理者は、別のテナント管理者を設定したり、削除したりすることが可能である。

【0015】

テナント管理者は、クライアント端末 0240 からウェブブラウザを介して管理するテナントにアクセスし、そのテナントを利用する複数の利用者アカウントを作成することができる。図 1 では、テナント A 管理者 0103 が、技術営業 A1 (0110)、技術営業 A2 (0111) から、技術営業 An (0112) までのアカウントを作成した例を示している。それぞれのアカウントには、ディスクリプションを格納しておくディスクリプションプールが割り当てられる。ここで、ディスクリプションとは、カスタムアプリケーションの振る舞い（動作手順）を記述したファイル（記述ファイル）であり、オーサリングシステム 0101 を用いて作成することができる。なお、カスタムアプリケーションの構成やディスクリプションの詳細については後述する。

20

【0016】

図 1 では、技術営業 A1 (0110)、技術営業 A2 (0111) から、技術営業 An (0112) までのアカウントに、それぞれディスクリプションプールが割り当てられている様子を示している。各アカウントにはそれぞれ、ディスクリプションプール A1 (0113)、ディスクリプションプール A2 (0114) から、ディスクリプションプール An (0115) (n は自然数) までが割り当てられている。なお、図 1 上では割愛するが、テナント B 管理者やテナント C 管理者も同様に、各テナントにおける利用者のアカウントを適宜作成することができる。また、テナント管理者は、共有ディスクリプションプールをテナント中に配置することで、1つのディスクリプションをテナント内の複数のアカウントで共有することを許可することができる。図 1 では、テナント A (0107) 内に共有ディスクリプションプール A (0116) が配置することで、テナント A (0107) 内でディスクリプションを共有する例を示している。テナント内のディスクリプションを共有する同様の構成として、テナント B (0108) 内には共有ディスクリプション

30

40

【0017】

なお、本実施例では、テナント内のすべてのアカウントが読み書き可能をおこなえるように共有ディスクリプションプールの設定がなされているものとして説明する。なお、アカウントに応じて書き込みや読み取りが制限されるように共有ディスクリプションプールの設定を変更しても構わない。また、オーサリングシステム 0101 は、すべてのテナントのアカウントがアクセス可能な共有ディスクリプションプールを備えていてもよい。この、共有ディスクリプションプールについても、アカウントに応じた制限を設定してもよい。

50

## 【 0 0 1 8 】

<システムの利用フロー>

図 9 を用いて、オーサリングシステムの主要な利用フローを説明する。図 9 はオーサリングシステムの利用フローを示す図である。

## 【 0 0 1 9 】

ステップ 9 0 1 (以降、S 9 0 1 等と表記する)において、システム管理者 0 1 0 2 は、オーサリングシステム 0 1 0 1 に対してテナント作成処理・再生アプリケーションの登録処理を行う。オーサリングシステム 0 1 0 1 には、カスタムアプリケーション用の再生アプリケーションの他、通常のアプリケーションの情報が登録されている。これらのアプリケーションは一意に識別可能な情報に基づいて他のアプリケーションと区別可能に管理

10

## 【 0 0 2 0 】

システム利用者 0 2 0 2 の指示によって、クライアント端末 0 2 4 0 では、WEB ツールを使用したディスクリプションの生成処理、登録処理がおこなわれる (S 9 0 4 ~ S 9 0 7)。オーサリングシステム 0 1 0 1 に登録されたディスクリプションは、一意に識別可能な情報に基づいて他のディスクリプションと区別可能に管理される。ディスクリプションの生成処理・登録処理の詳細については図 1 4 を用いて後述する。

20

## 【 0 0 2 1 】

S 9 0 8 において、クライアント端末 0 2 4 0 は、登録されたディスクリプションを使用するカスタムアプリケーションについて、インストールコードの発行依頼をオーサリングシステム 0 1 0 1 に送信する。

## 【 0 0 2 2 】

S 9 1 4 でインストールコード発行依頼を受信したオーサリングシステム 0 1 0 1 は、後述するディスクリプション署名付与 0 2 1 1 において署名用鍵情報 0 2 1 8 を用いることで、ディスクリプション 0 2 1 9 に署名情報を付与する。なお、署名用鍵情報 0 2 1 8 は、後述の再生アプリケーションの署名に使用する鍵情報とは異なる情報である。再生アプリケーションの署名に使用する鍵情報はシステム開発者 0 2 0 1 の開発環境において管理される。

30

## 【 0 0 2 3 】

S 9 1 5 において、オーサリングシステム 0 1 0 1 は、後述するカスタムアプリパッケージング 0 2 1 2 において、署名付与したディスクリプションと署名付与済みの再生アプリケーションのパッケージング処理を行う。

## 【 0 0 2 4 】

S 9 0 9 において、オーサリングシステム 0 1 0 1 は、インストールコード発行システム 0 2 1 4 においてインストールコードを発行し、パッケージングしたカスタムアプリケーションをカスタムアプリ配信システム 0 2 1 3 に登録する。発行されたインストールコードは、電話や Eメールなどの伝達手段によって利用者 0 2 0 2 から顧客 0 2 3 0 に伝達

40

## 【 0 0 2 5 】

S 9 1 0 において、顧客 0 2 3 0 が MFP 0 2 0 3 にインストールコードを入力すると、MFP 0 2 0 3 はオーサリングシステム 0 1 0 1 にインストールコードを送信する。

## 【 0 0 2 6 】

S 9 1 1 において、オーサリングシステム 0 1 0 1 は、カスタムアプリ配信システム 0 2 1 3 において、受信したインストールコードに対応したカスタムアプリケーションを MFP 0 2 0 3 に提供する。

## 【 0 0 2 7 】

S 9 1 2 において、カスタムアプリケーションを受信した MFP 0 2 0 3 は、カスタム

50

アプリケーションのインストールを行う。正常にインストールが完了すると、S 9 1 3 においてアプリケーションが実行され、アプリケーションが提供する機能を顧客 0 2 3 0 が利用可能な状態となる。

【 0 0 2 8 】

< オーサリングシステムのソフトウェア構成 >

次に、オーサリングシステム 0 1 0 1 のソフトウェア構成について説明する。図 2 ( A ) は、オーサリングシステムのソフトウェア構成を示す図である。

【 0 0 2 9 】

同図において、オーサリングシステム 0 1 0 1、テナント A ( 0 1 0 7 )、利用者アカウントである技術営業 A 1 ( 0 1 1 0 )、および、ディスクリプションプール A 1 ( 0 1 1 3 ) は、図 1 と同一のものである。 10

【 0 0 3 0 】

オーサリングシステム 0 1 0 1 は、フロントエンドリソース 0 2 0 4、バックエンドプログラム 0 2 0 5、再生アプリケーションリソース 0 2 0 6 といったソフトウェアモジュールを備える。

【 0 0 3 1 】

フロントエンドリソース 0 2 0 4 は、ディスクリプションエディタ 0 2 0 7、フレームワークエミュレータ 0 2 0 8、インストールコード発行画面 0 2 0 9 などのリソースを収容する。それぞれのリソースは、J a v a S c r i p t 等で記述されたプログラムで、利用者がオーサリングシステム 0 1 0 1 にアクセスする際に使う w e b ブラウザ上 ( ウェブブラウザ上 ) で動作する。 20

【 0 0 3 2 】

ディスクリプションエディタ 0 2 0 7 は、各テナントの利用者がディスクリプションの作成・編集等を行う際に w e b ブラウザ上で動作するプログラム ( 編集プログラム ) である。ディスクリプションエディタ 0 2 0 7 で作成されたディスクリプションは、利用者アカウントに紐づけられたディスクリプションプールに保存することができる。図 2 では、利用者である技術営業 A 1 ( 0 2 0 2 ) が、オーサリングシステム 0 1 0 1 にログインし、フロントエンドリソース 0 2 0 4 にアクセスしている例を示している。また、利用者である技術営業 A 1 ( 0 2 0 2 ) がフロントエンドリソース 0 2 0 4 に収容されているディスクリプションエディタ 0 2 0 7 を使って作成したディスクリプション 0 2 1 9 が、ディスクリプションプール A 1 ( 0 1 1 3 ) に保存された様子を示している。前述した通り、ディスクリプションプール A 1 ( 0 1 1 3 ) は、テナント A ( 0 1 0 7 ) に作成された利用者アカウントである技術営業 A 1 ( 0 1 1 0 ) に紐づけられている。 30

【 0 0 3 3 】

フレームワークエミュレータ 0 2 0 8 は、ディスクリプションエディタ 0 2 0 7 上で作成中 ( 編集中 ) ・作成済みのディスクリプションの振る舞いを同エディタ内で確認する際に呼び出されるプログラムである。フレームワークエミュレータ 0 2 0 8 は、W e b ブラウザ上で動作するよう構成されている。フレームワークエミュレータ 0 2 0 8 は、デバイスにおけるアプリケーションフレームワークの動作をエミュレーションし、作成されたディスクリプションを含むカスタムアプリケーションの振る舞いを画面情報として出力することで可視化する。 40

【 0 0 3 4 】

インストールコード発行画面 0 2 0 9 は、カスタムアプリケーションをデバイスにインストールする際に使われるインストールコードを、利用者が使う w e b ブラウザ上に表示するプログラムである。

【 0 0 3 5 】

バックエンドプログラム 0 2 0 5 は、ディスクリプションナンバー管理 0 2 1 0、ディスクリプション署名付与 0 2 1 1、カスタムアプリパッケージング 0 2 1 2、カスタムアプリ配信システム 0 2 1 3、インストールコード発行システム 0 2 1 4 などを収容する。

【 0 0 3 6 】

バックエンドプログラム 0205 に収容されたそれぞれのプログラムは、オーサリングシステム 0101 がホストされるクラウド上のサーバ装置 0220 で実行され、フロントエンドリソース 0204 の各プログラムと協調して動作する。

【0037】

ディスクリプションナンバー管理 0210 は、ひとつひとつのディスクリプションを識別するためのユニークなシリアルナンバーを発行するプログラムである。ディスクリプションナンバー管理 0210 は、利用者がディスクリプションエディタ 0207 を使って新たなディスクリプションの作成を開始するとき呼び出される。また、利用者が過去に作ったディスクリプションを破棄するときにも呼び出される。破棄したいディスクリプションがカスタムアプリケーションとして使用されていないならば、該ディスクリプションの作成時に割り当てたシリアルナンバーを再利用可能な状態に戻す等の処理を受け持つ。

10

【0038】

ディスクリプション署名付与 0211 は、ディスクリプションが改ざんされていないことを保証するための署名を該ディスクリプションに付与するプログラムである。署名の付与は、例えば、作成されたディスクリプションを含むカスタムアプリケーションのインストールパッケージの生成時におこなわれる。ディスクリプション署名付与 0211 は、その署名の際に用いる署名用鍵情報 0218 を備える。図 2 では、ディスクリプションプール A1 (0113) に保存されているディスクリプション 0219 がディスクリプション署名付与 0211 に引用されている例を示している。

【0039】

カスタムアプリパッケージング 0212 は、カスタムアプリケーションのインストールパッケージ化を行うプログラムである。インストールパッケージでは、ディスクリプション署名付与 0211 によって生成された署名付きディスクリプションと、後述する特定の再生アプリケーションとをひとつにまとめられている。インストールパッケージ化することによって、任意のデバイスにカスタムアプリケーションをインストールすることが可能になる。なお、インストールパッケージの構造の詳細については後述する。

20

【0040】

カスタムアプリ配信システム 0213 は、インストールコード発行画面 0209 で発行されたインストールコードに予め紐づけられた、ひとつ以上のカスタムアプリケーションのインストールパッケージを、デバイスに配信するシステムのプログラムである。ここで、ひとつ以上のカスタムアプリケーションのインストールパッケージをカスタムアプリケーションのインストールパッケージアーカイブと呼ぶこととする。カスタムアプリ配信システム 0213 は、インストールコードを伴った配信要求をデバイスから受け取ると、そのインストールコードに紐づけられたカスタムアプリケーションのインストールパッケージアーカイブをデバイスに送信する。図 2 では、デバイスとして MFP 0203 がオーサリングシステム 0101 のバックエンドプログラム 0205 に接続されている例を示しており、カスタムアプリケーションのインストールパッケージは MFP 0203 に順次送信されることになる。

30

【0041】

インストールコード発行システム 0214 は、インストールコード発行画面 0209 と協調して動作し、利用者によって選択されたひとつ以上のカスタムアプリケーションに対するインストールコードを発行するプログラムである。また、インストールコード発行システム 0214 は、選択されたひとつ以上のカスタムアプリケーション（すなわち、カスタムアプリケーションのインストールパッケージアーカイブ）と、発行されたインストールコードとの関係を示すテーブルを内部に保持する。詳細は後述する。

40

【0042】

ここで、カスタムアプリケーションのインストールパッケージアーカイブは、オンデマンドで生成されるように構成してもよいし、事前に生成しておくように構成してもよい。もしオンデマンドで生成されるように構成したならば、カスタムアプリ配信システム 0213 がインストールコードを伴った配信要求をデバイスから受け取った後に、カスタムア

50

プリパッケージング 0 2 1 2 によって生成される。一方、事前に生成しておくように構成したならば、インストールコード発行システム 0 2 1 4 がインストールコードを発行したときにカスタムアプリパッケージング 0 2 1 2 によって生成される。そして、インストールコード発行システム 0 2 1 4 内部に発行されたインストールコードとの関係を示すテーブルと共に保持される。

#### 【 0 0 4 3 】

再生アプリケーションリソース 0 2 0 6 は、作成されたディスクリプションを再生するためのアプリケーションであって、かつ、事前に署名が付加されたパッケージである署名付きアプリケーションを収容する。再生アプリケーションリソース 0 2 0 6 で管理される再生アプリケーションはひとつとは限らない。例えば、バージョン毎に別々に管理されるなど複数を管理されてもよい。図 2 では、事前に署名が付与された再生アプリケーション ver 1 ( 0 2 1 5 )、再生アプリケーション ver 2 ( 0 2 1 6 ) から再生アプリケーション ver N ( 0 2 1 7 ) を収容している例を示している。各再生アプリケーションには対応する署名「署名 1」「署名 2」から「署名 N」が付与される。それぞれの再生アプリケーションは、システム開発者 0 2 0 1 によって開発される。また、再生アプリケーションの署名は、その署名に用いられる鍵情報が厳重に管理されたシステム開発者 0 2 0 1 の開発環境で事前に付与される。そして、事前に署名が付与された再生アプリケーションは、システム開発者 0 2 0 1 によって再生アプリケーションリソース 0 2 0 6 へ適時アップロードされる。加えて、再生アプリケーションリソース 0 2 0 6 に収容されるそれぞれの再生アプリケーションは、カスタマイズ可能な情報を備える。利用者は、ディスクリプションエディタ 0 2 0 7 でディスクリプションの作成を開始するとき再生アプリケーションのバージョンを選択する。そして、ディスクリプションエディタ 0 2 0 7 は、選択されたバージョンの再生アプリケーションから、同再生アプリケーション固有のカスタマイズ可能な情報を取得する。これにより、ディスクリプションエディタ 0 2 0 7 は、カスタマイズに必要なメニュー項目のような情報を静的に保持しておく必要がなくなる。

#### 【 0 0 4 4 】

< サーバ装置 >

図 2 ( A ) で説明したオーサリングシステムのソフトウェア構成を実行するためのハードウェア構成について説明する。オーサリングシステム 0 1 0 1、クラウド上に構築するシステムであるため、複数の装置によって負荷を分散するように構成される。本実施例では説明を簡単にするため、1つのサーバ装置 0 2 2 0 で処理するケースについて説明する。図 2 ( B ) は、オーサリングシステムに用いられるサーバ装置のハードウェア構成を示す図である。

#### 【 0 0 4 5 】

サーバ装置 0 2 2 0 はオーサリングシステムを実行するための情報処理装置であり、コントローラユニット 2 5 0 を備える。コントローラユニット 2 5 0 は、CPU 2 5 2、RAM 2 5 3、HDD 2 5 5、ROM 2 5 6、ネットワーク I / F 2 5 7 を備える。CPU 2 5 2、RAM 2 5 3、HDD 2 5 5、ROM 2 5 6、ネットワーク I / F 2 5 7 は、システムバス 2 5 9 を介して接続されている。

#### 【 0 0 4 6 】

CPU 2 5 2 は、サーバ装置の動作を制御するものであり、RAM 2 5 3 に格納されたプログラムに基づいて動作する。ROM 2 5 6 はブート ROM であり、システムのブートプログラムが格納されている。ネットワーク I / F 2 5 7 は LAN に接続されて、ネットワーク経由での各種情報の入出力を司る。HDD 2 5 5 は、システムソフトウェアをはじめとした各種プログラム等が格納されている。HDD 2 5 5 に格納されたプログラムが RAM 2 5 3 にロードされ、CPU 2 5 2 はこれに基づいてサーバ装置 0 2 2 0 の動作を制御する。

#### 【 0 0 4 7 】

上述した管理者や利用者のアカウント情報、図 2 ( A ) 記載のテナント A ( 0 1 0 7 ) に代表される各テナントに関する情報などは、すべて HDD 2 5 5 上で保持・管理される

10

20

30

40

50

。図 2 ( A ) 記載のフロントエンドリソース 0 2 0 4、バックエンドプログラム 0 2 0 5、再生アプリケーションリソース 0 2 0 6 など、HDD 2 5 5 上に保存される。加えて、図示しないが、図 1 の説明で記述した管理者および利用者のユーザ認証の仕組みを実現するプログラムなども HDD 2 5 5 上に保持されている。ユーザ認証の仕組みを実現するプログラムは、サーバ装置 0 2 2 0 を起動すると自動的に RAM 2 5 3 上にロードされ、CPU 2 5 2 によって実行される。例えば、図 2 ( A ) 記載のシステム開発者 0 2 0 1 や利用者 ( 技術営業 A 1 ) 0 2 0 2 がオーサリングシステム 0 1 0 1 にアクセスしてログインを試みると、その要求はネットワーク I / F 2 5 7 を介してユーザ認証のプログラムに伝えられる。そして、ユーザ認証のプログラムによって、システム開発者 0 2 0 1 や利用者 ( 技術営業 A 1 ) 0 2 0 2 が使用している PC 等の web ブラウザ上にログイン画面が表示される。フロントエンドリソース 0 2 0 4 に収容されるディスクリプションエディタ 0 2 0 7 など、HDD 2 5 5 上に保持されている。例えば利用者 ( 技術営業 A 1 ) 0 2 0 2 がオーサリングシステム 0 1 0 1 にログインしてディスクリプションエディタの URL にアクセスすると、その要求はネットワーク I / F 2 5 7 を介してサーバ装置 0 2 2 0 に伝わる。そして、ディスクリプションエディタ 0 2 0 7 のプログラムは、利用者 ( 技術営業 A 1 ) 0 2 0 2 が使用している PC の web ブラウザに返され、web ブラウザ上で動作する。フレームワークエミュレータ 0 2 0 8 やインストールコード発行画面 0 2 0 9 についても同様の処理がなされる。

10

#### 【 0 0 4 8 】

バックエンドプログラム 0 2 0 5 に収容されるディスクリプションエディタ 0 2 0 7 など、HDD 2 5 5 上に保持されている。前述の通り、バックエンドプログラム 0 2 0 5 はフロントエンドリソース 0 2 0 4 の各プログラムと協調して駆動される。バックエンドプログラム 0 2 0 5 に収容されたディスクリプションナンバー管理 0 2 1 0 をはじめとする各モジュールは、駆動を要求されると HDD 2 5 5 から RAM 2 5 3 上にロードされ、CPU 2 5 2 によって実行される。

20

#### 【 0 0 4 9 】

< M F P >

MFP 0 2 0 3 は情報を処理する情報処理装置であり、画像を処理する画像処理装置であり、シートに画像を形成する画像形成装置である。図 3 ( A ) は、MFP のハードウェア構成を示す図である。図 3 ( B ) は、MFP のソフトウェア構成を示す図である。

30

#### 【 0 0 5 0 】

MFP 0 2 0 3 はコントローラユニット 3 0 0 と、プリンタ 3 9 5 と、スキャナ 3 7 0 と、操作部 3 1 2、USB ストレージ 3 1 4 とを備える。コントローラユニット 3 0 0 は、スキャナ 3 7 0 で読み取られた画像データをプリンタ 3 9 5 により印刷出力するコピー機能を実現するための制御を行う。スキャナ 3 7 0 は、原稿から画像を読み取る画像読取部である。プリンタ 3 9 5 は、シート ( 用紙 ) に画像を形成する画像形成部である。操作部 3 1 2 はユーザの指示を受け付ける受付部であり、ユーザに情報を表示する表示部である。

#### 【 0 0 5 1 】

コントローラユニット 3 0 0 は、CPU 3 0 1 を有し、CPU 3 0 1 は、ROM 3 0 3 に格納されているブートプログラムによりオペレーションシステム ( OS ) を立ち上げる。

40

#### 【 0 0 5 2 】

CPU 3 0 1 は、この OS 上で、ストレージ 3 0 4 に格納されているプログラムを実行し、これによって各種処理を実行する。この CPU 3 0 1 の作業領域としては RAM 3 0 2 が用いられる。RAM 3 0 2 は、作業領域を提供するとともに、画像データを一時記憶するための画像メモリ領域を提供する。ストレージ 3 0 4 は、プログラムや画像データを格納する。

#### 【 0 0 5 3 】

CPU 3 0 1 には、システムバス 3 0 7 を介して、ROM 3 0 3 および RAM 3 0 2、

50

操作部 I / F (操作部インターフェース) 306、ネットワーク I / F (ネットワークインターフェース) 310、USB ホスト I / F 313、画像バス I / F (画像バスインターフェース) 305 が接続される。操作部 I / F 306 は、タッチパネルを有する操作部 312 とのインターフェースであり、操作部 312 に表示すべき画像データを操作部 312 に対して出力する。また、操作部 I / F 306 は、操作部 312 においてユーザにより入力された情報を CPU 301 に送出する。ネットワーク I / F 310 は、MFP 0203 を LAN に接続するためのインターフェースである。USB ホスト I / F 313 は、USB ストレージ 314 と通信するインターフェース部である。USB ホスト I / F 313 は、ストレージ 304 に格納されているデータを USB ストレージ 314 に記憶させるための出力部である。また、USB ホスト I / F 313 は、USB ストレージ 314 に格納されているデータを入力し、CPU 301 にそれを伝える。USB ストレージ 314 は、データを格納する外部記憶装置であり、USB ホスト I / F 313 に対して着脱可能である。USB ホスト I / F 313 には、USB ストレージ 314 を含む複数の USB デバイスが接続可能である。RTC 315 は、現在時刻を制御する。この RTC 315 で制御する時刻情報はジョブ投入時間の記録などに利用する。画像バス I / F 305 は、システムバス 307 と、画像データを高速で転送する画像バス 308 とを接続し、データ形式を変換するためのバスブリッジである。画像バス 308 は、PCI バスまたは IEEE 3394 等によって構成される。画像バス 308 上には、デバイス I / F 320、スキャナ画像処理部 380、プリンタ画像処理部 390 が設けられる。デバイス I / F 320 には、スキャナ 370 およびプリンタ 395 が接続され、デバイス I / F 320 は、画像データの同期系 / 非同期系の変換を行う。スキャナ画像処理部 380 は、入力画像データに対し補正、加工、編集を行う。プリンタ画像処理部 390 は、プリント出力画像データに対してプリンタ 395 に応じた補正、解像度変換などを行う。

10

20

#### 【0054】

ストレージ 304 に記憶されているプログラムを、CPU 301 が RAM 302 にロードし、プログラムを実行することで図 3 の各モジュールが実現される。図 3 (B) は MFP のソフトウェア構成を示す図である。

#### 【0055】

MFP 0203 は、そのソフトウェア構成として、サブレットサービス 354、UI 制御部 353、アプリケーション実行制御部 355、ストレージ制御部 356、アプリケーション実行制御部 355、標準機能制御部 351、タイマー管理部 359 を備える。

30

#### 【0056】

サブレットサービス 354 は、ネットワーク I / F 310 を通して HTTP アクセスされた際に、そのリクエストを受け付ける。そして、アクセスされた URL によってモジュール (アプリケーション管理部 352 か標準機能制御部 351) に処理を振り分けるモジュールである。

#### 【0057】

UI 制御部 353 は、操作部 312 に画面を表示し、ユーザからの操作を受け付けて、その操作情報を適切なモジュール (アプリケーション管理部 352 か標準機能制御部 351) に通知を行うモジュールである。アプリケーション管理部 352 は、インストールされている拡張アプリケーション 440 のインストールやの起動等の管理を行うモジュールである。

40

#### 【0058】

アプリケーション実行制御部 355 は、アプリケーション管理部 352 で起動されたアプリケーションの実行制御を行うモジュールである。具体的には、VM スレッド 415、VM システムサービス 420、VM 430、拡張アプリケーション 440 を制御する。

#### 【0059】

ストレージ制御部 356 は拡張アプリケーション 440 のプログラムや、設定情報を記録管理するモジュールである。

#### 【0060】

50

アプリケーション実行制御部 3 5 5 は、ストレージ制御部 3 5 6 にアクセスしてアプリケーションのプログラムを読み出す。また、各モジュールは、ストレージ制御部 3 5 6 にアクセスし、設定値の参照、設定を行う。

【 0 0 6 1 】

標準機能制御部 3 5 1 は、M F P 0 2 0 3 の標準機能であるコピーや F A X の制御や、その他の M F P 0 2 0 3 に必要な制御（例えば U S B ホストパス I / F 3 1 3 の制御）を行うモジュールである。また、標準機能制御部 3 5 1 は、実行したコピーや F A Xなどをジョブとして管理し、そのジョブの実行結果の記録をジョブ実行履歴制御部 3 5 7 に依頼する。ジョブ実行履歴制御部 3 5 7 は、ジョブの実行結果を履歴としてストレージに残すため、ストレージ制御部 3 5 6 に履歴の記録を依頼する。

10

【 0 0 6 2 】

タイマー管理部 3 5 9 は、R T C 3 1 5 から現在時刻情報を取得し、規定の時刻が経過したことを検知し、アプリケーション管理部 3 5 2 に通知する。アプリケーション管理部 3 5 2 は、タイマー管理部 3 5 9 から通知を受け、アプリケーションのタイマー処理の実行をアプリケーション実行制御部 3 5 5 に依頼する。

【 0 0 6 3 】

ここで、O S、V M、アプリケーションの関係について詳細に説明する。図 4 は、M F Pにおけるソフトウェアの階層構造を示す図である。

【 0 0 6 4 】

オペレーティングシステムである O S 4 0 1 上では、プリンタや F A X、スキャナと言った画像処理ユニットを制御するためのネイティブプログラム 4 1 0 が動作する。また、オペレーティングシステムである O S 4 0 1 上では、拡張アプリケーション 4 4 0 の実行環境である仮想マシン ( V i r t u a l M a c h i n e : V M ) 4 3 0 が動作する。V M 4 3 0 は、拡張アプリケーション 4 4 0 を制御するプログラムを理解し実行するモジュールである。拡張アプリケーション 4 4 0 は、必ず V M 4 3 0 上で動作する。V M 4 3 0 で動作するプログラムは、C P U 3 0 1 で動作する命令とは異なり、V M 4 3 0 専用の命令で動作する。この命令をバイトコードと呼ぶ。一方で C P U 専用命令をネイティブコードと呼ぶ。V M がこのバイトコードを逐次解釈し処理することで、C P U 3 0 1 上ではネイティブコードと、V M のバイトコードが動作する。V M には、バイトコードをそのまま逐次解釈処理するタイプと、バイトコードをネイティブコードに変換して実行するタイプがある。本実施例の V M 4 3 0 は前者のタイプであるが、後者のタイプであってもよい。一般的に C P U の種別が異なると、C P U で動作する命令に互換性が無い様に、V M が異なると V M で動作する命令も互換性が無い。本実施例では、拡張アプリケーションは L u a 言語で構成されていることを前提とするが、他の言語でも構わない。L u a 言語は、L u a 言語で記載されたスクリプトを L u a 言語専用のバイトコードに変換し、L u a 言語専用の V M 4 3 0 上で実行する。もちろん L u a 言語のバイトコードは、C P U 3 0 1 が直接実行可能な命令コードとは異なるため、L u a 言語のバイトコードを C P U 3 0 1 で直接実行できない。この L u a 言語の V M 4 3 0 は、L u a 言語のバイトコードを逐次解釈実行するソフトウェアモジュールとして、C P U 3 0 1 の命令コードで構成されている。もちろん、L u a 言語のバイトコードを解釈実行する専用のハードウェアとして構成していてもよい。

20

30

40

【 0 0 6 5 】

ネイティブプログラム 4 1 0 内には、プリンタや F A X、スキャナと言った画像処理ユニットを制御するためのネイティブスレッド 4 1 4 と、V M 4 3 0 を動かすための V M スレッド 4 1 5 がある。V M スレッドは V M 4 3 0 の数に対応する数存在する。ここでは、V M スレッド A - 1 ( 4 1 1 )、V M スレッド A - 2 ( 4 1 2 )、V M スレッド B - 1 ( 4 1 3 ) の 3 つのスレッドが生成されている。

【 0 0 6 6 】

V M システムサービス 4 2 0 は、各拡張アプリケーション 4 4 0 から共通利用されるユーティリティライブラリである。拡張アプリケーション 4 4 0 から V M システムサービス

50

420の機能呼び出すことで、拡張アプリケーション440の開発スピードを加速することができる。また、VMシステムサービス420は、MFP0203の各モジュールへアクセスすることができる。VMシステムサービス420には、VMとして最低限動作させる標準VMシステムサービス421と、MFP0203の各モジュールにアクセスや、OSの機能を提供する拡張VMシステムサービス422がある。標準VMシステムサービス421は、拡張アプリケーション440をロードする機能も含む。VMシステムサービス420は、VM430が拡張アプリケーション440内のバイトコードで指示されているAPIを実行することで、APIに関連づけられているVMシステムサービス420が呼び出される。拡張VMシステムサービス422には、操作部312の画面に任意の描画を行う描画APIや、MFP0203のコピーやスキャンなどの任意のジョブの実行を依頼するジョブ制御APIが提供されている。

#### 【0067】

VM430は、拡張アプリケーション440を実行する。VM430は、拡張アプリケーション440のスレッド毎に生成される。この図では拡張アプリケーションA441で2つのスレッドを動かすためのVM A-1(431)と、VM A-2(432)を生成している。また、この図では、拡張アプリケーションB(442)で1つのスレッドを動かすためのVM B-1を生成している。

#### 【0068】

また、MFP0203の操作部312に表示されるメインメニュー画面には、拡張アプリケーション440ごとのアイコンが表示される。このアイコンをユーザが選択したことを、操作部312を通じて操作部I/F306が検知すると、操作部I/F306はその旨をCPU301に送信する。その旨を受け取ったCPU301はユーザによって選択された拡張アプリケーション440を起動する。

#### 【0069】

以降の説明において、拡張アプリケーション440をMFP0203上で動かすこの仕組み(VMシステムサービス420、VM430)を、アプリケーションフレームワークと記す。

#### 【0070】

<クライアント端末>

本システムに登場する一般的なクライアント端末を代表して、利用者0202が利用するクライアント端末0240について説明する。なお、ここで説明する構成をシステム管理者0102、テナントA管理者0103、テナントB管理者0104、テナントC管理者0105、システム開発者0201等が用いるクライアント端末に適用してもよい。図5(A)はクライアント端末のハードウェア構成を示す図である。

#### 【0071】

クライアント端末0240は、オーサリングシステム0101にアクセスする外部端末である。クライアント端末0240は、コントローラユニット500と、操作部510、表示部512とを備える。コントローラユニット500は、システムバス509によって各ハードウェアと接続され、情報をやり取りしている。コントローラユニット500は、ROM506あるいはHDD505に記録されているOSあるいはWebブラウザのようなプログラムをRAM503に展開する。展開されたプログラムはCPU502によって逐次処理される。操作部510は、ユーザの操作を受け付ける受付部である。操作部510は、操作部I/F501に接続されており、操作部510で受け付けたユーザ操作情報は操作部I/F501を通してCPU502に通知される。この操作情報は、CPU502上で動作するプログラムによって処理される。これによってユーザの操作が処理され、操作結果が表示部I/F504に接続されている表示部512に表示され、あるいはネットワークI/F507に接続されているネットワークに通知される。

#### 【0072】

本実施例では、このクライアント端末0240上でWebブラウザを実行し、そのWebブラウザで図1に示すオーサリングシステム0101にアクセスする。このクライアン

ト端末 0240 は一般的な PC でも良いし、スマートフォンのような携帯端末でも良い。

【0073】

次に、クライアント端末 0240 が Web ブラウザ上で、ディスクリプションエディタを実行するケースについて説明する。図 6 は、ディスクリプションエディタを表示している Web ブラウザのブロック図である。

【0074】

Web ブラウザ 601 は、Web サーバに接続するためのソフトウェアであり、ウェブページを表示したり、プログラムを実行したりすることができる。Web ブラウザ 601 は、Web ブラウザ自体を動かすためのネイティブプログラムである Web ブラウザネイティブプログラム 602 上において JavaScript VM 603 を実行可能である。JavaScript VM 603 は、Web ブラウザ上で JavaScript を動かすためのプログラムである。

10

【0075】

ディスクリプションエディタ 604 は、JavaScript で構成されたプログラムであり、図 2 で紹介したディスクリプションエディタ 0207 の一例である。ディスクリプションエディタ 604 は、JavaScript VM 603 上で実行される。ディスクリプションエディタ 604 は、アプリケーションの編集環境を提供するプログラムである。

【0076】

また、ディスクリプションエディタ 604 は、Application Framework Emulator (アプリケーションフレームワークエミュレータ 605) として機能する。アプリケーションフレームワークエミュレータ 605 は、ブラウザ上で JavaScript と異なる言語のプログラムを実行するためのアプリケーションフレームワークのエミュレーションを実行する。アプリケーションフレームワークエミュレータ 605 は、図 2 で紹介したフレームワークエミュレータ 0208 の一例である。本実施例において、アプリケーションフレームワークエミュレータ 605 は、MFP 0203 上で実行されるアプリケーションの言語と同言語の実行環境についてエミュレーションをおこなう。また、アプリケーションフレームワークエミュレータ 605 は、VM システムサービスエミュレータ 606 と、VM エミュレータ 607 とを含む。これらは、図 2 の VM 430、VM システムサービス 420 の動作を JavaScript 上でエミュレートするものであり、JavaScript で構成されるプログラムである。

20

30

【0077】

再生アプリケーション 608 は、ユーザによってカスタマイズされたディスクリプションに基づき動作するアプリケーション (再生プログラム) である。再生アプリケーション 608 は、MFP 0203 のアプリケーションフレームワーク上で実行されるアプリケーションと同様に構成されている。図 6 では、再生アプリケーション 608 がアプリケーションフレームワークエミュレータ 605 上で実行される様子を示している。再生アプリケーション 608 内には、VM システムサービスからのカスタマイズ可能な情報問い合わせの処理を行うカスタマイズ可能な情報制御 610 が含まれる。また、再生アプリケーション 608 内には、カスタマイズ可能な情報の定義 611 と、与えられたディスクリプションを再生するディスクリプション再生制御 609 が含まれる。

40

【0078】

ディスクリプション 612 はユーザが編集したカスタマイズ情報を、再生アプリケーションで再生可能な形式で保持するデータである。ディスクリプションエディタ 604 が提供する編集画面に対してユーザが編集内容を指示すると、ディスクリプションエディタ 604 によってディスクリプション 612 が生成される。ディスクリプションエディタ 604 は、アプリケーションフレームワークエミュレータ 605 上で動作する再生アプリケーション 608 において生成されたディスクリプション 612 を再生する。これにより、カスタマイズされたアプリケーションの動作確認画面を Web ブラウザ上でユーザに提供する。

50

## 【0079】

本実施例では、ディスクリプションエディタ604を構成するプログラム言語としてJavaScriptを利用しているが、他のWebブラウザ上で直接実行可能なコード、例えばWebAssemblyで構成してもよい。

## 【0080】

また、ディスクリプションエディタ604で実行する再生アプリケーションは、MFP0203で実行される再生アプリケーションと完全に同一でなくてもよい。例えば、再生アプリケーションのカスタマイズに直接関係するディスクリプション再生制御部分だけを切り出した動作確認用アプリケーションをロードする構成であってもよい。

## 【0081】

次に、ディスクリプションエディタ604の詳細について説明する。図5(B)はクライアント端末が実行するディスクリプションエディタのソフトウェア構成を示す図である。

## 【0082】

ディスクリプションエディタ604は、Webブラウザが所定のURLにアクセスしてサーバからダウンロードし、Webブラウザ内で実行されるプログラムであり、JavaScriptで記述される。ディスクリプションエディタ604は、全体制御部551、画面構成部552、フレームワークエミュレータ制御部556、サーバ通信部557、ディスクリプション生成部555、カスタマイズ部品情報管理部553、カスタマイズ部品情報編集部554を備える。

## 【0083】

全体制御部551は、このディスクリプションエディタ全体を制御する構成である。この全体制御部551が各ブロックと連携して動作することで、ディスクリプションエディタ604を制御する。画面構成部552は、このディスクリプションエディタ604の画面構成を制御する構成である。この画面構成部552は、図7で示す画面を構成するHTML element、css(画面情報)を更新する。また、ユーザの操作を受け付け、操作内容(編集操作)を全体制御部551に通知する。全体制御部551は、その操作通知を受けて適宜制御を行う。その操作通知がアプリケーション画面イメージの表示領域701に対する操作情報の場合、全体制御部551はフレームワークエミュレータ制御部556に操作情報を通知する。例えば、アプリケーション画面イメージの表示領域701をクリックした場合、アプリケーション実行画面の所定の場所をクリックされたことを示す情報がフレームワークエミュレータ制御部556に通知される。フレームワークエミュレータ制御部556は、カスタマイズしたディスクリプションを再生する再生アプリケーションをWeb上で実行するための構成である。再生アプリケーションは、MFP0203内のアプリケーションフレームワーク上で動作するように構成されているため、Webブラウザ上では直接実行することはできない。このフレームワークエミュレータは、MFP0203向けのアプリケーションをWeb上で実行するためのエミュレーションを行う。このフレームワークエミュレータ制御部556は、全体制御部551からの画面操作通知を受け、エミュレータ内で実行するアプリケーションに通知する。操作情報の通知を受けたアプリケーションは、操作に対する画面更新を行う。この画面更新は、アプリケーションフレームワーク内の画面描画APIによって実現されている。このAPIは、フレームワークエミュレータによって、Webブラウザ内の画像イメージレンダリングAPIに置き換えられ、アプリケーションの表示画面が図7のアプリケーション画面イメージの表示領域701に表示される。カスタマイズ部品情報編集部554は、再生アプリケーションからカスタマイズ可能な部品情報を取得し、その情報を元にディスクリプションエディタ604で利用可能なフォーマットに変換し保持する構成である。この部品情報を元に、画面構成部552は部品リスト702、カスタマイズ設定情報の表示領域706を構成する。カスタマイズ部品情報管理部553は、現在設定されているカスタマイズ部品の設定情報を管理する構成である。この情報を元に、画面構成部552はカスタマイズ設定情報の表示領域706を構成する。ディスクリプション生成部555は、カスタマイズ部品

10

20

30

40

50

情報編集部 554 で保持されている現在の設定情報を元にディスクリプションを生成する構成である。このディスクリプションはフレームワークエミュレータ制御部 556 を通じて再生アプリケーションに通知する。これにより、再生アプリケーションはディスクリプションを再生し、図 6 のアプリケーション画面イメージの表示領域 701 が更新される。サーバ通信部 557 は、サーバとの通信の制御を行う。サーバとの通信には、カスタマイズ部品情報編集部 554 で保持している部品設定情報をサーバに保存するための通信と、フレームワークエミュレータ制御部 556 で実行するための、再生アプリケーションのダウンロードが含まれる。これにより、再生アプリケーションのカスタマイズ可能な情報に増減や、動作に変更があっても、ディスクリプションエディタ 604 の変更なしに、再生アプリケーションに合わせた編集機能を提供できる。

10

**【0084】**

<ディスクリプションの管理>

図 9 の S904 ~ S907 において触れたディスクリプションの生成や登録について詳細を説明する。図 14 は、ディスクリプション生成 (S904 ~ S907) の詳細フローを示す図である。

**【0085】**

図 14 のフローにおいて、クライアント端末 0240 のウェブブラウザからのアクセスを受け付けると、オーサリングシステム 0101 は、フロントエンドリソース 0204 であるディスクリプションエディタ 0207 とフレームワークエミュレータ 0208 をクライアント端末 0240 に提供する (S903)。ディスクリプションエディタ 0207 は、ディスクリプションを新規に作成する場合、オーサリングシステム 0101 のバックエンドプログラム 0205 であるディスクリプションナンバー管理 0210 にディスクリプション新規作成通知が送信する (S1401)。通知を受けたディスクリプションナンバー管理 0210 は、既存アプリケーションの識別子情報、既存ディスクリプションの識別子情報と重複しない識別子情報を生成する (S1402)。次に、バックエンドプログラム 0205 は、アクセス情報からシステム利用者 0202 のテナントを特定し、ディスクリプション保存先を決定する (S1403)。次に、バックエンドプログラム 0205 は、作成するディスクリプションに対応する再生アプリケーションのバージョン情報を取得する (S1404)。次に、バックエンドプログラム 0205 は、これまでの識別子情報、再生アプリケーションのバージョン情報などから meta データを生成し (S1405)、クライアント端末 0240 のディスクリプションエディタへ送信する (S1406)。既存ディスクリプションを編集する場合、ディスクリプションエディタは、ディスクリプションプール A1 (0113) へディスクリプション読み出し要求を送信し (S1410)、オーサリングシステム 0101 は指定されたディスクリプションと meta データをディスクリプションエディタへ送信する (S1411)。既存ディスクリプションを元に新たなディスクリプションを作成する場合は、ディスクリプション新規作成時のフロー (S1401 ~ S1406) と同様に meta データの作成が行われる。

20

30

**【0086】**

ディスクリプションの新規作成または読み込みが行われた後、ディスクリプションエディタおよびフレームワークエミュレータでは、ディスクリプションのカスタマイズおよび動作確認が行われる (S1421 ~ S1424)。ディスクリプションのカスタマイズについては詳細を後述する。

40

**【0087】**

ディスクリプションのカスタマイズ後、システム利用者 0202 はカスタマイズを行ったディスクリプションの保存操作をおこなう (S1407)。保存操作を受けたディスクリプションエディタは、オーサリングシステム 0101 にディスクリプションを送信する (S1408)。オーサリングシステム 0101 は対応する保存先にディスクリプションを保存する (S1409)。

**【0088】**

上述したように、オーサリングシステム 0101 ではディスクリプションの管理が行わ

50

れる。図 16 は、ディスクリプションナンバー管理 0210 の管理テーブルの例を示す図である。

【0089】

図 16 の管理テーブルのように、ディスクリプションの関連情報を格納することと、オーサリングシステム 0101 はディスクリプションを適切に管理する。

【0090】

管理テーブルは、ディスクリプションナンバー欄 1606、ディスクリプション名欄 1607、バージョン欄 1608、更新日時欄 1609、再生アプリケーションバージョン欄 1610、作成者欄 1611、公開範囲欄 1612、インストール実績欄 1613 などの情報を管理する。

10

【0091】

ディスクリプションナンバー欄 1606、作成者欄 1611、再生アプリケーションバージョン欄 1610 は、新規ディスクリプションナンバーが発行された際に併せて登録される情報である。ディスクリプションナンバー欄 1606 には、ディスクリプションナンバー管理 0210 によって発行された情報が格納される。作成者欄 1611 には、ディスクリプションナンバーの発行を要求したディスクリプションエディタの操作ユーザの情報が登録される。再生アプリケーションバージョン欄 1610 には、ディスクリプションに対応する現在の再生アプリケーションバージョンの情報が登録される。

【0092】

ディスクリプション名欄 1607、更新日時欄 1609、再生アプリケーションバージョン欄 1610 は、ディスクリプションが編集されて保存された際に更新される情報である。ディスクリプション名欄 1607 には、ユーザによって指定された名称が登録される。更新日時欄 1609 は、ディスクリプションの更新段階を示す数値が登録される。

20

【0093】

公開範囲欄 1612 には、保存されたディスクリプションにアクセスし、編集や複製をおこなうことができるユーザの範囲情報が登録される。公開範囲としては、例えば、全てのユーザ、同テナントに所属するユーザ、個人のみ、といった範囲を指定することができる。公開範囲欄 1612、ディスクリプションの保存時等、任意のタイミングで変更することができる。

【0094】

インストール実績欄 1613、登録されたディスクリプションを使用してアプリケーションが発行され・顧客先装置でインストールされた数値が登録される。インストール実績欄 1613 は、市場のニーズ把握をおこなう材料として利用可能な他、ディスクリプションナンバーの棚卸処理に利用することができる。たとえば、オーサリングシステム 0101 は、ディスクリプションプール A1 (0113) に格納されたディスクリプションを削除する際、管理テーブルのインストール実績欄 1613 を確認する。そして、インストール実績欄 1613 が 0 の場合、そのディスクリプションナンバーは解放し、インストール実績欄 1613 が 0 でない場合、解放せずに維持する。解放されたディスクリプションナンバーは、新規ディスクリプションが生成された際に、割り当てることが可能となる。すなわち、ディスクリプションナンバーの再使用が可能となる。

30

40

【0095】

例えばディスクリプション情報 1603 はインストール実績がないため、ディスクリプションが削除されたときディスクリプションナンバー“F00C”は解放される。そして、別のディスクリプション情報 1605 が作成されたときディスクリプションナンバー“F00C”が再使用される。一方でインストール実績があるディスクリプション情報 (1601, 1602, 1604) が削除されていた場合でも、そのディスクリプションナンバーは再使用されることはない。

【0096】

<ディスクリプションの編集>

まず、カスタムアプリケーションを作成するためのディスクリプションエディタの実行

50

画面について説明する。図7(A)は、ディスクリプションエディタの画面を示す図である。ユーザがWebブラウザでオーサリングシステム0101にアクセスし、アプリケーションのカスタマイズを開始すると、Webブラウザ内に画面700が表示される。画面700はサーバからダウンロードしたHTML、JavaScript、cssによって構成されていて、Webブラウザがそれらを解釈し、画面をレンダリングした結果が表示される。本実施例では、顧客の要望を受け、Copyの実行ボタンを選択しやすくしたカスタムアプリケーションを作成する様子を例に説明する。画面700に複数のアイテム(オブジェクト)が配置されており、アプリケーションの編集環境をユーザに提供する。画面700の主要な構成を次に説明する。

#### 【0097】

表示領域701は、ユーザによってカスタマイズされた内容が反映されたアプリケーション(変種中のアプリケーション)の実行イメージを表示するためのイメージ表示領域である。本実施例では、この領域をHTMLのcanvas elementによって構成した例を用いて説明するが、この領域を他のレンダリング可能なHTML elementを使って構成しても良い。このような構成により、ユーザは、表示領域701を見て所望のカスタマイズが行えているか確認しながらカスタマイズを行うことができる。

#### 【0098】

部品リスト702は、カスタマイズ可能(追加可能)な表示部品のリストを表示するプルダウン形式の表示オブジェクトである。ユーザは、この部品リスト702から所望の部品を選択・追加することで、アプリケーションに対して部品を追加するカスタマイズをおこなうことができる。編集可能な部品としては、例えば、機能を紐づけ可能なボタン、タイトル情報、メッセージ情報等があげられる。

#### 【0099】

部品の追加した結果は表示領域701に即座に反映されるため、ユーザはカスタマイズ状況を容易に確認することができる。

#### 【0100】

表示領域706は、現在設定されている部品と、その部品に設定されている情報を表示するカスタマイズ設定情報表示領域である。図7(A)では、タイトル部品703、メッセージ部品704、ボタン部品705が設定されている様子が表示されている。ユーザは、表示領域706に表示されている部品の情報を編集することで、部品のカスタマイズをおこなうことができる。例えば、タイトル部品703に表示されている「カスタムアプリケーション」文字列を「テストアプリケーション」と編集したとすると、表示領域701においてその編集が反映され、「カスタムアプリケーション」と記載された箇所が「テストアプリケーション」に更新される。また、部品のパラメータを編集することで、部品のサイズやレイアウトを変更することも可能である。

#### 【0101】

次に、ディスクリプションエディタの実行フローについて説明する。図8は、ディスクリプションエディタのソフトウェアモジュール間のシーケンスを示す図である。各処理は、コントローラユニット500によって実現される。詳細には、WEBブラウザ601のプログラムをROM506から取得してRAM503に展開したCPU502が、JavaScript VM603を介してディスクリプションエディタ604を実行することで、各処理が実現される。

#### 【0102】

ディスクリプションエディタ604の実行フローでは、はじめに全体制御部551が、画面構成部552に画面構成要求する(S800)。画面構成部552は、図7(A)で示すような編集画面の構成をおこなう。この時、部品リスト702、カスタマイズ設定情報の表示領域706はカスタマイズ可能な部品情報が不定のため、空の状態で作成される。次に全体制御部551は、サーバ通信部557に再生アプリケーションのダウンロードを要求する(S801)。サーバ通信部557は、この要求を受けてサーバから再生アプ

10

20

30

40

50

リケーションをダウンロードする。サーバ通信部 557 は、ダウンロードした再生アプリケーションを全体制御部 551 に通知する (S802)。全体制御部 551 はフレームワークエミュレータ制御部 556 に、再生アプリケーションの実行を要求する (S803)。フレームワークエミュレータ制御部 556 は、要求された再生アプリケーションを実行する。続いて、全体制御部 551 はフレームワークエミュレータ制御部 556 に、カスタマイズ可能な部品情報を要求する (S804)。フレームワークエミュレータ制御部 556 は、この要求を受け再生アプリケーションにカスタマイズ可能な部品情報を要求し、取得できたカスタマイズ可能な部品情報を全体制御部 551 に返す (S805)。全体制御部 551 は、取得したカスタマイズ可能な部品情報をカスタマイズ部品情報管理部 553 に通知する (S806)。カスタマイズ部品情報管理部 553 は、通知された情報を元に、ディスクリプションエディタで扱うためのデータフォーマットに変換する。次に全体制御部 551 は、画面構成部 552 に画面更新を要求する (S808)。ここではカスタマイズ可能な部品情報が確定しているため、部品リスト 702、カスタマイズ設定情報の表示領域 706 に情報が反映され、ユーザがカスタマイズ可能な状態になる。次にユーザが部品リスト 702 から部品を追加する (S808)。画面構成部 552 は、この変更通知を受け、全体制御部 551 にユーザ操作情報を通知する (S809)。全体制御部 551 は、設定値が変更された事をカスタマイズ部品情報編集部に通知する (S810)。カスタマイズ部品情報編集部は、設定された値を確認し現在の設定値を全体制御部 551 に通知する (S811)。全体制御部 551 は、現在の設定値からディスクリプションを生成するようにディスクリプション生成部 555 に依頼する (S812)。ディスクリプション生成部 555 は、現在の設定値に対応するディスクリプションを生成して、全体制御部 551 に通知する (S813)。全体制御部 551 は、フレームワークエミュレータ制御部 556 にディスクリプションを通知する (S814)。フレームワークエミュレータ制御部 556 は、ディスクリプションを再生アプリケーションで再生する、これにより、アプリケーション画面イメージの表示領域 701 が更新される。最後に、全体制御部 551 は、サーバ通信部 557 に、編集されたカスタマイズ部品設定情報の保存を要求する (S815)。これを受け、サーバ通信部 557 は、サーバにカスタマイズ部品設定情報の保存を要求する。これによって、ユーザが設定したアプリケーションのカスタマイズ情報がサーバに保存される。次回にユーザがアプリケーションのカスタマイズを行う場合、この情報がロードされ、ユーザが編集した情報が反映された状態からカスタマイズを行うことが出来る。

#### 【0103】

次に、ディスクリプションの構造について説明する。図 11 はディスクリプションを示す図である。図 11 は、図 7 (A) でカスタマイズされたアプリケーション設定情報に対応するディスクリプションの例を示している。すなわち、Copy 実行確認画面を表示し、OK 押下時にコピージョブを実行するアプリケーションに対応するディスクリプションを示している。本実施例では、ディスクリプションの内容について Web ブラウザとの親和性の高い JSON フォーマット (JSON 形式) で表記する。ディスクリプションのフォーマットとして他のフォーマットを利用しても良い。

#### 【0104】

ディスクリプションは、大きく分けてパネル情報 1101、1102 から構成されている。パネル情報 1101 には、部品の情報および部品毎のカスタマイズ情報が格納されている。部品の情報としては、タイトル部品情報 1103、メッセージ部品情報 1109、OK ボタン部品情報 1110、キャンセルボタン部品情報 1111 が含まれる。各部品のカスタマイズ情報としては、部品 ID 情報 1104、部品設定情報 1105、部品パラメータ情報 1106、部品パラメータデータタイプ 1107、部品パラメータデータ 1108 が含まれる。部品 ID 情報 1104 は、そのカスタマイズ情報が何の部品の情報なのかを示す識別情報である。この例の場合、“title” 部品のカスタマイズ情報であることを示している。部品設定情報 1105 は、その部品の設定情報を示す。部品パラメータ情報 1106 は、部品のパラメータのカスタマイズ情報を示す。部品パラメータデータ

イブ1107は、部品パラメータのデータの型を示す。部品パラメータデータ1108は、その部品パラメータの設定値を示す。data\_typeが“string”の場合、ローカライズされた文字列の情報が指定される。

【0105】

したがって、タイトル部品情報1103には、“title”部品に、“title”パラメータがあり、そのパラメータの設定値はローカライズされた文字列“カスタムアプリケーション”が設定されていることがわかる。メッセージ部品情報1109、OKボタン部品情報1110、キャンセルボタン部品情報1111についても、同様に各カスタマイズ情報が設定されている。

【0106】

パネル情報1102はジョブを実行するための情報が格納されている。ジョブ実行部品情報1112には、ジョブ実行を行うための部品として“execute\_job”が指定されている。このジョブ実行部品情報1112についての情報として、何のジョブを実行するのかを示すジョブ識別ID1114と、そのジョブのジョブパラメータ1115と、パラメータ設定値として含まれている。この例はコピーを実行するアプリケーションであるため、ジョブ識別ID1114には“copy”が指定されている。ジョブパラメータ1115には、そのジョブを実行する際のパラメータを登録することができる。ここでは、部数情報1116と、カラーモード1117が登録されている。

【0107】

以上の説明で例に挙げたディスクリプションは、例えば再生アプリケーションのディスクリプション再生制御609によって読み込まれる。そして、再生アプリケーションは、ディスクリプションに応じた画面を構成し、ユーザ操作に応じて画面遷移やジョブの実行制御を行う。

【0108】

<カスタムアプリケーションのインストール>

図9のS912において触れたカスタムアプリケーションの作成処理について詳細を説明する。オーサリングシステム0101に登録されたディスクリプションは、パッケージングされ(S915)、カスタムアプリケーションとしてMFP0203に提供され(S911)、インストールされる(S912)。カスタムアプリケーションがインストールされると、MFP0203において再生アプリケーションが実行され、スクリプトによってディスクリプションが読み込まれ、ディスクリプションに指定されている情報が再生される。図7(B)で示すように実行画面を表示する。図7(B)は、MFPにおけるカスタムアプリケーションの実行画面を示す図である。この実行画面は、ディスクリプションの編集時に表示領域701に表示されたものと同じであり、正常に動作していることがわかる。以下では、カスタムアプリケーションのインストールに関する処理について詳細に説明する。

【0109】

まず、オーサリングシステム0101がMFP0203に提供するインストールパッケージの構造について説明する。図10はインストールパッケージのデータ構造を示す図である。

【0110】

インストールパッケージ1000は、署名付きディスクリプション1052と署名付きアプリケーション1001を含むパッケージデータである。

【0111】

署名付きアプリケーション1001は、スクリプト1004とリソース1005とmeta情報1006からなる再生アプリケーション1002と、アプリケーション署名1003を備えている。本実施例では、ディスクリプションエディタ604が保持している再生アプリケーションと同一のものが利用される。

【0112】

一方の署名付きディスクリプション1052は、ディスクリプション1053と、ディ

10

20

30

40

50

スクリプションmeta情報1054と、ディスクリプション署名1055によって構成される。ディスクリプション1053は、ディスクリプションエディタによって生成されたものであり、ユーザがカスタマイズしたアプリケーションの振る舞いを定義するデータである。ディスクリプション1053の詳細については、図11を用いて説明した通りである。ディスクリプションmeta情報1054は、カスタマイズアプリケーションの識別子(記述識別子)やバージョン情報、アプリケーション名を示す情報である。ディスクリプション署名1055は、ディスクリプション1053とディスクリプションmeta情報1054の完全性を保証するための署名データである。このディスクリプション署名1055は、アプリケーション署名1003とは異なる署名鍵を利用して生成される。この署名鍵はオーサリングシステム0101に保持されており、ディスクリプション署名付与0211によって署名データであるディスクリプション署名1055が生成される。

10

#### 【0113】

上述したインストールパッケージ1000をMFP0203にインストールする際の詳細な処理について説明する。図12は、MFPへカスタムアプリケーションをインストールした後にMFPのストレージで保持されるアプリケーション構造と、対応するインストールパッケージを示す図である。図12の左側がインストールパッケージ1000であり、図の右側がインストール済みアプリケーション1200である。インストールパッケージ1000の構成は上述した通りである。これに対し、インストール済みアプリケーション1200は、スクリプト1201、リソース1202、ディスクリプション1203、meta情報1204、および、第2のmeta情報1205から構成されている。MFP0203によって実行されるアプリケーションインストール処理では、インストールパッケージ1000内の各データが読み出され、MFP0203のストレージ上において再構成される。具体的には、インストールパッケージ内のスクリプト1004がスクリプト1201として配置される。インストールパッケージ内のリソース1005が、インストール後のリソース1202として配置される。また、インストールパッケージ内のディスクリプション1053が、インストール後のディスクリプション1203として配置される。

20

#### 【0114】

加えて、インストールされたアプリケーションを一意に識別するため情報(識別子)としてmeta情報1204が格納される。すなわち、meta情報1204は、MFP0203にインストールされた他のアプリケーションと区別するために用いられる情報である。あるいは、meta情報1204は、アプリケーションのバージョンアップがあった場合にバージョンアップ対象のアプリケーションを特定するのに用いられる情報である。

30

#### 【0115】

図15(A)は、アプリケーションパッケージのmetaデータとMFP0203のアプリケーションデータ管理を説明する図である。図15(B)は、各metaデータとMFP0203内のデータ対応を示す表である。MFP0203は、図15(B)に示すようなデータを保持することで、複数のアプリケーションそれぞれについて一意に識別可能に管理している。インストールするアプリケーションが通常の場合、インストールパッケージ内のアプリケーションに紐づけられたmeta情報をmeta情報1204として格納すればよい。すなわち、通常の場合とは、ディスクリプションと再生アプリケーションを組み合わせるカスタムアプリケーションではないアプリケーションを指す。例えば、図15(B)に示すように、アプリケーション1および2は、アプリケーションmetaデータ欄の情報(アプリケーション識別子、ID)として"AppNo:0001"および"AppNo:0002"を保持しており、両者を区別することができる。そのため、アプリケーションmetaデータをmeta情報1204に格納することで、アプリケーションを一意に識別することができるようになる。

40

#### 【0116】

一方で、インストールするアプリケーションがオーサリングシステム0101で作成されたカスタムアプリケーションである場合、アプリケーションに紐づけられたmeta情

50

報ではアプリケーションを一意に識別することができない。例えば、図15(B)に示すように、カスタムアプリケーションAおよびカスタムアプリケーションBは、アプリケーションmetaデータ欄1504の情報として"AppNo:FFFF"を保持しており、両者を区別することができない。この現象は、カスタムアプリケーションAおよびカスタムアプリケーションBが同一の再生アプリケーションを利用していることに起因している。そこで、本実施例では、カスタムアプリケーションをインストールする場合、meta情報1204に格納するアプリケーションのmeta情報1006をディスクリプションmeta情報1054で書き換える処理をおこなう。図15(B)に示すように、カスタムアプリケーションAおよびカスタムアプリケーションBは、ディスクリプションmetaデータ欄1505の情報として"DesNo:F00A"および"DesNo:F00B"を保持しており、両者を区別することができる。そのため、meta情報1204に書き込むアプリケーションmetaデータをディスクリプションmetaデータで書き換えることにより、カスタムアプリケーションを一意に識別可能にすることができる。また、本実施例では、MFP0203内データ欄1506に示すように、meta情報1204の情報に基づきアプリケーションのディレクトリを生成している。そのため、複数のアプリケーションでmeta情報1204に重複がある場合、ディレクトリの重複が発生してしまう。しかしながら、上記書き換え処理が事前におこなわれるため、すべてのアプリケーションについて異なるディレクトリで適切に管理することができる。

#### 【0117】

また、カスタムアプリケーションをインストールする際、インストールされたアプリケーションの識別子であるmeta情報1204とは別に、アプリケーションのmeta情報1006を第2のmeta情報1205として格納する。この情報は、再生アプリケーションのバージョン情報やアプリケーション名称を保持する目的で格納される。

#### 【0118】

次に、アプリケーションパッケージをインストールする際に実行される制御処理について説明する。図13は、アプリケーションパッケージのインストールフローを示す図である。図13中の処理はアプリケーション管理部352の処理を示す。すなわち、各処理は、コントローラユニット300によって実現される。詳細には、ROM303に格納されたプログラムをRAM302に展開してCPU301が処理することによって実現される。

#### 【0119】

まず、S1302において、アプリケーション管理部352は、任意のインストール手段でインストール要求を送信する。

#### 【0120】

カスタムアプリ配信システム0213からのインストールの場合(S1311)、アプリケーション管理部352は、S1312においてカスタムアプリケーションインストーラーを起動する。アプリケーション管理部352は、S1313においてシステム利用者0202またはMFP0203の利用者によるインストールコードの入力を受け付ける。これはS910に対応する。アプリケーション管理部352は、入力されたインストールコードを用いてカスタムアプリ配信システム0213にインストール要求を送信する。インストールコードの入力方法としては、操作部312のソフトウェアキーボードを用いた入力あげられる。しかしながら、別の方法でインストールコードを入力してもよい。たとえば、MFP0203が提供するWEBページにPC等の情報処理装置でアクセスし、キーボードで入力した内容をMFP0203に送信する構成であってもよい。また、インストールコードまたはこれに対応する情報が記載された用紙をスキャナ370で読み取ることによってインストールコードを入力してもよい。

#### 【0121】

通常アプリケーションのインストールの場合(S1315)、アプリケーション管理部352は、S1316においてアプリケーションインストーラーを起動する。S1317において、アプリケーション管理部352は、MFP0203の利用者から、インストー

ルするアプリケーションの選択を受け付ける。アプリケーション管理部 3 5 2 は、選択されたアプリケーションのインストール要求をアプリケーションサーバに送信する。

【 0 1 2 2 】

プレインストールアプリケーションを用いたインストールの場合 ( S 1 3 1 9 )、アプリケーション管理部 3 5 2 は、プレインストーラを実行する。S 1 3 2 0 において、起動したプレインストーラは内部ストレージへインストール要求を送信する。アプリケーション管理部 3 5 2 は、S 1 3 0 3 でのインストール要求への応答としてアプリケーションのインストールパッケージを受信する。S 1 3 0 4 において、アプリケーション管理部 3 5 2 は、受信したインストールパッケージに含まれるアプリケーションのアプリケーション形態を判別する。

10

【 0 1 2 3 】

アプリケーション形態が判別できない場合 ( 判別失敗 )、アプリケーション管理部 3 5 2 は、インストール失敗と判定してエラー終了する。

【 0 1 2 4 】

アプリケーション形態が通常のアプリケーションの場合 ( 第 1 の形態 )、アプリケーション管理部 3 5 2 は、S 1 3 0 5 において、アプリケーションの検証をおこなう。検証では、アプリケーション署名 1 0 0 3 を用いて、アプリケーションのスクリプト 1 0 0 4、リソース 1 0 0 5、meta 情報 1 0 0 6 に不正な情報が含まれないかの確認がおこなわれる。

【 0 1 2 5 】

検証に失敗した場合 ( 検証失敗 )、アプリケーション管理部 3 5 2 は、インストール失敗として判定してエラー終了する。検証に成功した場合 ( 検証成功 )、アプリケーション管理部 3 5 2 は、S 1 3 0 6 においてアプリケーションのインストールを行う。

20

【 0 1 2 6 】

アプリケーション形態がオーサリングシステム 0 1 0 1 で作成したアプリケーションの場合 ( 第 2 の形態 )、アプリケーション管理部 3 5 2 は、S 1 3 0 5 において、アプリケーションの検証をおこなう。検証では、再生アプリケーションのアプリケーション署名 1 0 0 3 を用いて、再生アプリケーションのディスクリプション 1 0 5 3、ディスクリプション meta 情報 1 0 5 4 が不正なものでないことの確認が行われる。

【 0 1 2 7 】

検証に失敗した場合 ( 検証失敗 )、インストール失敗としてエラー終了する。検証に成功した場合 ( 検証成功 )、アプリケーション管理部 3 5 2 は、S 1 3 0 8 においてディスクリプション署名 1 0 5 5 を用いて、ディスクリプション 1 3 0 3、ディスクリプション meta 情報 1 3 0 4 に不正な情報が含まれていないことを検証する。検証に失敗した場合 ( 検証失敗 )、アプリケーション管理部 3 5 2 は、インストール失敗と判定してエラー終了する。検証に成功した場合 ( 検証成功 )、アプリケーション管理部 3 5 2 は、S 1 3 0 9 において再生アプリケーションの meta データから取得された App No を、ディスクリプションの meta データ内の Des No で置き換える処理を行う。アプリケーション管理部 3 5 2 は、S 1 3 1 0 において、再生アプリケーションとディスクリプションのインストールを行う。

30

【 0 1 2 8 】

< 備考 >

以上で説明したように、オーサリングシステム 0 1 0 1 は、アプリケーションを容易に編集・配信できる環境を提供する。これにより、サービスマンや技術営業等の人員がカスタムアプリケーションを作成することができるようになるため、顧客が所望するアプリケーションを迅速に提供することが可能となる。

【 0 1 2 9 】

また、オーサリングシステム 0 1 0 1 は再生アプリケーションの情報 ( 特に識別子 ) とは別に、ディスクリプションの情報 ( 特に識別子 ) を管理する。これにより、顧客の要望の数に比例して種類が増大カスタムアプリケーションを適切に管理することができる。

40

50

## 【0130】

(他の実施例)

本発明は、上述の実施例の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、その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実行する処理でも実現可能である。また、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。また、実施例中においてHDDとして説明したストレージをSSD等の半導体ストレージに置き換えてもよい。

## 【0131】

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、1つの機器からなる装置に適用してもよい。 10

## 【0132】

本発明は上記実施例に限定されるものではなく、本発明の趣旨に基づき種々の変形が可能であり、それらを本発明の範囲から除外するものではない。

## 【0133】

なお、実施例中に登場する略称は次の意味である。

## 【0134】

ASICとは、Application Specific Integrated Circuitのことである。

## 【0135】

CPUとは、Central Processing Unitである。 20

## 【0136】

HDDとは、Hard Disk Driveのことである。

## 【0137】

IEEEとは、Institute of Electrical and Electronics Engineersのことである。

## 【0138】

MFPとは、Multi Function Peripheralのことである。

## 【0139】

OSとは、Operating Systemのことである。 30

## 【0140】

RAMとは、Random Access Memoryのことである。

## 【0141】

ROMとは、Read Only Memoryのことである。

## 【0142】

RTCとは、Real-Time Clockのことである。

## 【0143】

SSDとは、Solid State Driveのことである。

## 【0144】

UIとは、User Interfaceのことである。 40

## 【0145】

USBとは、Universal Serial Busのことである。

## 【0146】

VMとは、Virtual Machineのことである。

## 【符号の説明】

## 【0147】

0101 オーサリングシステム(情報処理システム)

0220 サーバ装置(情報処理装置)

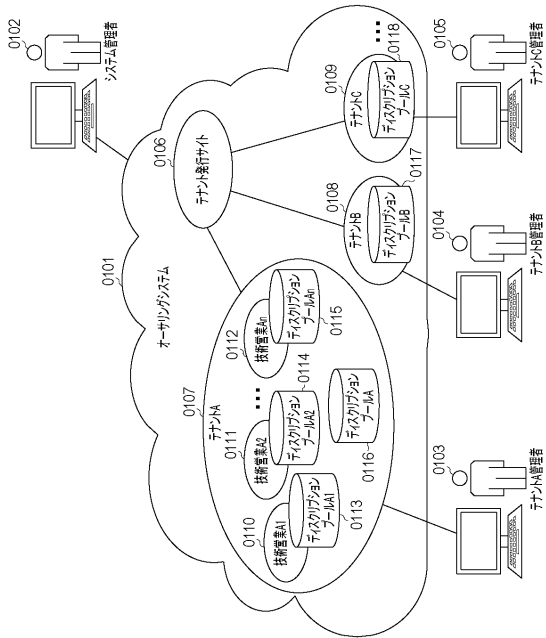
0203 MFP(画像処理装置)

0240 クライアント端末(外部端末) 50

- 0 2 0 7 ディスクリプションエディタ (編集プログラム)
- 6 0 8 再生アプリケーション
- 6 1 2 ディスクリプション (記述ファイル)
- 1 5 0 4 アプリケーションmetaデータ (アプリケーション識別子)
- 1 0 5 5 ディスクリプションmetaデータ (記述識別子)

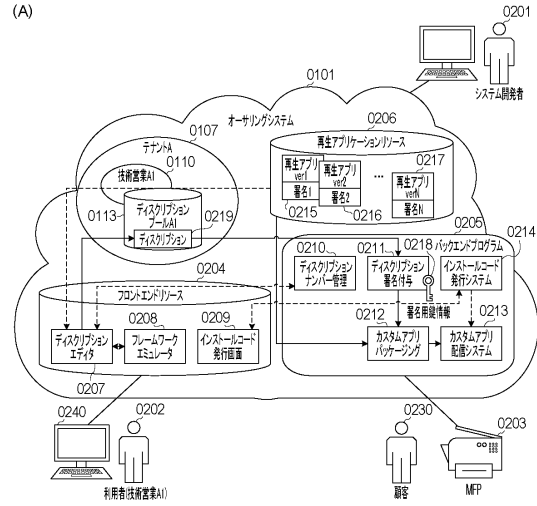
【図面】

【図 1】



【図 2】

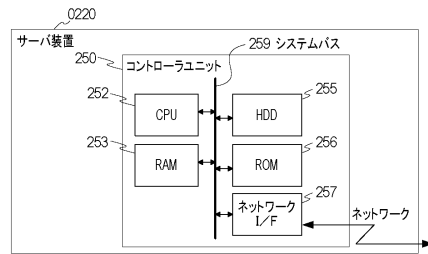
(A)



10

20

(B)

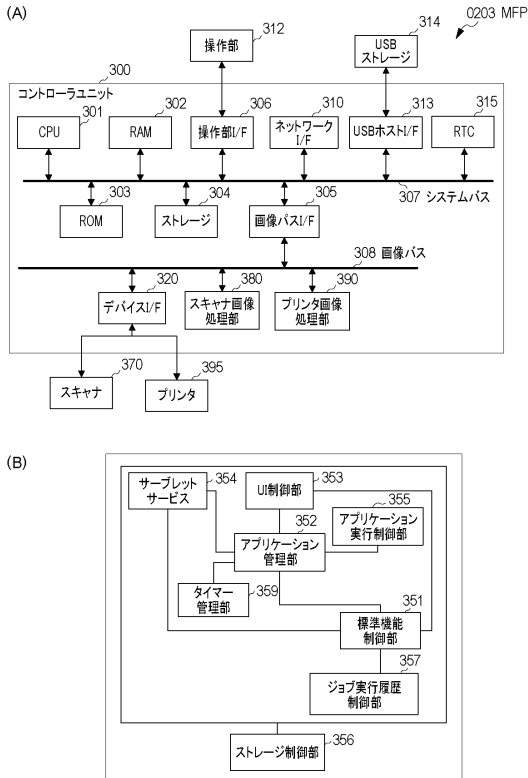


30

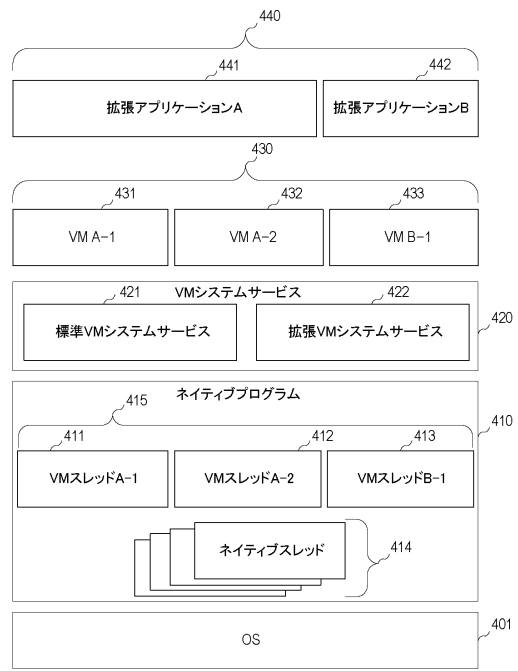
40

50

【 図 3 】



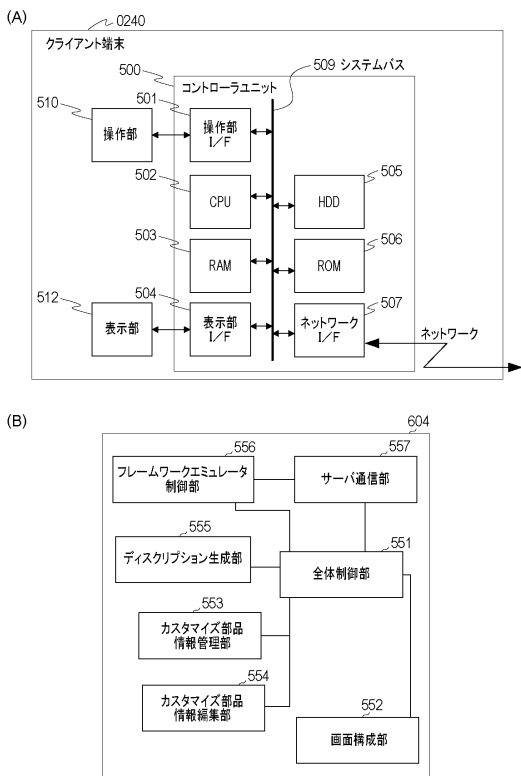
【 図 4 】



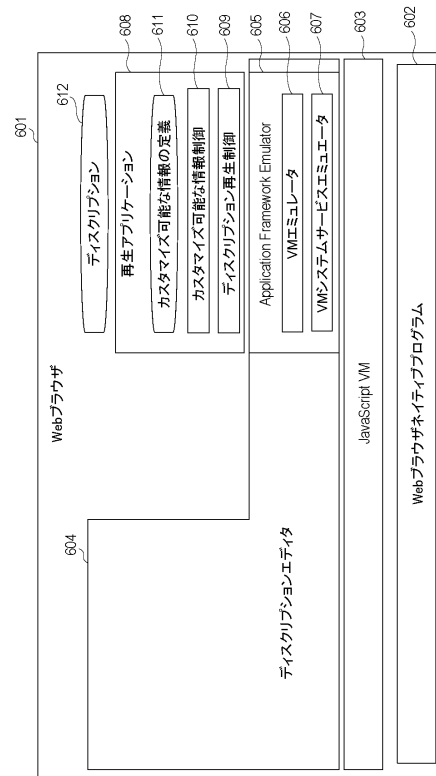
10

20

【 図 5 】



【 図 6 】

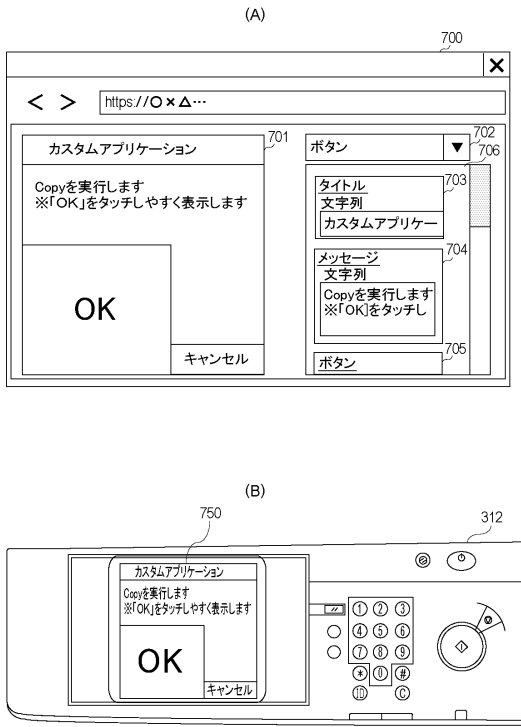


30

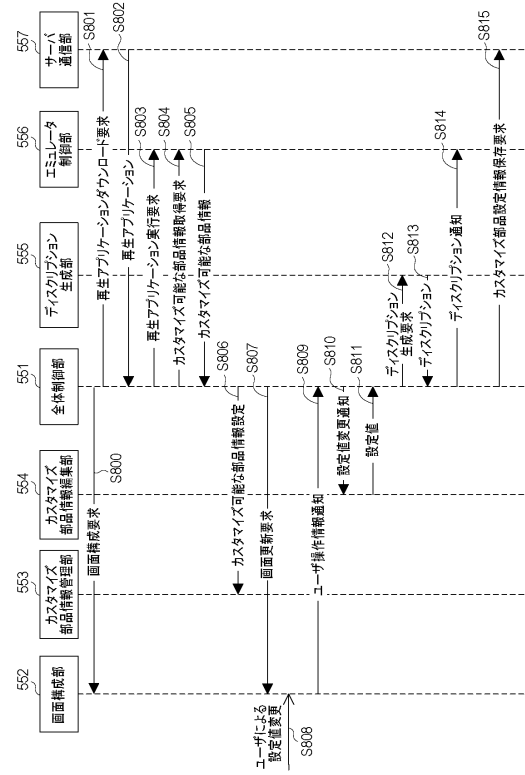
40

50

【 図 7 】



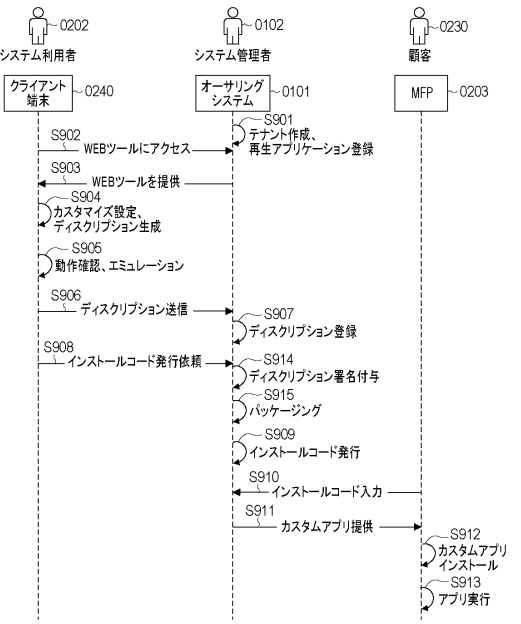
【 図 8 】



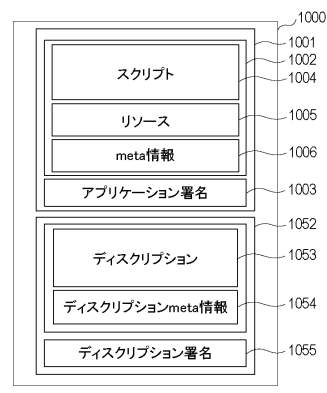
10

20

【 図 9 】



【 図 10 】

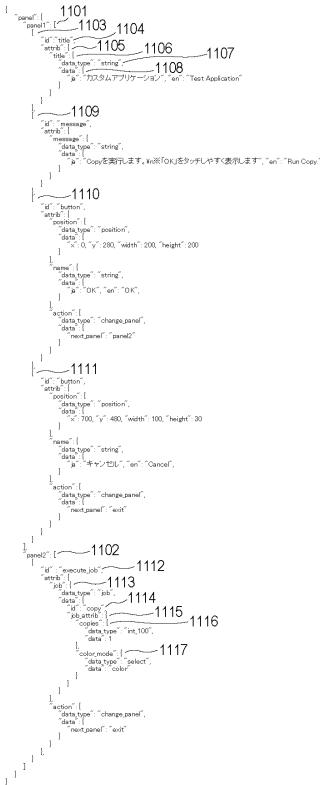


30

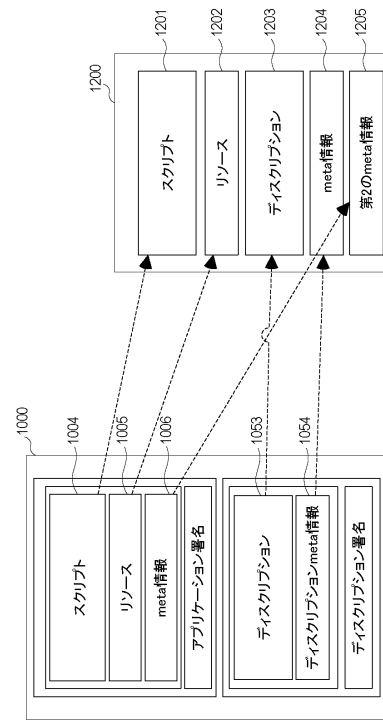
40

50

【図 1 1】



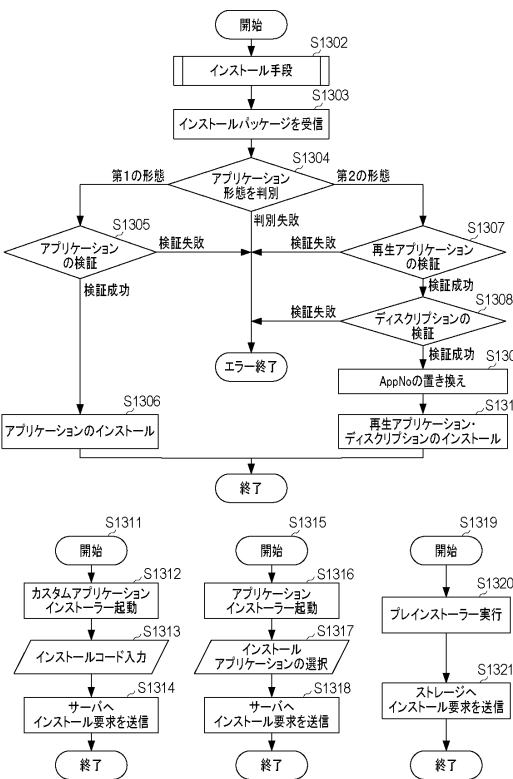
【図 1 2】



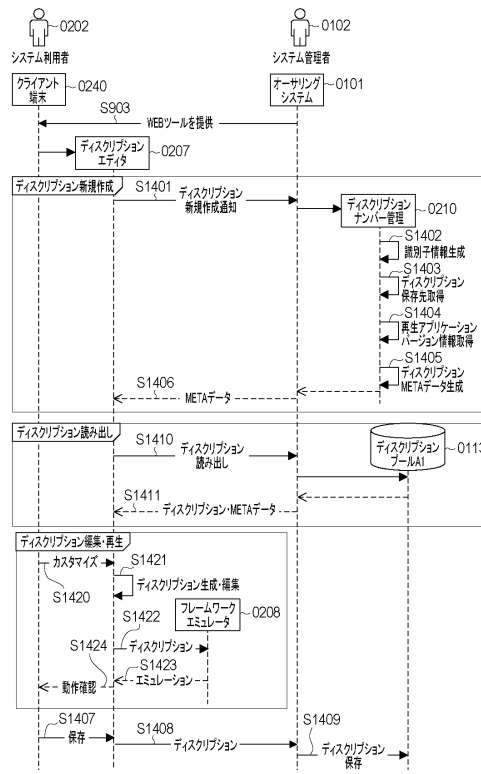
10

20

【図 1 3】



【図 1 4】



30

40

50



フロントページの続き

ヤノン株式会社内

(72)発明者 小池 宏明

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) 5B376 AC13