

(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2019-125308  
(P2019-125308A)

(43) 公開日 令和1年7月25日(2019.7.25)

(51) Int. Cl.	F I	テーマコード (参考)
<b>G06F 8/00 (2018.01)</b>	G06F 9/06 620A	5B376
<b>G06F 3/0481 (2013.01)</b>	G06F 3/0481	5E555

審査請求 未請求 請求項の数 9 O L (全 18 頁)

(21) 出願番号	特願2018-7228 (P2018-7228)	(71) 出願人	390002761 キヤノンマーケティングジャパン株式会社 東京都港区港南2丁目16番6号
(22) 出願日	平成30年1月19日 (2018.1.19)	(71) 出願人	592135203 キヤノンITソリューションズ株式会社 東京都品川区東品川2丁目4番11号
		(74) 代理人	100189751 弁理士 木村 友輔
		(72) 発明者	石田 知子 東京都品川区東品川2丁目4番11号 キヤノンITソリューションズ株式会社内
		(72) 発明者	高塚 剛 東京都品川区東品川2丁目4番11号 キヤノンITソリューションズ株式会社内
		Fターム(参考)	5B376 BC50 BC51 FA15
			最終頁に続く

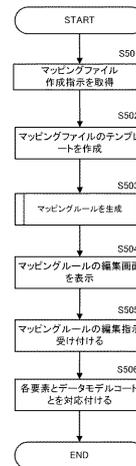
(54) 【発明の名称】 情報処理装置、情報処理装置の制御方法、およびコンピュータプログラム

(57) 【要約】

【課題】 Webサービスを利用するアプリケーションを容易に構築する仕組みを提供すること

【解決手段】 アプリケーションを構築する情報処理装置であって、Webサービスから取得した戻り値のデータ構造を要素単位で識別可能に表示部に表示する表示制御手段と、表示部に表示されたデータ構造を要素毎にデータモデル定義と対応づけるための入力を受け付ける入力受付手段と、を有することを特徴とする。

【選択図】 図5



**【特許請求の範囲】****【請求項 1】**

アプリケーションを構築する情報処理装置であって、  
Webサービスから取得した戻り値のデータ構造を要素単位で識別可能に表示部に表示する表示制御手段と、  
前記表示部に表示されたデータ構造を要素毎にデータモデル定義と対応づけるための入力を受け付ける入力受付手段と、  
を有することを特徴とする情報処理装置。

**【請求項 2】**

前記表示制御手段は、前記データ構造における要素同士の階層構造を識別可能に表示部に表示することを特徴とする請求項 1 に記載の情報処理装置。

10

**【請求項 3】**

前記表示制御手段は、前記データ構造に同一の要素が存在するか否かを識別可能に表示部に表示することを特徴とする請求項 1 または 2 に記載の情報処理装置。

**【請求項 4】**

前記表示制御手段は、前記データ構造の要素のうち単一で存在する要素と複数存在する要素とを識別可能に表示部に表示することを特徴とする請求項 3 に記載の情報処理装置。

**【請求項 5】**

前記入力受付手段は、前記表示部に表示されたデータ構造を要素単位で削除するための入力を受け付け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装置

20

**【請求項 6】**

前記Webサービスから取得した戻り値は、XML形式またはJSON形式の少なくともいずれか一方を含むことを特徴とする請求項 1 乃至 5 のいずれか 1 項に記載の情報処理装置。

**【請求項 7】**

前記Webサービスから取得した戻り値は、RESTによる通信を用いて取得されたデータであることを特徴とする請求項 1 乃至 6 のいずれか 1 項に記載の情報処理装置。

**【請求項 8】**

アプリケーションを構築する情報処理装置の制御方法であって、  
Webサービスから取得した戻り値のデータ構造を要素単位で識別可能に表示部に表示する表示制御工程と、  
前記表示部に表示されたデータ構造を要素毎にデータモデル定義と対応づけるための入力を受け付ける入力受付工程と、  
を有することを特徴とする情報処理装置の制御方法。

30

**【請求項 9】**

コンピュータを請求項 1 乃至 7 のいずれか 1 項に記載の各手段として機能させるためのコンピュータプログラム。

**【発明の詳細な説明】****【技術分野】**

40

**【0001】**

本発明は、情報処理装置、情報処理装置の制御方法、およびコンピュータプログラムに関する。

**【背景技術】****【0002】**

従来から、入出力定義やデータモデル定義などを設定し、Webアプリケーションを構築するWebアプリケーション開発ツールが存在する。

**【0003】**

また、一般に公開されているサーバにて提供される機能やリソースを活用するWebサービスの利用も高まっており、複数のWebサービスを組み合わせることによって、統合

50

的なWebアプリケーションを作成することができる。Webサービスの一例として、HTTP(HyperText Transport Protocol)などのインターネット関連技術を応用して、SOAP(Simple Object Access Protocol)と呼ばれるXML(Extensible Markup Language)形式のProtocolを用いてメッセージの送受信を行うサービスがある。このWebサービスを呼び出すWebサービスクライアント(クライアントアプリケーション)には、公開されているWSDL(Web Services Description Language)が利用される。WSDLは、Webサービスを記述している言語である。

【0004】

特許文献1には、WSDL等で記載されたWebサービスの定義情報を用いて、マッピングファイルと呼ばれる(データモデルとWebサービスの入出力データを対応付ける)定義ファイルを生成するWebアプリケーション開発ツールの仕組みが示されている。

【先行技術文献】

【特許文献】

【0005】

【特許文献1】特開2011-59751号公報

【発明の概要】

【発明が解決しようとする課題】

【0006】

ところで、SOAPにおけるWSDLのようにサービスの構造を定義しているものが存在しないRESTのようなProtocolも存在する。したがって、このようなProtocolを用いてWebサービスのリソースをユーザが取得しアプリケーションを開発する場合には、WSDLのような既存の定義を利用してマッピングファイルを作成することは困難であった。

【0007】

そこで本発明は、上記課題に鑑みてなされたものであり、Webサービスを利用するアプリケーションを容易に構築する仕組みを提供することである。

【課題を解決するための手段】

【0008】

そこで、本発明は、アプリケーションを構築する情報処理装置であって、Webサービスから取得した戻り値のデータ構造を要素単位で識別可能に表示部に表示する表示制御手段と、前記表示部に表示されたデータ構造を要素毎にデータモデル定義と対応づけるための入力を受け付ける入力受付手段と、を有することを特徴とする。

【発明の効果】

【0009】

本発明によれば、Webサービスを利用するアプリケーションを容易に構築する仕組みを提供することが可能となる。

【図面の簡単な説明】

【0010】

【図1】本実施形態における情報処理システムの一例を示す図である。

【図2】本実施形態における情報処理システムの各構成に適用可能なハードウェア構成の一例を示す図である。

【図3】本実施形態におけるプログラム開発装置の機能構成を示す図である。

【図4】本実施形態におけるプログラム開発装置の機能構成を示す図である。

【図5】マッピングファイルを生成するためのフローチャートの一例である。

【図6】マッピングルールを生成するためのフローチャートの一例である。

【図7】マッピングファイルの表示画面の一例を示す図である。

【図8】マッピングファイルの編集画面の一例を示す図である。

【図9】データモデル定義の表示画面の一例を示す図である。

10

20

30

40

50

【図10】マッピングファイルの編集画面の一例を示す図である。

【図11】REST通信メッセージの一例を示す図である。

【図12】マッピングファイルの一例を示す図である。

【図13】XML形式のメッセージから生成されたマッピングファイルとデータモデル定義との関係を示す図である。

【図14】JSON形式のメッセージから生成されたマッピングファイルとデータモデル定義との関係を示す図である。

【図15】Webアプリケーション/Webサービス生成のフローチャートの一例である。

【発明を実施するための形態】

【0011】

以下、本発明の実施形態を、図面を参照して詳細に説明する。

【0012】

図1は、本実施形態に係わるプログラム開発装置（開発者がアプリケーションの構築のために使用する情報処理装置）、プログラム開発サーバ、データベースサーバ、アプリケーションクライアント（クライアント装置）、アプリケーションサーバの構成の一例を示すシステム（情報処理システム）構成図である。

【0013】

プログラム開発装置101は、開発者の操作に従って画面レイアウト及びデータベース検索指示などを定義する。プログラム開発装置101単体では、開発者の入力受付を行い、後述するプログラム開発サーバ102に実際のプログラム生成処理、アプリケーション生成処理をさせてもよいし、プログラム開発装置101単体でプログラム生成、アプリケーション生成まで処理してもよい。

【0014】

なお、本実施形態においては、プログラム開発装置101で生成するアプリケーションはWebアプリケーションを一例として説明するが、これに限定されるものではない。アプリケーションには、携帯電話・スマートフォン・タブレットなどの情報処理装置で動作するアプリケーションや組み込みソフトウェアなどのWeb技術による通信を利用しないアプリケーションが含まれる。また、アプリケーションは、Webサービスも含まれる。

【0015】

更に、本実施形態においては、プログラム開発装置101はアプリケーションのプログラムを生成するものとして説明するが、この方法に限定するものではない。例えば、プログラム開発装置101はプログラムを生成することなく、プログラム開発サーバ102やクラウド環境等でアプリケーションが動作する様にアプリケーション環境を構築する方法であってもよい。

【0016】

また、以下の説明では、SOAPにおけるWSDLのようにWebサービスの構造を定義しているものが存在しないプロトコルの一態様としてRESTを例に説明するが、これに限られるものではない。

【0017】

プログラム開発サーバ102a～102b（情報処理装置）は、プログラム開発装置101により入力された開発者の指示に従って、プログラムを開発する。プログラム開発サーバ102aはLANなどのネットワーク106内に配置されてもよいし、プログラム開発サーバ102bはインターネット上やクラウド上に配置されてもよい。

【0018】

データベースサーバ103a～103b（情報処理装置）は、開発されたアプリケーションが使用するデータベースであり、開発の際に動作確認などのために利用してもよい。例えば、開発者が利用するためにデータベースサーバ103は、プログラム開発装置101と同一の装置で構成されていてもよいし、LANなどのネットワーク106内に配置されてもよい（データベースサーバ103a）。またインターネット上やクラウド上に配置

10

20

30

40

50

されてもよい(データベースサーバ103b)。また、プログラム開発装置101が、プログラム開発サーバ102と協調する場合には、プログラム開発サーバ102とデータベースサーバ103が同一の装置内に構成されていてもよい。

【0019】

アプリケーションサーバ105a~105b(情報処理装置)は、プログラム開発装置101で開発されたアプリケーションを実行する。LANなどのネットワーク106内に配置されてもよい(アプリケーションサーバ105a)し、またインターネット上やクラウド上に配置されてもよい(アプリケーションサーバ105b)。また、ネットワーク106、インターネット、クラウド上のデータベースサーバ103と接続して動作する可能である。

10

【0020】

アプリケーションクライアント104a~104b(情報処理装置)は、アプリケーションサーバ105と協調してプログラム開発装置101で開発したアプリケーションプログラムを動作させる、ユーザの入力端末である。LANなどのネットワーク106内に配置されてもよい(アプリケーションクライアント104a)し、またインターネット上やクラウド上に配置されてもよい(アプリケーションクライアント104b)。また、情報処理装置は、携帯端末やタブレット型の端末であってもよい。

【0021】

図2は、本実施形態に係わるプログラム開発装置101、プログラム開発サーバ102、データベースサーバ103、アプリケーションクライアント104、アプリケーションサーバ105として適用可能な各ハードウェア構成の一例を示すブロック図である。

20

【0022】

図2において、CPU201は、システムバス204に接続される各デバイスを統括的に制御する。

【0023】

また、ROM203あるいは外部メモリ211には、CPU201の制御プログラムであるオペレーティングシステム(OS)や、各サーバ、クライアント、装置など情報処理装置の後述する各種機能を実現するためのプログラムが記憶されている。

【0024】

RAM202は、CPU201の主メモリ、ワークエリア、一時待避領域等として機能する。

30

【0025】

入力コントローラ205は、入力部209からの入力を制御する。この入力部209としては、情報処理装置では、キーボード、マウス等のポインティングデバイスが挙げられる。

【0026】

出力コントローラ206は、出力部210の表示を制御する。この出力部210としては、例えば、CRTや液晶ディスプレイ等が挙げられる。つまり、出力部210は、画面を表示する表示部としての機能を備える。この場合、CPU201は、出力コントローラ206を介して表示部の表示を制御することが可能である。換言すると、CPU201は、表示制御手段の一態様である。

40

【0027】

外部メモリコントローラ207は、ブートプログラム、各種のアプリケーション、フォントデータ、ユーザーファイル、編集ファイル、プリンタドライバ等を記憶する外部メモリ211へのアクセスを制御する。加えて、各サーバ、クライアント、装置等の各種機能を実現するための各種テーブル、パラメータが記憶されている。この外部メモリ211としては、ハードディスク(HD)やフレキシブルディスク(FD)、PCMCIAカードスロットにアダプタを介して接続されるコンパクトフラッシュ(登録商標)、スマートメディア等が挙げられる。

【0028】

50

通信 I / F コントローラ 2 0 8 は、ネットワークを介して外部機器との通信制御処理を実行する。

【 0 0 2 9 】

図 3 および図 4 を用いて、プログラム開発装置 1 0 1 の機能構成を説明する。なお、プログラム開発装置 1 0 1 における各機能は、プログラム開発装置 1 0 1 の CPU 2 0 1 により実行されるものである。また、CPU 2 0 1 によって実行されるプログラム 2 1 2 は、外部メモリ 2 1 1 に記録されており、必要に応じて RAM 2 0 2 にロードされる。つまり、各機能は、CPU 2 0 1 とメモリ ( RAM 2 0 2 ) とが協業することにより実現される。

【 0 0 3 0 】

プログラム開発装置 1 0 1 は、Web アプリケーションを開発する開発者により設定されたリポジトリ定義部 3 0 4 の各定義を用いて、Web アプリケーション生成部 3 0 6 により Web アプリケーション 4 4 0 を生成する。当該処理については、図 1 5 を用いて後述する。

【 0 0 3 1 】

プログラム開発装置 1 0 1 は、入力受付部 3 0 1 と、マッピングファイル編集部 3 0 2 と、マッピングルール生成部 3 0 3 と、リポジトリ定義部 3 0 4 と、表示制御部 3 0 5 と、Web アプリケーション生成部 3 0 6 とを含む。

【 0 0 3 2 】

入力受付部 3 0 1 は、ユーザによる各表示画面への入力操作を受け付ける機能部である。入力受付部 3 0 1 は、本実施形態において、後述する図 7 ~ 図 1 0 に表示される各表示画面への入力操作を受け付ける。

【 0 0 3 3 】

リポジトリ定義部 3 0 4 は、アプリケーション定義 4 0 1、入出力定義 4 0 2、データモデル定義 4 0 3、ビジネスプロセス定義 4 0 4、拡張定義 4 0 5、マッピングファイル 4 0 6、を備える。これらの定義・ファイルは、入力受付部 3 0 1 で受け付けた入力内容を基に生成される。

【 0 0 3 4 】

入出力定義 4 0 2 は、プログラムへ引数として入力される項目を定義する入力定義情報と、プログラムから処理結果として出力する項目を定義する出力定義情報とを入出力定義情報として管理する。具体的に、入力定義情報は、生成された Web アプリケーションの画面を介して当該 Web アプリケーションのユーザが入力する入力項目を定義した情報である。また、出力定義情報は、生成された Web アプリケーションの画面として出力する出力項目を定義した情報である。

【 0 0 3 5 】

データモデル定義 4 0 3 は、データベースのスキーマ情報 ( すなわち、項目の属性や名称、桁数 ) を管理する情報である。

【 0 0 3 6 】

ビジネスプロセス定義 4 0 4 は、業務フローを管理する情報である。

【 0 0 3 7 】

拡張定義 4 0 5 は、マッピングファイル 4 0 6 を呼び出すための定義である。また、拡張定義 4 0 5 は入出力定義 4 0 2 から呼び出されて用いられる。

【 0 0 3 8 】

マッピングファイル 4 0 6 は、Web サービスの入出力データ項目と、Web アプリケーションの取り扱うデータモデルのデータ項目とを対応付けるファイルである。また、マッピングファイル 4 0 6 は、外部の Web サービスを REST により取得したデータ ( リソース ) とデータモデル定義 4 0 3 を対応づけるものである。なお、以下の説明では、マッピングファイルのテンプレートとは、マッピングファイルに対してマッピングルールおよびデータモデル定義とが対応づけられていないファイルを示す。また、マッピングルールは、REST 通信メッセージを解析し、マッピングファイルに対応づけが可能となった

10

20

30

40

50

データを示す。ここで、REST通信メッセージとは、RESTによって取得されたWebサービスの戻り値を示す。また、本実施形態において、REST通信メッセージは、なお、本実施形態では、REST通信メッセージは、アプリケーション作成者であるユーザがブラウザなどを用いて取得し、プログラム開発装置101に入力する。データを記述するためフォーマットの一例として、XML形式、およびJSON形式の場合を例に説明するが、これに限られるものでない。

**【0039】**

マッピングファイル編集部302は、マッピングファイルの生成・編集を行うとともに、編集したマッピングファイルとデータモデル定義403との対応づけを行う機能部である。マッピングファイル編集部302は、マッピングファイルGUI表示部421と、マッピングファイル生成部422と、マッピングファイル更新部423と、を備える。

10

**【0040】**

マッピングファイルGUI表示部421は、データを表示制御部305で表示可能なGUIを生成するための機能部である。マッピングファイルGUI表示部421は、図11から図14で示すように記述されたデータを、図7から図10で示す各表示画面を生成する。マッピングファイルGUI表示部421で生成された画面は、表示制御部305により表示される。

**【0041】**

マッピングファイル生成部422は、ユーザからの指示に応じてマッピングファイルを生成する機能部である。図7の710は、マッピングファイルGUI表示部421で生成されたマッピングファイルの一態様であり、図12の1200は、マッピングファイルの一態様である。

20

**【0042】**

マッピングファイル更新部423は、ユーザからの、図7～図10に表示される各表示画面への入力操作に応じてマッピングファイルを更新する機能部である。ここで、マッピングファイルの更新は、マッピングルールが対応づけられたマッピングファイルの一部の要素を削除する態様を含む。

**【0043】**

マッピングルール生成部303は、REST通信メッセージの取得・解析を行うことによりマッピングルールの生成を行う。マッピングルール生成部303は、REST通信メッセージ取得部431と、REST通信メッセージ解析部432とを備える。

30

**【0044】**

REST通信メッセージ取得部431は、RESTによってWebサービスの戻り値を取得する。Webサービスの戻り値は、例えば、図11の1100、および1110で示すデータである。REST通信メッセージ取得部431は、図7の取得部741に入力されたREST通信メッセージを取得することができる。

**【0045】**

REST通信メッセージ解析部432は、REST通信メッセージ取得部431で取得した戻り値のデータ構造を解析する。なお、当該解析処理については、図6を用いて後述する。また、当該解析処理の結果表示される画面については、図8, 10を用いて後述する。

40

**【0046】**

表示制御部305は、後述する図7～図10に表示される各表示画面の表示を制御する機能部である。具体的には、表示制御部305は、マッピングファイル生成部422で生成されたマッピングファイルの表示画面、およびマッピングファイル編集部302で編集されたマッピングファイルの表示画面を表示する。

**【0047】**

Webアプリケーション生成部306は、リポジトリ定義部304に基づきWebアプリケーションを生成する機能を備える。Webアプリケーション生成部306は、リポジトリ定義解析部461と、Webアプリケーションコード生成部462とを備える。

50

## 【 0 0 4 8 】

リポジトリ定義解析部 4 6 1 は、リポジトリ定義部 3 0 4 の各定義を読み込み解析する。Webアプリケーションコード生成部 4 6 2 は、当該解析した定義を用いてソースコードを生成する。Webアプリケーション生成部 3 0 6 は、ソースコードコンパイル部 4 1 0 を介し、Webアプリケーション 4 4 0 としてプログラムを生成することができる。

## 【 0 0 4 9 】

Webアプリケーション 4 4 0 は、コンパイル J a v a (登録商標)コード 4 4 1 と H T M L / J S P / J a v a S c r i p t (登録商標) 4 4 2 とを含む。

## 【 0 0 5 0 】

次に、図 5 を用いて、マッピングファイルを生成するための処理を説明する。図 5 は、マッピングファイルを生成するためのフローチャートの一例である。プログラム開発装置 1 0 1 の各機能部により実行される。

## 【 0 0 5 1 】

S 5 0 1 において、プログラム開発装置 1 0 1 は、マッピングファイルの作成指示を取得する。当該作成指示は、ユーザによる入力操作により取得される。

## 【 0 0 5 2 】

S 5 0 2 において、プログラム開発装置 1 0 1 は、マッピングファイルのテンプレートを作成する。当該マッピングファイルのテンプレートの一例を図 1 2 に示す。図 1 2 のマッピングファイル 1 2 0 0 は、マッピングファイルのテンプレートを X M L 形式で表示したものである。そして、マッピングファイル 1 2 0 0 を G U I で表示したものを図 7 の 7 1 0 に示す。ここで、図 7 のマッピングファイル表示画面 7 1 0 について説明する。マッピングファイル表示画面 7 1 0 は、操作一覧部 7 1 1 を含み、マッピングファイル 1 2 0 0 の内容をユーザが認識しやすい態様で表示する。操作一覧部 7 1 1 の各項目は、R E S T での呼び出し方法を定義するための項目である。具体的には、操作一覧部 7 1 1 は、コード 7 1 2、操作名 7 1 3、U R L パス 7 1 4、H T T P メソッド 7 1 5、メッセージ形式 7 1 6 の入力を受け付ける。

## 【 0 0 5 3 】

コード 7 1 2 および操作名 7 1 3 は、それぞれ、R E S T 操作を識別するコード、および名称である。U R L パス 7 1 4 は、R E S T による通信で呼び出す W e b サービスの U R L を示す。H T T P メソッド 7 1 5 は、H T T P におけるメソッドを示し、例えば、G E T、P O S T、P U T、D E L E T E のいずれか 1 つから選択することができる。メッセージ形式 7 1 6 は、R E S T による通信で取得するメッセージの形式を示し、X M L 形式、および J S O N 形式のいずれか 1 つから選択することができる。なお、これらの各選択項目は一例であり、R E S T での呼び出し方法を定義するための項目の一態様であり、これに限られるものでない。

## 【 0 0 5 4 】

マッピングファイル表示画面 7 1 0 における 1 つの行は、1 つの W e b サービスの呼び出し方法を定義している。そのため、ユーザは、マッピングファイル表示画面 7 1 0 から複数の W e b サービスの呼び出し方法の定義を行うことが可能である。つまり、1 つの W e b アプリケーションに対して、複数の W e b サービスを使用することも可能である。

## 【 0 0 5 5 】

また、タブ 7 1 7 は、プログラム開発装置 1 0 1 で生成する W e b アプリケーションと、利用する W e b サービスの入出力関係を規定する。タブ 7 1 7 で「入力」が選択されている場合、W e b サービスのリソースを、生成した W e b アプリケーションが使用する。一方、タブ 7 1 7 で「出力」が選択されている場合、生成した W e b アプリケーションを、W e b サービスとして利用させる場合を示す。

## 【 0 0 5 6 】

S 5 0 3 において、プログラム開発装置 1 0 1 は、マッピングルールを生成する。なお、当該マッピングルールの生成処理については、図 6 を用いて詳細に説明する。

## 【 0 0 5 7 】

10

20

30

40

50

S 5 0 4において、プログラム開発装置 1 0 1は、マッピングファイルの編集画面を表示する。ここで、図 8 を用いてマッピングファイルの編集画面について説明する。図 8 の編集画面 8 0 0 および編集画面 8 1 0 は、マッピングファイルを編集を受け付けるための表示画面である。編集画面 8 0 0 は、図 7 のマッピングファイル表示画面 7 2 0 で表示されるマッピングファイルのテンプレートに、S 5 0 3 で生成されたマッピングルールが対応づけて表示された画面である。なお、編集画面 8 0 0 は、R E S T 通信メッセージ 1 1 0 0 から生成されたマッピングルールの編集画面である。また、編集画面 8 1 0 は、R E S T 通信メッセージ 1 1 1 0 から生成されたマッピングルールの編集画面である。以下、編集画面 8 0 0 を用いて各項目について説明するが、編集画面 8 1 0 についても同様である。ここで、編集画面 8 0 0 は、マッピングルールのデータ構造を、要素単位で識別可能に表示している。また、編集画面 8 0 0 は、各要素の要素数も識別可能に表示している。一例として、編集画面 8 0 0 において、要素 8 0 4 の子要素は、要素 8 0 5 である。編集画面 8 0 0 は、各要素単位で編集、削除するための入力操作を受け付ける。

10

#### 【 0 0 5 8 】

S 5 0 5 において、プログラム開発装置 1 0 1 は、マッピングルールの編集指示を受け付ける。ここで、編集画面 8 0 0 から要素 8 0 7 の削除を受け付ける。編集画面 8 0 0 は、行単位あるいはプロパティ単位で削除のための操作入力を受け付ける。図 1 0 における編集画面 1 0 0 0 は、編集画面 8 0 0 から要素 8 0 7 が削除された後の編集画面である。編集画面 8 0 0 から所定の要素（要素 8 0 7）が削除されることにより、当該削除された要素よりも下の階層にある要素が編集画面上で繰り上がって表示される。同様に、図 1 0 における編集画面 1 0 1 0 は、編集画面 8 1 0 から要素 8 1 2 が削除された後の編集画面である。このため、編集画面は、編集の操作に応じてマッピングファイルの構造をより見やすく表示することができる。

20

#### 【 0 0 5 9 】

S 5 0 6 において、プログラム開発装置 1 0 1 は、マッピングルールとデータモデルコードとを対応づける。図 1 0 における編集画面 1 0 0 0 は、編集画面 8 0 0 にデータモデル定義 1 0 0 1 の対応づけがされた状態を示す。また、同様に、図 1 0 における編集画面 1 0 1 0 は、編集画面 8 1 0 にデータモデル定義 1 0 1 1 の対応づけがされた状態を示す。ここで、対応づけられたデータモデル定義は、図 9 に示すデータモデル定義表示画面 9 0 0 で示す定義である。データモデル定義表示画面 9 0 0 において、データモデル 9 0 1 を一覧で表示する表示画面である。

30

#### 【 0 0 6 0 】

次に、図 6 を用いて、マッピングルールを生成するためのフローを説明する。図 6 は、マッピングルールを生成するためのフローチャートの一例である。

#### 【 0 0 6 1 】

S 6 0 1 において、プログラム開発装置 1 0 1 は、図 7 のマッピングファイル表示画面 7 1 0 における操作一覧部 7 1 1 への入力を受け付ける。図 7 のマッピングファイル表示画面 7 2 0 は、操作一覧部 7 1 1 に各項目が入力された後の画面である。ここで、図 1 2 を用いて、マッピングファイル 1 2 1 0 およびマッピングファイル 1 2 2 0 について説明する。マッピングファイル 1 2 1 0 およびマッピングファイル 1 2 2 0 は、マッピングファイル 1 2 0 0 に、操作一覧部 7 1 1 を介して各種情報が入力された後のマッピングファイルである。ここで、マッピングファイル 1 2 1 0 は、メッセージ形式 7 1 6 で X M L 形式が選択された場合のマッピングファイルである。また、マッピングファイル 1 2 2 0 は、メッセージ形式 7 1 6 で J S O N 形式が選択された場合のマッピングファイルである。

40

#### 【 0 0 6 2 】

S 6 0 2 において、プログラム開発装置 1 0 1 は、R E S T 通信メッセージを受け付ける。プログラム開発装置 1 0 1 は、コンテキストメニュー表示 7 3 0 から R E S T 通信メッセージを読み込む旨の選択を受け付ける。プログラム開発装置 1 0 1 は、当該選択を受け付けたことに応じて入力ボックス 7 4 0 を表示する。入力ボックス 7 4 0 は、R E S T 通信メッセージの入力を受け付けるための取得部 7 4 1 を備える。プログラム開発装置 1

50

01は、取得部741にメッセージが入力され、「OK」ボタンが押下されることによりREST通信メッセージを受け付ける。ここで、REST通信メッセージの一例について、図11を用いて説明する。図11における1100は、XML形式で取得したREST通信メッセージの一例である。また、図11における1110は、JSON形式で取得したREST通信メッセージの一例である。なお、以下の説明では、REST通信メッセージ1100に基づいて説明し、REST通信メッセージ1110については説明を省略する。

#### 【0063】

S603において、プログラム開発装置101は、取得部741に入力されたREST通信メッセージの形式が適正か否かを判定する。具体的には、プログラム開発装置101は、取得部741に入力されたREST通信メッセージの形式とメッセージ形式716とが一致するか否かを判定する。プログラム開発装置101は、メッセージ形式同士が一致しないと判定する場合には、S604の処理を実行する。その一方で、プログラム開発装置101は、メッセージ形式同士が一致しないと判定した場合には、マッピングルールの生成を終了する。

10

#### 【0064】

S604において、プログラム開発装置101は、REST通信メッセージにおける最初の要素を親に設定する。ここで、具体的な処理について、図11のREST通信メッセージ1100を用いて説明する。REST通信メッセージ1100において、最初の要素1101は、「ns:ST1C\_\_IO306\_\_getDataResponse」である。

20

#### 【0065】

S605において、プログラム開発装置101は、上述の最初の要素における要素数を1に設定する。

#### 【0066】

S606において、プログラム開発装置101は、親の要素名をプロパティ名として設定する。REST通信メッセージ1100において、プロパティ名として「ns:ST1C\_\_IO306\_\_getDataResponse」が設定される。

#### 【0067】

ここで、S605とS606の処理が行われた後の表示画面について図8を用いて説明する。プログラム開発装置101は、図8で示すようにプロパティ名801に「ns:ST1C\_\_IO306\_\_getDataResponse」が表示される。また、プロパティ名801の「ns:ST1C\_\_IO306\_\_getDataResponse」に対応する要素数803の欄に「1」が表示される。

30

#### 【0068】

S607において、プログラム開発装置101は、S608からS614までの処理を子要素の数だけ繰り返される。REST通信メッセージ1100において、親の要素を「ns:ST1C\_\_IO306\_\_getDataResponse」としている場合には、子要素は「ns:return」である。

#### 【0069】

S608において、プログラム開発装置101は、要素がすでに設定済みの要素であるか否かを判定する。具体的には、プログラム開発装置101は、同じフルパスがすでに出力されている場合はS611に進む。一例として、REST通信メッセージ1100における子要素「ns:return」のフルパスは「ns:ST1C\_\_IO306\_\_getDataResponse/ns:return」となる。プログラム開発装置101は、現在の子要素が1102である場合には、先に同一の要素が設定されていないため、S610に進む。その一方で、プログラム開発装置101は、現在の子要素が1102である場合には、先に同一の要素が設定されているため、S609に進む。

40

#### 【0070】

S609において、プログラム開発装置101は、子の要素数を複数に設定する。図8

50

において、要素数の項目において、複数の要素数が設定された例を 806 に示す。複数となる場合は、例えば、図 11 の 1100 における要素 1105 と要素 1106 のように同一の要素が複数存在する場合である。

【0071】

S610 において、プログラム開発装置 101 は、子の要素数を 1 に設定する。

【0072】

S611 において、プログラム開発装置 101 は、子の要素名をマッピングファイルに出力する。ここで、図 8 で示すようにプロパティ名欄の 805 に「ns: return」が設定される。また、「ns: return」は、親の要素である「ns: ST1C\_\_IO306\_getDataResponse」の下部に表示される。

10

【0073】

S612 において、プログラム開発装置 101 は、現在の子の要素にさらに子の要素があるかを判定する。プログラム開発装置 101 は、現在の子の要素にさらに子の要素がある場合には、S613 に進む。その一方で、プログラム開発装置 101 は、現在の子の要素に子がない場合には S614 に進む。ここで、REST 通信メッセージ 1100 を用いて当該処理について具体的に説明する。プログラム開発装置 101 は、現在の子の要素が 1102 の場合、さらに、子の要素 1103 (「ns: queryResult」) が存在するため、S613 へ進む。一方で、プログラム開発装置 101 は、現在の子要素が 1104 である場合には、更なる子要素が存在しないため S614 へ進む。

【0074】

S613 において、プログラム開発装置 101 は、子の要素を親と設定して S608 から S614 の処理を繰り返す。

20

【0075】

S614 において、プログラム開発装置 101 は、子の要素全ての処理が終了した場合にマッピングルールの生成を終了する。

【0076】

上述の処理により、REST 通信メッセージのデータ構造が要素単位で識別可能に表示される。そのため、ユーザは REST 通信メッセージ階層構造を容易に識別することが可能となる。具体的には、プログラム開発装置 101 は、REST 通信メッセージにおける要素同士の階層構造を識別可能に表示する。また、プログラム開発装置 101 は、REST 通信メッセージにおける同一の要素が存在するか否かを識別可能に表示している。更に、プログラム開発装置 101 は、REST 通信メッセージにおける同一の要素が存在するか否かを識別可能に表示する。更に、プログラム開発装置 101 は、単一で存在する要素と複数存在する要素とを識別可能に表示しているともいえる。

30

【0077】

次に、図 12 から図 14 を用いて、マッピングファイルおよびマッピングルールについて説明する。初めに、図 12 を用いてマッピングファイルについて説明する。

【0078】

マッピングファイル 1200 は、図 7 のマッピングファイル表示画面 710 で表示されるデータである。すなわち、マッピングファイル 1200 は、マッピングファイルのテンプレートである。

40

【0079】

マッピングファイル 1210 は、図 7 のマッピングファイル表示画面 720 で表示されるデータであって、メッセージ形式 716 で XML 形式が選択された場合のマッピングファイルである。マッピングファイル 1210 は、マッピングファイル 1200 から、要素 1201 に要素 1211 が追加されたデータである。

【0080】

マッピングファイル 1220 は、図 7 のマッピングファイル表示画面 720 で表示されるデータであって、メッセージ形式 716 で JSON 形式が選択された場合のマッピングファイルである。マッピングファイル 1220 は、マッピングファイル 1200 から、要

50

素 1 2 0 1 に要素 1 2 2 1 が追加されたデータである。

【 0 0 8 1 】

次に、図 1 3 を用いて X M L 形式で呼び出された R E S T 通信メッセージに基づくマッピングファイルについて説明する。

【 0 0 8 2 】

マッピングファイル 1 3 0 0 は、図 8 の編集画面 8 0 0 で表示されるマッピングファイルである。なお、要素 1 3 0 2 は、編集画面 8 0 0 における要素 8 0 7 に対応する。また要素 1 3 0 1 の前後は、マッピングファイル 1 2 1 0 に対応するため省略する。

【 0 0 8 3 】

マッピングファイル 1 3 1 0 は、編集画面 1 0 0 0 で表示されるマッピングファイルである。ここで、要素 1 3 1 1 は、編集画面 1 0 0 0 で表示されるデータモデル定義 1 0 0 1 に対応する。

10

【 0 0 8 4 】

次に、図 1 4 を用いて J S O N 形式で呼び出された R E S T 通信メッセージに基づくマッピングルールについて説明する。

【 0 0 8 5 】

マッピングファイル 1 4 0 0 は、図 8 の編集画面 8 1 0 で表示されるマッピングファイルである。なお、要素 1 4 0 1 は、編集画面 8 1 0 における要素 8 1 2 に対応する。

【 0 0 8 6 】

マッピングファイル 1 4 1 0 は、編集画面 1 0 1 0 で表示されるマッピングファイルである。ここで、要素 1 4 1 1 は、データモデル定義 1 0 1 1 に対応する。

20

【 0 0 8 7 】

次に、図 1 5 を用いて W e b アプリケーション / W e b サービス生成のフローチャートの一例を示す図である。

【 0 0 8 8 】

ステップ S 1 5 0 1 において、プログラム開発装置 1 0 1 は、ユーザによる W e b アプリケーション / W e b サービス生成指示を受け付けると、外部メモリ 2 1 1 に記憶されているリポジトリ定義部 3 0 4 のアプリケーション定義 4 0 1 を R A M 2 0 2 に読み込む。

【 0 0 8 9 】

ステップ S 1 5 0 2 において、プログラム開発装置 1 0 1 は、外部メモリ 2 1 1 に記憶されているリポジトリ定義部 3 0 4 のデータモデル定義 4 0 3 を R A M 2 0 2 に読み込む。

30

【 0 0 9 0 】

ステップ S 1 5 0 3 において、プログラム開発装置 1 0 1 は、外部メモリ 2 1 1 に記憶されているリポジトリ定義部 3 0 4 の入出力定義 4 0 2 を R A M 2 0 2 に読み込む。

【 0 0 9 1 】

ステップ S 1 5 0 4 において、プログラム開発装置 1 0 1 は、外部メモリ 2 1 1 に記憶されているリポジトリ定義部 3 0 4 のビジネスプロセス定義 4 0 4 を R A M 2 0 2 に読み込む。

【 0 0 9 2 】

ステップ S 1 5 0 5 において、プログラム開発装置 1 0 1 は、外部メモリ 2 1 1 に記憶されているリポジトリ定義部 3 0 4 のデータモデル定義 4 0 3 を R A M 2 0 2 に読み込む。

40

【 0 0 9 3 】

ステップ S 1 5 0 6 において、プログラム開発装置 1 0 1 は、R A M 2 0 2 に読み込んだリポジトリ定義部 3 0 4 の各定義・各ファイルから情報を取得し、ソースコードを生成する。なお、生成されたソースコードにはプログラミング言語が記載されたファイルに加え、HTML、JavaScript (登録商標)、JSP、Java (登録商標) 等の W e b アプリケーションや W e b サービスの提供に利用されるファイルも含まれる。

【 0 0 9 4 】

50

以上、上述の実施形態によれば、情報処理装置は、Webサービスから取得した戻り値のデータ構造を要素単位で識別可能に表示部に表示し、当該表示部に表示されたデータ構造を要素毎にデータモデル定義と対応づけるための入力を受け付けることができる。そのため、Webサービスを利用するアプリケーションを容易に構築する仕組みを提供することが可能となる。したがって、ユーザは、RESTのようなサービスの定義を取得できないプロトコルにおいても、取得したリソースを容易にアプリケーションの構築に利用することができる。特に、Webサービスのリソースの階層構造を容易に識別できることによりユーザが、必要なリソースを選択的に編集することができる。

【0095】

また、情報処理装置は、データ構造の要素のうち単一で存在する要素と複数存在する要素とを識別可能に表示部に表示することができる。そのため、全ての要素を階層構造で提示せず同一の要素についてはまとめてプロパティ単位で表示できる。したがって、Webサービスのリソースの階層構造の全体を把握することに寄与する。

【0096】

以上、本発明について、上述した実施の形態を用いて説明したが、本発明は上述した実施の形態に限定されるものではない。特に、プログラム開発装置101の各機能は、本実施の形態に限定されるものでなく、複数のプログラム開発装置で協業して処理されるものであってもよい。更に、プログラム開発装置は、オンプレミス環境で用いられるサーバであってもよいし、クラウドサービスにおける仮想的なサーバであってもよい。

【0097】

本発明は、例えば、システム、装置、方法、プログラム若しくは記憶媒体等としての実施形態も可能であり、具体的には、複数の機器から構成されるシステムに適用してもよいし、また、1つの機器からなる装置に適用してもよい。

【0098】

なお、本実施形態は、コンピュータや制御コンピュータがプログラム(コンピュータプログラム)を実行することによって実現することもできる。また、プログラムをコンピュータに供給するための手段、例えばかかるプログラムを記録したCD-ROM等のコンピュータが読み取り可能な記録媒体又はかかるプログラムを伝送するインターネット等の伝送媒体も実施例として適用することができる。また、上記のプログラムも実施形態として適用することができる。上記のプログラム、記録媒体、伝送媒体およびプログラムプロダクトは、本発明の範疇に含まれる。

【0099】

以上、実施形態に基づいて詳述してきたが、特定の実施形態に限られるものではなく、発明の要旨を逸脱しない範囲の様々な形態も本発明の範疇に含まれる。さらに、上述した実施形態は一実施の形態を示すものにすぎず、上述した実施形態から容易に想像可能な発明も本発明の範疇に含まれる。

【符号の説明】

【0100】

- 101 プログラム開発装置
- 301 入力受付部
- 302 リポジトリ定義部
- 303 マッピングファイル編集部
- 304 マッピングルール生成部
- 305 表示制御部
- 306 Webアプリケーション生成部

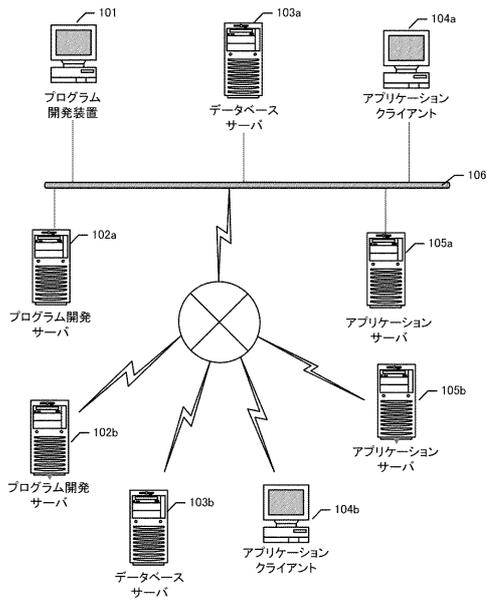
10

20

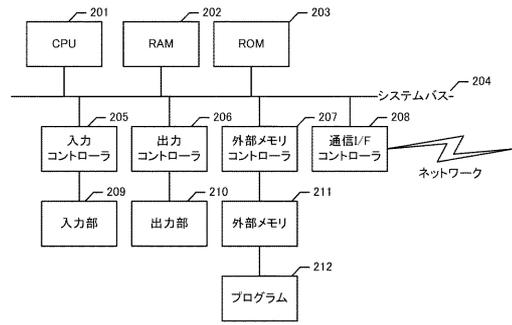
30

40

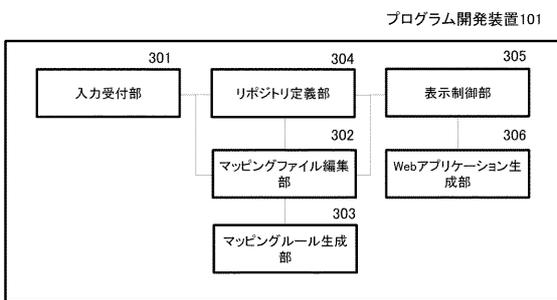
【図1】



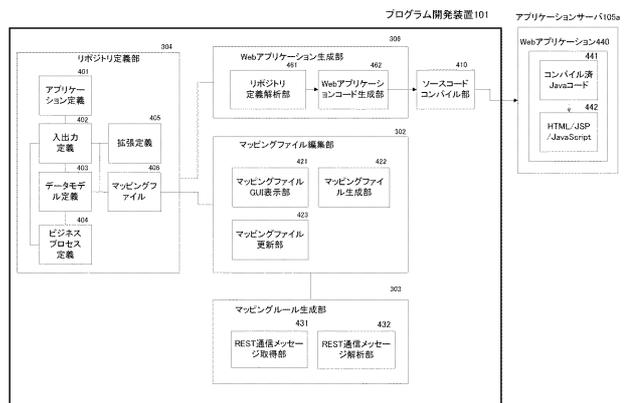
【図2】



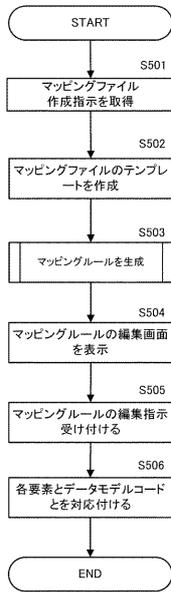
【図3】



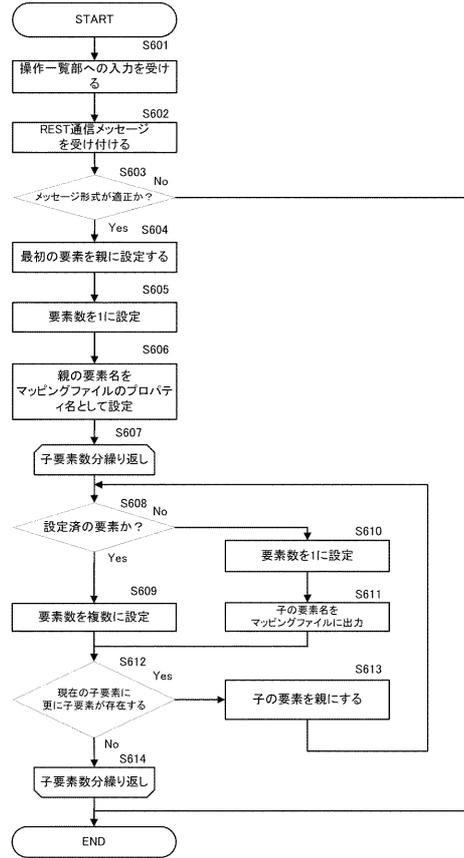
【図4】



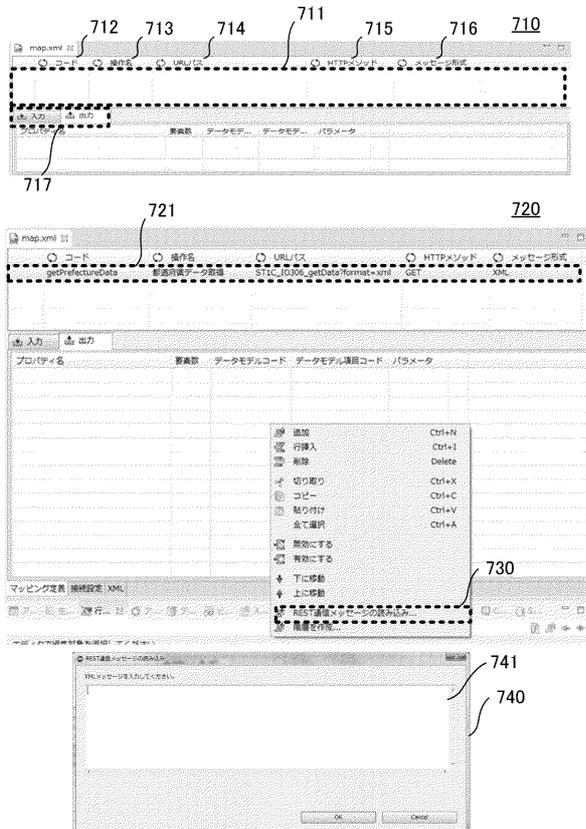
【図5】



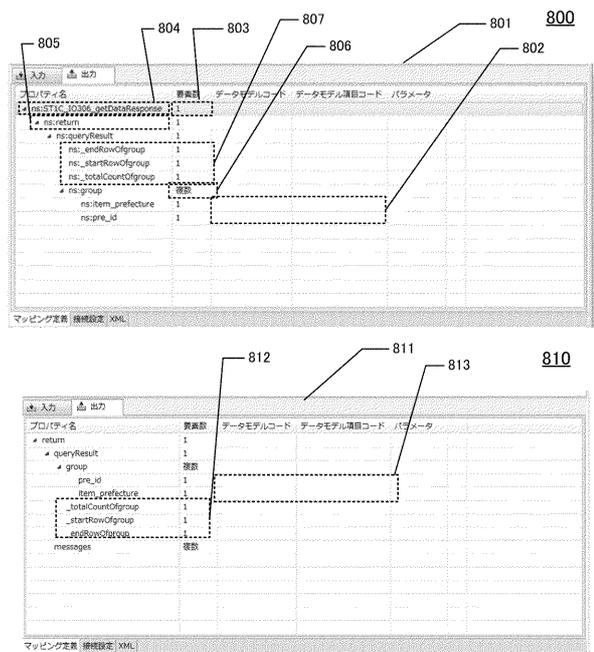
【図6】



【図7】



【図8】







---

フロントページの続き

Fターム(参考) 5E555 AA28 AA30 BA02 BA45 BB02 BC17 DB41 DB58 DC15 DC18  
DD07 EA07 EA08 FA00