

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7541227号
(P7541227)

(45)発行日 令和6年8月28日(2024.8.28)

(24)登録日 令和6年8月20日(2024.8.20)

(51)国際特許分類

F I

G 0 6 F 8/34 (2018.01)

G 0 6 F 8/34

請求項の数 10 (全22頁)

(21)出願番号	特願2020-72673(P2020-72673)	(73)特許権者	390002761
(22)出願日	令和2年4月15日(2020.4.15)		キヤノンマーケティングジャパン株式会
(65)公開番号	特開2021-170199(P2021-170199		社
	A)		東京都港区港南2丁目16番6号
(43)公開日	令和3年10月28日(2021.10.28)	(73)特許権者	592135203
審査請求日	令和5年4月14日(2023.4.14)		キヤノンITソリューションズ株式会
			社
			東京都港区港南2丁目16番6号
		(74)代理人	100189751
			弁理士 木村 友輔
		(72)発明者	鈴木 智子
			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	柴本 文洋
			東京都品川区東品川2丁目4番11号
			最終頁に続く

(54)【発明の名称】 情報処理装置、情報処理システム、制御方法、プログラム及び記録媒体

(57)【特許請求の範囲】

【請求項1】

情報処理装置を、
処理が定義された部品を対応付けることが可能なオブジェクトを編集画面上に配置する配置手段と、
前記配置手段により配置されたオブジェクトと前記処理が定義された部品とを対応付ける対応付手段と、
前記配置手段により前記オブジェクトが配置された編集画面に対応するアプリケーションの画面において、前記対応付手段により当該オブジェクトに対応付けられた部品に定義された処理の実行が可能な前記アプリケーションの構築を制御する構築手段
として機能させるためのプログラムであって、
前記対応付手段は、
前記配置手段により配置されたオブジェクトと、前記処理が定義された部品であって、前記プログラムを提供する者以外の者が作成した部品または前記プログラムを提供する者以外の者が提供する部品とを対応付けること
を特徴とするプログラム。

【請求項2】

前記配置手段は、
前記処理が定義された部品であって、前記プログラムを提供する者以外の者が作成した部品または前記プログラムを提供する者以外の者が提供する部品を対応付けることが可能な

オブジェクトを編集画面上に配置すること
を特徴とする請求項 1 に記載のプログラム。

【請求項 3】

前記情報処理装置を、
前記配置手段により配置されたオブジェクトと、前記処理が定義された部品であって、前記プログラムを提供する者以外の者が作成した部品または前記プログラムを提供する者以外の者が提供する部品とを対応付けるための画面を表示する画面表示手段
として機能させるための請求項 1 又は 2 に記載のプログラム。

【請求項 4】

前記情報処理装置を、
前記配置手段により配置されたオブジェクトと、前記処理が定義された部品であって、前記プログラムを提供する者以外の者が作成した部品または前記プログラムを提供する者以外の者が提供する部品とを対応付けるために当該部品の識別情報を受け付ける受付手段
として機能させるための請求項 1 乃至 3 のいずれか 1 項に記載のプログラム。

【請求項 5】

前記プログラムを提供する者以外の者が作成した部品または前記プログラムを提供する者以外の者が提供する部品は、前記情報処理装置にテンプレートとして記憶されていない部品であること
を特徴とする請求項 1 乃至 4 のいずれか 1 項に記載のプログラム。

【請求項 6】

前記構築手段を、
前記配置手段により前記オブジェクトが配置された編集画面に対応する前記アプリケーションの画面において、前記処理が定義された部品であって、前記プログラムを提供する者以外の者が作成した部品または前記プログラムを提供する者以外の者が提供する部品の表示が可能な前記アプリケーションの構築を制御する手段
として機能させるための請求項 1 乃至 5 のいずれか 1 項に記載のプログラム。

【請求項 7】

処理が定義された部品を対応付けることが可能なオブジェクトを編集画面上に配置する配置手段と、
前記配置手段により配置されたオブジェクトと前記処理が定義された部品とを対応付ける対応付手段と、
前記配置手段により前記オブジェクトが配置された編集画面に対応するアプリケーションの画面において、前記対応付手段により当該オブジェクトに対応付けられた部品に定義された処理の実行が可能な前記アプリケーションの構築を制御する構築手段
として機能させるためのプログラムであって、
前記処理が定義された部品とは、前記プログラムを提供する者以外の者が作成した部品または前記プログラムを提供する者以外の者が提供する部品であること
を特徴とするプログラムが読み込まれた情報処理装置。

【請求項 8】

情報処理装置を制御するための制御方法であって、
処理が定義された部品を対応付けることが可能なオブジェクトを編集画面上に配置する配置ステップと、
前記配置ステップにより配置されたオブジェクトと前記処理が定義された部品とを対応付ける対応付ステップと、
前記配置ステップにより前記オブジェクトが配置された編集画面に対応するアプリケーションの画面において、前記対応付ステップにより当該オブジェクトに対応付けられた部品に定義された処理の実行が可能な前記アプリケーションの構築を制御する構築ステップとを含む制御方法であって、
前記処理が定義された部品とは、前記制御方法をコンピュータに実行させるプログラムを提供する者以外の者が作成した部品または前記制御方法をコンピュータに実行させるプロ

10

20

30

40

50

グラムを提供する者以外の者が提供する部品であること
を特徴とする制御方法。

【請求項 9】

処理が定義された部品を対応付けることが可能なオブジェクトを編集画面上に配置する配置手段と、

前記配置手段により配置されたオブジェクトと前記処理が定義された部品とを対応付ける対応付手段と、

前記配置手段により前記オブジェクトが配置された編集画面に対応するアプリケーションの画面において、前記対応付手段により当該オブジェクトに対応付けられた部品に定義された処理の実行が可能な前記アプリケーションの構築を制御する構築手段

10

として機能させるためのプログラムであって、

前記処理が定義された部品とは、前記プログラムを提供する者以外の者が作成した部品または前記プログラムを提供する者以外の者が提供する部品であること

を特徴とするプログラムが読み込まれた情報処理システム。

【請求項 10】

情報処理装置を、

処理が定義された部品を対応付けることが可能なオブジェクトを編集画面上に配置する配置手段と、

前記配置手段により配置されたオブジェクトと前記処理が定義された部品とを対応付ける対応付手段と、

20

前記配置手段により前記オブジェクトが配置された編集画面に対応するアプリケーションの画面において、前記対応付手段により当該オブジェクトに対応付けられた部品に定義された処理の実行が可能な前記アプリケーションの構築を制御する構築手段

として機能させるためのプログラムであって、

前記処理が定義された部品とは、前記プログラムを提供する者以外の者が作成した部品または前記プログラムを提供する者以外の者が提供する部品であること

を特徴とするプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

30

情報処理装置、情報処理システム、制御方法、プログラム及び記録媒体に関する。

【背景技術】

【0002】

近年、アプリケーション開発プログラムを用いた開発の効率化が行われている。

【0003】

例えば、画面の設計に際し、ソースコードを手打ちして画面を作成するのではなく、入力フィールドやボタン等の部品（開発中のアプリケーションにおける画面に配置可能な項目）を画面上に配置し画面設計ができる技術が普及している。

【0004】

しかし、1つのアプリケーション開発プログラムが提供している部品や当該部品が持つ機能には限りがある。ときには当該アプリケーション開発プログラムが提供している部品以外の、第三者が作成・公開（提供）している部品、および、当該部品により実現される機能を開発中のアプリケーションに組み込みたいことがある。

40

【0005】

当該部品を組み込むための方法の1つとして、当該部品の見た目（表示）・機能を参考に、同等の見た目と機能を持つ部品を1から作成する方法が考えられるが、非常に手間であり、これはアプリケーション開発の工程全体における効率の低下に繋がる。そこで、既に公開されている当該部品の見た目（表示）にかかるコードや、機能のコードを流用することが望まれている。

【0006】

50

例えば特許文献 1 には、プログラム開発・編集用のエディタにおいて、エディタに未登録の特殊命令を含むプログラムを開発する際、既存の命令に識別子を付与したプログラムを作成しておき、当該プログラムを、当該識別子が付された命令を変換ルールに従って特殊命令に置き換えたプログラムに変換する技術が記載されている。また、プログラム中の要素が当該識別子に該当するとき、当該要素を変換ルールに登録された関数に変換することで、特殊命令の引数の数や内容を容易に変更、設定できることが記載されている。

【先行技術文献】

【特許文献】

【0007】

【文献】特開平 11 - 24905 号公報

10

【発明の開示】

【発明が解決しようとする課題】

【0008】

特許文献 1 の技術によれば、既にある機能を簡単に呼び出して実現することができる。しかしながら、特許文献 1 の技術では、プログラムのコードに直接、特殊命令の識別子を付与しなければならない。

【0009】

例えば、部品のデータの出力処理を、異なるデータ処理に置き換える場合、当該データの出力処理にあたる部分をコードの中から探し出す必要がある。つまり、ユーザが、特殊命令を付与したい出力命令がどれか認識し、コードを追記・書き換えできる程度には、プログラム開発アプリケーションにより作成されたコードの読解力を要し、プログラミングそのものに精通していなければならない。

20

【0010】

本発明は、アプリケーション開発プログラムを提供する者以外の者が作成した部品またはアプリケーション開発プログラムを提供する者以外の者が提供する部品を用いて容易にアプリケーション開発が可能な仕組みを提供することを目的とする。

【課題を解決するための手段】

【0011】

本発明は、処理が定義された部品を対応付けることが可能なオブジェクトを画面上に配置する配置手段と、前記配置手段により配置されたオブジェクトと前記処理が定義された部品とを対応付ける対応付手段と、前記配置手段により前記オブジェクトが配置された画面に対応するアプリケーション画面において、前記対応付手段により当該オブジェクトに対応付けられた部品に定義された処理の実行が可能なプログラムの生成を制御する生成手段として機能させるためのプログラムであって、前記対応付手段は、前記配置手段により配置されたオブジェクトと、前記処理が定義された部品であって、前記プログラムを提供する者以外の者が作成した部品または前記プログラムを提供する者以外の者が提供する部品とを対応付けることを特徴とする。

30

【発明の効果】

【0012】

本発明によれば、既存の部品の表示や機能を用いて容易にアプリケーション開発が可能な仕組みを提供することができる。

40

【図面の簡単な説明】

【0013】

【図 1】本発明の実施形態における、情報処理システムの構成の一例を示す図である。

【図 2】本発明の実施形態における、各種装置のハードウェア構成の一例を示す図である。

【図 3】本発明の実施形態における、各種装置の機能構成の一例を示す図である。

【図 4】本発明の実施形態における、各種装置の機能構成の詳細を示す図である。

【図 5】本発明の実施形態における、カスタム部品の登録処理の流れを示すフローチャートである。

【図 6】本発明の実施形態における、アプリケーション生成処理の流れを示すフローチャ

50

ートである。

【図 7】本発明の実施形態における、カスタム部品にかかるコード生成処理の流れを示すフローチャートである。

【図 8】本発明の実施形態における、表示画面の構成の一例を示す図である。

【図 9】本発明の実施形態における、データ構成の一例を示す図である。

【図 10】本発明の実施形態における、ソースコードの一例を示す図である。

【図 11】本発明の実施形態における、ソースコードの一例を示す図である

【発明を実施するための形態】

【0014】

以下、図面を参照して、本発明の実施の形態の一例について説明する。

10

【0015】

まず図 1 を参照して、本発明の実施形態における、情報処理システムの構成の一例について説明する。

【0016】

プログラム開発装置 100、データベースサーバ 200、アプリケーションサーバ 300 は、ネットワーク 101 を介して通信可能に接続されている。

【0017】

プログラム開発装置 100 には、プログラム開発アプリケーションがインストールされており、プログラム（アプリケーションプログラム）を開発、生成することができる。

20

【0018】

例えば、対象のアプリケーションの画面を編集できる。具体的には、当該画面上に部品（アプリケーションが表示するボタンやテキスト入力フィールド等）を配置したり、当該部品のサイズ、形状、位置を設定・変更したりすることが可能である。

【0019】

また、当該部品の名称や、部品に対する操作を受け付けた場合の動作、入出力するデータ（入出力値 / 入出力データとも言う）や入出力先の画面、テーブル等を設定・編集可能である。

【0020】

データベースサーバ 200 には、検索アプリケーションがインストールされており、データベースサーバ 200 が参照可能なデータベース内のデータを検索することができる。

30

【0021】

アプリケーションサーバ 300 は、プログラム開発装置 100 によって生成されたアプリケーションが稼働するサーバである。プログラム開発装置 100 は、生成したアプリケーションのコードを自機に記憶した後、ユーザ操作に従って、当該コードをアプリケーションサーバ 300 に記憶する。以上が図 1 の説明である。

【0022】

次に図 2 を参照して、本発明の実施形態における、各種装置のハードウェア構成について説明する。

【0023】

図 2 において、201 は CPU で、システムバス 204 に接続される各デバイスやコントローラを統括的に制御する。また、ROM 202 あるいは外部メモリ 211 には、CPU 201 の制御プログラムである BIOS（Basic Input / Output System）やオペレーティングシステムプログラム（以下、OS）や、各サーバ或いは各 PC の実行する機能を実現するために必要な後述する各種プログラム等が記憶されている。

40

【0024】

203 は RAM で、CPU 201 の主メモリ、ワークエリア等として機能する。CPU 201 は、処理の実行に際して必要なプログラム等を ROM 202 あるいは外部メモリ 211 から RAM 203 にロードして、該ロードしたプログラムを実行することで各種動作を実現するものである。

50

【 0 0 2 5 】

また、205は入力コントローラで、キーボード（KB）209や不図示のマウス等のポインティングデバイス等からの入力を制御する。

【 0 0 2 6 】

206はビデオコントローラで、ディスプレイ210（例えばCRTディスプレイ）等の表示器への表示を制御する。なお、表示器はCRTディスプレイだけでなく、液晶ディスプレイ等の他の表示器であってもよい。これらは必要に応じてユーザが使用するものである。

【 0 0 2 7 】

207はメモリコントローラで、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、ユーザファイル、編集ファイル、各種データ等を記憶するハードディスク（HD）や、フレキシブルディスク（FD）、或いは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）メモリ等の外部メモリ211へのアクセスを制御する。

10

【 0 0 2 8 】

208は通信I/Fコントローラで、ネットワーク（例えば、図1に示したネットワーク101）を介して外部機器と接続・通信するものであり、ネットワークでの通信制御処理を実行する。例えば、TCP/IPを用いた通信等が可能である。

【 0 0 2 9 】

なお、CPU201は、例えばRAM203内の表示情報用領域へアウトラインフォントの展開（ラスターライズ）処理を実行することにより、ディスプレイ210上での表示を可能としている。また、CPU201は、ディスプレイ210上の不図示のマウスカーソル等でのユーザ指示を可能とする。

20

【 0 0 3 0 】

本発明を実現するための後述する各種プログラムは、外部メモリ211に記録されており、必要に応じてRAM203にロードされることによりCPU201によって実行されるものである。さらに、上記プログラムの実行時に用いられる定義ファイル及び各種情報テーブル等も、外部メモリ211に格納されており、これらについての詳細な説明も後述する。以上が図2の説明である。

【 0 0 3 1 】

次に図3を参照して、本発明の実施形態における、各種装置の機能構成について説明する。

30

【 0 0 3 2 】

部品記憶部311は、プログラム開発装置100にインストールされているプログラム開発アプリケーションが、テンプレートとして記憶管理している部品（テンプレート部品）と、それ以外の部品（非テンプレート部品）を記憶する記憶部である。

【 0 0 3 3 】

非テンプレート部品とは、例えばユーザが独自に作成した部品や、プログラム開発アプリケーションの提供元とは異なる企業、すなわち、プログラム開発アプリケーションの開発元企業から見た第三者が公開している部品等である。

40

【 0 0 3 4 】

非テンプレート部品の特徴を説明する。テンプレート部品が、そのテンプレート部品を提供しているプログラム開発アプリケーションに最適化されているのに対して、非テンプレート部品は必ずしもそうではない。

【 0 0 3 5 】

例えば、非テンプレート部品自体の、部品に入力された情報（例：テキスト）の登録先のデータ項目名や、当該部品に出力する（例：グラフ等の部品に表示するためにデータベースから当該部品に出力する）データのデータ項目名、これらのデータのデータ型や制限（最大表示桁数、全角・半角の制限）等は、プログラム開発装置100にインストールされているプログラム開発アプリケーションの扱う、あるいは、当該アプリケーションで開

50

発するプログラム（プログラム開発アプリケーションにより生成されるアプリケーション）が扱うデータ項目名、データ型、許容する制限等とは必ずしも一致しない。

【 0 0 3 6 】

非テンプレート部品を、テンプレート部品と同じように、確実に、当該プログラム開発アプリケーション、およびプログラム開発アプリケーションにより生成されるアプリケーションで利用するためには、当該非テンプレート部品を利用するアプリケーション用に最適化する必要があるのである。

【 0 0 3 7 】

図 3 の説明に戻る。カスタム I / O 項目記憶部 3 1 2 は、カスタム I / O 項目の画像や、カスタム I / O 項目に設定された非テンプレート部品、および非テンプレート部品からの / 非テンプレート部品へのデータの入出力に際して利用するデータの変換ルールを記憶する機能部である。

【 0 0 3 8 】

I / O 項目（入出力項目）とは、データの入力または出力、あるいはその両方を実行可能な機能を備える部品である。例えば、テキストフィールドや、グラフオブジェクト等がこれにあたる。

【 0 0 3 9 】

カスタム I / O 項目（カスタム入出力項目）とは、非テンプレート部品と、その非テンプレート部品のデータの入出力をするための設定がされた入出力項目（部品）である。カスタム I / O 項目に設定された部品のことをカスタム部品と呼ぶ。

【 0 0 4 0 】

カスタム I / O 項目配置処理部 3 1 3 は、開発対象のアプリケーションの画面上に、カスタム I / O 項目を配置する機能部である。例えば、画面エディタの脇に表示されているカスタム I / O 項目の画像を、画面上にドラッグ＆ドロップすることで、ドロップされた位置にカスタム I / O 項目の画像を配置する。

【 0 0 4 1 】

指定受付画面表示部 3 1 4 は、カスタム I / O 項目と、どの非テンプレート部品を対応付けるかの指定を受け付けるための画面を表示する。指定受付部 3 1 5 は、カスタム I / O 項目と対応付ける、非テンプレート部品の指定を受け付ける機能部である。

【 0 0 4 2 】

アプリケーション生成部 3 1 6 は、アプリケーションのコードを生成する機能部である。例えば、カスタム I / O 項目のコードを含む、アプリケーション全体のソースコードを生成する。例えば、後述する図 4 のアプリケーション生成部 4 1 0 と、カスタム部品ソース生成部 4 3 0 の機能を備える。以上が図 3 の説明である。

【 0 0 4 3 】

次に図 4 を参照して、本発明の実施形態における、詳細な各種装置の機能構成について説明する。

【 0 0 4 4 】

プログラム開発装置 1 0 0 はアプリケーションを開発する開発者がプログラム開発装置を操作することにより定義した（プログラム開発装置のプログラム開発アプリケーションにより生成された）定義ファイルを管理するリポジトリ定義部 4 0 0 と、それらをもとにアプリケーションのコードを生成するアプリケーション生成部 4 1 0 を備える。

【 0 0 4 5 】

さらに、拡張ファイルを記憶管理する拡張ファイル管理部 4 2 0 を備える。

【 0 0 4 6 】

さらに、カスタム I / O 項目のソースコードを生成するカスタム部品ソース生成部 4 3 0 を備える。すなわち、カスタム I / O 項目に設定されたカスタム部品に関連するソースを生成する生成部である。

【 0 0 4 7 】

アプリケーションサーバ 3 0 0 は、各種コードの生成部によって生成されたコードによ

10

20

30

40

50

り構成されるアプリケーションを記憶する。

【 0 0 4 8 】

以下、各機能部の詳細について説明する。リポジトリ定義部 4 0 0 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、データベース定義 4 0 5を管理している。

【 0 0 4 9 】

これらのファイルはプログラム開発アプリケーションを介して開発者によって入力され、作成され、プログラム開発装置 1 0 0 の外部メモリに記憶される。

【 0 0 5 0 】

アプリケーション定義 4 0 1 とは、プログラム開発アプリケーションにおいて開発するアプリケーションごとに保持される定義であって、そのアプリケーションで使用する設定、データ等を定義した情報である。具体的には、アプリケーションを生成する際に使用する入出力定義や、アプリケーション定義そのものの識別コード、アプリケーション名を定義可能である。

【 0 0 5 1 】

入出力定義 4 0 2 とは、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含む定義情報である。

【 0 0 5 2 】

例えば、当該定義の種類（項目タイプ）や、当該定義自体の識別コード（項目コード）、名前、データを入出力するデータ項目の項目コード、当該データ項目を備えるデータモデルの識別コード（データモデルコード）、当該入出力定義によって定義される部品を表示画面に表示するか否かを示す設定（表示設定）、入出力するデータの桁数や少数点以下何桁までを入出力するかの設定（桁数、少数桁等）等が定義されている。定義の編集画面（定義の一覧を表示した画面）の一例を図 9 に示す。

【 0 0 5 3 】

データモデル定義 4 0 3 とは、スキーマ情報として定義される定義である。各データモデル定義は、識別コードと名前を保持する。データモデル定義には複数のデータ項目が定義されている。具体的には、各種データ項目の項目コード、項目名、そのデータ項目で使用する桁数やデータタイプ（文字、数値、日付等のデータの型）、記憶するデータとして `null` を許可するか否かの許可設定等が定義されている。

【 0 0 5 4 】

ビジネスプロセス定義 4 0 4 とは、アプリケーションで動作するビジネスロジックを定義した情報である。例えば、ビジネスロジックごと操作タイプ（「IN」はビジネスプロセスへの入力データの変数格納、「CALL」はデータモデルコードが保有する機能コードの内容を呼び出す、等）、データモデルコード、機能を示す機能コード等を含む。

【 0 0 5 5 】

データベース定義 4 0 5 とは、プログラム開発アプリケーションにより生成されたプログラム（アプリケーション）がアクセスするサーバの設定を記憶した定義である。具体的には、アクセスするデータベースを保持しているサーバの識別コード、サーバ名、接続先 URL、データベース名、データベースへの接続に必要なユーザ名およびパスワード等が定義されている。リポジトリ定義部 4 0 0 は、これらの定義を管理する手段の一例である。

【 0 0 5 6 】

拡張ファイル管理部 4 2 0 とは、プログラム開発アプリケーションの機能拡張をするための拡張ファイルを管理する機能部である。拡張ファイルとは、非テンプレート部品のファイルである。非テンプレート部品のファイルは、プログラム開発装置 1 0 0 が、ユーザからの操作等に従って、自機の外部メモリ 2 1 1 に記憶する。例えば、インターネット上において、商用利用可能なデータとして公開されている部品を、ユーザ操作に従ってダウンロードする等である。

10

20

30

40

50

【 0 0 5 7 】

拡張ファイル管理部 4 2 0 の詳細を説明する。

【 0 0 5 8 】

インクルードファイル管理部 4 2 1 は、ＪＳＰファイルに取り込む（呼び出す）するファイル（ＣＳＳファイル、ＪＳファイル等）を管理する管理機能部である。

【 0 0 5 9 】

インクルードするＪＳファイル 4 2 2（ＪａｖａＳｃｒｉｐｔ（登録商標）ファイル）は、インクルードする対象として指定されたスクリプトファイルである。ＪＳファイルには、非テンプレート部品の動作（機能）に関するコードが記載されている。例えば、当該非テンプレート部品のテキストフィールドに対するマウスクリック操作がされるまでは、テキストフィールドの中に「入力してください」という文字列を表示しておき、テキスト入力フィールドに対するマウスクリック操作がされると、当該文字列を非表示にする、等の動作である。

10

【 0 0 6 0 】

また、当該非テンプレートファイルがグラフの場合、例えば、非テンプレート部品がグラフの場合、当該グラフにマウスカーソルがホバーされた際に、グラフを強調表示したり、対象のグラフの数値を表示したりするための動作（機能）に関するコードが記載されている。

【 0 0 6 1 】

インクルードするＣＳＳファイル 4 2 3（Ｃａｓｃａｄｉｎｇ Ｓｔｙｌｅ Ｓｈｅｅｔｓ ファイル）は、インクルードする対象として指定されたスタイルシートである。例えば、非テンプレート部品がグラフである場合、そのグラフの表示領域自体や、そのグラフ自体に対してスタイルを指定したＣＳＳファイルが記憶されているものとする。

20

【 0 0 6 2 】

コード管理部 4 2 4 は、各種コードを管理する機能部である。

【 0 0 6 3 】

部品描画コード 4 2 5 は、ＪＳＰファイルの中のカスタムＩ／Ｏ項目として設定された部品を描画する部分のコードである。例えば、マウスホバー等によりある要素がフォーカスされた場合に、どのようなイベントを実行するかが記述される。当該イベントの内容はＪＳファイルに記述されている。一例を図 1 0 の 1 0 0 2 に示す。なお、当該説明は一例であり、グラフの部品において、当該説明とは異なる描画に関するコードが記載されることもある。

30

【 0 0 6 4 】

データ処理部と描画部品の関連付けコード 4 2 6 は、非テンプレート部品の描画とプログラム開発アプリケーションが管理するデータとを関連付けるコードである。例えば、非テンプレと部品がグラフの場合、当該非テンプレート部品の入出力定義にどのような設定がされている（どのようなパラメータが指定されている）場合に、どのデータを取得して、グラフに描画するかを記述したコードである。本発明の実施形態において、当該関連付けコードはＪＳファイルである。一例を図 1 1 に示す。なお、当該説明は一例であり、グラフの部品において、当該説明とは異なる内容のコードが記載されることもある。

40

【 0 0 6 5 】

アプリケーション生成部 4 1 0 は、リポジトリ定義解析部 4 1 1 によってリポジトリ定義部 4 0 0 の各種定義を解析し、アプリケーションコード生成部 4 1 2 を介し、ソースコードコンパイル部 4 1 3 にて、コンパイル済みコード 4 4 3 とＨＴＭＬ／ＪＳＰ／ＪａｖａＳｃｒｉｐｔ（登録商標） 4 4 4 からなるアプリケーションコード 4 4 2 を生成する。例えばＷｅｂアプリケーション 4 4 1（Ｗｅｂアプリケーションのコード）を生成する。

【 0 0 6 6 】

アプリケーションコードは、当該コードによって構成されるアプリケーションがその機能を実行する際に利用するデータベース定義 4 0 5 に指定されたデータベース 4 4 5 にアクセスする。すなわち、アプリケーション生成部 4 1 0 は、アプリケーションに用いられ

50

るプログラムを生成するルールを記憶する手段の一例である。

【 0 0 6 7 】

カスタム部品ソース生成部 4 3 0 は、カスタム部品を描画し、そのカスタム部品にかかる処理の内容を示すソースを生成する処理部である。

【 0 0 6 8 】

インクルード部生成部 4 3 1 は、カスタム I / O 項目が配置されている画面の J S P ファイルの中に、当該カスタム I / O 項目に設定された非テンプレート部品（カスタム部品）の C S S、J S ファイルを取り込む処理を示すコードを記述する機能部である。インクルード部生成部で生成した J S P ファイルの一例を、図 1 0 の 1 0 0 1 に示す。

【 0 0 6 9 】

ボディ部生成部 4 3 2 は、カスタム I / O 項目が配置されている画面の J S P ファイルの中に、当該カスタム I / O 項目に関する描画コードやデータ処理コードを記述する機能部である。ここでは、描画部コードとデータ処理コードを記述した部分をボディ部という。一例を図 1 0 の 1 0 0 3 に示す。

【 0 0 7 0 】

データ処理部コード生成部 4 3 3 は、カスタム I / O 項目の内部でデータを保持する部分であるデータ処理部を生成する機能部である。一例を図 1 0 の 1 0 0 3 に示す。例えば 1 0 0 3 は、図 9 に示すカスタム I / O 項目（ B A R _ V E R T I C A L ）の内部値を保持するための隠れ項目「 d i s p 」である。すなわち、カスタム I / O 項目の内部値を保持するためのデータ処理部が、J S P に生成されていることを示している。

【 0 0 7 1 】

例えば、テキスト入力フィールドの部品に入力されたテキストを、不図示のデータベース上のデータ項目 X に登録する場合について説明する。

【 0 0 7 2 】

テキスト入力フィールドの部品に入力されたテキストを「 d i s p 」に書き込む旨が「データ処理部と描画部品の関連付けコード 4 2 6」に記述されている場合、テキスト入力フィールドの部品に入力された部品は、一度「 d i s p 」に書き込まれる。その後、別途アプリケーションコード 4 4 2 に記述された「「 d i s p 」に書き込まれたデータをデータ項目 X に書き込む」というコードに従って、当該データがデータ項目 X に書き込まれることとなる。

【 0 0 7 3 】

描画部コード生成部 4 3 4 は、カスタム I / O 項目が配置されている画面の J S P ファイルの中に、カスタム I / O 項目に設定された部品（カスタム部品）を描画するコードである描画部を生成（記述）する機能部である。以上が図 4 の説明である。

【 0 0 7 4 】

以下、図 5 ~ 図 1 1 を参照して、本発明の実施形態における処理の流れについて説明する。なお、各種フローチャートの処理は、各装置に搭載された C P U 2 0 1 が、各装置が備える機能部の機能を用いて実行するものとする。

【 0 0 7 5 】

まず、図 5 を参照して、本発明の実施形態におけるカスタム部品の登録処理の流れを説明する。

【 0 0 7 6 】

ステップ S 5 0 1 で、プログラム開発装置 1 0 0 は、ユーザ受け付けた操作に従って、プログラム開発アプリケーションを起動する。

【 0 0 7 7 】

ステップ S 5 0 2 で、プログラム開発装置 1 0 0 は、プログラム開発アプリケーションの提供する、アプリケーション画面の編集機能（画面エディタ機能）を起動して、画面編集画面（画面エディタ画面）を表示する。エディタ画面には、カスタム I / O 項目や、テンプレート部品を示すオブジェクトを表示する。つまり、画面上に項目表示を行なう。

【 0 0 7 8 】

10

20

30

40

50

ステップ S 5 0 3 で、プログラム開発装置 1 0 0 のプログラム開発アプリケーションは、カスタム I / O 項目を画面上に生成する。具体的には、図 8 の 8 0 0 に示すように、カスタム I / O 項目のオブジェクト 8 0 1 を、編集集中の画面上にドラッグ & ドロップすることで、ドロップされた画面上の場所にカスタム I / O 項目を配置する。

【 0 0 7 9 】

配置した時点では、まだ、当該カスタム I / O 項目を、どの非テンプレート部品として表示し、動作させるかが未決定の状態であるため、配置先には、カスタム I / O 項目を示すダミーオブジェクトを表示する（ダミー画像を内包したオブジェクト表示をする）。

【 0 0 8 0 】

ダミーオブジェクトのサイズのデフォルト値は、プログラム開発装置 1 0 0 の外部メモリ 2 1 1 に記憶されている。当該サイズは、ダミーオブジェクトの端をドラッグする等の操作により変更する。プログラム開発アプリケーションは、当該サイズで当該カスタム I / O 項目に対応付けられた部品（カスタム部品）を表示するプログラムを生成する。具体的には、当該画面の J S P ファイルにおけるボディ部の `div` タグの `class` 属性に、設定されているスタイルクラスに変更後のサイズを書き込む。例えば、図 1 0 の例の場合は、「`field__BAR__VERTICAL`」に書き込む。

【 0 0 8 1 】

なお、テンプレート部品に対応する入出力項目（テンプレート項目）が配置された場合は、当該入出力項目に対応するテンプレート部品の画像を記憶装置から取得して、オブジェクトとして表示する。8 0 2 に示すリストがその一例である。

【 0 0 8 2 】

ステップ S 5 0 4 で、プログラム開発装置 1 0 0 のプログラム開発アプリケーションは、配置したカスタム I / O 項目と、非テンプレート部品を対応付ける設定を行なう。

【 0 0 8 3 】

具体的には、ユーザ操作に従って、プログラム開発アプリケーションの提供する定義編集機能を起動し、定義一覧画面（図 9 の 9 0 0 に例示）を表示する。そして、ステップ S 5 0 3 で配置されたカスタム I / O 項目のプロパティに、ユーザが所望する非テンプレート部品の識別情報と、当該非テンプレート部品に関する処理のうちどの処理を実行するかの設定値の入力を受け付ける。

【 0 0 8 4 】

具体例を図 9 に示す。図 9 の 9 0 0 においては、デフォルトの項目コードである「`XXX`」であり、当該非テンプレート部品の識別情報の登録先であるプロパティ = 「`compName`」9 0 1 と、パラメータの設定先であるプロパティ = 「`compParam`」9 0 2 が空欄の入出力項目が選択されている。ユーザは、当該選択した入出力項目の各項目を編集可能である。つまり、設定値の入力・設定が可能である。

【 0 0 8 5 】

設定後の画面イメージを図 9 の 9 1 0 に示す。図 9 の 9 1 0 において、項目コード = 「`BAR__VERTICAL`」には、「`GRAPH`」という非テンプレート部品の識別情報が設定されている。また、当該「`GRAPH`」の部品に設定する値（パラメータ）として「`"type": "PROCEED"`」が設定されている。

【 0 0 8 6 】

当該非テンプレート部品の識別情報の登録先であるプロパティ = 「`compName`」と、パラメータの設定先であるプロパティ = 「`compParam`」は、カスタム I / O 項目にのみ表示する。実際には、カスタム I / O 項目にこれらのプロパティ欄が対応付けて外部メモリに記憶されており、カスタム I / O 項目が画面に配置された場合に、入出力項目の一覧に、これらのプロパティ欄が空のカスタム I / O 項目を新規生成する。

【 0 0 8 7 】

なお、本実施形態における非テンプレート部品とは、非テンプレート部品の識別情報と同一名称のフォルダに格納されている、C S S ファイル、J S ファイル、非テンプレート部品の描画コード、当該非テンプレート部品のから構成される情報である。非テンプレ

10

20

30

40

50

ト部品の例を説明する。

【 0 0 8 8 】

非テンプレート部品のCSSファイルには、当該非テンプレート部品の表示に関する設定（描画コード）が記述されている。例えば、グラフの種類、色、サイズ等である。

【 0 0 8 9 】

また、非テンプレート部品のJSファイルには、当該非テンプレート部品の描画とデータ処理を関連付けるコード（図 1 1 ）が記述されている。

【 0 0 9 0 】

例えば、1 1 0 1 には、「GRAPH」のCSSファイルによる描画（ここではグラフの描画）をする場合に、設定値を参照し、設定値が「“ type ” : “ PROCEED ”」である場合には、定義一覧（図 9 ）の中の「年月」（MONTH__AN...）と「売上」（PROCEEDS）列からデータを取得し、グラフに出力する」という処理のコードが記述されている。また、1 1 0 2 には、「GRAPH」のCSSファイルによる描画（ここではグラフの描画）をする場合に、設定値を参照し、設定値が「“ type ” : “ EVENT ”」である場合には、定義一覧（図 9 ）の中の「年月」（MONTH__AN...）と「イベント回数」（EVENT）列からデータを取得し、グラフに出力する」という処理のコードが記述されている。以上が非テンプレート部品の説明である。

【 0 0 9 1 】

図 5 の説明に戻る。ステップ S 5 0 5 で、プログラム開発装置 1 0 0 のプログラム開発アプリケーションは、ステップ S 5 0 4 で入力を受け付けた情報を、外部メモリに記憶する。つまり、「BAR__VERTICAL」というカスタム I / O 項目のプロパティの各項目に、「GRPH」、「“ type ” : “ PROCEED ”」を記憶する。以上が図 5 の説明である。

【 0 0 9 2 】

次に、図 6 を参照して、アプリケーション生成処理の流れを説明する。

【 0 0 9 3 】

ステップ S 6 0 1 で、プログラム開発装置 1 0 0 のプログラム開発アプリケーションは、不図示のアプリケーション生成実行画面において、これからコードを生成するアプリケーションの定義（アプリケーション定義）の指定を受け付ける。

【 0 0 9 4 】

ステップ S 6 0 2 で、プログラム開発装置 1 0 0 のプログラム開発アプリケーションは、当該アプリケーション定義に基づいたアプリケーションコードの生成指示を受け付ける。例えば、当該アプリケーション生成実行画面における「生成実行」ボタンの押下操作を受け付ける。

【 0 0 9 5 】

ステップ S 6 0 3 で、プログラム開発装置 1 0 0 のプログラム開発アプリケーションは、ステップ S 6 0 1 で指定を受け付けたアプリケーション定義を、外部メモリ 2 1 1 から取得する。

【 0 0 9 6 】

ステップ S 6 0 4 で、当該アプリケーション定義によって特定されるデータモデル定義を、外部メモリ 2 1 1 から取得する。

【 0 0 9 7 】

ステップ S 6 0 5 で、当該アプリケーション定義によって特定される入出力定義を、外部メモリ 2 1 1 から取得する。

【 0 0 9 8 】

ステップ S 6 0 6 で、当該アプリケーション定義によって特定されるビジネスプロセス定義を、外部メモリ 2 1 1 から取得する。

【 0 0 9 9 】

ステップ S 6 0 7 で、当該アプリケーション定義によって特定されるデータベース定義を、外部メモリ 2 1 1 から取得する。

【 0 1 0 0 】

ステップ S 6 0 8 で、プログラム開発装置 1 0 0 のプログラム開発アプリケーションは、アプリケーション定義によって特定される拡張ファイルを、外部メモリ 2 1 1 から取得する。

【 0 1 0 1 】

ステップ S 6 0 9 で、プログラム開発装置 1 0 0 のプログラム開発アプリケーションは、アプリケーションコードを生成する。例えば、アプリケーションの J a v a（登録商標）ファイルや、各画面の J S P ファイル、C S S ファイル、J a v a S c r i p t ファイル（図 4 の 4 4 4 等参照）を生成する。

【 0 1 0 2 】

ステップ S 6 1 0 で、プログラム開発装置 1 0 0 のプログラム開発アプリケーションは、カスタム部品ソースを生成する。ステップ S 6 1 0 の詳細は、図 7 の説明で後述する。

【 0 1 0 3 】

ここで図 7 を参照して、本発明の実施形態における、カスタム部品にかかるコード生成処理の流れについて説明する。

【 0 1 0 4 】

ステップ S 7 0 1 で、プログラム開発装置 1 0 0 のプログラム開発アプリケーションは、入出力定義の一覧を取得する。そして、当該入出力項目の数だけ、ステップ S 7 0 2 ~ S 7 0 8 の処理を繰り返す。

【 0 1 0 5 】

ステップ S 7 0 2 で、プログラム開発装置 1 0 0 のプログラム開発アプリケーションは、未処理の入出力定義を 1 つ取得する。

【 0 1 0 6 】

ステップ S 7 0 3 で、プログラム開発装置 1 0 0 のプログラム開発アプリケーションは、取得した入出力定義の「c o m p N a m e」に、値が設定されているか判定する。つまり、当該入出力定義に、非テンプレート部品の識別情報が設定されているか判定する。

【 0 1 0 7 】

ここでは、非テンプレート部品の識別情報が設定されていると判定された部品を、カスタム I / O 項目として特定する。すなわち、ステップ S 7 0 3 の判定処理は、当該入出力定義が、カスタム I / O 項目の定義か判定する処理である。

【 0 1 0 8 】

入出力定義の「c o m p N a m e」に値が設定されていないと判定された場合、未処理の入出力定義を取得するべく、処理をステップ S 7 0 2 に戻す。未処理の入出力定義がない場合は、図 7 の処理を終了する。

【 0 1 0 9 】

入出力定義の「c o m p N a m e」に値が設定されていると判定された場合、処理をステップ S 7 0 4 に移行する。

【 0 1 1 0 】

ステップ S 7 0 4 で、プログラム開発装置 1 0 0 のプログラム開発アプリケーションは、取得した入出力定義の「c o m p P a r a m」に、値が設定されているか判定する。つまり、当該入出力定義にパラメータが設定されているか判定する。

【 0 1 1 1 】

「c o m p P a r a m」に、値が設定されていないと判定された場合は、処理をステップ S 7 0 6 に移行する。

【 0 1 1 2 】

「c o m p P a r a m」に、値が設定されていると判定された場合は、処理をステップ S 7 0 5 に移行し、設定されている値を取得して、処理をステップ S 7 0 6 に移行する。

【 0 1 1 3 】

ステップ S 7 0 6 で、プログラム開発装置 1 0 0 のプログラム開発アプリケーションは、カスタム部品のインクルード部を生成する。

10

20

30

40

50

【 0 1 1 4 】

具体的には、当該入出力定義が適用される入出力項目（カスタム I / O 項目）を保持する画面の J S P ファイルの中に、非テンプレート部品の C S S ファイルと J S ファイルをインクルードするためのコードを記述する。具体例を図 1 0 の 1 0 0 1 に示す。

【 0 1 1 5 】

ステップ S 7 0 7 で、プログラム開発装置 1 0 0 のプログラム開発アプリケーションは、カスタム部品のデータ処理部のコードを生成する。データ処理部は、カスタム部品の内部でデータを保持する部分である。一例を図 1 0 の 1 0 0 3 に示す。

【 0 1 1 6 】

ステップ S 7 0 8 で、プログラム開発装置 1 0 0 のプログラム開発アプリケーションは、カスタム部品の描画部のコードを生成する。描画部とは、カスタム部品を描画する部分である。描画部の一例を図 1 0 の 1 0 0 2 に示す。

【 0 1 1 7 】

プログラム開発装置 1 0 0 のプログラム開発アプリケーションは、未処理の入出力定義がある場合は、処理をステップ S 7 0 2 に戻し、入出力定義を 1 つ取得して、ステップ S 7 0 3 ~ S 7 0 8 の処理を実行する。未処理の入出力定義がなくなった場合に図 7 の処理を終了する。以上が図 7 の説明である。

【 0 1 1 8 】

図 6 の説明に戻る。ステップ S 6 1 1 で、プログラム開発装置 1 0 0 のプログラム開発アプリケーションは、ステップ S 6 0 9、S 6 1 0 で生成したコードをデプロイする。以上が図 6 の説明である。

【 0 1 1 9 】

以上説明したように、本発明によれば、既存の部品の表示や機能を用いて容易に、また、容易にアプリケーション開発が可能な仕組みを提供することができる。

【 0 1 2 0 】

当該部品の表示例を図 8 の 8 1 0 に示す。当該 8 1 0 に示す画面は、ステップ S 6 0 9、S 6 1 0 で生成され、ステップ S 6 1 0 でデプロイされたアプリケーションが、8 0 0 に示す画面を表示した場合の一例である。図 8 の 8 1 1 には、8 0 1 に対応付けられた非テンプレート部品（図 9 の 9 2 0 において「B A R _ V E R T I C A L」というカスタム I / O 項目に対応付けられた「G R A P H」）の表示例を示す。また、8 1 2 には、8 0 1 とは別に配置されたカスタム I / O 項目に対応付けられた非テンプレート部品（図 9 の 9 2 0 において「B A R _ V E R T I C A L 2」というカスタム I / O 項目に対応付けられた「G R A P H」）の表示例を示している。

【 0 1 2 1 】

これらの部品は、当該「G R A P H」という非テンプレート部品の C S S に準じたスタイルで表示され、「G R A P H」に対応する J S ファイル（図 1 0 の 1 0 0 1 で J S P ファイルにインクルードされた情報）に従った機能を保持する。

【 0 1 2 2 】

また、図 1 0 の描画コード 1 0 0 2 およびデータ処理部 1 0 0 3、図 1 1 の関連付けコードに従って描画、データの入出力を実行可能である。

【 0 1 2 3 】

なお、上述した実施形態の説明においては、カスタム I / O 項目として設定された非テンプレートファイルの C S S ファイル（部品の表示に関する情報の 1 つ）と、J S ファイル（部品の機能に関する情報の 1 つ）の両方をインクルードする例を記載したが、どちらか一方をインクルードするよう発明を構成してもよい。

【 0 1 2 4 】

また、上述した実施形態の説明においては、ユーザ操作に従って、プログラム開発アプリケーションの提供する定義編集機能を起動し、定義一覧画面 9 0 0（図 9 に例示）を表示するものとしたが、自動で表示するようにしてもよい。例えば、プログラム開発アプリケーションが、S 5 0 3 でカスタム I / O 項目が配置（生成）されたことを検知した場合

10

20

30

40

50

に、表示画面に定義一覧画面 900 を表示するようにしてもよい。また、当該カスタム I / O 項目に対応付ける部品を指定可能な状態の画面を表示するべく、表示した定義一覧画面 900 において、S503 で生成したカスタム I / O 項目に自動でフォーカスする（または当該カスタム I / O 項目にフォーカスされた状態で定義一覧画面 900 を表示する）ようにしてもよい。

【0125】

当該定義一覧画面 900 を自動で表示することにより、ユーザが、定義一覧画面を表示するための操作をする手間を軽減することができる。

【0126】

また、当該カスタム I / O 項目に対応付ける部品を指定可能な状態の画面を表示することにより、ユーザが、配置したカスタム I / O 項目を一覧から探し出す手間を軽減することができる。

10

【0127】

また、上述した実施形態の説明においては、カスタム I / O 項目として設定された非テンプレート部品からの入力値や当該非テンプレート部品への出力値を、例えばデータ処理部 1003 を介して行う旨を、描画とデータ処理の関連付けコード（図 11）に記述しているものとした。つまり、部品に対する操作を受け付けた場合のデータ処理の内容（その全部または一部）と、値の入力先や出力元の項目の識別情報が当該関連付けコードに記述されているものとした。

【0128】

20

しかし、第三者が公開する非テンプレート部品においては、当該値の入力先や出力元の項目の識別情報が、前記プログラム開発アプリケーションが扱うデータの項目名、データ型と必ずしも一致しないことがある。

【0129】

これは、描画とデータ処理の関連付けコードに相当するような、部品に入力されたデータをどのデータ項目に入力するか、や、部品に対してどのデータ項目からデータを出力するか等のコードが記述されたファイルを、前記プログラム開発アプリケーションの利用者ではなく、非テンプレート部品の提供者等の第三者により作成・公開されている場合にしばしば起こりうる。

【0130】

30

また、関連付けコード（図 11）の作成者が間違った項目名（例えば、データ処理部 1003 の隠れ項目と異なる項目名）を指定してしまった場合や、データモデル定義内のどこにも存在しない項目名を指定してしまった場合等に起こりうる。

【0131】

例えば、データ処理部である隠れ項目の項目名が「disp」であるが、当該関連付けコード（図 11）により指定されている項目名（例えば、グラフのデータの出力元のデータ項目名）が、データモデルの中に存在せず、また、「disp」でもない場合である。

【0132】

この場合、当該部品のデータはどこからも出力できず、どこにも入力できなくなってしまう。つまり、動作しない部品となってしまう。

40

【0133】

この問題を回避し、動作する部品を効率的に登録するため、すなわち、動作する部品で構築されたアプリケーションプログラムを効率的に作成するために、自動で、当該項目を更新・生成するようにしてもよい。

【0134】

例えば、S610 の後に、プログラム生成アプリケーションは、図 7 の処理で生成したデータ処理部の隠れ項目の項目名、桁数、少数桁、データ型を取得する。また、描画とデータ処理の関連付けコード（図 11）を取得する。

【0135】

プログラム生成アプリケーションは、当該関連付けコードを解析し、当該コードの中の

50

、 1 または複数の、データの入力先および出力元の項目名を特定する。そして、各データ項目名が、S 6 0 4 で取得したデータモデル定義またはデータ処理部の隠れ項目名のいずれかと一致するか判定する。つまり、データの入力先、および出力元が存在するか判定する。

【 0 1 3 6 】

一致する項目名がない場合（つまり、データの入力先、または出力元が存在しない場合）は、関連付けコード内の当該項目名を、当該関連付けコードに対応する非テンプレート部品のデータ処理部 1 0 0 3 の隠れ項目名に変更・更新する。一致する項目があった場合は、当該関連付けコード内の未処理の項目名を取得する。これを、関連付けコード内のデータの入力先、および出力元の項目名に対して繰り返す。その後、処理を S 6 1 1 に移行する。

10

【 0 1 3 7 】

当該変更・更新の処理により、データの入力先・出力元を、前記プログラム開発アプリケーションの扱うデータ項目名とすることができる。

【 0 1 3 8 】

すなわち、動作する部品で構築されたアプリケーションプログラムを容易に作成することができる。

【 0 1 3 9 】

また、桁数、小数点桁、データ型等のデータ項目の設定が、部品とデータ入力先および出力元とで違う場合、当該違いが原因でデータの登録エラーが発生することが考えられる。

20

【 0 1 4 0 】

よって、例えば S 6 1 0 の段階で、関連付けコードの中に記述されているデータの入力先のデータ項目名から、当該データ項目における桁数、少数桁、データ型を取得する。そして、関連付けコードの中に、部品に入力された値を、当該入力先のデータ項目の桁数、少数桁、データ型に変更する旨のコードを自動で生成し・挿入する。

【 0 1 4 1 】

また、例えば S 6 1 0 の段階で、関連付けコードの中に記述されているデータの出力元のデータ項目名を取得する。また、非テンプレート部品の J S ファイルに記述されている、当該部品に出力するデータの桁数、少数桁、データ型を取得する。そして、関連付けコードの中に、当該出力元のデータ項目から部品に出力する値を、当該部品の桁数、少数桁、データ型に変更する旨のコードを自動で生成し・挿入する。

30

【 0 1 4 2 】

当該変換用のコードの生成・挿入によって、桁数、小数点桁、データ型等の設定が異なる部品、項目間でもデータの入出力が可能となる。また、当該データの入出力が可能なプログラムを効率的に作成することができる。

【 0 1 4 3 】

すなわち、既存の部品の表示や機能を用いて効率的にアプリケーション開発が可能な仕組みを提供することができる。

【 0 1 4 4 】

なお、上述した実施形態においては、関連付けコード内の項目名をデータ処理部 1 0 0 3 の隠れ項目に更新する処理のコードを、当該関連付けコードの中に記述するものとしたが、例えば、どのような項目名をどのような項目名に変更・更新するかを、変更前の項目名と変更後の項目名を対応付けたルールとして外部メモリに記憶しておき、当該ルールに従って、関連付けコード内の、データモデル定義及び隠れ項目に一致しない項目名を変換・更新するようにしてもよい。

40

【 0 1 4 5 】

また、テンプレート部品や非テンプレート部品は、ファイルとして記憶・管理するのではなく、例えば、プログラム生成アプリケーション自体のコードの中に、各部品の C S S や、J S ファイル、描画コード等が記述されていることで記憶・管理されるようにしてもよい。

50

【 0 1 4 6 】

なお、テンプレート部品は入出力項目とペアで記憶されているものとする。つまり、テンプレート部品とペアになっている入出力項目が画面上に配置された場合には、部品と当該入出力項目を対応付けるための指示部（C o m p N a m e 等）は不要である。プログラム生成アプリケーションは、当該配置に応じて、既にテンプレート部品とペアになっている（対応付けられている）入出力項目を、入出力項目定義に追加・生成することになる。

【 0 1 4 7 】

また、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。

10

【 0 1 4 8 】

具体的には、プログラム開発装置 1 0 0 とデータベースサーバ 2 0 0、アプリケーションサーバ 3 0 0 が一体であり、上述した実施形態の各処理を実行するようにしてもよい。

【 0 1 4 9 】

また、拡張ファイル管理部 4 2 0 を、データベースサーバ 2 0 0 が備えるよう構成してもよい。すなわち、カスタム部品としてインクルードする部品のファイル（C S S ファイル、J S ファイル）や、当該部品の描画コード、データ処理と描画の対応付けコードを、プログラム開発装置 1 0 0 とは別の記憶装置が記憶するようシステムを構成してもよい。この場合、指定されたカスタム I / O 項目の設定値に指定された部品の情報は、プログラム開発装置 1 0 0 が、データベースサーバ 2 0 0 に要求して取得し、プログラム開発装置 1 0 0 自身の R A M に当該情報を記憶して、インクルードするパスを設定する。

20

【 0 1 5 0 】

なお、本発明は、前述した実施形態の機能を実現するソフトウェアのプログラムを、システム或いは装置に直接、或いは遠隔から供給するものを含む。そして、そのシステム或いは装置のコンピュータが前記供給されたプログラムコードを読み出して実行することによっても達成される場合も本発明に含まれる。

【 0 1 5 1 】

したがって、本発明の機能処理をコンピュータで実現するために、前記コンピュータにインストールされるプログラムコード自体も本発明を実現するものである。つまり、本発明は、本発明の機能処理を実現するためのコンピュータプログラム自体も含まれる。

30

【 0 1 5 2 】

プログラムを供給するための記録媒体としては、例えば、フレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、M O、C D - R O M、C D - R、C D - R W などがある。また、磁気テープ、不揮発性のメモリカード、R O M、D V D（D V D - R O M、D V D - R）などもある。

【 0 1 5 3 】

その他、プログラムの供給方法としては、クライアントコンピュータのブラウザを用いてインターネットのホームページに接続する。そして、前記ホームページから本発明のコンピュータプログラムそのもの、若しくは圧縮され自動インストール機能を含むファイルをハードディスク等の記録媒体にダウンロードすることによっても供給できる。

40

【 0 1 5 4 】

また、本発明のプログラムを構成するプログラムコードを複数のファイルに分割し、それぞれのファイルを異なるホームページからダウンロードすることによっても実現可能である。つまり、本発明の機能処理をコンピュータで実現するためのプログラムファイルを複数のユーザに対してダウンロードさせる W W W サーバも、本発明に含まれるものである。

【 0 1 5 5 】

また、本発明のプログラムを暗号化して C D - R O M 等の記憶媒体に格納してユーザに配布し、所定の条件をクリアしたユーザに対し、インターネットを介してホームページから暗号化を解く鍵情報をダウンロードさせる。そして、ダウンロードした鍵情報を使用することにより暗号化されたプログラムを実行してコンピュータにインストールさせて実現

50

することも可能である。

【 0 1 5 6 】

また、コンピュータが、読み出したプログラムを実行することによって、前述した実施形態の機能が実現される。その他、そのプログラムの指示に基づき、コンピュータ上で稼動しているOSなどが、実際の処理の一部又は全部を行い、その処理によっても前述した実施形態の機能が実現され得る。

【 0 1 5 7 】

さらに、記録媒体から読み出されたプログラムが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれる。その後、そのプログラムの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部又は全部を行い、その処理によっても前述した実施形態の機能が実現される。

10

【 0 1 5 8 】

尚、前述した実施形態は、本発明を実施するにあたっての具体化の例を示したものに過ぎず、これらによって本発明の技術的範囲が限定的に解釈されてはならないものである。即ち、本発明はその技術思想、又はその主要な特徴から逸脱することなく、様々な形で実施することができる。

【 符号の説明 】

【 0 1 5 9 】

- 1 0 0 プログラム開発装置
- 2 0 0 データベースサーバ
- 3 0 0 アプリケーションサーバ

20

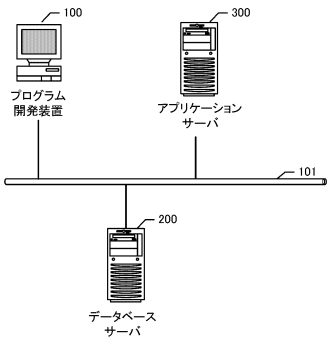
30

40

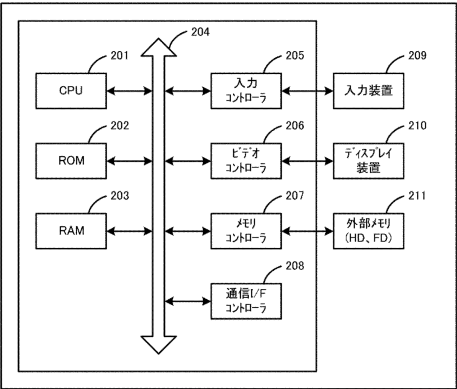
50

【図面】

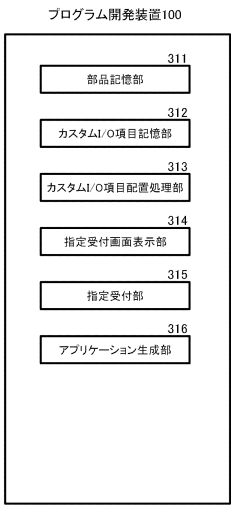
【図 1】



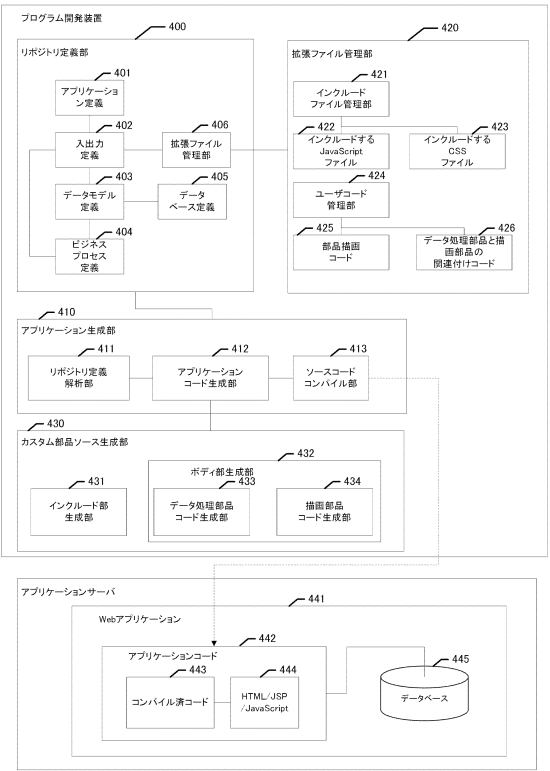
【図 2】



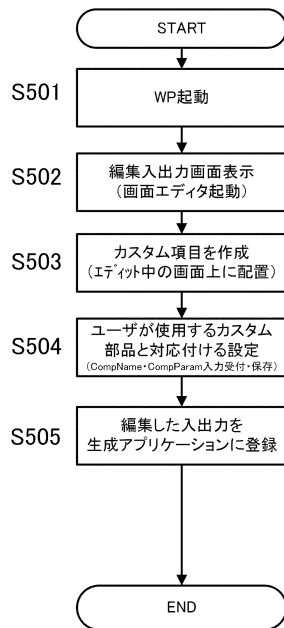
【図 3】



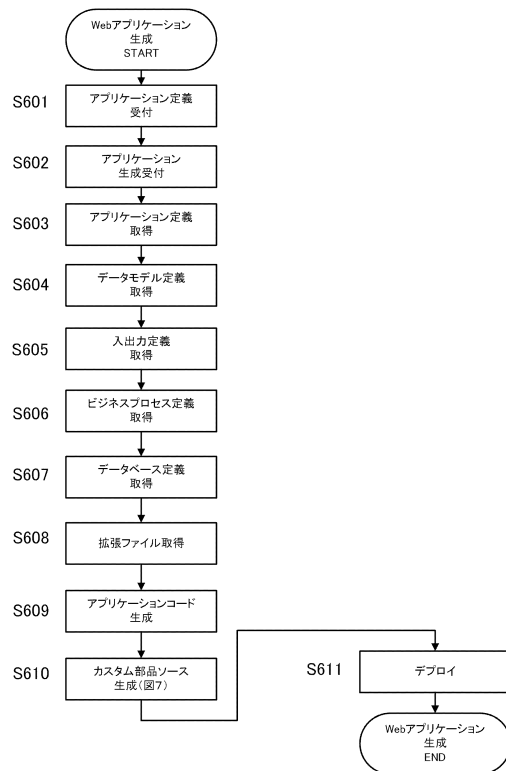
【図 4】



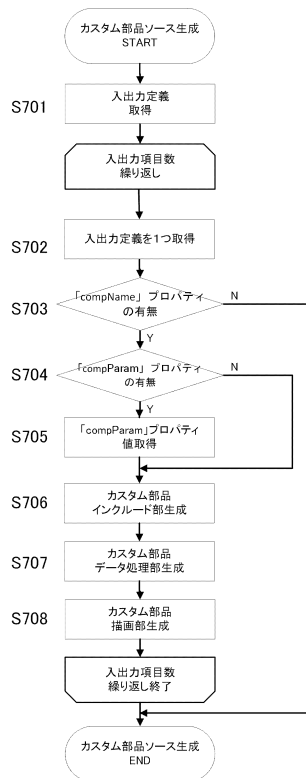
【 図 5 】



【 図 6 】



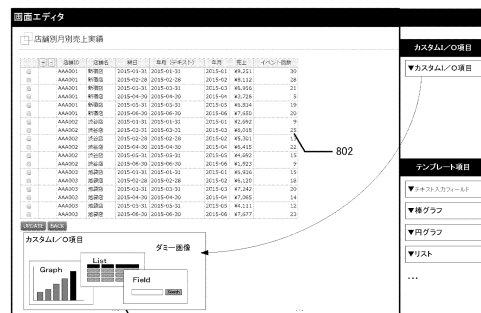
【圖 7】



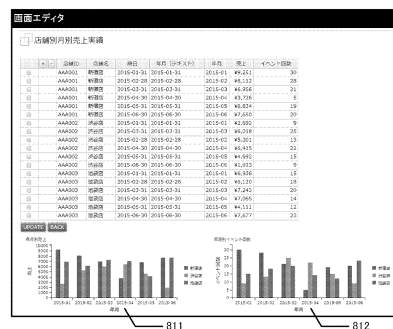
【圖 8】

■カスタム部品を含む画面イメージ

800



810



【 図 9 】

【 図 1 0 】

定常編集画面									
機能項目	画面項目	名称	表示	編集	印刷	検索	読み込み	データリンク	データリンクコメント
グループ	G	定常編集作業表	表示	1	0	0	0	0	0
IO上の表	CLOCKING	定常表	表示	2	0	0	0	0	STORE_BI_表名
IO上の表	STORE_NAME	定常表	表示	2	0	0	0	0	STORE_BI_表名
IO上の表	CLOCKING	定常表	表示	2	0	0	0	0	STORE_BI_表名
IO上の表	MONTHLY_AENT	定常表	表示	2	0	0	0	0	CLOCKING上の項目
IO上の表	PROCESSING	定常表	表示	2	0	0	0	0	STORE_BI_表名
IO上の表	EVENT	定常表	表示	2	0	0	0	0	STORE_BI_表名
IO上の表	BASE_VERTICAL	定常表	表示	1	0	0	0	0	STORE_BI_表名
IO上の表	BASE_VERTICAL	定常表	表示	1	0	0	0	0	STORE_BI_表名

B10 変換編集画面									
番号	変換コード	名前	属性	必須	レバ	短	データ型	デフォルト値	デフォルト値コメント
01	OFF	G	数値	1	0	-			
02	STORL	STORL	数値	2	0	-	STORE_BI_MAND	STORE_BI_MAND	
03	STORL	STORL	数値	2	0	-	STORE_BI_MAND	STORE_BI_MAND	
04	STORL	STORL	数値	2	0	-	STORE_BI_MAND	STORE_BI_MAND	
05	CLONING	CLONING	数値	2	0	-	STORE_BI_MAND	CLONING	CLONING
06	MONTHLY	MONTHLY	数値	2	0	-	STORE_BI_MAND	MONTHLY	MONTHLY
07	PRECISION	PRECISION	数値	2	0	-	STORE_BI_MAND	PRECISION	PRECISION
08	EVENT	EVENT	数値	2	0	-	STORE_BI_MAND	EVENT	EVENT
09	FACE	FACE	数値	1	0	-			
10	BACK	BACK	数値	1	0	-			
11	VERTICAL	VERTICAL	数値	1	0	-			
12	FACE	FACE	数値	1	0	-			
13	FACE	FACE	数値	1	0	-			
14	FACE	FACE	数値	1	0	-			
15	FACE	FACE	数値	1	0	-			
16	FACE	FACE	数値	1	0	-			
17	FACE	FACE	数値	1	0	-			
18	FACE	FACE	数値	1	0	-			
19	FACE	FACE	数値	1	0	-			
20	FACE	FACE	数値	1	0	-			
21	FACE	FACE	数値	1	0	-			
22	FACE	FACE	数値	1	0	-			
23	FACE	FACE	数値	1	0	-			
24	FACE	FACE	数値	1	0	-			
25	FACE	FACE	数値	1	0	-			
26	FACE	FACE	数値	1	0	-			
27	FACE	FACE	数値	1	0	-			
28	FACE	FACE	数値	1	0	-			
29	FACE	FACE	数値	1	0	-			
30	FACE	FACE	数値	1	0	-			
31	FACE	FACE	数値	1	0	-			
32	FACE	FACE	数値	1	0	-			
33	FACE	FACE	数値	1	0	-			
34	FACE	FACE	数値	1	0	-			
35	FACE	FACE	数値	1	0	-			
36	FACE	FACE	数値	1	0	-			
37	FACE	FACE	数値	1	0	-			
38	FACE	FACE	数値	1	0	-			
39	FACE	FACE	数値	1	0	-			
40	FACE	FACE	数値	1	0	-			
41	FACE	FACE	数値	1	0	-			
42	FACE	FACE	数値	1	0	-			
43	FACE	FACE	数値	1	0	-			
44	FACE	FACE	数値	1	0	-			
45	FACE	FACE	数値	1	0	-			
46	FACE	FACE	数値	1	0	-			
47	FACE	FACE	数値	1	0	-			
48	FACE	FACE	数値	1	0	-			
49	FACE	FACE	数値	1	0	-			
50	FACE	FACE	数値	1	0	-			
51	FACE	FACE	数値	1	0	-			
52	FACE	FACE	数値	1	0	-			
53	FACE	FACE	数値	1	0	-			
54	FACE	FACE	数値	1	0	-			
55	FACE	FACE	数値	1	0	-			
56	FACE	FACE	数値	1	0	-			
57	FACE	FACE	数値	1	0	-			
58	FACE	FACE	数値	1	0	-			
59	FACE	FACE	数値	1	0	-			
60	FACE	FACE	数値	1	0	-			
61	FACE	FACE	数値	1	0	-			
62	FACE	FACE	数値	1	0	-			

[illegible]

■インクルード部生成(1001)

JSPサンプル(一部抜粋)

```
<link rel="stylesheet" type="text/css" href="<%=request.getContextPath()%>/components/GRAPH/style/graph_style.css"
rel="stylesheet" />

<script src="<%=request.getContextPath()%>/components/GRAPH/scripts/vendor/graph.js" type="text/javascript"></script>
```

■ボディ部(データ処理部、描画部)生成

JSPサンプル(一部抜粋)

```

<div class="field field_bar_VERTICAL" o-field="title" id="sidBAR_VERTICAL">
  <!-- 描画コード -->
  <div id="sidBAR_VERTICAL" on_focus="setupEvent(event);event.onFocus='_bar_vertical'/"
  onblur="setupEvent(event);event.onBlur=updated_bar_vertical'/"event.onBlur=checkIfBAR_VERTICAL'; TEXT", 0, 0)"
  onkeypress="setupEvent(event);return event_onkeyPressId(BAR_VERTICAL', isTextKeyCode)"
  onkeyup="setupEvent(event);return event_onkeyPressId(BAR_VERTICAL', isTextKeyCode)"
  onkeydown="setupEvent(event);cursor_onkeyUp(BAR_VERTICAL', tabIndex=0)"
  <!--
  String path = "/WEB-INF/page/graph_userSnipnet.include";
  java.net.URL url = request.getSession().getServletContext().getResource(path);
  if (url != null) {
  <!--
  <!-- カスタム部品 <GRAPH> -->
  <!-- import url="/WEB-INF/page/graph_userSnipnet.include" charEncoding="UTF-8"/>
  <!--
  }
  <!--
  </div>
</div>

```

④ データ処理コード →

```

<!-- @html: hidden name="disp" property="_bar_vertical" styleId="sidBAR_VERTICAL" styleClass="TEXT"
write=true"/>
</div>

```

10

20

【 図 1 1 】

■描画とデータ処理の関連付けコード

カスタム部品名「GRAPH」の場合の例

```
function init_GRAPH(itemElem, itemValue, itemCode, compParam, isGroupItem, index)
{
    if(compParam["type"] == "PROCEED") {
        //compParamに"Type":"PROCEED"が設定されている場合
        //一覧の「年月」と「売上」列からのデータを取得し、グラフに描画
        ...
    }
    else if(compParam["type"] == "EVENT") {
        //compParamに"Type":"EVENT"が設定されている場合
        //一覧の「年月」と「イベント」列からのデータを取得し、グラフに描画
        ...
    }
    else {
        //compParamが上記以外の場合、一覧のどの列からデータを取得し、グラフに描画するかは記述
        ...
    }
}
```

30

40

50

フロントページの続き

キヤノンＩＴソリューションズ株式会社内
(72)発明者 飯沼 賢治
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
審査官 北川 純次
(56)参考文献 特開２０１７－１８２６７５（ＪＰ，Ａ）
特開２００４－１５７９２７（ＪＰ，Ａ）
特開２００７－０４２０８５（ＪＰ，Ａ）
特開２０１６－０２９５５８（ＪＰ，Ａ）
米国特許出願公開第２０１９／００１２１５０（ＵＳ，Ａ１）
(58)調査した分野 (Int.Cl.，ＤＢ名)
Ｇ０６Ｆ ８／３４