

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号  
特許第7233342号  
(P7233342)

(45)発行日 令和5年3月6日(2023.3.6)

(24)登録日 令和5年2月24日(2023.2.24)

(51)国際特許分類 F I  
G 0 6 F 8/60 (2018.01) G 0 6 F 8/60  
G 0 6 F 8/34 (2018.01) G 0 6 F 8/34

請求項の数 12 (全36頁)

|          |                             |          |   |
|----------|-----------------------------|----------|---|
| (21)出願番号 | 特願2019-146252(P2019-146252) | (73)特許権者 | 000005108<br>株式会社日立製作所<br>東京都千代田区丸の内一丁目6番6号 |
| (22)出願日  | 令和1年8月8日(2019.8.8)          | (74)代理人  | 110002365<br>弁理士法人サンネクスト国際特許事務所             |
| (65)公開番号 | 特開2021-26660(P2021-26660A)  | (72)発明者  | 井出 貴也<br>東京都千代田区丸の内一丁目6番6号<br>株式会社日立製作所内    |
| (43)公開日  | 令和3年2月22日(2021.2.22)        | (72)発明者  | 長沼 佑樹<br>東京都千代田区丸の内一丁目6番6号<br>株式会社日立製作所内    |
| 審査請求日    | 令和4年2月21日(2022.2.21)        | (72)発明者  | 畑崎 恵介<br>東京都千代田区丸の内一丁目6番6号<br>株式会社日立製作所内    |
|          |                             | 審査官      | 杉浦 孝光                                       |

最終頁に続く

(54)【発明の名称】 サービス連携支援システムおよびその方法

(57)【特許請求の範囲】

【請求項1】

ビジュアルプログラミングシステムが提供するフロー編集画面におけるフローが、前記ビジュアルプログラミングシステムの外部サービス呼び出すノードであるサービス利用ノードを含んでいて、且つ、前記サービス利用ノードに前記外部サービスを接続するユーザ要求を検出した場合、前記外部サービスの作成機能を有するサービス管理システムに前記外部サービスの作成要求を送信し、当該作成要求に回答して前記サービス管理システムにより作成された前記外部サービスに接続するための情報であり前記サービス管理システムにより作成された情報である接続情報を、前記サービス管理システムから取得する第1の機能と、

10

前記取得された接続情報を前記サービス利用ノードのプロパティに設定する第2の機能とを備えるサービス連携支援システム。

【請求項2】

前記第2の機能が、前記外部サービスの作成と削除のうちの少なくとも一つに関する条件を含んだポリシーの指定をユーザから受け付け、

前記第1の機能が、前記ユーザにより指定されたポリシーを表すポリシー情報に従う、請求項1に記載のサービス連携支援システム。

【請求項3】

前記第1の機能が、前記ユーザ要求を検出した場合、前記ポリシー情報に指定されている、前記外部サービスの作成する条件である作成条件が満たされたときに、前記外部サー

20

ピスの作成要求を前記サービス管理システムに送信する、  
請求項 2 に記載のサービス連携支援システム。

【請求項 4】

前記第 1 の機能が、前記ポリシー情報に指定されている、前記外部サービスの削除する条件である削除条件が満たされたときに、前記外部サービスの削除要求を前記サービス管理システムに送信する、

請求項 2 に記載のサービス連携支援システム。

【請求項 5】

前記第 2 の機能は、前記取得された接続情報を前記ビジュアルプログラミングシステムにより管理される情報である接続情報一覧に書き込むようになっており、

前記第 2 の機能は、前記サービス利用ノードが前記プロパティに設定された前記接続情報をを用いて前記外部サービスを呼び出すことができなくなったことを検出した場合、

前記外部サービスの接続情報を前記接続情報一覧から取得し、当該取得された接続情報をを用いて前記外部サービスに対するアクセスを行い、

当該アクセスに失敗の場合、前記外部サービスの削除する条件である削除条件が満たされたときに前記外部サービスの削除要求を前記サービス管理システムに送信し、その後、前記外部サービスの作成する条件である作成条件が満たされたときに前記外部サービスの作成要求を前記サービス管理システムに送信する、

請求項 1 に記載のサービス連携支援システム。

【請求項 6】

前記第 2 の機能は、前記取得された接続情報を前記ビジュアルプログラミングシステムにより管理される情報である接続情報一覧に書き込むようになっており、

前記第 2 の機能は、前記サービス利用ノードが前記プロパティに設定された前記接続情報をを用いて前記外部サービスを呼び出すことができなくなったことを検出した場合、前記外部サービスの接続情報を前記接続情報一覧から取得し、当該取得された接続情報をを用いて前記外部サービスに対するアクセスを行う、

請求項 1 に記載のサービス連携支援システム。

【請求項 7】

前記第 2 の機能は、前記外部サービスにアクセスが可能か監視し、

前記第 2 の機能は、前記外部サービスにアクセス不可の場合、前記サービス利用ノードの処理を待機または中断する、

請求項 1 に記載のサービス連携支援システム。

【請求項 8】

前記第 2 の機能は、前記フローが有する全てのサービス利用ノードの各々について、当該サービス利用ノードに接続される外部サービスにアクセスが可能か監視し、

前記フローにおける実行中ノードの後に実行される一つ以上のサービス利用ノードから呼び出される少なくとも一つの外部サービスにアクセス不可の場合、前記フローの実行が待機または中断する、

請求項 1 に記載のサービス連携支援システム。

【請求項 9】

前記第 1 の機能は、前記ビジュアルプログラミングシステムの外部に設けられており、

前記第 2 の機能は、前記ビジュアルプログラミングシステムの内部に設けられており、

前記第 2 の機能が、前記サービス利用ノードに対する前記外部サービスの接続がユーザにより要求された場合に前記ユーザ要求の発生を表すイベントの通知を前記第 1 の機能に送信し、

前記通知を前記第 1 の機能により受信することが、前記ユーザ要求を前記第 1 の機能により検出することである、

請求項 1 に記載のサービス連携支援システム。

【請求項 10】

前記フローを表すフロー情報が、前記サービス利用ノードの所定種類の ID を表す情報

10

20

30

40

50

を含んでいて、

前記作成要求には、前記サービス利用ノードの前記所定種類のIDが関連付けられており、

前記サービス管理システムにより作成され前記サービス管理システムにおいて管理される接続情報には、前記所定種類のIDが関連付けられ、

前記第1の機能により取得され前記ビジュアルプログラミングシステムにおいて管理される接続情報には、前記所定種類のIDが関連付けられる、

請求項1に記載のサービス連携支援システム。

【請求項11】

ビジュアルプログラミングシステムが提供するフロー編集画面におけるフローが、前記ビジュアルプログラミングシステムの外部サービス呼び出すノードであるサービス利用ノードを含んでいて、且つ、前記サービス利用ノードに前記外部サービスを接続するユーザ要求を検出した場合、前記外部サービスの作成機能を有するサービス管理システムに前記外部サービスの作成要求を送信し、

当該作成要求に回答して前記サービス管理システムにより作成された前記外部サービスに接続するための情報であり前記サービス管理システムにより作成された情報である接続情報を、前記サービス管理システムから取得し、

前記取得された接続情報を前記サービス利用ノードのプロパティに設定する、サービス連携支援方法。

【請求項12】

ビジュアルプログラミングシステムが提供するフロー編集画面におけるフローが、前記ビジュアルプログラミングシステムの外部サービス呼び出すノードであるサービス利用ノードを含んでいて、且つ、前記サービス利用ノードに前記外部サービスを接続するユーザ要求を検出した場合、前記外部サービスの作成機能を有するサービス管理システムに前記外部サービスの作成要求を送信し、

当該作成要求に回答して前記サービス管理システムにより作成された前記外部サービスに接続するための情報であり前記サービス管理システムにより作成された情報である接続情報を、前記サービス管理システムから取得し、

前記取得された接続情報を前記サービス利用ノードのプロパティに設定する、ことをコンピュータに実行させるコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、概して、アプリケーションの開発を支援する技術に関し、特に、ビジュアルプログラミングシステムを用いた開発を支援する技術に関する。

【背景技術】

【0002】

アプリケーションの機能性を高めるため、サービスとして提供される様々なソフトウェア、例えば、データベースサービスや音声認識サービス等を用いて、アプリケーションを構築することがある。これらサービスはアプリケーションを構成するプログラムから、API (Application Programming Interface) を用いて呼び出され、その機能が利用される。これにより、開発者は様々なサービスが持つ機能を実装なしに利用できる。

【0003】

一方、市場価値にあったアプリケーションを作るため、小さな単位でプログラミングを行い、試行錯誤しながらアプリケーションを開発することが行われる。このとき、アプリケーションのプログラムの開発に、ビジュアルプログラミングシステムが用いられることがある。ビジュアルプログラミングとは、アイコンにより視覚化された処理の部品を配置及び接続することによりプログラムを構築する方法である。ビジュアルプログラミングシステムを用いることにより、開発者は直感的かつ少ない作業で迅速にプログラムを開発できる。

10

20

30

40

50

## 【 0 0 0 4 】

そこで、ビジュアルプログラミングシステムを用いて様々なサービスを呼び出すアプリケーションを開発すれば、様々なサービスの機能を活用したアプリケーションを高速に試行錯誤しながらプログラミングでき、結果として、より市場価値の高いアプリケーションを開発できる可能性がある。

## 【 0 0 0 5 】

しかし、ビジュアルプログラミングシステムからサービスの機能を使うには、開発者が前もってサービスを作成（プロビジョニングのように利用可能な状態にすることも含む）し、サービスの接続情報（例えば、エンドポイントを表す情報と認証情報とを含んだ情報）を取得し、その接続情報をビジュアルプログラミングシステムに設定として登録する事前作業が必要になる。利用するサービスの数が多くなるほど、この作業負荷は増大し、高速なアプリケーション開発の試行錯誤を妨げてしまう問題がある。

10

## 【 0 0 0 6 】

このため、ビジュアルプログラミングシステムにてサービスを利用するには、開発者によるビジュアルプログラミングシステムの操作に応じて動的にサービスが作成され、接続情報の設定なしにビジュアルプログラミングシステムからアクセス可能になることが望ましい。この実現に利用可能な技術として、特許文献 1 が挙げられる。なお、ビジュアルプログラミングシステムは、「モデル開発環境」と呼ばれてもよい。ソフトウェアの構成要素や処理単位がノードであり、ノード同士の結線は「エッジ」と呼ばれてもよい。ノードとエッジは有向グラフにおける用語であり、フローダイアグラムも有向グラフとして捉えることができる。本明細書において、フローダイアグラムとブロックダイアグラムと有向グラフは同義とし、フローダイアグラムは単に「フロー」と呼ぶことができ、「フロー」と「グラフ」は同義とできる。フローは、典型的には、DAG（Directed Acyclic Graph）である。

20

## 【 先行技術文献 】

## 【 特許文献 】

## 【 0 0 0 7 】

【 文献 】 US 2 0 1 8 / 0 0 6 9 8 0 4 A 1

## 【 発明の概要 】

## 【 発明が解決しようとする課題 】

30

## 【 0 0 0 8 】

ビジュアルプログラミングシステムから動的にサービスの作成および接続を行う場合、特許文献 1 には例えば以下の課題 1 および課題 2 がある。

## 【 0 0 0 9 】

（課題 1）ビジュアルプログラミングシステムに相当するユーザデバイスが、サービスの作成要求（create service request）をオンデマンドサービスブローカーに出す方法が開示されていない。高速に試行錯誤しながらアプリケーションの開発を行う場合、プログラム変更のたびに必要となるサービスの作成や削除を開発者が行っている場合は、高速な試行錯誤が阻害される。プログラムの変更や実行を検知して動的にサービスを作成できることが望ましいが、特許文献 1 ではできない。

40

## 【 0 0 1 0 】

（課題 2）サービスの接続情報とノード（アイコンにより視覚化された処理の部品。ビジュアルプログラミングシステムではノードを繋ぎ合わせるによりフローを構築する）の対応関係が管理されていない。また、接続情報を動的にノードに反映することができない。ビジュアルプログラミングシステムでは無数のノードが様々なサービスに接続しており、更に複数ノードが同一のサービスに接続する場合もあるため、このサービスとノードの接続関係をユーザが管理し、サービス作成のたびに設定を手動登録することは困難である。

## 【 課題を解決するための手段 】

## 【 0 0 1 1 】

50

ビジュアルプログラミングシステムが提供するフロー編集画面におけるフローが、ビジュアルプログラミングシステムの外部サービス呼び出すノードであるサービス利用ノードを含んでいて、且つ、サービス利用ノードに対する外部サービスの接続がユーザにより要求されたことであるユーザ要求を検出した場合、サービス連携支援システムが、外部サービスの作成機能を有するサービス管理システムに外部サービスの作成要求を送信し、当該作成要求にตอบสนองしてサービス管理システムにより作成された外部サービスに接続するための接続情報（サービス管理システムにより作成された情報）を、サービス管理システムから取得する。サービス連携支援システムは、当該取得された接続情報をサービス利用ノードのプロパティに設定する。

【発明の効果】

10

【0012】

本発明の代表的な実施形態によれば、ビジュアルプログラミングシステムのフローの編集において、動的に外部サービスが作成されると共に、当該作成された外部サービスの接続情報が、当該外部サービス呼び出すサービス利用ノードのプロパティに自動で設定される。このため、ユーザ（例えば、開発者）は事前に外部サービスを作成したりその外部サービスの接続情報を手動で設定したりする必要がなくなる。結果として、アプリケーションの開発を効率化できる。具体的には、外部サービス呼び出すサービス利用ノードを含んだフローを用いた試行錯誤を高速に行うことができる。構成および効果の一例は、以下の実施形態の説明により明らかにされる。

【図面の簡単な説明】

20

【0013】

【図1】本発明の第1実施形態におけるシステム構成の一例を示すブロック図である。

【図2A】ビジュアルプログラミングシステムのGUI画面の一例を示す図である。

【図2B】プロパティ設定の操作の一例を示す図である。

【図3】計算機のハードウェア構成の一例を示すブロック図である。

【図4】フロー情報の一例を示す図である。

【図5】サービス連携システム内の各種情報のデータ構造の一例を示す図である。

【図6】サービス定義情報のうちのCustomPolicyおよびMonitoringのデータ構造の一例を示す図である。

【図7】接続情報一覧のデータ構造の一例を示す図である。

30

【図8】イベント通知処理を示す図である。

【図9】作成処理を示す図である。

【図10】削除処理を示す図である。

【図11】ノード別の一時停止処理を示す図である。

【図12】フロー全体の一時停止処理を示す図である。

【図13】サービス再起動処理のフローチャートである。

【図14】本発明の第2実施形態におけるシステム構成の一例を示すブロック図である。

【発明を実施するための形態】

【0014】

以下、図面を用いて、本発明の一実施形態を詳述する。

40

【0015】

ただし、本発明は後述する実施形態に限定されるものではなく、添付した特許請求の範囲の趣旨内における様々な変形例および同等の構成が含まれる。例えば、前述した実施形態は本発明を分かりやすく説明するために詳細に説明したものであり、必ずしも説明した全ての構成を備えるものに本発明は限定されない。

【0016】

なお、以下の説明では、「インターフェース装置」は、一つ以上のインターフェースデバイスでよい。当該一つ以上のインターフェースデバイスは、下記のうちの少なくとも一つでよい。

・一つ以上のI/O（Input/Output）インターフェースデバイス。I/O（Input/Output）

50

ut) インターフェースデバイスは、I/Oデバイスと遠隔の表示用計算機とのうちの少なくとも一つに対するインターフェースデバイスである。表示用計算機に対するI/Oインターフェースデバイスは、通信インターフェースデバイスでよい。少なくとも一つのI/Oデバイスは、ユーザインターフェースデバイス、例えば、キーボードおよびポインティングデバイスのような入力デバイスと、表示デバイスのような出力デバイスとのうちのいずれでもよい。

・一つ以上の通信インターフェースデバイス。一つ以上の通信インターフェースデバイスは、一つ以上の同種の通信インターフェースデバイス(例えば一つ以上のNIC(Network Interface Card))であってもよいし二つ以上の異種の通信インターフェースデバイス(例えばNICとHBA(Host Bus Adapter))であってもよい。

10

【0017】

また、以下の説明では、「メモリ」は、一つ以上のメモリデバイスであり、典型的には主記憶デバイスでよい。メモリにおける少なくとも一つのメモリデバイスは、揮発性メモリデバイスであってもよいし不揮発性メモリデバイスであってもよい。

【0018】

また、以下の説明では、「永続記憶装置」は、一つ以上の永続記憶デバイスである。永続記憶デバイスは、典型的には、不揮発性の記憶デバイス(例えば補助記憶デバイス)であり、具体的には、例えば、HDD(Hard Disk Drive)またはSSD(Solid State Drive)である。

【0019】

20

また、以下の説明では、「記憶装置」は、メモリと永続記憶装置の少なくともメモリでよい。

【0020】

また、以下の説明では、「プロセッサ」は、一つ以上のプロセッサデバイスである。少なくとも一つのプロセッサデバイスは、典型的には、CPU(Central Processing Unit)のようなマイクロプロセッサデバイスであるが、GPU(Graphics Processing Unit)のような他種のプロセッサデバイスでもよい。少なくとも一つのプロセッサデバイスは、シングルコアでもよいしマルチコアでもよい。少なくとも一つのプロセッサデバイスは、プロセッサコアでもよい。少なくとも一つのプロセッサデバイスは、処理の一部または全部を行うハードウェア回路(例えばFPGA(Field-Programmable Gate Array)またはASIC(Application Specific Integrated Circuit))といった広義のプロセッサデバイスでもよい。

30

【0021】

また、以下の説明では、各情報を「テーブル」または「JSON(JavaScript Object Notation)フォーマットのテキストデータ」形式にて説明するが(JavaScriptは登録商標)、これら情報は必ずしもテーブルによるデータ構造で表現されていなくてもよく、リスト、DB、キュー等のデータ構造や、YAML、XML等フォーマットのテキストデータや、またそれ以外で表現されていてもよい。そのため、データ構造に依存しないことを示すために「テーブル」、「リスト」、「DB」、「キュー」等について単に「情報」と呼ぶことがある。また、各情報の内容を説明する際に、「識別情報」、「識別子」、「ID(Identification)」という表現を用いることが可能であり、これらについてはお互いに置換が可能である。

40

【0022】

また、以下の説明では、GUI(Graphical User Interface)上のボタンの押下を起点に実行される処理は、対応するAPIの呼び出しを起点に実行されてもよい。

【0023】

また、以下の説明では、各機能を実現するプログラム、テーブル、ファイル等の情報は、メモリ、ハードディスク、SSD(Solid State Drive)等の記憶装置、又は、ICカード、SDカード、DVD等の記録媒体に格納することができる。

【0024】

50

また、以下の説明では、前述した各構成、機能、処理手段等は、それらの一部又は全部を、例えば集積回路で設計する等により、ハードウェアで実現してもよく、プロセッサがそれぞれの機能を実現するプログラムを解釈し実行することにより、ソフトウェアで実現してもよい。

【0025】

また、以下の説明では、「xxx機能」の表現にて機能を説明することがあるが、機能は、一つ以上のコンピュータプログラムがプロセッサによって実行されることで実現されてもよいし、一つ以上のハードウェア回路（例えばFPGAまたはASIC）によって実現されてもよい。プログラムがプロセッサによって実行されることで機能が実現される場合、定められた処理が、適宜に記憶装置および/またはインターフェース装置等を用いながら行われるため、機能はプロセッサの少なくとも一部とされてもよい。機能を主語として説明された処理は、プロセッサあるいはそのプロセッサを有する装置が行う処理としてもよい。プログラムは、プログラムソースからインストールされてもよい。プログラムソースは、例えば、プログラム配布計算機または計算機が読み取り可能な記録媒体（例えば非一時的な記録媒体）であってもよい。各機能の説明は一例であり、複数の機能が一つの機能にまとめられたり、一つの機能が複数の機能に分割されたりしてもよい。

10

【0026】

また、以下の説明では、異なる計算機間にてデータを取得したりプログラムの機能呼び出ししたりする際は、実際にはWeb API等の通信プロトコルを用いたりリモートプロシージャコールを行っている場合がある。

20

【0027】

また、以下の説明では、同種の要素を区別しないで説明する場合には、参照符号のうちの共通部分を使用し、同種の要素を区別する場合は、参照符号を使用することがある。例えば、ビジュアルプログラミングシステムを区別しない場合には、「ビジュアルプログラミングシステム10」と言い、ビジュアルプログラミングシステムを区別する場合には、「ビジュアルプログラミングシステム10A」、「ビジュアルプログラミングシステム10B」のように言うことがある。

[第1実施形態]

【0028】

<システムの説明>

30

【0029】

図1は、本発明の第1実施形態におけるシステム構成の一例を示すブロック図である。

【0030】

ネットワーク80に、ビジュアルプログラミングシステム10と、サービス連携システム50と、サービス管理システム60とが接続される。

【0031】

本実施形態では、ビジュアルプログラミングシステム10のユーザ40が設定したポリシーを表すポリシー情報に従い、ビジュアルプログラミングシステム10での特定のイベントに応じて外部サービス（以下、サービス）70が作成される。作成されたサービス70が、ビジュアルプログラミングシステム10から接続可能になる。同様に、ビジュアルプログラミングシステム10での特定のイベントに応じて、サービス70の削除および再作成が可能である。更に、サービス70が利用可能か監視され、利用不可能なときは、当該サービス70を呼び出すフローの実行が待機または中断される。

40

【0032】

ビジュアルプログラミングシステム10とサービス管理システム60のいずれも、サービス連携システム50一つに対し複数存在してよい。また、サービス70は、サービス管理システム60一つに対し複数存在してよい。また、1つのビジュアルプログラミングシステム10を複数のユーザが利用してもよいし、一人のユーザが複数のビジュアルプログラミングシステム10を利用してもよい。また、各コンポーネントの接続形態はネットワークに限らず、例えば、個々のビジュアルプログラミングシステム10の内部にサービス

50

連携システム 50 が設けられてもよく、また、サービス連携システム 50 の内部にサービス管理システム 60 が設けられてもよい。

【0033】

ビジュアルプログラミングシステム 10 は、フローを実装および実行する機能を有する。フローはビジュアルプログラミングシステム 10 を用いて実装されたプログラムである。ビジュアルプログラミングシステム 10 はフローを実装したり、実行したりするために、GUI を用いることができ、フローを編集する GUI 画面を出力デバイス 42 に出力することができる。ビジュアルプログラミングシステム 10 の利用イメージの一例は図 2 にて詳述する。また、ビジュアルプログラミングシステム 10 内のコンポーネントの説明は図 2 説明の後に詳述する。

10

【0034】

ユーザ 40 は、ビジュアルプログラミングシステム 10 を用いてプログラミングを行うことができる。プログラミングのために、ユーザ 40 は、液晶ディスプレイなどの出力デバイス 42 を用いビジュアルプログラミングシステム 10 の GUI 画面を確認し、キーボードおよびポインティングデバイスなどの入力デバイス 41 を用いて、その GUI 画面を操作できる。ユーザ 40 の計算機（図示せず）が、入力デバイス 41 と出力デバイス 42 を有してよく、ビジュアルプログラミングシステム 10 と通信してよい。

【0035】

サービス連携システム 50 は、ビジュアルプログラミングシステム 10 から通知されるイベントに応じてサービス 70 を作成または削除する機能と、ビジュアルプログラミングシステム 10 にサービス 70 の接続情報を通知する機能を有する。

20

【0036】

具体的には、サービス連携システム 50 は、イベント検知機能 51 と、サービス作成機能 52 と、サービス削除機能 53 と、情報取得機能 54 とを有する。また、サービス連携システム 50 は、ポリシー情報 55 と、サービス定義情報 56 と、管理システム情報 57 とを管理する。

【0037】

イベント検知機能 51 は、ビジュアルプログラミングシステム 10 から送付されるイベントを基に、サービス作成機能 52 またはサービス削除機能 53 にサービス 70 の作成または削除を依頼したり、サービス 70 の作成または削除の後に当該サービス 70 の接続情報をビジュアルプログラミングシステム 10 に送付したりする。イベント検知機能 51 の具体的な処理は図 8 にて詳述する。また、イベントの種類およびデータ形式は後述する。

30

【0038】

ここで、「接続情報」とは、サービス 70 の API をネットワーク 80 を介して操作するために必要な情報であり、例えば、サービス 70 にアクセスするためのエンドポイントを表す情報とサービス 70 を操作するための認証情報とを含んだ情報である。エンドポイントは、例えば、サービス 70 の URL (Uniform Resource Locator) や IP アドレス (Internet Protocol) である。認証情報は、例えば、サービス 70 の認証を通過するためのユーザアカウント名とパスワードの組、もしくはアクセストークンや証明書などの文字列を表す。

40

【0039】

サービス作成機能 52 は、サービス管理システム 60 にサービス 70 の作成を依頼する。サービス削除機能 53 は、サービス管理システム 60 にサービス 70 の削除を依頼する。サービス作成機能 52 の処理は図 9 にて詳述する。サービス削除機能 53 の処理は図 10 にて詳述する。

【0040】

情報取得機能 54 は、サービス管理システム 60 からサービス 70 の接続情報を取得する。情報取得機能 54 は、ビジュアルプログラミングシステム 10 の接続管理機能 34 により使用される。

【0041】

50

ポリシー情報 55 は、ユーザ 40 が設定した各サービス 70 の利用ポリシーを表す情報である。ポリシー情報 55 のデータ構造は、図 5 にて詳述する。

【0042】

サービス定義情報 56 は、サービス 70 の利用方法をビジュアルプログラミングシステム 10 やユーザ 40 に提示するための情報である。サービス定義情報 56 は、例えば、サービス 70 を提供するプロバイダにより作成される。サービス定義情報 56 のデータ構造は、図 5 にて詳述する。

【0043】

管理システム情報 57 は、サービス連携システム 50 がサービス管理システム 60 を利用するための情報である。管理システム情報 57 のデータ構造は、図 5 にて詳述する。

10

【0044】

サービス管理システム 60 は、サービス連携システム 50 からの要求を基にサービス 70 を作成または削除する機能、および、サービス 70 への接続情報を保持する機能を有する。サービス管理システム 60 は、例えば Open Service Broker の仕様 に 則 っ た インターフェイスを持つサービス作成システム (ServiceBroker) でよい。Open Service Broker とは、コンポーネント化されたサービスを作成し、サービスを利用するための設定値をやりとりするための命令のインターフェイスを標準化した仕様である。サービス管理システム 60 は、サービス 70 の種類ごとに存在してもよいし、サービスを提供する企業ごとに存在してもよい。図 1 の例によれば、サービス管理システムにより作成された 60 X および 60 Y がある。

20

【0045】

ここで、「サービス 70 の作成」とは、ネットワーク 80 を介して人やプログラムがサービス 70 を操作し、その機能を利用できる状態にすることを意味する。本実施形態では、サービス 70 の作成方法は大きく 2 種類ある。一つ目は、サービス 70 のアカウントを発行した後、そのアカウントのために新規にインスタンスを起動し、インスタンス上でサービス 70 の実態となるプログラムを稼働させた後、発行したアカウントと稼働させたインスタンスとを紐付ける方法である。このインスタンスは、専用のハードウェアで構築あってもよいし、計算機上や計算機上に仮想的技術やコンテナ技術によって構築された仮想的な計算機であってもよい。二つ目は、サービス 70 に新規アカウントを追加した後、そのアカウントを稼働済みのサービス 70 に登録する方法である。

30

【0046】

また、「サービス 70 の削除」とは、例えば、下記の少なくとも一つを意味する。

- ・サービス 70 から対応するアカウント情報を削除した後、サービス 70 が個別にインスタンスを作成する場合、インスタンスを停止 (物理計算機の場合) または削除 (仮想的な計算機の場合) すること。
- ・サービス 70 が一つのインスタンスで複数のアカウントに対応する場合、当該アカウントの使用していたデータや設定情報を削除すること。

【0047】

サービス管理システム 60 は、サービス作成機能 61 と、サービス削除機能 62 とを有する。また、サービス管理システム 60 は、接続情報一覧 63 を管理する。

40

【0048】

サービス作成機能 61 は、前述の方法を用いてサービス 70 を作成する。サービス削除機能 62 は、前述の方法を用いてサービス 70 を削除する。

【0049】

接続情報一覧 63 は、作成されたサービス 70 の接続情報 (例えば、エンドポイント情報および認証情報を含んだ情報) を保持する。接続情報一覧 63 が保持する情報は、Web API 等を通じて、サービス連携システム 50 の情報取得機能 54 が取得できる。接続情報一覧 63 のデータ構成は、図 7 にて詳述する。ただし、接続情報一覧 63 は、図 7 に示す Status フィールド 704 は保持しなくてもよい (一方で、接続情報一覧 35 は、Status フィールド 704 を保持する)。

50

## 【 0 0 5 0 】

サービス70は、一つ以上の計算機で実行されるプログラム（インスタンス）である。サービス70は、ネットワーク80を介してプログラムの機能を人や他のプログラムに提供する機能を有する。サービス70には、例えば、データベースサービスや音声認識サービスがある。人や他のプログラムは、例えばWeb APIを用いてサービス70の機能を利用する。サービス70は、ネットワーク80を介して当該サービス70にアクセスするためのURLやIPアドレス等のエンドポイント情報と、アクセス元がサービス70を利用可能か判定するための認証情報とを持つ。エンドポイント情報は、同じサービス70であったとしても、ユーザごとやインスタンスごとにより異なる場合がある。例えば、図1では、サービス70Xについて、エンドポイントが異なるサービスX1とサービスX2とがある。

10

## 【 0 0 5 1 】

< GUIの説明 >

## 【 0 0 5 2 】

図2Aは、ビジュアルプログラミングシステム10のGUI画面の一例を示す図である。図2を用いて、ユーザ40がビジュアルプログラミングシステム10を利用する際の操作の一例を説明する。

## 【 0 0 5 3 】

GUI画面200は、ビジュアルプログラミングシステム10のフロー編集機能11が生成したGUI画面の一例である。本GUI画面200は、出力デバイス42に表示される。ユーザ40は、本GUI画面200を用いて、フローの作成、フローの実行、および、サービス70の利用のためのポリシーの設定を行う。GUI画面200が、フロー編集画面の一例である。

20

## 【 0 0 5 4 】

GUI画面200は、実行ボタン201、ユーザ情報UI202、ノードパレット210、編集UI220、プロパティ設定UI230、および、ノード説明UI234を有する。

## 【 0 0 5 5 】

ノードパレット210には、編集UI220に配置可能な（フロー221の要素になり得る）一つ以上のノード211が表示される。ノード211は、アイコン等により視覚化された処理部品（例えば、ソフトウェアの部品となり得るプログラムモジュール）のうちインスタンス化されていない処理部品を表す。ノード211の定義情報は、ノード定義情報20に格納される。

30

## 【 0 0 5 6 】

ノード211は、外部からの入力を受け付けるための入力ノードと、データを別形式のデータに変換する処理ノードと、外部に出力するための出力ノードの3種に大別できる。入力ノードには、例えば、ファイルからデータを入力するファイル読込ノードや、HTTP（Hyper Text Transfer Protocol）リクエストを受けるHTTP受付ノード、ユーザ40により設定された時間間隔で定期的にフローを実行する定期実行ノードがある。処理ノードには、例えば、サービスXの機能をAPI経由で呼び出すサービスX利用ノードや、JSON等の形式で入力された値をXML形式に変換するXMLノードや、任意のJavaScript関数を実行するファンクションノード等がある。出力ノードには、例えば、データをファイルに保存する結果保存ノードや、HTTPレスポンスを返すHTTP出力ノードがある。

40

## 【 0 0 5 7 】

ユーザ40は、ノードパレット210からノード211を編集UI220に配置する。これにより、ノード211が表す処理部品がインスタンス化される。編集UI220におけるノード222は、インスタンス化された処理部品を表す。ノード222間を線223で繋ぎ合わせることでフロー221を作成できる。フロー221は、複数のノード222と、ノード222間を結ぶ線223で表すことができる。フロー221に配置されたノー

50

ド 2 2 2 が表す処理部品には、一意の node\_id ( node\_id 4 0 1 ( 図 4 ) が表すノード ID ) が付与される。結果として、処理部品がインスタンス化される。このため、ユーザ 4 0 は、フロー 2 2 1 上に、同じノード定義 2 1 ( 図 1 参照 ) を持つ処理部品を表すノード 2 2 2 を複数配置できる。

#### 【 0 0 5 8 】

ノード 2 2 2 同士の接続は、あるノード 2 2 2 の出力が別のノード 2 2 2 の入力となる接続である。ノード 2 2 2 の左端が入力を表し、右端が出力を表す。例えば、フロー 2 2 1 において、定期実行ノード 2 2 2 - 1 の出力が、サービス X 利用ノード 2 2 2 - 2 の入力となり、サービス X 利用ノード 2 2 2 - 2 の出力は、もう一つのサービス X 利用ノード 2 2 2 - 3 の入力となる。

10

#### 【 0 0 5 9 】

また、フロー 2 2 1 では、ノード 2 2 2 間の接続関係に基づいてフロー実行機能 1 2 による処理の実行順序が決定される。例えば、定期実行ノード 2 2 2 - 1 の処理の実行の次にサービス X 利用ノード 2 2 2 - 2 の処理が実行される。例えば、サービス X 利用ノード 2 2 2 - 2 または 2 2 2 - 3 ( サービス利用ノードの一例 ) には、当該ノードのステータス 2 2 4 を表示できる。各サービス利用ノードについて、ステータス表示の有無、内容および変更条件は、後述するノード定義 2 1 の概要定義 2 2 に記述してある。概要定義 2 2 をフロー編集機能 1 1 が解釈してサービス利用ノード 2 2 2 のステータスを表示する。

#### 【 0 0 6 0 】

ユーザ 4 0 は、編集 UI 2 2 0 中のノード 2 2 2 を、入力デバイス 4 1 を用いて選択すると、フロー編集機能 1 1 が、当該ノード 2 2 2 に対応した情報に、プロパティ UI 2 3 0 に表示される情報と、ノード説明 UI 2 3 4 に表示される情報とを更新する。プロパティ設定 UI 2 3 0 を介して、ユーザ 4 0 は、選択したノード 2 2 2 のプロパティを設定できる。プロパティ設定 UI 2 3 0 は、UI 2 3 1 ~ 2 3 3 を有する。なお、「プロパティ」とは、ノード 2 2 2 の一種の属性であり、ノード 2 2 2 のプロパティに、一つ以上のパラメータが設定される。ノード 2 2 2 が、例えば定期実行ノード 2 2 2 - 1 の場合、プロパティの値として設定され得るパラメータ値として、実行間隔の値がある。プロパティの値は、ノード定義 2 1 ごとに異なる。プロパティ設定 UI 2 3 0 が有する UI としては、テキストボックス、リストボックス、ボタンなど、一般的な Web アプリケーションで使用される GUI 部品 ( コンポーネントの一例 ) でよく、そのような UI を用いてノード 2 2 2 のプロパティの設定が行われる。

20

30

#### 【 0 0 6 1 】

また、編集 UI 2 2 0 上に配置されたノード 2 2 2 の概要、利用方法およびプロパティの説明が、ノード説明 UI 2 3 4 に表示される。

#### 【 0 0 6 2 】

ユーザ 4 0 が実行ボタン 2 0 1 を押下すると、フロー実行機能 1 2 にてフロー 2 2 1 が実行される。フロー 2 2 1 の実行処理は図 2 の説明後の図 1 の説明にて詳述する。

#### 【 0 0 6 3 】

ユーザ情報 UI 2 0 2 には、ビジュアルプログラミングシステム 1 0 にログイン中のユーザ 4 0 のプロフィール情報 ( 例えば、アカウント名、所属プロジェクト名 ) とユーザ 4 0 を表すアイコンとが表示される。

40

#### 【 0 0 6 4 】

ノード 2 2 2 の中には、前述のサービス X 利用ノード 2 2 2 - 2 および 2 2 2 - 3 のように、ビジュアルプログラミングシステム 1 0 の外部にあるサービス 7 0 を呼び出しサービス 7 0 の機能を利用するノード 2 2 2 が存在する。このようなノードを、「サービス利用ノード」と呼称する。一つのサービス利用ノード 2 2 2 が、複数のサービス 7 0 を操作する場合もあれば、一つのサービス 7 0 を操作する場合もある。また、データ格納ノードとデータ取得ノードのように、一つのサービス 7 0 に対して複数種類のノード 2 2 2 が存在する場合もある ( 例えば、サービス Y 学習実行ノード、サービス Y 推論実行ノード ) 。このように、サービス利用ノード 2 2 2 とサービス 7 0 との対応関係は、1 : 1 でもよいし

50

、 1 : 多でもよいし、多 : 1でもよいし、多 : 多でもよい。

【 0 0 6 5 】

サービス利用ノード 2 2 2 のプロパティは、通常ノード（サービス利用ノード以外のノード）のプロパティが持つパラメータ項目以外に、サービス 7 0 の接続情報やサービス 7 0 の作成または削除のポリシーが設定される接続設定プロパティを含む。接続設定プロパティの設定のため、サービス利用ノードのプロパティ設定 UI 2 3 0 には、当該サービス利用ノードの接続先を選択するためのリストボックス 2 3 1 と、接続設定を編集することを指定するためのボタン 2 3 2 とがある。一つのサービス利用ノードが、複数の接続設定プロパティを保有してもよい。

【 0 0 6 6 】

サービス利用ノードの接続設定プロパティのユーザ 4 0 による入力を補助するため、ノード設定 UI 2 3 4 には、ノードが使用するサービス 7 0 の概要や課金形式などのサービス仕様を表す情報が表示されてよい。

【 0 0 6 7 】

また、ノード 2 1 1 として、稼働確認ノード 2 1 1 が用意されてもよい。稼働確認ノード 2 1 1 は、当該稼働確認ノード 2 1 1 の実行中において、当該ノード 2 1 1 より後にフロー 2 2 1 の実行において呼び出される可能性のあるサービス 7 0 が全てアクセス可能か確認し、アクセスできないサービス 7 0 が一つでもあれば当該フロー 2 2 1 の実行を待機または中断するノードである。稼働確認ノードは、入力ノードや処理ノードとして構築されてよい。稼働確認ノードの処理は、図 1 2 にて詳述する。

【 0 0 6 8 】

図 2 B は、サービス利用ノード 2 2 2 についてのプロパティ設定 UI 2 3 0 の幾つかの例を示す。

【 0 0 6 9 】

プロパティ設定 UI 2 3 0 - 1 は、例えば初めのプロパティ設定 UI 2 3 0 である。ユーザ 4 0 が、リストボックス 2 3 1 を入力デバイス 4 1 を用いて選択すると、編集 UI 2 2 0 で選択されたサービス利用ノードに接続可能なサービス 7 0 の一覧である接続設定一覧 2 3 5 が、フロー編集機能 1 1 により表示される（プロパティ選択 UI 2 3 0 - 2 ）。

【 0 0 7 0 】

接続設定一覧 2 3 5 において、選択可能な接続設定は、「新規追加」以外の文字列、例えば、「X1-9dbc4b3b」や「X1-c8dc722j」といった文字列により表される。接続設定を表す文字列は、ユーザ 4 0（またはユーザ 4 0 が所属するプロジェクト）が既に作成しているサービス 7 0 の接続情報を表す ID（例えば、後述の ServiceId と ResourceId との組合せ）か、もしくはユーザ 4 0 が後述の接続名 2 4 2 にて登録した任意の文字列である。「ユーザ 4 0（またはユーザ 4 0 が所属するプロジェクト）が既に作成しているサービス 7 0」は、例えば、接続情報一覧 3 5 のうち、ユーザ 4 0 の ID（または、ユーザ 4 0 の ID とプロジェクトの ID）に適合する後述の Credential 7 0 3 に対応するサービス 7 0 でよい。「X1-9dbc4b3b」または「X1-c8dc722j」といった接続設定を選択することは、例えば、エンドポイントを変えて作成済のサービスを利用することを意味する。ここで、「接続設定」とは、サービス 7 0 の接続情報とポリシー情報との両方の設定を意味する。一方、接続設定一覧 2 3 5 の「新規追加」という項目は、ユーザ 4 0 が新規にサービス 7 0 を作成したい場合に選択する。

【 0 0 7 1 】

ユーザ 4 0 が接続設定一覧 2 3 5 から既存の接続設定、もしくは新規追加を選択後に編集ボタン 2 3 2 を押下すると、フロー編集機能 1 1 により、プロパティ設定 UI 2 3 0 に、接続設定の一つ以上の入力項目が表示される（プロパティ選択 UI 2 3 0 - 3 ）。

【 0 0 7 2 】

接続設定の一つ以上の入力項目にそれぞれ対応した一つ以上の UI の一例は、接続名 2 4 2、エンドポイント 2 4 3、認証情報 2 4 4、自動作成条件 2 4 5、自動削除条件 2 4 6、リスタート条件 2 4 7、および、カスタムポリシー 2 4 8 がある。ユーザ 4 0 は、こ

10

20

30

40

50

これらのUI 242 ~ 248を介して情報を入力後、設定ボタン240を押下することにより、新しい接続設定を生成できる。UI 242 ~ 248の少なくとも一つについて、入力情報は、当該UIについて予め設定された特定の情報(値)が、ユーザ40からの特段の入力がなかったときに採用されてもよい。また、ユーザ40は、削除ボタン241を押下することにより既存の接続設定を削除できる。更に、プロパティ設定UI 230-3には、ユーザ40が手動でサービス70を作成または削除するために、手動作成ボタン249や手動削除ボタン250が表示されてよい。

**【0073】**

接続名242は、接続設定一覧235に表示される接続設定の名前の入力を受け付ける入力UIである。ユーザ40は、接続名として分かりやすい名前を設定できる。接続名242の初期値には、例えば、後述のServiceIdとResourceIdとの組合せでよい。

10

**【0074】**

エンドポイント243は、サービス70のエンドポイント(例えば、サービス70にアクセスするためのURLやIPアドレス)の入力UIである。認証情報244は、サービス70の機能を使用するための認証情報(例えば、サービス70上のアカウント名とパスワードの組や、アクセストークン情報)の入力UIである。エンドポイントおよび認証情報は、本実施形態では、サービス70の作成時に自動で設定される。ただし、ユーザ40が手動でこれらの情報を入力できてもよい。

**【0075】**

自動作成条件245は、サービス70を自動的に作成する条件である作成条件の入力を受け付ける入力UIである。作成条件が設定されることで、自動で、ユーザ40がサービス70を必要とするタイミングでそのサービス70を作成し、且つ、そのサービス70の接続情報をサービス利用ノードのプロパティに設定できる。

20

**【0076】**

自動作成条件245は、例えば、作成条件の一覧を表示するリストボックスである。選択可能な作成条件として、例えば、接続設定追加時、ノード使用時、サービス呼び出し時、OFF等がある。これらの作成条件は、後述するイベント通知機能32のイベントに対応している。また、自動作成条件245には、初期値として「ノード使用時」などの特定の作成条件を設定しておくことができる。これにより、ユーザ40が手動で作成条件(ポリシーの少なくとも一部)を入力しなくても、サービス70が必要となるタイミングでサービス70が自動作成される。

30

**【0077】**

「接続設定追加時」とは、ユーザ40が設定ボタン240を押下し、接続設定が接続設定一覧235に追加されたときを意味する。「ノード使用時」は、接続設定を使用したサービス利用ノードがフロー221の構成要素である状態でフローの実行ボタン201が押されたときを意味する。「サービス呼び出し時」は、サービス利用ノードの処理が実行され、実際にビジュアルプログラミングシステム10がサービス70にアクセスするときを意味する。「OFF」は、サービス70の自動作成を行わないことを意味する。ただし、「ノード使用時」において、サービス利用ノードが入力ノードなしに単体で置かれている場合など、編集UI 220上に配置されていても明らかに実行されない場合は「ノード使用時」のタイミングとみなされなくてもよい。

40

**【0078】**

「接続設定追加時」により、フロー221の実行前にサービス70が作成されるため、フロー実行時にサービス70の作成を待たなくてよくなる。一方で、サービス70の利用料金が、サービス70が利用可能な状態にある時間長に応じた従量課金に従う場合、フロー221の作成中の段階でも(別の言い方をすれば、サービス70が実際に利用されていなくても利用可能な状態にある限り)、サービス70の利用料金が発生する恐れがある。また、「ノード使用時」(言い換えれば「フロー実行時」と「サービス呼び出し時」とでサービス70の作成タイミングが異なるので、例えば、フロー221にサービス利用ノードは組み込まれているものの、条件分岐の背後にある等の理由により当該サービス利用

50

ノードが滅多に実行されないような場合、「サービス呼び出し時」が選択されれば、サービス70の不要な稼働を回避できる。一方で、サービス70の作成に時間がかかる場合、サービス70が作成されるまで、フロー221の処理が停止するので、そのような場合には、「接続設定追加時」または「ノード使用時」が選択されることが好ましいと考えられる。

#### 【0079】

自動削除条件246は、サービス70を自動的に削除する条件である削除条件の入力を受け付ける入力UIである。削除条件が設定されることで、ユーザ40にとってサービス70が不要となったタイミングで自動でそのサービス70が削除される。結果として、不必要なサービス70の課金を回避することや、消費される記憶容量を低減することが期待できる。

10

#### 【0080】

自動削除条件246は、例えば、削除条件の一覧を表示するリストボックスである。選択可能な削除条件として、例えば、装置停止時、接続設定削除時、ノード削除時、呼び出し終了時、OFF等がある。これらの削除条件は、後述するイベント通知機能32のイベントに対応している。また、自動削除条件246には、初期値として「ノード削除時」などの削除条件を設定しておくことができる。これにより、ユーザ40が手動で削除条件（ポリシーの少なくとも一部）を入力しなくても、サービス70が不要になったタイミングでサービス70が自動削除される。

#### 【0081】

「装置停止時」とは、ビジュアルプログラミングシステム10が停止するタイミングを意味する。「接続情報削除時」とは、ユーザ40により削除ボタン241が押下され、対応する接続設定が削除された状態でフローが実行されたタイミングを意味する。「ノード削除時」とは、当該接続設定を使用するサービス利用ノードが全て削除されたタイミングを意味する。「呼び出し終了時」とは、サービス利用ノードからのサービス70の呼び出しが終わったタイミングを意味する。「OFF」は、サービス70の自動削除を行わないことを意味する。

20

#### 【0082】

リスタート条件247は、サービス70の異常をビジュアルプログラミングシステム10が検知したときの挙動の条件であるリスタート条件の入力を受け付ける入力UIである。リスタート条件が設定されることで、サービス70が異常状態になったとき自動でサービス70を再作成したり、もしくはサービス70内部のデータの保護を優先して、サービス70が異常状態になってもサービス70の再作成を控えたりなど、ユーザ40は挙動を選択できる。ここで、再作成とは、サービス70を一度削除し、改めて作成することを意味する。ただし、サービス70の機能として、再作成やインスタンス再起動など、サービス70の異常解消につながる処理を実行する機能を有している場合は、その機能が実行されてもよい。

30

#### 【0083】

リスタート条件247は、例えば、リスタート条件の一覧を表示するリストボックスである。選択可能なリスタート条件として、例えば、常時、自動作成条件に合わせる、OFF等がある。「常時」とは、サービス70の異常を検知したときは常にサービス70を再作成することを意味する。「自動作成条件に合わせる」とは、サービス70の異常を検知したとき、サービス70の削除だけ行い、その後の再作成は、設定されている自動作成条件に従うことを意味する。「OFF」は、サービス70の自動リスタート（自動での削除および再作成）を行わないことを意味する。

40

#### 【0084】

カスタムポリシー248は、サービス70ごとに個別のポリシーの入力を受け付ける入力UIである。つまり、サービス利用ノードについて、ポリシーは、接続されるサービスに依存しない共通のポリシーと、接続されるサービスに依存する個別のポリシーとのうちの少なくとも共通のポリシーを含む。個別のポリシーとしては、例えば、課金プランがあ

50

る。個別のポリシーは、課金プランに代えてまたは加えて、例えば、サービスのインスタンスの作成先データセンタや、サービスに割り当てるコンピュータリソースの量などが採用されてよい。カスタムポリシー 248 はなくてもよいし、複数存在してもよい。カスタムポリシー 248 が受け付ける入力項目は、例えば、サービス 70 の提供者がサービス定義情報 56 の CustomPolicy 511 として登録する。フロー編集機能 11 は、その CustomPolicy 511 を解釈し、カスタムポリシー 248 を生成する。CustomPolicy 511 のデータ構造およびカスタムポリシー 248 の生成方法は、図 5 および図 6 にて詳述する。

【0085】

<システムの説明(続き)>

【0086】

図 1 の説明に戻る。

【0087】

ビジュアルプログラミングシステム 10 は、フロー編集機能 11 と、フロー実行機能 12 と、ログイン機能 14 と、連携機能 30 とを有する。ビジュアルプログラミングシステム 10 は、フロー情報 13 と、ノード定義情報 20 とを管理する。

【0088】

フロー編集機能 11 は、GUI 画面 200 を提供し、GUI 画面 200 に対するユーザ操作に応答して、フローを生成、変更または削除したり、フロー実行機能 12 を呼び出したりする。フロー編集機能 11 は、ノードパレット 210、プロパティ設定 230 およびノード説明 234 の各々の表示を、後述のノード定義情報 20 にあるノード定義 21 を解釈して決定する。フロー編集機能 11 は、後述の第 2 実施形態の通り、Web アプリケーションとしてユーザ 40 に提供されてもよい。Web アプリケーションとしてのフロー編集機能を含むシステムの構成例は、図 14 にて詳述する。

【0089】

フロー実行機能 12 は、フロー編集機能 11 からの要求を受けてフロー 221 を解釈および実行する。フロー 221 を解釈するとは、フロー情報 13 を解釈し、フロー情報 13 が表すフロー 221 を構成するノード 222 の種類とプロパティと接続関係を認識することを含む。フロー 221 を実行するとは、フロー 221 における前述の入力ノードを見つけ、入力ノードからノード 222 の接続に沿って順番にノード 222 の処理定義 24 のプログラムを実行することを含む。入力ノードが、定期的に実行されたり、HTTP リクエスト受け付けのように他入力をもとに処理を実行したりする場合は、そのたびに入力フローに接続されている一連のフローが実行される。処理定義 24 プログラムの仕様は後述する。

【0090】

フロー情報 13 は、フロー 221 の実態としてのデータである。フロー情報 13 は、例えば、JSON で記載されている。フロー情報 13 は、フロー 221 内の全ノードを表す情報と、各ノードのプロパティを表す情報と、ノード間の接続関係を表す情報とを含む。フロー情報 13 のデータ構造の一例は、図 4 にて詳述する。

【0091】

ログイン機能 14 は、ユーザ 40 にログイン画面を提示し、ログインされたのち、そのアカウント名と、所属プロジェクトと、ユーザ 40 を表すアイコンとを表す情報を含むプロフィール情報を管理する。プロフィール情報が、フロー編集機能 11 により、GUI 画面 200 のユーザ情報 UI 202 に表示される。

【0092】

ノード定義情報 20 は、各ノード 211 の定義であるノード定義 21 を含む。ノード定義 21 は、定期実行ノード、サービス X 利用ノードのようにノードごとに存在し、例えば HTML と JavaScript のような形式言語とプログラミング言語により記述される。ノード定義 21 は、概要定義 22 と、プロパティ定義 23 と、処理定義 24 とを有する。ノード定義 21 の以下の説明において、以下、一つのノードを例に取る(図 1 の説明において「注目ノード」)。

10

20

30

40

50

## 【 0 0 9 3 】

概要定義 2 2 は、例えば、注目ノードの概要を表す情報である。概要定義 2 2 は、例えば、注目ノードの名称を表す情報、注目ノードの種類（例えば、“処理ノード”）を表す情報、ノード説明 2 3 4 の表示内容を表すノード説明情報、および、注目ノードのステータス定義を表す情報、を含む。ノード説明情報は、表示する文章のテキストデータでもよいし、特定のプログラムを実行した結果を表す情報でもよい。注目ノードのステータス定義は、条件分岐を用いて特定の条件を満たす場合に、特定の文字列をステータスとして表示するというプログラムでよい。当該プログラムは、注目ノード 2 1 1 が編集 UI 2 2 0 に配置されたタイミングで一度のみ実行される。当該プログラム内に定期実行処理が記述されていれば、定期的にステータスを更新することができる。

10

## 【 0 0 9 4 】

注目ノードのプロパティ定義 2 3 は、プロパティのデータ構造やプロパティ設定 UI 2 3 0 を構築するためプログラムの定義を表す。

## 【 0 0 9 5 】

処理定義 2 4 は、フロー 2 2 1 の実行時に当該フロー 2 2 1 内の注目ノードが行うべき処理のプログラムの定義を表す。処理定義 2 4 のプログラムは、例えば JavaScript のオブジェクト形式など、ハッシュマップの形式でデータの入出力ができる。この処理で出力されたデータが、フロー 2 2 1 にて注目ノードの出力側に接続しているノードの入力値となる。処理定義 2 4 のプログラム中では、プロパティ設定 UI 2 3 0 にて設定されたプロパティの値が変数として利用できる。

20

## 【 0 0 9 6 】

ここで、サービス利用ノードのノード定義 2 1 の特徴の一例を説明する。サービス利用ノードについてノード説明 UI 2 3 4 に表示される情報は、ポリシー設定機能 3 1 により実行されたノード説明取得処理の戻り値でよい。また、ステータス定義の値は、接続情報一覧 3 5（図 7 参照）の対応するレコードの Status 7 0 4 の値でよい。対応するレコードとは、ResourceId 7 0 0 が接続設定プロパティ内の ResourceId と同じレコードである。また、プロパティ定義 2 3 が、接続設定プロパティと ServiceId プロパティを持つ。接続設定プロパティの入力 UI は、例えば、呼び出されたポリシー設定機能 3 1 により生成される。ServiceId の入力 UI は無い。ユーザ 4 0 により ServiceId が変更されないためである。また、処理定義 2 4 は、サービス 7 0 を呼び出す際は、接続管理機能 3 4 に ResourceId を引数として接続情報取得処理を実施し、その戻り値として得た接続情報でサービス 7 0 を呼び出す。

30

## 【 0 0 9 7 】

連携機能 3 0 は、ビジュアルプログラミングシステム 1 0 からサービス連携システム 5 0 を呼び出したり、サービス 7 0 の接続情報を管理したり、サービス 7 0 の状態を監視したりする。連携機能 3 0 は、ポリシー設定機能 3 1、イベント通知機能 3 2、サービス監視機能 3 3、および、接続管理機能 3 4 を含む。連携機能 3 0 は、接続情報一覧 3 5 を管理する。

## 【 0 0 9 8 】

ポリシー設定機能 3 1 は、サービス利用ノードのノード説明 UI 2 3 4、および、プロパティ設定 UI 2 3 0 を生成したり、プロパティ設定 UI 2 3 0 に対するユーザ操作にตอบสนองして値をポリシー情報 5 5 に追加したりポリシー情報 5 5 から削除したり、手動作成ボタン 2 4 9 もしくは手動削除ボタン 2 5 0 の押下時にサービスの作成または削除を表すイベントをイベント通知機能 3 2 に送付したりする。

40

## 【 0 0 9 9 】

例えば、ポリシー設定機能 3 1 は、ポリシー情報 5 5 から、サービス利用ノードの ServiceId と一致する ServiceId 5 0 1 を持つレコードを全て取得し、取得した各レコード内の Name 5 0 3 の値を、接続設定一覧 2 3 5 に含める。

## 【 0 1 0 0 】

また、例えば、ポリシー設定機能 3 1 は、プロパティ設定 UI 2 3 0 - 3 を生成する際

50

は、カスタムポリシー 248 以外の入力 UI を生成した後、サービス定義情報 56 から、サービス利用ノードの ServiceId と一致する ServiceId 510 を持つレコードを取得し、そのレコード内の CustomPolicy 511 を解釈して、カスタムポリシー 248 の入力 UI を生成する。CustomPolicy 511 を「解釈」とするとは、例えば、CustomPolicy 511 内部の各ポリシー設定項目の「type」が“string”や“number”の場合はテキストボックスを生成し、「type」が“Boolean”の場合はチェックボックスを生成し、「type」が“string”や“number”に加え“enum”を持つ場合は“enum”に定義された項目値を持つリストボックスを生成する等を行うことでよい。このような処理は、CustomPolicy 511 内のポリシーの数だけ繰り返し行われる。

#### 【0101】

また、例えば、ポリシー設定機能 31 は、サービス定義情報 56 のうち ServiceId と一致する ServiceId 510 を持つレコード内の Overview 513、Charges 514 および CustomDescription 515 の値を、ノード説明 UI 234 に表示する。また、当該レコードの CustomPolicy 511 内にカスタムポリシーの「description」が定義されている場合は、追加で、「description」の値が、ノード説明 UI 234 に表示されてよい。

#### 【0102】

ユーザ 40 が設定ボタン 240 を押下したとき、ポリシー設定機能 31 は、プロパティ設定 UI 230 に、ResourceId を含みサービス連携システム 50 内で一意の値を追加した上で、その値を含んだレコードをポリシー情報 55 に新規レコードとして追加する。また、削除ボタン 241 が押下され、既存の接続設定が削除された場合は、ポリシー設定機能 31 は、ポリシー情報 55 のうち、削除対象についての ResourceId と一致する ResourceId 500 を持つレコードを削除する。また、例えば、接続設定が追加または削除される際は、ポリシー設定機能 31 は、上記の処理の後で、イベント通知機能 32 に、「接続先追加」または「接続先削除」を表すイベントを通知する。

#### 【0103】

イベント通知機能 32 は、ビジュアルプログラミングシステム 10 内の特定の処理の発生をイベントとしてサービス連携システム 50 に通知する。このイベントを起点として、サービス連携システム 50 は、サービス 70 の作成もしくは削除を行う。イベントの発生を、ビジュアルプログラミングシステム 10 内の各種プログラム（各種機能）が明示的にイベント通知機能 32 に通知してもよいし、イベント通知機能 32 が、各種プログラムの状態を監視することでイベントの発生を検知してもよい。

#### 【0104】

イベントとしては、例えば、接続設定追加、接続設定削除、ノード使用、ノード削除、サービス呼び出し、サービス呼び出し終了、装置停止、強制作成、および、強制終了がある。

#### 【0105】

「接続設定追加」は、設定ボタン 240 の押下時に発生するイベントであり、設定ボタン 240 の押下を表すイベントである。「接続設定削除」は、削除ボタン 241 の押下時に発生するイベントであり、削除ボタン 241 の押下を表すイベントである。「ノード使用」は、フロー実行ボタン 201 の押下時に発生するイベントであり、フロー実行ボタン 201 が押下された時にサービス 70 の接続情報がプロパティに未設定のサービス利用ノードをフロー 221 が有することを表すイベントである。「ノード削除」は、フロー実行ボタン 201 の押下時に発生するイベントであり、フロー実行ボタン 201 が押下された時に接続情報一覧 35 内にフロー 221 内で未使用の接続情報があることを表すイベントである。

#### 【0106】

また、「サービス呼び出し」は、フロー 221 の実行時に発生するイベントであり、フローの 221 の実行においてサービス利用ノードの処理によりビジュアルプログラミングシステム 10 がサービス 70 にアクセスすることを表すイベントである。「サービス呼び出し終了」は、フロー 221 の実行時に発生するイベントであり、ビジュアルプログラミ

10

20

30

40

50

ングシステム 10 からサービス 70 へのアクセスが終了したことを表すイベントである。「装置停止」は、フロー編集機能 11 やフロー実行機能 12 のプログラムが停止する際に発生するイベントであり、フロー編集機能 11 やフロー実行機能 12 の停止を表すイベントである。

#### 【0107】

また、「強制作成」は、手動作成ボタン 249 の押下時やサービス監視機能 33 でのリスタート処理の際に発生するイベントであり、サービスの強制作成を表すイベントである。「手動終了」は、手動削除ボタン 250 の押下時やサービス監視機能 33 でのリスタート処理の際に発生するイベントであり、手動によりサービスを削除することを表すイベントである。

10

#### 【0108】

イベントの発生時に、イベント通知機能 32 がサービス連携システム 50 に送付するデータであるイベント通知は、例えば、JSON形式のデータであり、“event”、“serviceld” および “resourceld” を含む。

#### 【0109】

“event” は、発生したイベントの名前もしくは識別子を表す。具体的には、“event” は、前述の「接続設定追加」などのイベントの種類を表す。“event” は、接続設定追加などの文字列を含んでもよいし、イベントの種類を識別子を含んでもよい。“event” は、イベントの発生原因を表す情報を含んでもよい。

#### 【0110】

“serviceld” は、イベントの対象となるサービス 70 の識別子を表し、例えば、サービス利用ノードのプロパティが持つ Serviceld の値や、接続情報一覧の Serviceld 701 の値を含んでよい。

20

#### 【0111】

“resourceld” は、イベントに関係する接続設定または接続情報の Resourceld を表す。例えば、“resourceld” は、イベントを発行したサービス利用ノードの接続設定が持つ Resourceld や、サービス監視機能 33 により異常を検知された接続情報の Resourceld 700 の値を含んでよい。対象となる Resourceld が複数ある場合は、Resourceld ごとにイベントが発行されてよい。

#### 【0112】

サービス監視機能 33 は、定期的にサービス 70 の監視処理を行うことでサービス 70 の作成完了や異常を検知し、図 7 で示される接続情報一覧 35 の Status 704 を更新する。また、サービス 70 の異常を検知した際は、サービス監視機能 33 は、接続管理機能 34 にサービス 70 の接続情報の再取得を依頼したり、それでもなおサービス 70 の異常を検知したときはポリシー情報 55 に従い、サービス 70 を再起動したりする。接続情報の再取得及びリスタートの処理は、図 13 にて詳述する。

30

#### 【0113】

サービス監視機能 33 により、ユーザ 40 が GUI 画面 200 からサービス 70 のステータスを把握できるようになる。また、未稼働やエラー（サービス 70 のエラー以外にネットワークの到達性が無い場合のエラーも含んでよい）によりサービス 70 が利用できないとき、接続管理機能 34 がサービス利用ノードや稼働確認ノードの処理を止めることや、アクセス不可の状態で行われるフロー 221 でエラーが発生することを、防げる。更に、サービス 70 のエラー時にユーザ 40 が手動でサービス 70 を再起動する必要がなくなる。

40

#### 【0114】

サービス監視機能 33 は、接続情報一覧 35 に含まれる接続情報を持つサービス 70 を監視対象とする。また、サービス 70 のステータスとしては、例えば、サービス 70 が作成されていない状態を意味する “Pending”、作成中を意味する “Provisioning”、稼働中を意味する “Running”、および、異常発生中を意味する “Error” などがある。

#### 【0115】

50

ここで、サービス監視機能 3 3 によるサービス 7 0 のステータス検知の具体的な動作の一例を説明する。サービス監視機能 3 3 が、接続情報一覧 3 5 への新規接続情報が追加されたことを検知した場合、ポリシー情報 5 5 から当該接続情報の ResourceId と同じ ResourceId 7 0 0 を持つレコードの Monitoring 5 1 2 の値を取得する。その後、サービス監視機能 3 3 が図 6 に示す方法で、Monitoring 5 1 2 の内容を基にサービス 7 0 の監視処理を実行する。サービス監視機能 3 3 が、サービス 7 0 の監視処理の応答に合わせて、監視処理において検出されたステータスを、当該サービス 7 0 に対応した Status 7 0 5 として登録する。

#### 【 0 1 1 6 】

ここで、「監視処理」とは、一定時間ごとにサービス 7 0 の指定された URL にアクセスし、正しい応答が返るかチェックする処理を意味する。「正しい応答」とは、例えば、HTTP のステータスコードが 2 0 0 番代の HTTP レスポンスを意味してよい。ただし、HTTP に対応していないサービス 7 0 の監視処理は、TCP、UDP、ICMP 等により指定されたエンドポイントからの応答が返るかチェックしてもよい。また、監視処理にて応答が返らない場合、サービス監視機能 3 3 は、接続管理機能 3 4 に接続情報の再取得を依頼し、接続情報が更新された後改めて監視処理を実行してよい。具体的には、例えば、サービス 7 0 から応答が返らない、もしくはエラーが返る場合、サービス監視機能 3 3 は、そのサービス 7 0 の ResourceId を接続情報一覧 3 5 から取得し、その ResourceId を引数として接続管理機能 3 4 に接続情報の再取得を依頼し、その完了通知を取得後、改めて監視処理を実行してよい。

#### 【 0 1 1 7 】

接続管理機能 3 4 は、接続情報一覧 3 5 を作成、変更および削除したり、サービス利用ノードに接続情報を伝達したりする。本機能 3 4 により、ユーザ 4 0 による接続情報の管理や、サービス利用ノードへの手動で接続情報設定が不要になり、ユーザ 4 0 の負荷が削減される。

#### 【 0 1 1 8 】

接続管理機能 3 4 は、サービス連携システム 5 0 からの通知を基に、作成されたサービス 7 0 の接続情報を接続情報一覧 3 5 に追加したり、削除されたサービス 7 0 の接続情報を接続情報一覧 3 5 から削除したりする。この処理は、図 8 にて詳述する。

#### 【 0 1 1 9 】

また、接続管理機能 3 4 は、サービス監視機能 3 3 からの依頼を受け、サービス 7 0 の接続情報の再取得を行う。具体的には、例えば、接続管理機能 3 4 は、サービス監視機能 3 3 から対象となるサービス 7 0 の ResourceId 7 0 0 を引数としてサービス 7 0 の接続情報再取得を依頼されると、情報取得機能 5 4 に ResourceId 7 0 0 の接続情報の取得を依頼する。接続管理機能 3 4 は、その応答として取得した接続情報にて、接続情報一覧 3 5 の ResourceId 7 0 0 が一致するレコードの Endpoint 7 0 2 と Credential 7 0 3 の値を更新し、サービス監視機能 3 3 に応答を返す。

#### 【 0 1 2 0 】

また、サービス利用ノードから ResourceId を引数としてサービス 7 0 の接続情報の取得依頼を受けると、接続管理機能 3 4 は、接続情報一覧 3 5 から、引数の ResourceId と一致する ResourceId 7 0 0 を持つレコードを取得する。当該レコード内の Status 7 0 4 が “Running” のときは、接続管理機能 3 4 は、当該レコード内の Endpoint 7 0 2 と Credential 7 0 3 の値をサービス利用ノードに返り値として渡す。当該レコード内の Status 7 0 4 が “Error” のときは、接続管理機能 3 4 は、サービス利用ノードに Error を通知する。当該レコード内の Status 7 0 4 が “Running” および “Error” のいずれでもないときは、接続管理機能 3 4 は、Status 7 0 4 が “Running” になるまで一定時間応答を待つ。本処理は、図 1 1 にて詳述する。

#### 【 0 1 2 1 】

接続情報一覧 3 5 は、サービス 7 0 の接続情報を保持する。接続情報一覧 3 5 のデータ構造は、図 7 にて詳述する。

## 【 0 1 2 2 】

<ハードウェア構成>

## 【 0 1 2 3 】

図 3 は、計算機のハードウェア構成の一例を示すブロック図である。

## 【 0 1 2 4 】

計算機 3 0 0 は、ビジュアルプログラミングシステム 1 0、サービス連携システム 5 0、サービス管理システム 6 0、および、サービス 7 0 のうちの少なくとも一つの実行基盤の構成要素の一例である。すなわち、ビジュアルプログラミングシステム 1 0、サービス連携システム 5 0、サービス管理システム 6 0、および、サービス 7 0 のうちの少なくとも一つは、一つ以上の計算機 3 0 0 を基に実現される。なお、実行基盤は、単一の計算機でもよいし、複数の計算機を含むクラウド基盤のようなシステムでもよい。ビジュアルプログラミングシステム 1 0、サービス連携システム 5 0、サービス管理システム 6 0、および、サービス 7 0 のうちの少なくとも一つは、計算機 3 0 0 のような物理的な計算機上に実現されてもよいが、仮想的な計算機上で実現されてもよい。フロー情報 1 3、ノード定義情報 2 0、接続情報一覧 3 5、ポリシー情報 5 5、サービス定義情報 5 6、管理システム情報 5 7 および接続情報一覧 6 3 が、一つ以上の記憶装置に格納されてよい。一つ以上の記憶装置の少なくとも一つは、いずれかの実行基盤内に存在してもよいし、実行基盤が通信可能な外部の記憶装置でもよい。

10

## 【 0 1 2 5 】

図 3 において、計算機 3 0 0 には、プロセッサ 3 0 1、通信インターフェースデバイス 3 0 3、主記憶デバイス 3 0 4 および補助記憶デバイス 3 0 5 が設けられている。プロセッサ 3 0 1、通信インターフェースデバイス 3 0 3、主記憶デバイス 3 0 4 および補助記憶デバイス 3 0 5 は、内部バス 3 0 6 を介して相互に接続されている。

20

## 【 0 1 2 6 】

また、計算機 3 0 0 のヒューマンインターフェースとして、入力デバイス 3 0 7 および出力デバイス 3 0 8 が設けられていてもよい。入力デバイス 3 0 7 および出力デバイス 3 0 8 は、内部バス 3 0 6 に接続されてよい。

## 【 0 1 2 7 】

プロセッサ 3 0 1 は、計算機 3 0 0 全体の動作制御を司るハードウェアである。主記憶デバイス 3 0 4 は、例えば、半導体メモリから構成され、各種プログラムやデータを保持する。具体的には、例えば、ビジュアルプログラミングシステム 1 0 の主記憶デバイス 3 0 4 には、フロー編集機能 1 1、フロー実行機能 1 2、ログイン機能 1 4、連携機能 3 0 を実現するための一つ以上のプログラム、およびフロー情報 1 3、ノード定義情報 2 0 を保持することができる。また、サービス連携システム 5 0 の主記憶デバイス 3 0 4 には、イベント検知機能 5 1、サービス作成機能 5 2、サービス削除機能 5 3 および情報取得機能 5 4 を実現するための一つ以上のプログラム、および、ポリシー情報 5 5、サービス定義情報 5 6 および管理システム情報 5 7 を保持することができる。また、サービス管理システム 6 0 の主記憶デバイス 3 0 4 には、サービス作成機能 6 1 およびサービス削除機能 6 2 を実現するための一つ以上のプログラム、および、接続情報一覧 6 3 を保持することができる。

30

40

## 【 0 1 2 8 】

補助記憶デバイス 3 0 5 は、大容量の記憶容量を有する記憶デバイスであり、例えば、HDD (Hard Disk Drive) やSSD (Solid State Drive) である。補助記憶デバイス 3 0 5 は、各種プログラムの実行ファイルを保持する。補助記憶デバイス 3 0 5 および補助記憶デバイス 3 0 5 は、プロセッサ 3 0 1 からアクセス可能である。

## 【 0 1 2 9 】

<データ構造>

## 【 0 1 3 0 】

図 4 は、フロー情報 1 3 のデータ構造の一例を示す。

## 【 0 1 3 1 】

50

フロー情報 13 は、例えば JSON 形式のデータである。ただし、フロー情報 13 は、XML や YAML など他のデータ構造のデータであってもよい。

【0132】

フロー情報 13 は、フロー 221 内のノードごとにノード設定 400 のブロックを持つ。一つのノードを例に取る（図 4 の説明において「注目ノード」）。

【0133】

ノード設定 400 は、注目ノードをフロー上で一意に識別するための識別子を表す nodeId 401、ノードの種別（例えば、オブジェクト指向でのクラスに相当）を表す.nodeType 402、GUI 画面 200 での注目ノードの表示名を表す nodeName 403、注目ノードのプロパティとして設定された値を表す properties 404、GUI 画面 200 での注目ノードの表示位置としての座標を表す座標 405、注目ノードと別ノード間の接続関係を表す wires 406 などを含む。

10

【0134】

また、注目ノードがサービス利用ノードの場合、properties 404 は、当該サービス利用ノードに紐付けられた接続設定の resourceId を表す resourceId 407 を持つ。サービス利用ノードは、接続管理機能 34 から、resourceId 407 を引数として、対応するサービス 70 の接続情報を取得することができる。

【0135】

図 5 は、サービス連携システム 50 が持つ各種情報のデータ構造の一例を示す。

【0136】

ポリシー情報 55、サービス定義情報 56 および管理システム情報 57 の各々は、例えばテーブルである。

20

【0137】

ポリシー情報 55 は、例えば、サービス 70 のポリシー毎に、レコードを有する。各レコードは、ResourceId 500、ServiceId 501、ProjectId 502、Name 503、Provision 504、Delete 505、Restart 506 および CustomPolicy 507 といった情報を含む。ポリシー情報 55 の 1 レコードが、接続設定のポリシーに対応する。一つのフローに関し、例えば、一つのポリシーを複数のサービス利用ノードが参照する。以下、一つのポリシーおよびそのポリシーを使用するサービス利用ノードを例に取る（図 5 の説明において、「注目ポリシー」および「注目ノード」）。

30

【0138】

ResourceId 500 は、注目ポリシーの識別子に相当し、注目ノードの ResourceId である。ServiceId 501 は、注目ノードについてのプロビジョニング済みサービス 70 の識別子を表す。なお、「プロビジョニング済みサービス 70」とは、注目ノードのプロパティに未だ接続情報が設定されていないがやがて当該接続情報が設定されることになるサービス 70 と、注目ノードのプロパティに接続情報が設定済みのサービス 70 とのいずれかを意味する。ProjectId 502 は、注目ポリシーを設定したユーザ 40 の所属プロジェクトの識別子を表す。Name 503 は、ユーザ 40 が入力した接続名を表し、接続設定一覧 235 やプロパティ設定 UI 230 - 3 の表示に使用される。Provision 504 は、プロビジョニング済みサービス 70 の自動作成条件である。Delete 505 は、プロビジョニング済みサービス 70 の自動削除条件である。Restart 506 は、プロビジョニング済みサービス 70 のリスタート条件である。CustomPolicy 507 は、注目ポリシーにおけるカスタムポリシーの設定値である。CustomPolicy 507 は、例えば JSON 形式で表されるが、XML や YAML など他のデータ構造であってもよい。

40

【0139】

サービス定義情報 56 は、例えば、サービス 70 毎に、レコードを有する。各レコードは、ServiceId 510、CustomPolicy 511、Monitoring 512、Overview 513、Charges 514 および CustomDescription 515 といった情報を含む。サービス定義情報 56 の各レコードにおける情報は、例えば、当該レコードに対応したサービス 70 の提供者により入力される。一つのサービス 70 を例に取る（図 5 の説明において、「注目サ

50

ービス70」)。

【0140】

ServiceId510は、注目サービス70の識別子を表す。CustomPolicy511は、注目サービス70のカスタムポリシーの定義を表す。CustomPolicy511の詳細は、図6にて説明する。Monitoring512は、注目サービス70の監視処理の定義を表す。Monitoring512の詳細は、図6にて説明する。Overview513は、注目サービス70についてノード説明UI234に表示される概要を表す。Charges514は、注目サービス70についてノード説明UI234に表示される課金形式を表す。CustomDescription515は、注目サービス70についてノード説明UI234に表示される個別説明(例えば、サービス70の提供者が個別に設定する説明項目)を表す。CustomDescription515は、例えば、Markdown形式で表されるが、HTML形式や平文など他のデータ構造であってもよい。

10

【0141】

CustomPolicy511、Monitoring512、Overview513、Charges514、および、CustomDescription515の少なくとも一つの値は、空であってもよい。

【0142】

管理システム情報57は、例えば、サービス管理システム60毎に、レコードを有する。各レコードは、SystemId520、Endpoint521、Credential522およびServices523といった情報を含む。以下、一つのサービス管理システム60を例に取る(図5の説明において「注目管理システム60」)。

20

【0143】

SystemId520は、注目管理システム60の識別子を表す。Endpoint521は、注目管理システム60にアクセスするためのエンドポイント(例えば、URLやIPアドレス)を表す。Credential522は、注目管理システム60の認証情報(例えば、アカウント名とパスワードの組や、アクセストークン)である。Services523は、注目管理システムから作成または削除可能なサービス70のServiceIdのリストを表す。Services523は、例えばJSON形式で表されるが、XMLやYAMLなど他のデータ構造であってもよい。

【0144】

図6は、サービス定義情報56のうちのCustomPolicy511およびMonitoring512のデータ構造の一例を示す。

30

【0145】

上述したように、CustomPolicy511は、カスタムポリシーの定義を表し、Monitoring512は、監視処理の定義を表す。これらは、例えば、JSON形式で表されるが、XMLやYAMLなど他のデータ構造であってもよい。

【0146】

CustomPolicy511は、カスタムポリシーの設定項目毎に、項目定義600を持つ。項目定義600は、例えば、設定項目の名前を表すname601、設定項目の説明を表すdescription602、および、設定項目のデータ構造を表すscheme603を含む。scheme603は、例えば、データ型を表すtype604、データとして入力可能な値であるenum605、および、初期値を表すdefault606を含む。

40

【0147】

description602は、例えばMarkdown形式で表されるが、HTML形式や平文など他のデータ構造であってもよい。description602が設定されている場合は、その値がカスタムポリシーの説明としてノード説明UI234に表示される。type604の値は、例えば、文字列を意味する“string”、数値を意味する“number”、論理値を意味する“Boolean”がある。また、description602とenum605とdefault606は無くてもよい。また、カスタムポリシーが無い場合、CustomPolicy511は空であってもよい。

【0148】

Monitoring512は、例えば、監視のための通信の通信プロトコルを表すprotocol6

50

1 1、監視対象のエンドポイント（例えば、URLやIPアドレス）を表すendpoint 6 1  
 2、監視のための通信である監視通信に設定するHTTPヘッダを表すhttpHeaders 6 1  
 3、サービス7 0の作成後にサービス7 0の起動時間を考慮し初回の監視通信を待つ時間を表すinitialDelaySeconds 6 1 6、監視処理の実行間隔を表すperiodSeconds 6 1 7、および、監視通信の応答のタイムアウト時間を表すtimeoutSeconds 6 1 8を含む。また、httpHeader 6 1 3は、HTTPヘッダの項目がリストとして設定されており、各項目はヘッダ名であるname 6 1 4と値であるvalue 6 1 5で構成される。

【0 1 4 9】

Monitoring 5 1 2には、connection変数として接続情報一覧3 5の値を利用できる。使用する際はconnection.のあとに接続情報一覧3 5のフィールド名が指定されればよい。例えば、connection.Endpointとすると、接続情報一覧3 5にある監視対象のサービス7 0を表すレコードのEndpoint 7 0 2の値を使用できる。変数は、例えば、JavaScriptのテンプレートリテラル内での変数使用時と同じ形式で文字列に埋め込む。各項目は無くてもよく、その場合は、サービス監視機能に定義されているデフォルトの設定値が用いられてよい。

10

【0 1 5 0】

サービス監視機能3 3は、endpoint 6 1 2が表す監視対象にprotocol 6 1 1が表すプロトコルで通信しその返り値を確認する処理を含む監視処理、periodSeconds 6 1 7が表す時間間隔で行う。また、protocol 6 1 1がhttpまたはhttpsを表す場合、HTTPヘッダとしてhttpHeaders 6 1 3の値が用いられる。ただし、サービス7 0の作成後、初回の通信は、initialDelaySeconds 6 1 6が表す時間だけ待ち、その後、監視通信がされる。通信の応答が、timeoutSeconds 6 1 8が表す時間以上返らなかった場合（もしくは、HTTPやHTTPSの場合に通信の応答のステータスコードが所定番代以外のとき）、サービス監視機能3 3は、サービス7 0に異常があると判断する。

20

【0 1 5 1】

図7は、接続情報一覧3 5のデータ構造の一例を示す。

【0 1 5 2】

接続情報一覧3 5は、例えばテーブルである。接続情報一覧3 5は、例えば、サービス7 0の接続情報毎にレコードを有する。各レコードは、接続情報の識別子に相当するResourceIdであるResourceId 7 0 0と、サービス7 0の識別子を表すServiceId 7 0 1と、サービス7 0のエンドポイントを表すEndpoint 7 0 2と、サービス7 0の認証情報であるCredential 7 0 3と、サービス7 0のステータスを表すStatus 7 0 4とを含む。各レコードにおいて、Endpoint 7 0 2およびCredential 7 0 3が、本実施形態において接続情報に相当する。

30

【0 1 5 3】

<各種処理>

【0 1 5 4】

図8は、イベント通知処理を示す。

【0 1 5 5】

本処理では、ビジュアルプログラミングシステム1 0でのフロー実行などのイベントを起点として、サービス7 0が作成または削除され、サービス7 0の接続情報がビジュアルプログラミングシステム1 0に動的に反映される。これにより、ユーザ4 0が事前にサービス7 0を作成または削除し、サービス7 0の接続情報の設定または削除を個別にノード2 1 1のプロパティに対して行なわなくてよくなり、ユーザ4 0の作業負荷が削減される。

40

【0 1 5 6】

まず、フロー編集機能1 1もしくはフロー実行機能1 2が、イベントの発生を検知した場合に当該イベントの通知であるイベント通知をイベント通知機能3 2に送信する（P 9 0 1）。

【0 1 5 7】

次に、当該イベント通知を受けたイベント通知機能3 2が、当該通知が表すイベントの

50

イベント通知を、サービス連携システム50に送信する(P902)。

【0158】

当該イベント通知を受けたサービス連携システム50において、イベント検知機能51が、当該イベント通知とポリシー情報55とを比較する(P903)。イベント検知機能51が、イベント通知が持つResourceId(およびServiceId)と一致するResourceId500(およびServiceId501)に対応したProvision504またはDelete505が表す条件にイベント通知が表すイベント(例えばイベント発生原因)に適合するか否かを判断する。以下、図8の説明において、イベント通知が持つResourceIdに対応したサービス利用ノードを「注目利用ノード」と言い、イベント通知が持つServiceIdに対応したサービス70を「注目サービス70」と言う。

10

【0159】

Provision504との一致が得られた場合(P904:Yes)、イベント検知機能51は、作成処理を実行する(P905)。

【0160】

Delete505との一致が得られた場合(P906:Yes)、イベント検知機能51は、削除処理を実行する(P907)。

【0161】

どちらとの一致も得られなかった場合(P906:No)、イベント検知機能51は、ビジュアルプログラミングシステム10に、処理が完了したという応答を返す(P910)。

20

【0162】

なお、イベント通知が、「強制作成」イベントを表す場合、P904およびP906の判断無しに、P905が行われてよい。また、イベント通知が、「強制終了」イベントを表す場合、P904およびP906の判断無しに、P907が行われてよい。

【0163】

作成処理(P905)では、サービス作成機能52が、注目利用ノードに接続される注目サービス70の作成を、注目サービスを作成可能なサービス管理システム60(注目サービスに対応したSystemId520が表すサービス管理システム60)に依頼し、その結果として取得された接続情報(注目サービス70の接続情報)を、イベント検知機能51に通知する。作成処理P905の詳細は、図9にて説明する。

30

【0164】

削除処理(P907)では、サービス削除機能53が、注目サービス70の削除を、注目サービスを作成したサービス管理システム60(注目サービスに対応したSystemId520が表すサービス管理システム60)に依頼し、削除されたサービス70のResourceIdをイベント検知機能51に通知する。削除処理の詳細は、図10にて説明する。

【0165】

その後、イベント検知機能51は、作成もしくは削除した注目サービス70の接続情報(例えば、接続情報一覧35のレコードのうちStatus704以外のデータ)を、ビジュアルプログラミングシステム10に送付する(P908)。ただし、サービス70の削除処理が実行された場合は、ResourceId以外のレコードの値は空でよい。

40

【0166】

ビジュアルプログラミングシステム10では、接続管理機能34が、サービス連携システム50から受け取った接続情報を接続情報一覧35に反映(例えば追加)する(P909)。具体的には、例えば、受けた接続情報が、接続情報一覧35のレコードのうちStatus以外のレコード部分の場合、接続管理機能34が、そのレコード部分を接続情報一覧35に追加する。一方、例えば、接続情報が、ResourceId以外の値が空の情報の場合、接続管理機能34が、受けた接続情報のResourceIdと一致するResourceId700を持つレコードを接続情報一覧35から削除する。

【0167】

以上の処理により、ビジュアルプログラミングシステム10でのフロー実行などのイベ

50

ントを起点として、サービス70が作成または削除され、その接続情報がビジュアルプログラミングシステム10の接続情報一覧35に動的に反映される。

【0168】

図9は、サービス70の作成処理を示す。

【0169】

本処理は、図8のP905の詳細である。

【0170】

まず、イベント検知機能51が、イベントの発生をサービス作成機能52に通知する(P1001)。

【0171】

次に、サービス作成機能52が、その通知に回答して、管理システム情報57にアクセスし、Services523内にイベント通知のServiceIdを含むレコードを取得する。その後、サービス作成機能52が、そのレコードのEndpoint521およびCredential522を取得し、Endpoint521が表すサービス管理システム60に、Credential522が表す認証情報を用いてアクセスし、イベント通知のResourceIdおよびServiceIdに対応したサービス70のインスタンスの作成を依頼する(P1002)。サービス70の識別子とは、例えばOpenServiceBrokerのinstance\_idである。

【0172】

その後、サービス管理システム60のサービス作成機能61が、依頼されたサービス70を作成する(P1003)。この作成により、サービス管理システム60に定義された方法でサービス70の接続情報(例えば、エンドポイントおよび認証情報を含んだ情報)が発行される。

【0173】

その後、サービス作成機能61は、サービス70の接続情報を接続情報一覧63に保存する(P1004)。その後、サービス作成機能61は、作成処理終了の通知として、サービス70の接続情報(P1004で保存した接続情報)を、サービス連携システム50に送付する(P1005)。

【0174】

サービス連携システム50のサービス作成機能52は、送付された接続情報と、イベント検知機能51から通知されたイベントのServiceIdとResourceIdとを(すなわち、Status704を除いた、接続情報一覧35のレコード部分)を、イベント検知機能32に送付する(P1006)。

【0175】

以上の処理によりサービス70の作成が実現される。

【0176】

図10は、サービス70の削除処理を示す。

【0177】

本処理は、図8のP907の詳細である。

【0178】

まず、イベント検知機能51が、イベントの発生をサービス削除機能53に通知する(P1101)。

【0179】

次に、サービス削除機能53が、その通知に回答して、管理システム情報57にアクセスし、Services523内にイベント通知のServiceIdの値を含むレコードを取得する。その後、サービス削除機能53が、そのレコードのEndpoint521およびCredential522を取得し、Endpoint521が表すサービス管理システム60に、Credential522が表す認証情報を用いてアクセスし、イベント通知のResourceIdおよびServiceIdに対応したサービス70のインスタンスの削除を依頼する(P1102)。

【0180】

その後、サービス管理システム60のサービス削除機能62が、依頼されたサービス7

10

20

30

40

50

0 を削除する ( P 1 1 0 3 )。

【 0 1 8 1 】

その後、サービス削除機能 6 2 は、削除されたサービス 7 0 に対応したレコードを接続情報一覧 6 3 から削除する ( P 1 1 0 4 )。その後、サービス削除機能 6 2 は、削除処理終了の通知をサービス連携システム 5 0 に送付する ( P 1 1 0 5 )。

【 0 1 8 2 】

サービス連携システム 5 0 のサービス削除機能 5 3 は、削除されたサービス 7 0 の ResourceId をイベント検知機能 5 1 に送付する ( P 1 1 0 6 )。

【 0 1 8 3 】

以上の処理によりサービス 7 0 の削除が実現される。

10

【 0 1 8 4 】

図 1 1 は、ノード別の一時停止処理を示す。

【 0 1 8 5 】

各サービス 7 0 のステータスがサービス監視機能 3 3 により監視され Status 7 0 4 に反映されるが、ノード別の一時停止処理とは、サービス 7 0 の Status 7 0 4 に基づきノードの処理を待機または中断する処理である。本処理により、ビジュアルプログラミングシステム 1 0 は、サービス 7 0 の作成前や当該サービス 7 0 のエラー等により当該サービス 7 0 を利用できない状態でサービス利用ノードが実行されてしまうことを避けることができ、以って、フローの処理に悪影響を及ぼすことを避けることができる。

【 0 1 8 6 】

20

フロー 2 2 1 の実行において、サービス利用ノードの処理定義 2 4 がフロー実行機能 1 2 により実行されると、処理定義 2 4 ( プログラム ) は、サービス利用ノードのプロパティが持つ ResourceId を引数として、接続管理機能 3 4 に接続情報の取得を依頼する ( P 1 3 0 1 )。

【 0 1 8 7 】

次に、接続管理機能 3 4 は、接続情報一覧 3 5 から、引数の ResourceId と一致する ResourceId 7 0 0 を持つレコードから、Endpoint 7 0 2、Credential 7 0 3 および Status 7 0 4 の値を取得する ( P 1 3 0 2 )。

【 0 1 8 8 】

その後、Status 7 0 4 が “ Running ” であれば ( P 1 3 0 3 : Y e s )、接続管理機能 3 4 は、取得した Endpoint 7 0 2 と Credential 7 0 3 を処理定義 2 4 に送付する ( P 1 3 0 4 )。

30

【 0 1 8 9 】

また、Status 7 0 4 が “ Error ” であれば ( P 1 3 0 5 : Y e s )、接続管理機能 3 4 は、処理定義 2 4 にエラーを通知する ( P 1 3 0 7 )。エラーを通知された処理定義 2 4 は、サービス 7 0 へのアクセスを中断する。中断の場合、フローが最初から実行されてもよい。

【 0 1 9 0 】

また、Status 7 0 4 が “ Running ” でも “ Error ” でもなければ ( 例えば、 “ Provisioning ” ) ( P 1 3 0 5 : N o )、接続管理機能 3 4 は、予め定められた時間待機し ( P 1 3 0 6 )、その後再び P 1 3 0 2 を実行する。すなわち、サービス 7 0 がエラーではないがサービス 7 0 にアクセス不可の場合には、サービス 7 0 へのアクセスが待機される。

40

【 0 1 9 1 】

以上の処理により、サービス 7 0 の Status 7 0 4 が “ Running ” 以外のときは、サービス 7 0 へのアクセスが待機または中断される。このため、サービス 7 0 が利用可能ではない状態でサービス 7 0 が呼び出されることがなくなり、フローへの悪影響を避けることができる。

【 0 1 9 2 】

図 1 2 は、フロー全体の一時停止処理を示す。

【 0 1 9 3 】

50

フロー全体の一時停止処理とは、稼働確認ノードがフロー内で後続の全てのサービス利用ノードにそれぞれ接続される全サービス70の各々のStatus704を基にフローを待機または中断する処理である。本処理により、ビジュアルプログラミングシステム10は、稼働確認ノード以降の全サービス70が利用可能な状態であることを確認してからフローを実行できるため、いずれかのサービス70が利用不可能な状態で呼び出され、フローの実行に悪影響を及ぼすことを避けることができる。

【0194】

フロー221の実行において、稼働確認ノードの処理定義24がフロー実行機能12により実行されると、処理定義24は、稼働確認ノード以降にある全サービス利用ノード（稼働確認ノードを含むフローにおいて、当該稼働確認ノードの後に実行され得る全サービス利用ノード）がアクセスする全サービス70のステータスを接続管理機能34に確認する（P1401）。具体的には、例えば、次の通りである。すなわち、処理定義24が、稼働確認ノードのフロー情報13にアクセスする。処理定義24が、フロー情報13中のwires407を辿って、稼働確認ノードの出力側に接続しているノードの情報をフローの終端まで全て取得する。その後、処理定義24が、取得したノード群からサービス利用ノードを抽出し、抽出した全サービス利用ノードの各々のResourceIdを取得する。その後、処理定義24は、取得した全ResourceIdを引数として、接続管理機能34に接続情報の取得を依頼する。

10

【0195】

次に、接続管理機能34は、接続情報一覧35から、引数として与えられた各ResourceIdについて、当該ResourceIdと一致するResourceId700を持つレコードから、Endpoint702、Credential703およびStatus704の値を取得する（P1402）。

20

【0196】

その後、取得した全てのStatus704が“Running”であれば（P1403：Yes）、接続管理機能34は、取得した全ての接続情報（Endpoint702とCredential703）を処理定義24に送付し（P1404）、処理定義24が、処理を後続のノード211に渡す（P1408）。

【0197】

取得したStatus704の一つでも“Error”であれば（P1405：Yes）、接続管理機能34は、“Error”に対応するResourceIdのリストを処理定義24に通知する（P1407）。その後、処理定義24は、ResourceIdのリストをフロー編集機能11に通知する（P1409）。また、当該リストが通知されたフロー編集機能11は、フローの実行を中断してよい。例えば、フロー編集機能11は、GUI画面200（または別のGUI画面）に、通知されたリストにおけるResourceIdに対応したサービス利用ノードのリストを表示してよい。

30

【0198】

取得したStatus704に“Error”が無く“Running”および“Error”以外の値（例えば“Provisioning”）が存在するとき（P1405：No）、接続管理機能34は、予め定められた時間待機し（P1406）、その後再びP1402を実行する。すなわち、全てのサービス70のいずれもエラーではないがいずれかのサービス70にアクセス不可の場合には、サービス70へのアクセスが待機される。

40

【0199】

以上の処理により、稼働確認ノード以降の全サービス70のStatusが“Running”以外のときは、フロー0の実行が待機または中断される。このため、サービス70が利用可能ではない状態でサービス70が呼び出されることがなくなり、フローへの悪影響を避けることができる。

【0200】

図13は、サービス再起動処理を示す。

【0201】

サービス再起動処理は、サービス監視機能33がサービス70のエラーを検知したとき

50

に、ユーザ 40 が設定したポリシー情報 55 を基にサービス 70 をリスタートする処理である。本処理により、ユーザ 40 はサービス 70 にエラーが起きた時、手動でのサービス 70 のリスタートが不要になる。また、このときのリスタートのタイミングをユーザ 40 が指定することもできる。

#### 【0202】

サービス監視機能 33 が、監視処理にて、正常な値がサービス 70 から返らない等の理由によりサービス 70 の状態がエラーであると検知すると (P1501)、サービス 70 の ResourceId を引数に、接続管理機能 34 に、サービス 70 の接続情報の再取得を依頼する (P1502)。接続管理機能 34 は、その依頼に応答して、引数とされた ResourceId に一致する ResourceId 700 に対応した接続情報を接続情報一覧 35 から取得し、取得した接続情報をサービス監視機能 33 に渡す。サービス監視機能 33 は、その渡された接続情報を用いて、サービス 70 にアクセスしてみる。

10

#### 【0203】

しかし、それでもサービス 70 への監視処理が失敗する (もしくは接続管理機能 34 にて接続情報の再取得に失敗した通知を受ける) 等の理由によりサービス 70 のエラーを検知すると (P1503)、サービス監視機能 33 は、ポリシー情報 55 から、サービス 70 の ResourceId と一致する ResourceId 500 に対応した Restart 506 の値を確認する。

#### 【0204】

Restart 506 の値が “OFF” の場合 (P1504: Yes)、サービス監視機能 33 は、再起動処理を終了する。

20

#### 【0205】

Restart 506 の値が “OFF” 以外の場合 (P1504: No)、サービス監視機能 33 は、イベント通知機能 32 に、強制終了イベントの通知を送付する。強制終了イベントの通知をイベント通知機能 32 が受けた場合、図 8 の P907 以降の処理が行われる。すなわち、サービス 70 が削除され、サービス 70 の接続情報が接続情報一覧 35 から削除される (P1505)。

#### 【0206】

その後、Restart 506 の値が “常時” の場合 (P1506: Yes)、サービス監視機能 33 は、イベント通知機能 32 に、強制作成イベントの通知を送付する。強制作成イベントの通知をイベント通知機能 32 が受けた場合、図 8 の P905 以降の処理が行われる。すなわち、サービス 70 が作成され、サービス 70 の接続情報が接続情報一覧 35 に登録される (P1507)。

30

#### 【0207】

P1505 の後、Restart 506 の値が “常時” ではない場合 (例えば、“自動作成条件に合わせる” の場合) (P1506: No)、再起動処理は終了する。P1505 で削除されたが本再起動処理において作成されなかったサービス 70 は、後に、ポリシー情報 55 内の Provision 504 の条件を満たすイベントがビジュアルプログラミングシステム 10 で発生した際に作成されることになる。

#### 【0208】

以上の処理により、サービス 70 が異常なとき、ビジュアルプログラミングシステム 10 は接続情報の更新およびポリシー情報 55 に基づくサービスのリスタートが可能となる。

40

#### [第2実施形態]

#### 【0209】

以下、第2実施形態を説明する。その際、第1実施形態との相違点を主に説明し、第1実施形態との共通点については説明を簡略または省略する。

#### 【0210】

図 14 は、第2実施形態におけるシステム構成の一例を示すブロック図である。

#### 【0211】

図 14 に示されるシステムでは、図 1 のビジュアルプログラミングシステム 10 の代わ

50

りにビジュアルプログラミングシステム 1600 が設けられている。ビジュアルプログラミングシステム 1600 にはフロー編集機能 11 の代わりに Web 通信機能 1610 が設けられている。

【0212】

計算機 1620 には、Web ブラウザ 1621 が設けられている。Web ブラウザ 1621 には、フロー編集機能 1622 が設けられている。フロー編集機能 1622 は、Web ブラウザ 1621 にて動作することができる。計算機 1620 は、出力デバイス 1630 および入力デバイス 1640 を有する。さらに、計算機 1620 は、ネットワーク 80 を介して Web 通信機能 1610 に接続されている。

【0213】

Web ブラウザ 1621 は、計算機 1620 の上で動作するアプリケーションである。Web ブラウザ 1621 は、ビジュアルプログラミングシステム 1600 に HTTP または HTTPS リクエストを発行し、その応答として取得したデータを計算機 1620 の出力デバイス 1630 に表示する。この時、Web ブラウザ 1621 は、フロー編集機能 1622 を出力デバイス 1630 に表示することができる。

【0214】

Web 通信機能 1610 は、Web ブラウザ 1621 の HTTP リクエストを基にフロー編集機能 1622 の処理を定義したデータを Web ブラウザ 1621 に送付する。そして、Web ブラウザ 1621 上で動作するフロー編集機能 1622 からの HTTP リクエストを基に、Web ブラウザ 1621 上のフロー編集が実現できるよう、Web 通信機能 1610 は、各種処理の制御命令およびデータを中継する。

【0215】

フロー編集機能 1622 は、フロー編集機能 11 を Web ブラウザ 1621 で動作可能にしたものである。フロー編集機能 1622 が図 14 のシステム内の各種データや機能を使用する時は、Web 通信機能 1610 を介して使用する。

【0216】

以上により、図 14 のシステムは Web アプリケーションとして、遠隔の計算機 1620 を用いてフロー 221 を作成することができる。

【0217】

上述した説明を、例えば下記のように総括することができる。

【0218】

サービス連携システム 50 と通知機能 30 が、サービス連携システムの一例である。サービス連携システム 50 は、第 1 の機能の一例である。通知機能 30 が、第 2 の機能の一例である。サービス連携システム 50 が、ビジュアルプログラミングシステム 10 または 1600 が提供する GUI 画面 200 (フロー編集画面の一例) におけるフロー 221 が、サービス利用ノード 222 を含んでいて、且つ、当該サービス利用ノード 222 にサービス 70 (外部サービスの一例) を接続するユーザ要求を検出した場合、サービス 70 の作成機能 61 を有するサービス管理システム 60 に、サービス 70 の作成要求を送信する。サービス連携システム 50 が、当該作成要求に回答してサービス管理システム 60 により作成されたサービス 70 に接続するための接続情報 (当該サービス管理システム 60 により作成された接続情報) を、サービス管理システム 60 から取得する。連携機能 30 が、サービス連携システム 50 により取得された接続情報をサービス利用ノード 222 のプロパティに設定する。以上のように、フロー 221 の編集において、動的にサービス 70 が作成されると共に、当該作成されたサービス 70 の接続情報が、サービス利用ノード 222 のプロパティに自動で設定される。このため、ユーザは事前にサービス 70 を作成したりそのサービス 70 の接続情報を手動で設定したりする必要がなくなる。これにより、サービス利用ノード 222 を含んだフロー 221 を用いた試行錯誤を高速に行うことができる。

【0219】

連携機能 30 が、サービス 70 の自動作成条件 (「Provision」) と自動削除条件 (「D

10

20

30

40

50

delete」)のうちの少なくとも一つを含んだポリシーの指定をユーザ40から受け付けてよい。サービス連携システム50が、ユーザ40により指定されたポリシーを表すポリシー情報に従ってよい。これにより、サービス70の作成または削除をユーザ40の所望のポリシーに従って行うことができる。

【0220】

サービス連携システム50が、サービス70に関しユーザ要求を検出した場合、ポリシー情報に指定されている自動作成条件が満たされたときに、サービス70の作成要求をサービス管理システム60に送信してよい。これにより、サービス70が作成されるタイミングを、ユーザ40の所望のタイミングとすることができる。

【0221】

サービス連携システム50が、ポリシー情報に指定されている自動削除条件が満たされたときに、サービス70の削除要求をサービス管理システム60に送信してよい。これにより、サービス70が作成されるタイミングを、ユーザ40の所望のタイミングとすることができる。

【0222】

連携機能30は、取得された接続情報を接続情報一覧35に書き込むようになっている。連携機能30は、サービス利用ノード222のStatus704が“Running”でない場合(サービス利用ノード222がプロパティに設定された接続情報を用いてサービス70を呼び出すことができなくなったことを検出したことの一例)、サービス70の接続情報を接続情報一覧35から取得し当該取得された接続情報を用いてサービス70に対するアクセスを行い、それでもアクセスに失敗した場合、削除条件が満たされたときに削除要求をサービス管理システム60に送信し、その後、作成条件が満たされたときにサービス70の作成要求をサービス管理システム60に送信してよい。この段落で言う「削除条件」は、アクセス失敗に回答して直ちに削除する強制削除でもよいし、ユーザ40の任意の条件でもよい。同様に、この段落で言う「作成条件」は、強制削除の後直ちに作成する強制作成でもよいし、ユーザ40の任意の条件でもよい。これにより、サービス70のエラーまたはその他の理由によりサービス70へのアクセスが不可の場合、手動でのサービス70のリスタートが不要になる。

【0223】

連携機能30は、サービス利用ノード222のStatus704が“Running”でない場合、サービス70の接続情報を接続情報一覧35から取得し、当該取得された接続情報を用いてサービス70に対するアクセスしてよい。これにより、サービス70のエラーまたはその他の理由によりサービス70へのアクセスが不可になった場合、サービス利用ノード222に代えて連携機能30がサービス70へのアクセスが可能か否かのチェックを行うことができる。

【0224】

連携機能30は、サービス70にアクセスが可能か監視してよい。連携機能30は、サービス70にアクセス不可の場合、当該サービス70に接続されるサービス利用ノード222の処理を待機または中断してよい。これにより、サービス70が作成前や当該サービス70のエラー等により当該サービス70にアクセス不可の状態ですらサービス利用ノード222が実行されてしまうことを避けることができる。

【0225】

連携機能30は、フロー221が有する全てのサービス利用ノード222の各々について、当該サービス利用ノード222に接続されるサービス70にアクセスが可能か監視してよい。フロー221における稼働確認ノード(実行中ノードの一例)の後に実行される一つ以上のサービス利用ノード222から呼び出される少なくとも一つのサービス70にアクセス不可の場合、フロー221の実行が待機または中断されてよい。これにより、稼働確認ノード以降の全サービス70が利用可能な状態なことを確認してからフローを実行できる。

【0226】

10

20

30

40

50

サービス連携システム 50 は、ビジュアルプログラミングシステム 10 または 1600 の外部に設けられていてよい。連携機能 30 は、ビジュアルプログラミングシステム 10 または 1600 の内部に設けられていてよい。連携機能 30 が、サービス利用ノード 222 に対するサービス 70 の接続がユーザ 40 により要求された場合にユーザ要求の発生を表すイベントの通知をサービス連携システム 50 に送信してよい。当該通知をサービス連携システム 50 により受信することが、ユーザ要求をサービス連携システム 50 により検出することであってよい。これにより、複数のビジュアルプログラミングシステム 10 または 1600 に対しサービス連携システム 50 を一つとすることができる。

【0227】

フロー 221 を表すフロー情報 13 が、サービス利用ノード 222 の ResourceId (所定種類の ID の一例) を表す情報を含んでよい。サービス管理システム 60 への作成要求には、当該サービス利用ノード 222 の ResourceId が関連付けられてよい。当該サービス管理システム 60 により作成され接続情報一覧 63 に格納される接続情報には、ResourceId (例えば、ResourceId と ServiceId との組のようなキー) が関連付けられてよい。サービス連携システム 50 により取得され接続情報一覧 35 において管理される接続情報には、ResourceId (例えば、上記キーと同一のキー) が関連付けられてよい。これにより、サービス利用ノードの ResourceId を用いて接続情報を接続情報一覧 35 から取得できる。

10

【0228】

以上、本発明の幾つかの実施形態を説明したが、本発明は上述した実施形態に限定されるものではなく、添付した特許請求の範囲の趣旨内における様々な変形例および同等の構成が含まれる。例えば、上述した実施形態は本発明を分かりやすく説明するために詳細に説明したものであり、必ずしも説明した全ての構成を備えるものに本発明は限定されない。

20

【符号の説明】

【0229】

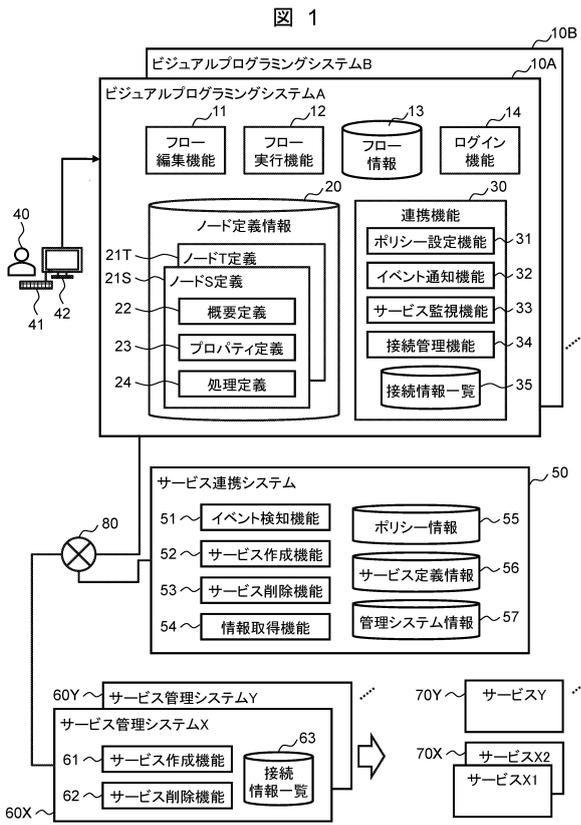
10 : ビジュアルプログラミングシステム、30 : 連携機能、50 : サービス連携システム、60 : サービス管理システム

30

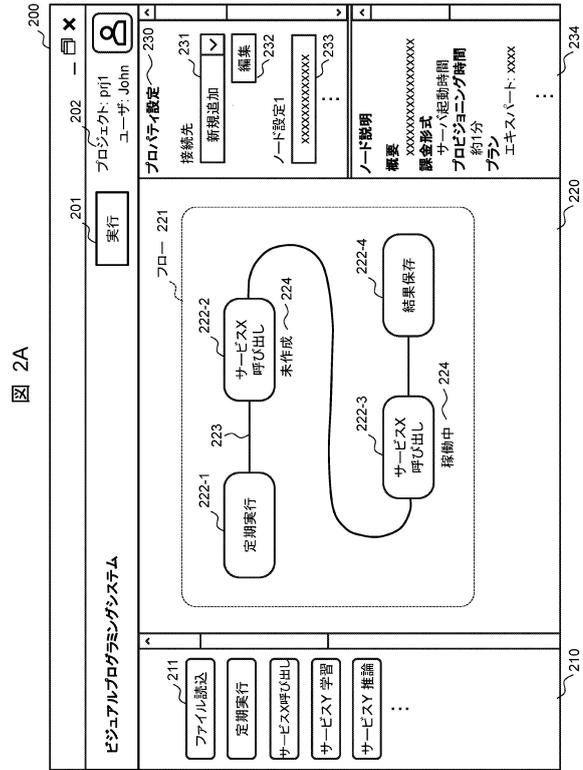
40

50

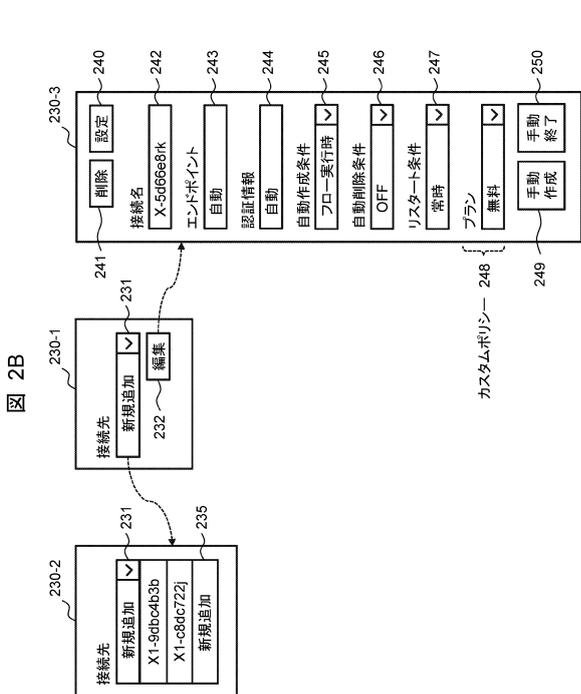
【図面】  
【図 1】



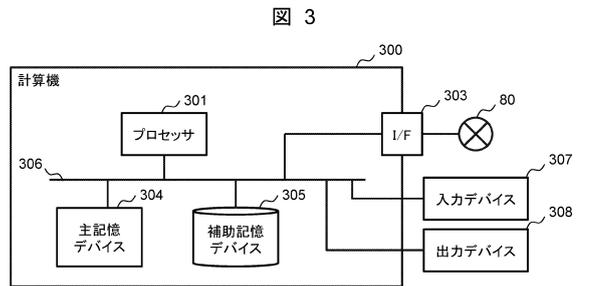
【図 2 A】



【図 2 B】



【図 3】



10

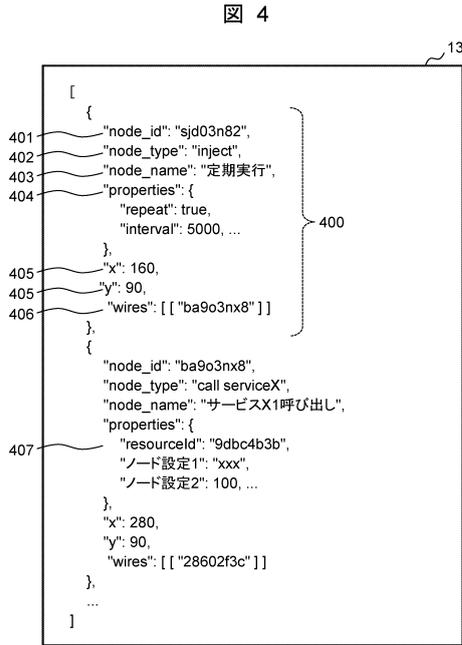
20

30

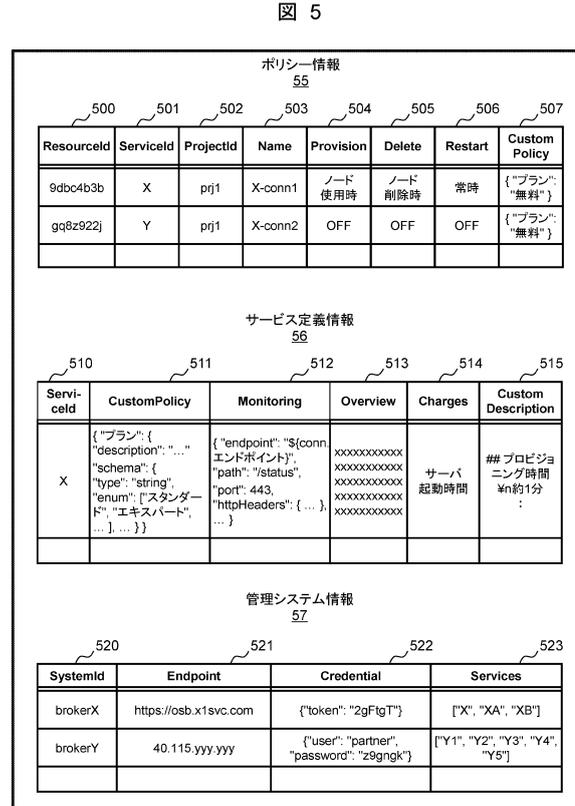
40

50

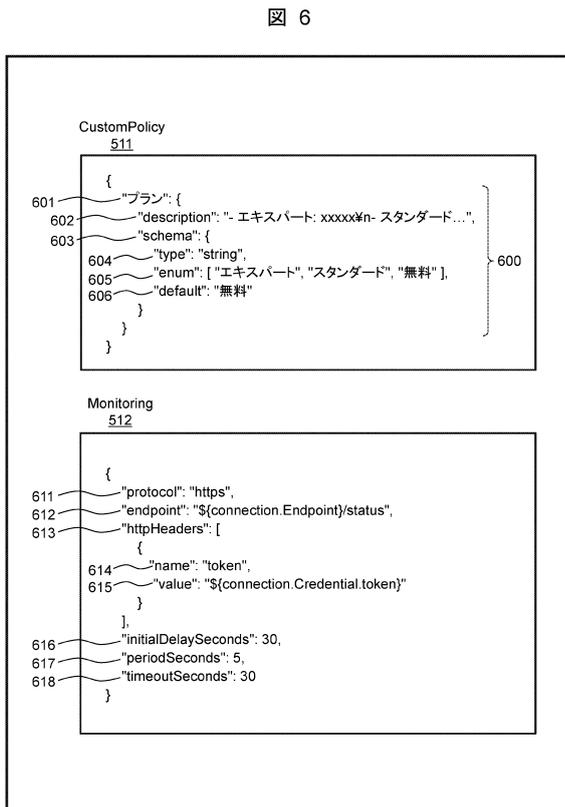
【 図 4 】



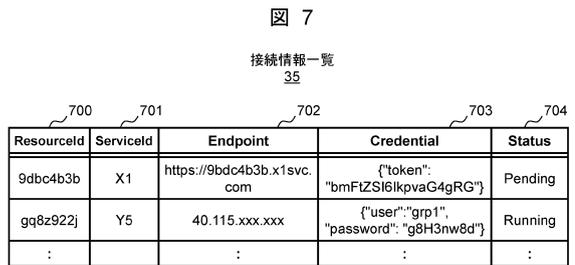
【 図 5 】



【 図 6 】

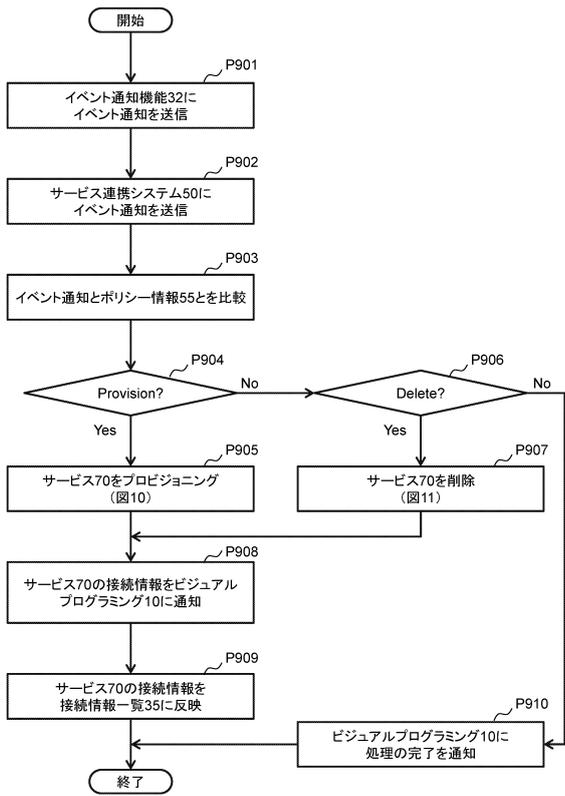


【 図 7 】



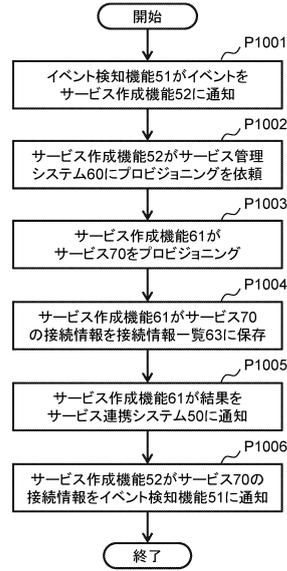
【 図 8 】

図 8



【 図 9 】

図 9

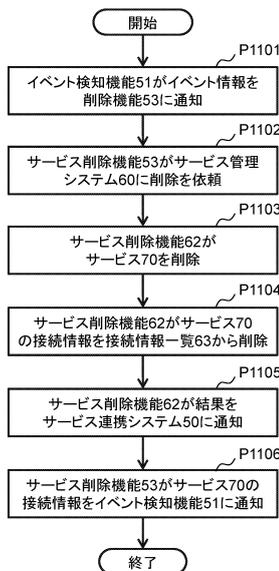


10

20

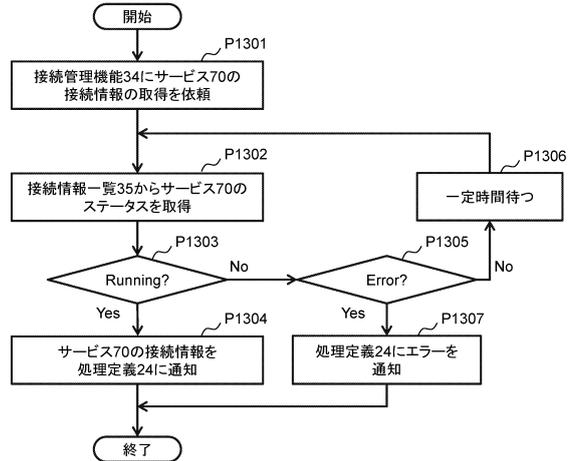
【 図 10 】

図 10



【 図 11 】

図 11



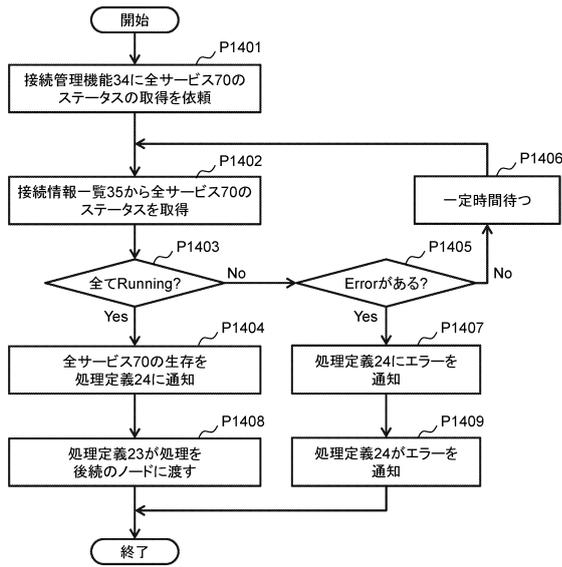
30

40

50

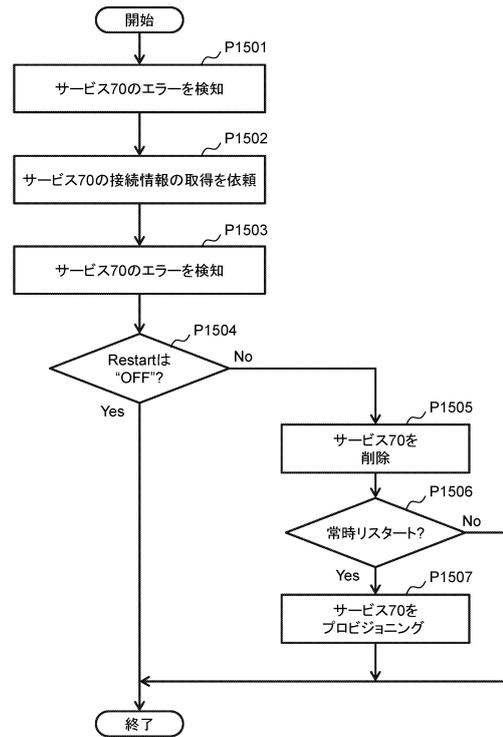
【 図 1 2 】

図 12



【 図 1 3 】

図 13

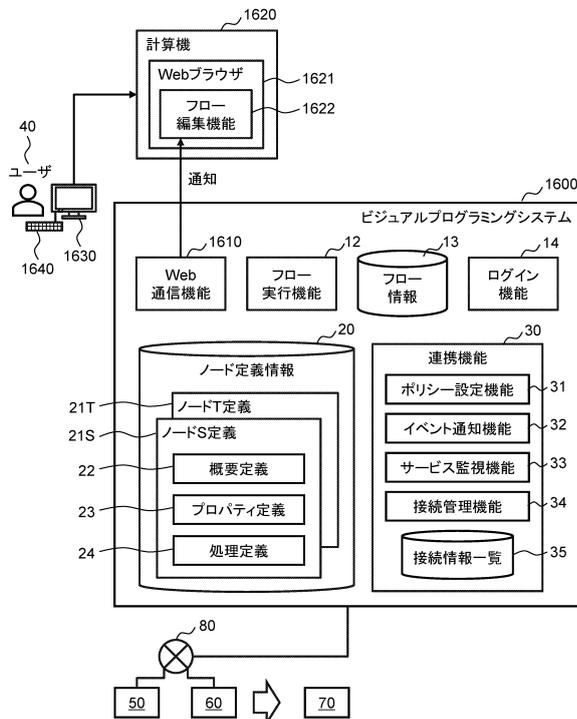


10

20

【 図 1 4 】

図 14



30

40

50

---

フロントページの続き

- (56)参考文献 米国特許出願公開第2018/0081642 (US, A1)  
米国特許出願公開第2018/0069804 (US, A1)
- (58)調査した分野 (Int.Cl., DB名)
- |      |              |
|------|--------------|
| G06F | 8/00 - 8/38  |
| G06F | 8/60 - 8/77  |
| G06F | 9/44 - 9/445 |