

(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2004-86355

(P2004-86355A)

(43) 公開日 平成16年3月18日(2004.3.18)

(51) Int.CI.⁷

G 06 F 9/445

G 06 F 17/60

F 1

G 06 F 9/06 61 O B

G 06 F 17/60 1 O 8

テーマコード(参考)

5 B 0 7 6

審査請求 有 請求項の数 13 O L (全 16 頁)

(21) 出願番号

特願2002-244004 (P2002-244004)

(22) 出願日

平成14年8月23日 (2002.8.23)

(71) 出願人 501223342

タイトラストシステムズ株式会社
神奈川県横浜市中区尾上町3-43

(74) 代理人 100088155

弁理士 長谷川 芳樹

(74) 代理人 100089978

弁理士 塩田 辰也

(74) 代理人 100092657

弁理士 寺崎 史朗

(74) 代理人 100114270

弁理士 黒川 朋也

(72) 発明者 上野 進

神奈川県横浜市中区尾上町3-43 横浜エ
クセレント内5階 タイトラストシス
テムズ株式会社内

F ターム(参考) 5B076 DD01 DF06

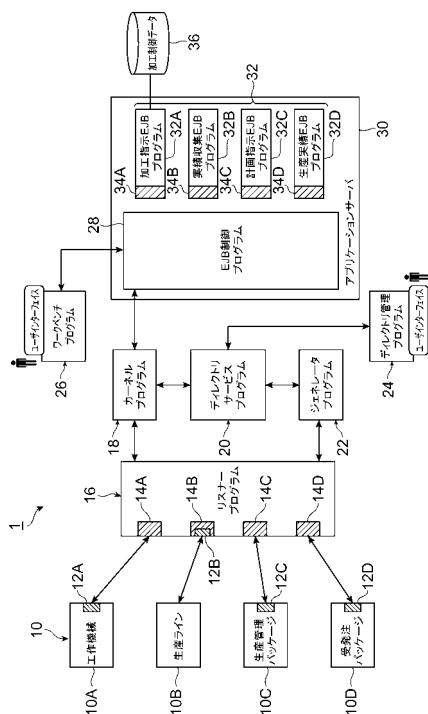
(54) 【発明の名称】統合化システム

(57) 【要約】

【課題】データフォーマットやアクセス方法等が異なる複数のデバイスやパッケージソフト等を円滑に統合化する。

【解決手段】入出力機器10に関する入出力データ構造、プロコトル及びアクセス手段を自己説明型XML文書として送信可能なデバイスエージェント12と、入出力データ構造等に基づいて入出力機器10からの入力データを受け取り、入力データに対し所定の処理を行い、当該処理結果を出力する機能を持つ、オブジェクト指向言語で記述されたEJBプログラム32と、入出力機器10からの自己説明型XML文書を解釈し、解釈で得られた入出力データ構造等の情報及び入出力機器10からの入力データをEJBプログラム32へ送信するとともに、EJBプログラム32からの処理結果を入出力機器10へ送信するミドルソフトウェアシステム16、18、20、22、24、26、28とを備えた統合化システム1を構築する。

【選択図】 図2



【特許請求の範囲】**【請求項 1】**

レガシーデバイス、ディジタルデバイス及び業務アプリケーションプログラムを含んで構成される入出力機器に関する入出力データ構造、入出力プロコトル及びアクセス手段を、自己説明型 XML 文書として送信可能なデバイスエージェントモジュールと、

前記入出力データ構造、入出力プロコトル及びアクセス手段に基づいて前記入出力機器からの入力データを受け取り、当該入力データに対し所定の処理を行い、当該処理結果としての出力データを出力する機能を持つ、オブジェクト指向言語で記述されたバックエンドアプリケーションと、

前記入出力機器からの自己説明型 XML 文書を解釈し、当該解釈で得られた前記入出力データ構造、入出力プロコトル及びアクセス手段の情報及び前記入出力機器からの入力データを前記バックエンドアプリケーションへ送信するとともに、前記バックエンドアプリケーションからの出力データを前記入出力機器へ送信するミドルソフトウェアシステムと、10
を備えた統合化システム。

【請求項 2】

前記ミドルソフトウェアシステムは、

前記デバイスエージェントモジュールから自己説明型 XML 文書を受信するリスナープログラムと、

受信された自己説明型 XML 文書を解釈し、解釈結果に基づいて Java 言語によるクラスオブジェクトを生成するジェネレータプログラムと、20

生成されたクラスオブジェクトを格納するディレクトリサービスプログラムと、

前記ディレクトリサービスプログラムを管理するディレクトリ管理プログラムと、

当該ミドルソフトウェアシステムにおける処理実行を制御するカーネルプログラムと、

前記バックエンドアプリケーション間のワークフローを定義するワークベンチプログラムと、

を含んで構成されることを特徴とする請求項 1 記載の統合化システム。

【請求項 3】

前記デバイスエージェントモジュールは、

前記入出力機器の内部に常駐し、当該入出力機器に関する入出力データ構造、入出力プロコトル及びアクセス手段を解析し、解析結果に基づいて前記自己説明型 XML 文書を生成することを特徴とする請求項 1 又は 2 に記載の統合化システム。30

【請求項 4】

前記デバイスエージェントモジュールには、それぞれ一意のデバイス識別情報が割り当てられることを特徴とする請求項 1 ~ 3 の何れか 1 項に記載の統合化システム。

【請求項 5】

前記自己説明型 XML 文書は、

生成したデバイスエージェントモジュールに割り当てられたデバイス識別情報が記述された識別情報部と、

対象の入出力機器の入出力データ構造が記述されたデータ構造部と、

対象の入出力機器の入出力動作に必要なプロトコルが記述されたプロトコル情報部と、40
前記データ構造部と前記プロトコル情報部とをマッピングするための情報が記述されたメッセージ情報部と、

を含んで構成されることを特徴とする請求項 4 記載の統合化システム。

【請求項 6】

前記入出力機器の入出力データ構造又は入出力プロトコルが変化した場合、前記デバイスエージェントモジュールは、新たに最新の情報に基づいて自己説明型 XML 文書を生成することを特徴とする請求項 1 ~ 5 の何れか 1 項に記載の統合化システム。

【請求項 7】

前記リスナープログラムは、

特定のプロトコルを解釈するためのプロトコル毎のアダプタモジュールを実装し、50

前記アダプタモジュールによって、前記デバイスエージェントモジュールから伝送される自己説明型 XML 文書を解釈可能とされていることを特徴とする請求項 2 記載の統合化システム。

【請求項 8】

前記リストナープログラムは、

前記入出力機器がデバイスエージェントモジュールを内蔵できない場合、当該入出力機器に対応するデバイスエージェントモジュールをアダプタモジュールとして内蔵していることを特徴とする請求項 2 記載の統合化システム。

【請求項 9】

前記ミドルソフトウェアシステムでは、

前記デバイスエージェントモジュールから受信した自己説明型 XML 文書を解析し、解析で得られたデバイス識別情報、入出力機器と通信を行うための入出力データ構造、プロトコル及びネットワークアドレスを、前記ディレクトリサービスプログラムに蓄積することを特徴とする請求項 2 記載の統合化システム。

【請求項 10】

前記ミドルソフトシステムでは、

デバイスエージェントモジュールから受信した自己説明型 XML 文書から、データ入出力検証用のオブジェクト指向言語に基づくクラスオブジェクトが生成され、生成されたクラスオブジェクトは前記ディレクトリサービスプログラムによって蓄積される、

ことを特徴とする請求項 2 記載の統合化システム。

【請求項 11】

最新のクラスオブジェクトは、ミドルソフトウェアシステム内にキャッシュ化されることを特徴とする請求項 10 記載の統合化システム。

【請求項 12】

前記ミドルソフトウェアシステムでは、

入出力機器のデータ構成が変化した場合に送信される新たな自己説明型 XML 文書がクラスオブジェクト化され、

最新の自己説明型 XML 文書と前回の自己説明型 XML 文書とのハッシュ値の相違に基づいて、自己説明型 XML 文書がバージョン管理される、

ことを特徴とする請求項 2 記載の統合化システム。

【請求項 13】

前記ミドルソフトウェアシステムは、

前記自己説明型 XML 文書に基づいて、入出力機器とのプロトコル及びデータ構造に依存しない前記バックエンドアプリケーションのためのアクセス手段として、オブジェクト指向言語仕様に基づくアプリケーションプログラムインターフェースを生成する、

ことを特徴とする請求項 2 記載の統合化システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、工作機械やセンサ機器等のレガシーデバイス、PDA 等のデジタルデバイス、及び業務アプリケーションプログラムを含めた企業内の様々なシステムリソースの統合を図るための統合化システムに関する。

【0002】

なお、本明細書の「入出力機器」は、レガシーデバイスやデジタルデバイス等のハードウェアリソースと、業務アプリケーションプログラム等のソフトウェアリソースの両方を含む概念とされている。

【0003】

【従来の技術】

インターネットは PC、UNIX サーバ、メインフレーム、そして携帯電話や PDA など

10

20

30

40

50

あらゆる情報機器を結ぶネットワークに成長している。これを支えているのは目覚しい進歩を遂げたコンピュータシステムであり、企業各社は何らかの形でコンピュータシステム、インターネットに代表されるネットワークシステムを取り入れている。

【0004】

ここ10年の間にネットワークは目覚しい発達を遂げたにも関わらず、進化した情報機器が扱うコンテンツは、人間に理解できる文字、音声、画像であり、あまり変化していない。昨今の最新情報技術動向から鑑みて、これら情報機器とインターネットを利用して、Web Serviceという概念を用いた大規模企業システムが主流となると思われる。

【0005】

一方、人間が直接理解する必要の無いデータをコンテンツとするセンサ、バーコード、RFID、ICカードなどの情報デバイスが激増している。これら情報デバイスは、その数においても、発生するであろう情報量においても、現在インターネットに接続される情報機器に比較して桁違いに大きく、今後10年間でのこの分野は1990年代のインターネット旋風を上回る爆発的な発展が期待される。

【0006】

ディジタルデバイスは技術革新と量産効果によって小型化、低価格化、高性能化が進み、これによってあらゆるもののがディジタルデバイス化されつつあり、温度センサ、工業用機械など従来からあるレガシーなデバイスもネットワークに接続されるケースが増えてきている。

【0007】

各種のデバイスがネットワーク化されると、これらから発生する情報を取り込んだり、制御したりする業務アプリケーションとの接続が発生する。例えば、コンピュータ化された工作機械からの工程進捗データに基づいて生産ライン全体の調整を行うような生産管理システムや、温度センサからの出力値である温度情報に基づいて予め設定された設定された温度に調整するための環境アプリケーションシステム等、さまざまなアプリケーションシステムが存在する。例えば、図1は代表的な製造業におけるシステム構成を示す。これらはサブシステム（例えば生産管理システム92、受発注システム94、工作機械稼働システム96、生産ラインシステム98）の各々に、相互に異なるデータ構造とプロトコルを実装している。

【0008】

上記のように発生する情報を取り込んだり、制御情報を出力したりするアプリケーションパッケージにはSCADAがある。SCADAソフトウェアはFA分野における代表的なソリューションと言える。このSCADAソフトウェアは、一般にパーソナルコンピュータで動作し、PLC（プログラマブル・ロジック・コントローラ）からの生産ライン情報を収集し、予め設定された条件を満足すると、他のPLCに接続された生産ラインに制御情報を出力したり、他のアプリケーションに情報を出力したりする。

【0009】

【発明が解決しようとする課題】

しかし、このようなソリューションはそれぞれが別々の生き立ちを持って確立されてきた技術であり、他のソリューションとは互換性の点で考慮されていない場合が多い。例えば先のSCADAは対象とするPLCやプラントのハードウェア仕様やDeviceNet等の業界通信プロトコルと強く結びついているため、製造業全体をカバーするような汎用的製品となりにくいのが実情である。すなわち、ソリューション間におけるデータ、プロトコル、操作の非互換性が管理コスト、実装コストの増加を招いている。

【0010】

そのため、現在においては、アプリケーションを含めた企業内の様々なシステムリソースの統合が可能であるミドルソフトウェアが必要とされる。このミドルソフトウェアとしては、はディジタルデバイス、レガシーデバイス、業務アプリケーションそれぞれのデータ構造（スキーマ）、プロトコル、アクセス手法に依存せずに、データ形式の変換、プロコトル変換、ルーティングが可能であることが強く求められる。即ち、ディジタルデバイス

10

20

30

40

50

、レガシーデバイス及び業務アプリケーションのそれぞれにおいて、データ形式やプロコトル等の互換性を持たせるための改変を施すことは非常に処理負担が大きく、コスト増加を招く。特にレガシーデバイス等への改変は技術的に困難であることが多く、現実的な対応とは考えがたい。

【0011】

本発明は、自己説明型XMLと、これを解釈し業務アプリケーションへプロトコルや物理的なデータ構造に依存しないAPI（アプリケーションプログラムインターフェース）を提供するためのミドルソフトウェアシステムを構築し、データフォーマットやアクセス方法等が相互に異なる複数のコンピュータシステム、ディジタルデバイス、レガシーデバイス等を円滑に統合化することができる統合化システムを提供することを目的とする。10

【0012】

【課題を解決するための手段】

上記目的を達成するために、本発明に係る統合化システムは、請求項1に記載したように、レガシーデバイス、ディジタルデバイス及び業務アプリケーションプログラムを含んで構成される入出力機器に関する入出力データ構造、入出力プロコトル及びアクセス手段を、自己説明型XML文書として送信可能なデバイスエージェントモジュールと、前記入出力データ構造、入出力プロコトル及びアクセス手段に基づいて前記入出力機器からの入力データを受け取り、当該入力データに対し所定の処理を行い、当該処理結果としての出力データを出力する機能を持つ、オブジェクト指向言語で記述されたバックエンドアプリケーションと、前記入出力機器からの自己説明型XML文書を解釈し、当該解釈で得られた前記入出力データ構造、入出力プロコトル及びアクセス手段の情報及び前記入出力機器からの入力データを前記バックエンドアプリケーションへ送信するとともに、前記バックエンドアプリケーションからの出力データを前記入出力機器へ送信するミドルソフトウェアシステムとを備えたことを特徴とする。20

【0013】

本発明に係る統合化システムでは、特に、デバイスエージェントモジュールが入出力機器に関する入出力データ構造、入出力プロコトル及びアクセス手段を、自己説明型XML文書として送信可能とされた点、及び、デバイスエージェントモジュールとバックエンドアプリケーションとを中継するミドルソフトウェアシステムを設けた点をポイントとする。

【0014】

即ち、デバイスエージェントモジュールが、各種の入出力機器に関する入出力データ構造、入出力プロコトル及びアクセス手段を、自己説明型XML文書として送信すると、ミドルソフトウェアシステムが、当該自己説明型XML文書を受け取って解釈し、当該解釈で得られた入出力データ構造、入出力プロコトル及びアクセス手段の情報及び入出力機器からの入力データをバックエンドアプリケーションへ送信する。30

【0015】

これらを受信したバックエンドアプリケーションは、入出力データ構造、入出力プロコトル及びアクセス手段に基づいて入出力機器からの入力データを受け取り、当該入力データに対し所定の処理を行い、当該処理結果としての出力データを出力する。そして、ミドルソフトウェアシステムが、バックエンドアプリケーションからの出力データを入出力機器へ送信する。40

【0016】

上記のようにミドルソフトウェアシステムが、各種の入出力機器に関する入出力データ構造、入出力プロコトル及びアクセス手段を表す自己説明型XML文書を自動的に解釈し、入出力機器とバックエンドアプリケーション間のデータを中継するため、入出力機器又はバックエンドアプリケーションが通信相手のデータフォーマットやアクセス方法等に適合させるための処理を行う必要がなく、データフォーマットやアクセス方法等が相互に異なる複数のコンピュータシステム、ディジタルデバイス、レガシーデバイス等を円滑に統合化することができる。

【0017】

10

20

30

40

50

本発明に係る統合化システムは、具体的には、以下の構成とすることができます。

【0018】

本発明に係る統合化システムでは、請求項2に記載したように、前記ミドルソフトウェアシステムは、前記デバイスエージェントモジュールから自己説明型XML文書を受信するリスナープログラムと、受信された自己説明型XML文書を解釈し、解釈結果に基づいてJava言語によるクラスオブジェクトを生成するジェネレータプログラムと、生成されたクラスオブジェクトを格納するディレクトリサービスプログラムと、前記ディレクトリサービスプログラムを管理するディレクトリ管理プログラムと、当該ミドルソフトウェアシステムにおける処理実行を制御するカーネルプログラムと、前記バックエンドアプリケーション間のワークフローを定義するワークベンチプログラムとを含んで構成することができる。10

【0019】

また、請求項3に記載したように、前記デバイスエージェントモジュールは、前記入出力機器の内部に常駐し、当該入出力機器に関する入出力データ構造、入出力プロコトル及びアクセス手段を解析し、解析結果に基づいて前記自己説明型XML文書を生成する構成とすることができます。

【0020】

一般に、入出力機械はさまざまなデータソースを持つ。例えば工作機械では、各種のセンサ情報例えば、油温、水温、ガス圧、冷却水伝導率等を持ち、これらはそれぞれの工作機械機種別、工作機械メーカーごとにさまざまである。また、業務アプリケーションプログラムである生産管理パッケージソフトウェアや受発注パッケージソフトウェアもさまざまな形式の入出力値を持つ。このため、上記のようにデバイスエージェントモジュールが入出力機器に関する入出力データ構造、入出力プロコトル及びアクセス手段を解析し、解析結果に基づいて自己説明型XML文書を生成する構成とすることで、既存のシステムに追加・修正を加えることなく、各入出力機器に関する入出力データ構造、入出力プロコトル及びアクセス手段の情報を得るしくみを構築することができる。20

【0021】

また、請求項4に記載したように、前記デバイスエージェントモジュールに、それぞれ一意のデバイス識別情報が割り当てられた構成とすることができます。これにより、各入出力機器が統合化システム内でユニークであることを保証することができる。30

【0022】

また、請求項5に記載したように、前記自己説明型XML文書は、生成したデバイスエージェントモジュールに割り当てられたデバイス識別情報が記述された識別情報部と、対象の入出力機器の入出力データ構造が記述されたデータ構造部と、対象の入出力機器の入出力動作に必要なプロトコルが記述されたプロトコル情報部と、前記データ構造部と前記プロトコル情報部とをマッピングするための情報が記述されたメッセージ情報部とを含んで構成することができる。

【0023】

また、請求項6に記載したように、前記入出力機器の入出力データ構造又は入出力プロトコルが変化した場合、前記デバイスエージェントモジュールは、新たに最新の情報に基づいて自己説明型XML文書を生成する構成とすることができます。この場合、入出力機器の入出力データ構造又は入出力プロトコルが変化しても、最新の情報に基づいて自己説明型XML文書がデバイスエージェントモジュールにより生成されることとなる。40

【0024】

また、請求項7に記載したように、前記リスナープログラムは、特定のプロトコル（例えば、HTTP、SMTP等）を解釈するためのプロトコル毎のアダプタモジュールを実装し、前記アダプタモジュールによって、前記デバイスエージェントモジュールから伝送される自己説明型XML文書を解釈可能な構成とすることができます。この場合、リスナープログラムは、プロトコル毎のアダプタモジュールによって、さまざまなプロトコルによる自己説明型XML文書の解釈が可能となる。50

【 0 0 2 5 】

また、請求項 8 に記載したように、前記リスナープログラムは、前記入出力機器がデバイスエージェントモジュールを内蔵できない場合、当該入出力機器に対応するデバイスエージェントモジュールをアダプタモジュールに内蔵した構成とすることができます。この場合、入出力機器とリスナープログラムは、入出力機器独自のプロトコルで通信を行い、デバイスエージェントモジュールを内蔵したアダプタモジュールが、自己説明型 XML 文書を生成することができる。即ち、入出力機器がデバイスエージェントモジュールを内蔵できない場合であっても、当該入出力機器も含めて複数システムを円滑に統合化することができる。

【 0 0 2 6 】

また、請求項 9 に記載したように、前記ミドルソフトウェアシステムは、前記デバイスエージェントモジュールから受信した自己説明型 XML 文書を解析し、解析で得られたデバイス識別情報、入出力機器と通信を行うための入出力データ構造、プロトコル及びネットワークアドレスを、前記ディレクトリサービスプログラムに蓄積する構成とすることができます。

【 0 0 2 7 】

また、請求項 10 に記載したように、前記ミドルソフトシステムでは、デバイスエージェントモジュールから受信した自己説明型 XML 文書から、データ入出力検証用のオブジェクト指向言語に基づくクラスオブジェクトが生成され、生成されたクラスオブジェクトは前記ディレクトリサービスプログラムによって蓄積される構成とすることができます。

【 0 0 2 8 】

また、請求項 11 に記載したように、最新のクラスオブジェクトは、ミドルソフトウェアシステム内にキャッシュ化される構成とすることができます。この場合、ミドルソフトウェアシステムによる処理のパフォーマンス効率を向上させることができます。

【 0 0 2 9 】

また、請求項 12 に記載したように、前記ミドルソフトウェアシステムでは、入出力機器のデータ構成が変化した場合に送信される新たな自己説明型 XML 文書がクラスオブジェクト化され、最新の自己説明型 XML 文書と前回の自己説明型 XML 文書とのハッシュ値の相違に基づいて、自己説明型 XML 文書がバージョン管理される構成とすることができます。即ち、ハッシュ値が前回の自己説明型 XML 文書と異なる場合にのみ、クラスオブジェクトの生成をすることで、統合化システムにおける効率良い運用を実現し、無駄な処理を省くことができる。

【 0 0 3 0 】

また、請求項 13 に記載したように、前記ミドルソフトウェアシステムは、前記自己説明型 XML 文書に基づいて、入出力機器とのプロトコル及びデータ構造に依存しない前記バックエンドアプリケーションのためのアクセス手段として、オブジェクト指向言語仕様に基づくアプリケーションプログラムインターフェースを生成する構成とすることができます。この場合、バックエンドアプリケーションが、入出力機器毎に存する入出力の特性を考慮する必要がなくなるという利点がある。

【 0 0 3 1 】**【発明の実施の形態】**

以下、添付図面に基づき、本発明に関する実施の形態について説明する。尚、各図において同一要素には同一符号を付して説明を省略する。また、本実施形態では、入出力機器間のデータ符号化形式、データ交換形式として XML 文書を利用する。XML (拡張可能型マークアップ言語: extensible Markup Language) は、データソース間で共通のデータ形式を定義する。XML はデータソースに独自の「型」の定義を可能とするが、本発明においては「型」は W3C (World Wide Web Consortium) において定義された XML スキーマ言語の型に従う。

【 0 0 3 2 】

図 2 には、本実施形態に係る製造業企業内のサブシステム間を統合するための統合シス

10

20

30

40

50

テム 1 の構成例を示す。この統合化システム 1 における入出力機器 1 0 としては、工作機械 1 0 A、生産ライン 1 0 B、生産管理パッケージソフトウェア 1 0 C 及び受発注パッケージソフトウェア 1 0 D から構成される。原則的に入出力機器 1 0 の内部には、デバイスエージェントモジュール 1 2 A 等（デバイスエージェントモジュール 1 2 と総称する）が常駐し、デバイスエージェントモジュール 1 2 は、これらの入出力データ構造と入出力プロトコルを解析して、自己説明型 XML 文書を生成する。入出力機器 1 0 の入出力データ構造や入出力プロトコルが変化した場合、デバイスエージェントモジュール 1 2 は、新たに最新の情報に基づいて自己説明型 XML 文書を生成する。なお、デバイスエージェントモジュール 1 2 には、それぞれ一意のデバイス識別情報が割り当てられる。

【 0 0 3 3 】

10

統合化システム 1 におけるバックエンドプログラムは、Java 言語による EJB (Enterprise Java Bean) プログラム 3 2 として構築され、EJB プログラム 3 2 はアプリケーションサーバ 3 0 上で実行される。EJB プログラム 3 2 のうち加工指示 EJB プログラム 3 2 A は工作機械や生産ラインを動作させるための EJB プログラムであり、データベース 3 6 に存在する加工制御データを統合化システム 1 に送り込み、工作機械や生産ラインを実行制御する。ここでの加工制御データは工作機械や生産ライン装置が判読できる数値制御データでも良いし、装置に送り込めるマイクロプログラムでも良い。また、実績収集 EJB プログラム 3 2 B は工作機械や生産ラインからの動作実績、障害情報を収集する EJB プログラムであり、計画指示 EJB プログラム 3 2 C は新規受注データを生産管理パッケージに送信するための EJB プログラムである。更に、生産実績 EJB プログラム 3 2 D は、収集された実績データを生産管理パッケージに送信するための EJB プログラムである。

20

【 0 0 3 4 】

統合化システム 1 におけるミドルソフトウェアシステムは、Java 言語によって構築され、リストナープログラム 1 6、ジェネレータプログラム 2 2、ディレクトリサービスプログラム 2 0、ディレクトリ管理プログラム 2 4、カーネルプログラム 1 8、EJB 制御プログラム 2 8 及びワークベンチプログラム 2 6 から構成される。

【 0 0 3 5 】

30

このうちリストナープログラム 1 6 は、入出力機器と通信を行うプログラムであり、特定のプロトコルを解釈するためのアダプタモジュール 1 4 A、1 4 B、1 4 C、1 4 D（アダプタモジュール 1 4 と総称する）を実装し、特定のプロコトルによって入出力機器 1 0 から伝送される自己説明型 XML 文書を受信可能とされている。

【 0 0 3 6 】

但し、リストナープログラム 1 6 は、入出力機器 1 0 がデバイスエージェントモジュール 1 2 を内蔵できない場合、例えば、図 2 の例で生産ライン 1 0 B がデバイスエージェントモジュール 1 2 を内蔵できない場合、生産ライン 1 0 B に対応するデバイスエージェントモジュール 1 2 B をアダプタモジュール 1 4 B に内蔵している。上記構成により、生産ライン 1 0 B とリストナープログラム 1 6 は、生産ライン 1 0 B 独自のプロトコルで通信を行い、アダプタモジュール 1 4 B が自己説明型 XML 文書を生成することが可能である。

【 0 0 3 7 】

40

ジェネレータプログラム 2 2 は、入出力機器 1 0 が送信する自己説明型 XML 文書を解釈しクラスオブジェクトを生成するプログラムである。また、ディレクトリサービスプログラム 2 0 は、ユーザインターフェースを持ったディレクトリ管理プログラム 2 4 により制御され、カーネルプログラム 1 8 は、ミドルソフトウェアシステム全体の動作を制御するプログラムである。また、EJB 制御プログラム 2 8 は、EJB プログラム間のワークフロー制御とミドルソフトウェアシステム間の通信を行うプログラムである。この EJB 制御プログラム 2 8 は、ユーザインターフェースを持ったワークベンチプログラム 2 6 からの指示に基づいてワークフロー制御の調整を行う。ワークフローが変更された場合には EJB 制御プログラム 2 8 は、ワークベンチプログラム 2 6 からの変更指示に基づいて、呼び出す対象の EJB プログラムを変更する。更に、ワークベンチプログラム 2 6 は自己説明型

50

X M L 文書から入出力クラスライブラリ 3 4 A、3 4 B、3 4 C、3 4 D（入出力クラスライブラリ 3 4 と総称する）を生成できる。入出力クラスライブラリ 3 4 は J a v a 言語によるテンプレートプログラムであり、コンパイル時に E J B プログラム 3 2 とリンクすることができる。

【 0 0 3 8 】

図 3 及び図 4 には、デバイスエージェントモジュール 1 2 が生成する自己説明型 X M L 文書 4 0 の構成を示す。自己説明型 X M L 文書 4 0 は、X M L スキーマ言語によって記述された入力用構造と出力用構造が記述されたデータ構造部 4 0 A と、発信元を一意に特定するためのデバイス I D (D I D) が記述されている識別情報部 4 0 B と、入出力動作に必要なプロトコルが記述されるプロトコル情報部 4 0 C と、入出力において使用するデータ構造部 4 0 A とプロトコル情報部 4 0 C をマッピングするための情報を記述されるメッセージ情報部 4 0 D とを含んで構成される。10

【 0 0 3 9 】

なお、データ構造部 4 0 A に含まれるデータ構造には、X M L スキーマ言語に合致したそれぞれのデータ型（文字型、整数型等）、とりうる値の範囲、及び出現回数の指定を行うことができる。データ構造部 4 0 A に記述するデータ構造は図 4 のように親子関係を持つても良いし、持っていないなくてもよい。また他の親構造を子構造として取り込んでも良い。

【 0 0 4 0 】

また、識別情報部 4 0 B のデバイス I D (D I D) は管理者によって与えられるものであっても良いし、ネットワークアドレスのような一意の識別子でもよい。20

【 0 0 4 1 】

また、プロトコル情報部 4 0 C にはデータ構造をどのような通信プロトコルを使用して配信するのかが記述される。プロトコル情報部 4 0 C にはプロトコルを識別するためのプロトコル名と、使用ネットワークにおける実際の伝送に必要なパラメータが記述される。例えば H T T P (H y p e r T e x t T r a n s m i s s i o n P r o t o c o l) であれば U R L である。

【 0 0 4 2 】

本実施形態の統合化システム 1 における処理は、図 1 0 に示すように、入出力機器 1 0 が自己説明型 X M L 文書をミドルソフトウェアシステムに送信する告知フェーズ S 1 0 0 と、入出力機器 1 0 とミドルソフトウェアシステム間が実際の入出力データを送受信する入出力フェーズ S 2 0 0 とから構成される。以下、これら告知フェーズ及び入出力フェーズを順に説明する。30

【 0 0 4 3 】

図 5 には本実施形態に係る告知フェーズにおける処理フローを示す。まず、工作機械や生産管理パッケージの管理者は、入出力機器 1 0 のデバイスエージェントモジュール 1 2 に一意の D I D を設定する（図 5 の S 1 ）。デバイスエージェントモジュール 1 2 は独自のユーザインターフェースを持って管理者から入力された D I D を保持しても良い。また、デバイスエージェントモジュール 1 2 は D I D を、予めコンピュータシステム上のメモリに格納したり、ある種のデータベースに格納してもよい。40

【 0 0 4 4 】

次に、工作機械や生産管理パッケージの管理者はデバイスエージェントモジュール 1 2 を動作させ（S 2）、デバイスエージェントモジュール 1 2 は対象の工作機械やアプリケーションの定義情報から図 3 及び図 4 のような自己説明型 X M L を生成する（S 3）。この定義情報とは、ハードディスク装置に格納されたテキストファイルであったり、コンピュータシステム上のメモリに格納されていたり、ある種のデータベースに格納されていてもよい。

【 0 0 4 5 】

生成された自己説明型 X M L は、T C P / I P 等のプロトコルを用いてインターネットや L A N を経由してリスナープログラム 1 6 に配信される（S 4）。リスナープログラム 150

6は工作機械や生産管理パッケージといった入出力機器10の内部に存在していてもよく、この場合には、内部プロセス間通信機構によって配信される。デバイスエージェントモジュール12とリストナープログラム16間にファイアウォールが存在する場合にはHTTPプロトコル等のファイアウォール透過プロトコルによってラッピングされてもよい。

【0046】

リストナープログラム16は、受信した自己説明型XML文書中の識別情報部のDIDと、入出力機器10のネットワークアドレスと、リストナープログラム16自身の識別子(リストナーハッシュ)とをディレクトリサービスプログラム20に登録する(S5)。ディレクトリサービスプログラム20は、DIDを登録キーにして入出力機器10のネットワークアドレスとリストナーハッシュとを自身のデータベースに登録する(S6)。登録された情報はディレクトリ管理プログラム24のユーザインターフェースから参照したり、削除したりすることができます。

【0047】

さらにリストナープログラム16は、受信した自己説明型XML文書をジェネレータプログラム22に引き渡す(S7)。

【0048】

ジェネレータプログラム22は自己説明型XML文書から一方向ハッシュ関数を用いて、一意のハッシュ値を生成する(S8)。ジェネレータプログラム22は自己説明型XML文書をDOM(Document Object Model)に基づいたオブジェクトとしてメモリに展開する。そして、DIDを検索キーとしてハッシュ値を検索し(S9)、ハッシュ値がDIDと合致しない場合(S10で否定判断の場合)に、以下のようにしてメッセージ処理クラスオブジェクトを生成する(S11)。

【0049】

ここで、図6を用いてジェネレータプログラム22による解析処理及びメッセージ処理クラスオブジェクト生成処理を説明する。ジェネレータプログラム22は、図6に示す以下の処理T1~T6を実行する。

【0050】

即ち、ジェネレータプログラム22は、情報解析処理T1において、自己説明型XML文書を解析してメモリ上に各種の解析情報(識別情報、プロトコル情報、メッセージ情報、データ構造)を作成する。そして、ジェネレータプログラム22は、メッセージ依存関係解析処理T2において、メッセージ情報の情報同士の依存関係を解析し、その際にデータ構造が関係するか否かを調べる。

【0051】

ここで、データ構造が関係する場合には、ジェネレータプログラム22は、データ依存関係解析処理T3において、データ構造の依存関係を解析する。また、ジェネレータプログラム22は、メッセージデータ型解析処理T4において、メッセージ情報からメッセージ情報に含まれるデータの型の構成情報を解析してデータ構成情報を作成する。

【0052】

また、データ構造が関係する場合には、ジェネレータプログラム22は、データ構造データ型解析処理T5において、データ構造のデータの型の構成情報を解析してデータ構成情報を追加する。更に、ジェネレータプログラム22は、データチェック情報作成処理T6において、データ構成情報と識別情報からメッセージ処理クラスオブジェクトを動的に生成する。

【0053】

以上のようにしてメッセージ処理クラスオブジェクトを生成した後、図5のようにジェネレータプログラム22は、動的に生成したメッセージ処理クラスオブジェクトとハッシュ値とを、識別情報部40Bに格納されているDIDを登録キーにして登録するべく、メッセージ処理クラスオブジェクト、ハッシュ値及びDIDを一組にしてディレクトリサービスプログラム20に渡す(図5のS12)。

【0054】

10

20

30

40

50

そして、ディレクトリサービスプログラム20は、DIDを登録キーにしてメッセージ処理クラスオブジェクトとハッシュ値を自身のデータベースにデバイス情報として登録する(S13)。登録されたデバイス情報はディレクトリ管理プログラム24のユーザインターフェースから参照したり、削除したりすることができる。

【0055】

ワークベンチプログラム26は、登録されたメッセージ処理クラスオブジェクトから入出力クラスライブラリ34を生成する(S14)。入出力クラスライブラリ34はJava言語によるテンプレートプログラムであり、コンパイル時にEJBプログラム32とリンクすることができる。

【0056】

ここで、図7及び図8を用いてディレクトリ管理プログラム24及びワークベンチプログラム26の処理を補足的に説明する。図7に示すように、ディレクトリ管理プログラム24はユーザインターフェースを利用して、アプリケーションサーバ情報としてアプリケーションサーバ30(図2)のネットワークアドレスと、EJBプログラム情報としてアプリケーションサーバ30上のEJBプログラム32のオブジェクトクラス名とオブジェクトメソッド名とをディレクトリサービスプログラム20に登録することができる。

【0057】

図8に示すように、これらのアプリケーションサーバ情報(アプリケーションサーバ30のネットワークアドレス情報)とEJBプログラム情報(アプリケーションサーバ30上のEJBプログラム32のオブジェクトクラス名とオブジェクトメソッド名)は、EJBクラスオブジェクト名からクラスメソッド名及びアプリケーションサーバ30のネットワークアドレスへのリンクリストとして、ディレクトリサービスプログラム20内のデータベースに保存される。但し、アプリケーションサーバのネットワークアドレスからEJBクラスオブジェクト名へのリンク情報も保存されている。

【0058】

一方、ワークベンチプログラム26は、ディレクトリ管理プログラム24が登録したEJBプログラム情報を読み込み、ユーザインターフェース上にアイコンとして表示できる。ワークベンチプログラム26の操作者は、入出力機器10が出力したデータをどのEJBプログラム32に引き渡すかを定義できる。また、EJBプログラム32のアイコン同士をグラフィカルな線で接続することによって、EJBプログラム32間の実行順序を定義できる。ワークベンチプログラム26が定義したワークフロー情報はディレクトリサービスプログラム20に格納される。

【0059】

次に、図9を用いて入出力フェーズ(図10のS200)を説明する。この入出力フェーズは、図2の受発注パッケージソフトウェア10Dから発生した出力データを計画指示EJBプログラム32Cが処理し、生産管理パッケージソフトウェア10Cへ送出するフェーズである。

【0060】

入出力機器としての受発注パッケージソフトウェア10Dがデータの出力をを行う場合、告知フェーズで送信した自己説明型XML文書の出力構造と合致するデータ構造とプロトコルを使用してデータの出力をを行う。また、出力パケットには、出力データと入出力機器(受発注パッケージソフトウェア10D)のDIDとが含まれている。出力パケットはインターネットやLANを経由してリスナープログラム16が受信する(S21)。出力データを受信したリスナープログラム16はカーネルプログラム18に出力パケットを渡す(S22)。

【0061】

例えば、受発注パッケージソフトウェア10Dは、受注情報と受発注パッケージソフトウェア10Dに定義されたDIDとを、出力データとしてミドルソフトウェアシステムに送信する(S21)。ミドルソフトウェアシステム内のリスナープログラム16は出力データを受信し、カーネルプログラム18に送信する(S22)。カーネルプログラム18は

10

20

30

40

50

、 D I D をキーとしてメッセージ処理クラスを取得する (S 2 3)。具体的には、カーネルプログラム 1 8 は、ディレクトリサービスプログラム 2 0 から D I D をキーとしてメッセージ処理クラスを検索することで (S 2 4)、 J A V A メッセージ処理クラスを得る。

【 0 0 6 2 】

S 2 4 の検索によりメッセージ処理クラスが得られると、カーネルプログラム 1 8 は、当該メッセージ処理クラスをインスタンス化し (S 2 5)、出力データが正常ならば (S 2 6 で肯定判断ならば)、 D I D をキーとして、計画指示 E J B プログラム 3 2 C を呼び出すために必要な情報 (E J B 呼出情報) を取得する (S 2 7)。具体的には、カーネルプログラム 1 8 は、ディレクトリサービスプログラム 2 0 から D I D をキーとして E J B 呼出情報を検索することで (S 2 8)、 E J B 呼出情報を得る。 E J B 呼出情報としては、例えば、計画指示 E J B プログラム 3 2 C のクラスオブジェクト名、クラスメソッド名、及び、計画指示 E J B プログラムが動作するアプリケーションサーバのネットワークアドレスが挙げられる。10

【 0 0 6 3 】

そして、カーネルプログラム 1 8 は、受注情報を引数として計画指示 E J B プログラム 3 2 C を呼び出す (S 2 9)。

【 0 0 6 4 】

計画指示 E J B プログラム 3 2 C は、製品ごとの生産数量を算出し生産基礎データを生成する (S 3 0)。そして、生産基礎データをパラメータとして入出力クラスライブラリ 3 4 C をコールする (S 3 1)。20

【 0 0 6 5 】

入出力クラスライブラリ 3 4 C は、カーネルプログラム 1 8 を呼び出し生産量基礎データを渡す (S 3 2)。また、この際に、入出力クラスライブラリ 3 4 C は、あて先となる生産管理パッケージソフトウェア 1 0 C を特定するための D I D も同時にカーネルプログラム 1 8 に渡す (S 3 2)。

【 0 0 6 6 】

カーネルプログラム 1 8 は、 D I D をキーとしてリスナー I D を取得する (S 3 3)。具体的には、カーネルプログラム 1 8 は、ディレクトリサービスプログラム 2 0 から D I D をキーとしてリスナー I D を検索することで (S 3 4)、 D I D を持った生産管理パッケージソフトウェア 1 0 C が自己説明型 X M L を送信した先のリスナープログラムのリスナー I D を得る。30

【 0 0 6 7 】

そして、カーネルプログラム 1 8 は、当該リスナー I D に該当するリスナープログラム 1 6 を呼び出すと (S 3 5)、当該リスナープログラム 1 6 に生産量基礎データを渡す。このリスナープログラム 1 6 は生産量基礎データを生産管理パッケージソフトウェア 1 0 C に送信する (S 3 6)。

【 0 0 6 8 】

以上説明した本発明の実施形態により、図 2 のミドルソフトウェアシステム 1 6 、 1 8 、 2 0 、 2 2 、 2 4 、 2 6 、 2 8 が、各種の入出力機器 1 0 に関する入出力データ構造、入出力プロコトル及びアクセス手段を表す自己説明型 X M L 文書を自動的に解釈し、入出力機器 1 0 と、バックエンドアプリケーションである E J B プログラム 3 2 との間のデータを中継するため、入出力機器 1 0 又は E J B プログラム 3 2 が通信相手のデータフォーマットやアクセス方法等に適合させるための処理を行う必要がなく、データフォーマットやアクセス方法等が相互に異なる複数のコンピュータシステム、デジタルデバイス、レガシーデバイス等を円滑に統合化することができる。40

【 0 0 6 9 】

なお、上記のような統合化システムにおけるミドルソフトウェアシステム内のプログラムの各々は、複数のコンピュータシステムに分散してインストールされていてもよいし、一台のコンピュータシステムに集中的にインスト - ルされていてもよい。

【 0 0 7 0 】

また、ミドルソフトウェアシステム内のプログラム（リストナープログラム、ジェネレータプログラム、ディレクトリサービスプログラム、ディレクトリ管理プログラム、カーネルプログラム）は、Java言語、C++言語等のオブジェクト指向言語で作成され、磁気ディスクにインストールされている構成を探ることができる。

【0071】

また、ディレクトリ管理プログラムは、ディレクトリサービスプログラムに蓄積された入出力機器のDID、データ構造、使用プロトコル及びネットワークアドレスをデバイス情報としてディレクトリ管理プログラムの画面上のアイコンとして表示する構成を探ることができる。

【0072】

また、ディレクトリ管理プログラムを使用して、ディレクトリサービスプログラムに対してバックエンドアプリケーションのネットワークアドレス、オブジェクトクラス名、オブジェクトメソッド名をバックエンドアプリケーション情報として登録でき、これらをディレクトリ管理プログラムの画面上のアイコンとして表示する構成を探ることができる。

【0073】

また、ワークベンチプログラムはディレクトリサービスプログラムのデバイス情報とバックエンドアプリケーション情報を参照して、アイコンのクリックアンドドラッグ操作により、デバイス情報アイコンとバックエンドアプリケーション情報アイコンを結びつけることができる。このような操作によって入出力機器とバックエンドアプリケーション間の経路制御（ルーティング）が可能になる。

【0074】

また、デバイスエージェントモジュールは、Java言語、C++言語等のオブジェクト指向言語で作成され、ネットワークサーバから供給される様、入出力機器の一部として取り込まれている様、独立したプログラムとして磁気ディスクにインストールされている様の何れの様を採用してもよい。

【0075】

【発明の効果】

以上説明したように、本発明によれば、ミドルソフトウェアシステムが、各種の入出力機器に関する入出力データ構造、入出力プロコトル及びアクセス手段を表す自己説明型XML文書を自動的に解釈し、入出力機器とバックエンドアプリケーション間のデータを中継するため、入出力機器又はバックエンドアプリケーションが通信相手のデータフォーマットやアクセス方法等に適合させるための処理を行う必要がなく、データフォーマットやアクセス方法等が相互に異なる複数のコンピュータシステム、デジタルデバイス、レガシーデバイス等を円滑に統合化することができる。

【図面の簡単な説明】

【図1】代表的な製造業におけるシステム構成例を示す図である。

【図2】本実施形態における統合化システムのシステム構成図である。

【図3】自己説明型XML文書の構成図である。

【図4】自己説明型XML文書のデータ構造を説明するための図である。

【図5】告知フェーズにおける処理を示すフロー図である。

【図6】ジェネレータプログラムにおける解析処理を説明するための図である。

【図7】ディレクトリ管理プログラム及びワークベンチプログラムの処理を示す図である。

。

【図8】EJBクラスオブジェクト名及びEJBクラスメソッド名のリンク構造を示す図である。

【図9】入出力フェーズにおける処理を示すフロー図である。

【図10】統合化システムにおける処理概要を示すフロー図である。

【符号の説明】

1...統合化システム、10...入出力機器、10A...工作機械、10B...生産ライン、10C...生産管理パッケージソフトウェア、10D...受発注パッケージソフトウェア、12...

10

20

30

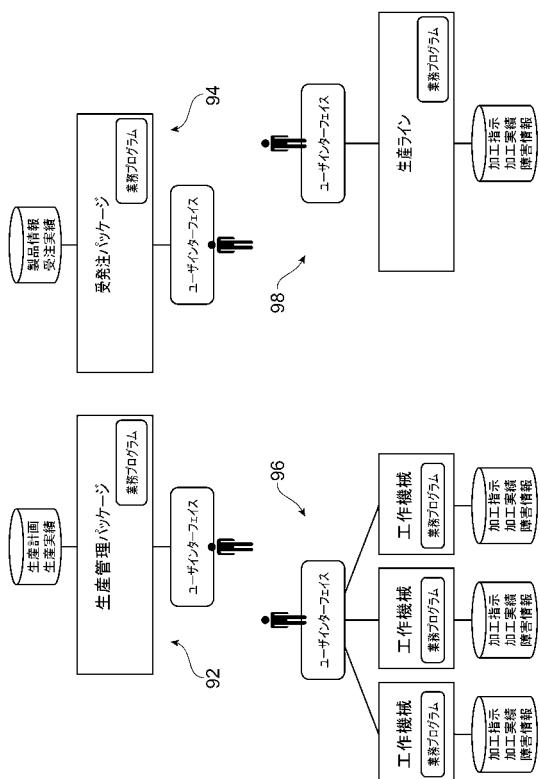
40

50

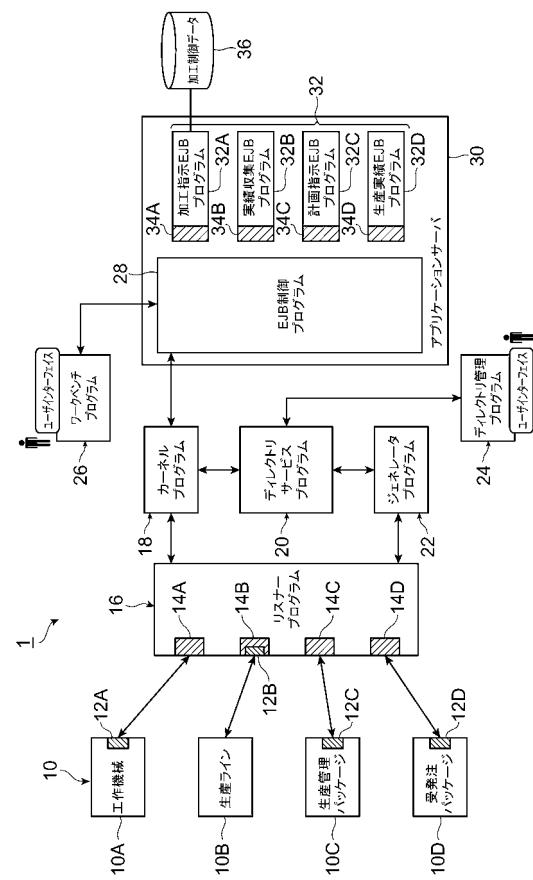
デバイスエージェントモジュール、14...アダプタモジュール、16...リストナープログラム、18...カーネルプログラム、20...ディレクトリサービスプログラム、22...ジェネレータプログラム、24...ディレクトリ管理プログラム、26...ワークベンチプログラム、28...制御プログラム、30...アプリケーションサーバ、32...EJBプログラム、34...入出力クラスライブラリ、36...データベース、40...自己説明型XML文書、40A...データ構造部、40B...識別情報部、40C...プロトコル情報部、40D...メッセージ情報部、92...生産管理システム、94...受発注システム、96...工作機械稼働システム、98...生産ラインシステム、S100...告知フェーズ、S200...入出力フェーズ、T1...情報解析処理、T2...メッセージ依存関係解析処理、T3...データ依存関係解析処理、T4...メッセージデータ型解析処理、T5...データ構造データ型解析処理、T6...データチェック情報作成処理。

10

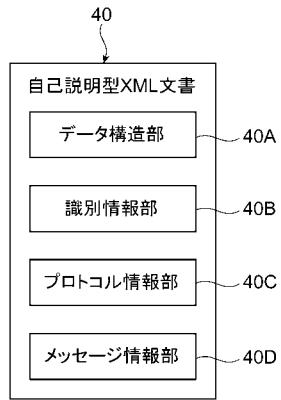
【図1】



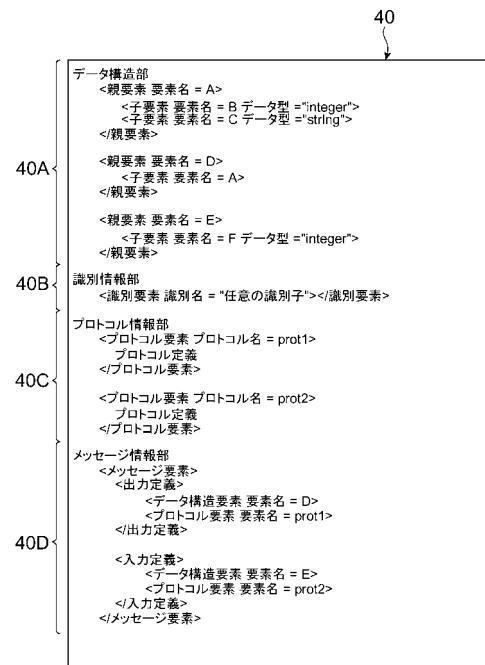
【図2】



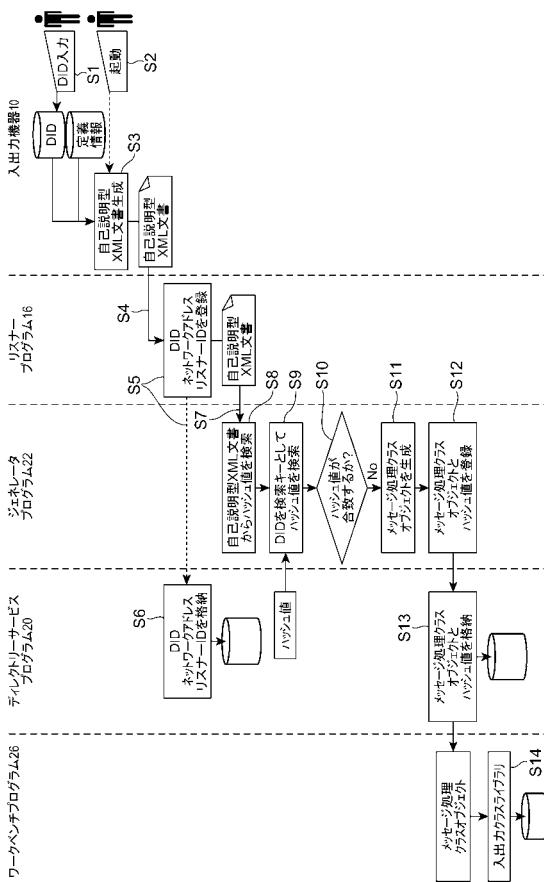
【図3】



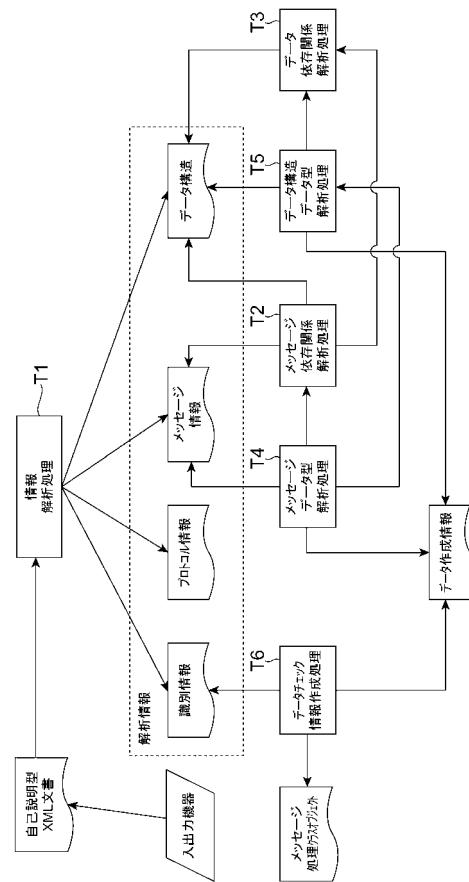
【図4】



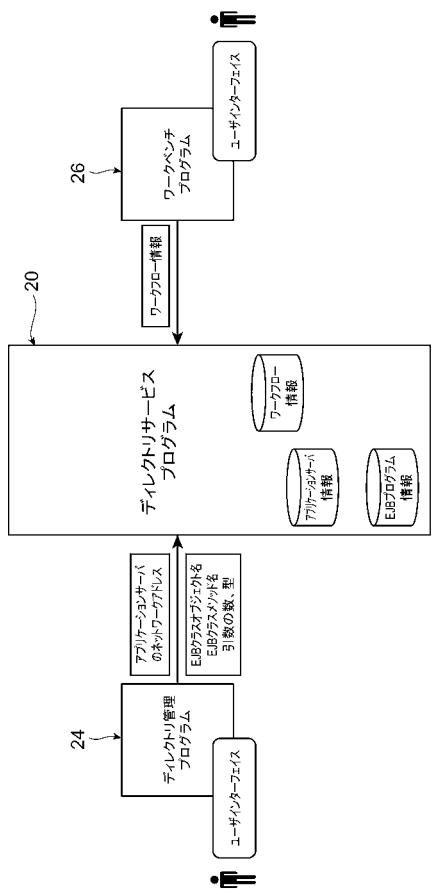
【図5】



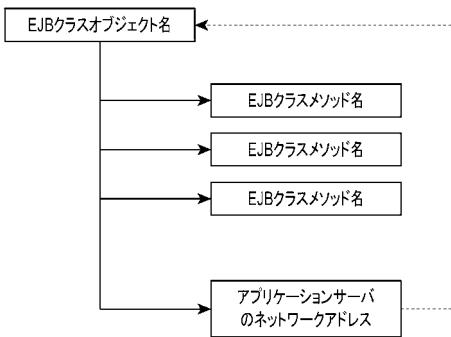
【図6】



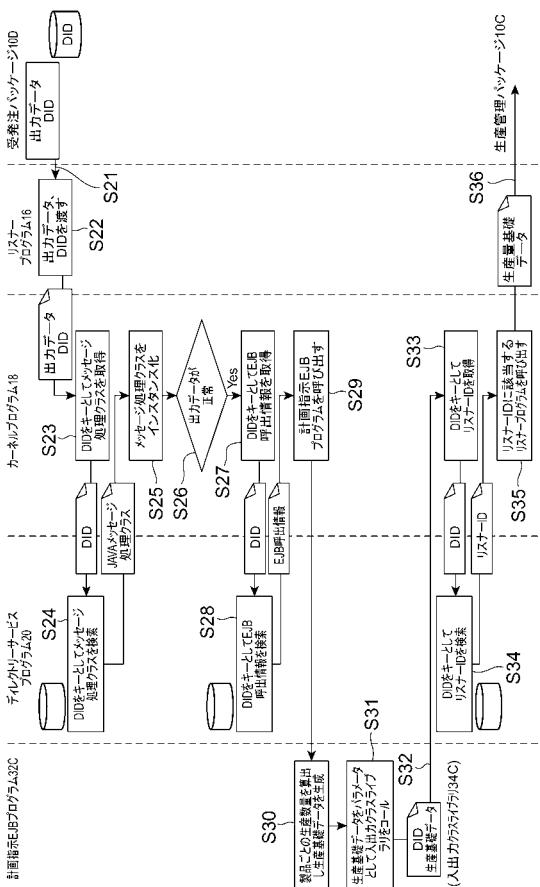
【 図 7 】



【 叁 8 】



〔 四 9 〕



【 図 1 0 】

