

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4691895号
(P4691895)

(45) 発行日 平成23年6月1日(2011.6.1)

(24) 登録日 平成23年3月4日(2011.3.4)

(51) Int.Cl. F I
G06F 9/44 (2006.01) G06F 9/06 620K

請求項の数 5 (全 25 頁)

(21) 出願番号	特願2004-104090 (P2004-104090)	(73) 特許権者	393031586
(22) 出願日	平成16年3月31日 (2004.3.31)		株式会社国際電気通信基礎技術研究所
(65) 公開番号	特開2005-292982 (P2005-292982A)		京都府相楽郡精華町光台二丁目2番地2
(43) 公開日	平成17年10月20日 (2005.10.20)	(74) 代理人	100099933
審査請求日	平成19年3月28日 (2007.3.28)		弁理士 清水 敏
前置審査		(72) 発明者	葦苜 豊
			京都府相楽郡精華町光台二丁目2番地2
			株式会社国際電気通信基礎技術研究所内
		(72) 発明者	中村 哲
			京都府相楽郡精華町光台二丁目2番地2
			株式会社国際電気通信基礎技術研究所内
		(72) 発明者	伊藤 玄
			京都府相楽郡精華町光台二丁目2番地2
			株式会社国際電気通信基礎技術研究所内

最終頁に続く

(54) 【発明の名称】 データ処理システム設計装置及びそのためのコンピュータプログラム

(57) 【特許請求の範囲】

【請求項1】

コンピュータ上で動作可能なモジュールを組合せることによりデータ処理システムを設計するためのデータ処理システム設計装置であって、

各モジュールの仕様、及びオプション又はパラメータ情報を含むモジュール情報を記憶するためのモジュール情報記憶手段と、

前記モジュール情報記憶手段に記憶された各モジュールの仕様、及びオプション又はパラメータ情報に基づいて、前記データ処理システム内の複数のモジュール間の接続関係をユーザが定義すること、及び当該データ処理システム内の各モジュールのオプション又はパラメータ値をユーザが指定することを補助し、1又は複数のプロセスの定義と、定義されたモジュールの入力と出力との間の接続関係と、指定された各モジュールのオプション又はパラメータ値とを含むフロー定義情報を生成するためのフロー定義生成手段とを含み、

前記1又は複数のプロセスの定義の各々は、当該プロセスを実行するホスト名と、当該プロセスを実現するプログラム名と、当該プロセス内で実行されるべきモジュールに関する情報とを含み、

さらに、前記フロー定義生成手段により生成されたフロー定義情報を記憶するためのフロー定義情報記憶手段を含む、データ処理システム設計装置。

【請求項2】

さらに、

前記フロー定義生成手段により生成されたフロー定義情報のうちの任意のものをユーザに選択させるフロー定義情報選択機能と、

前記フロー定義情報選択機能により選択されたフロー定義情報、及び前記モジュール情報記憶手段に記憶されたモジュール情報に従って、コンピュータシステム上で前記選択されたフロー定義情報に従うデータ処理システムを実現するために必要なモジュールのインスタンスを生成するインスタンス生成機能と、

前記インスタンス生成機能により生成されたモジュールのインスタンス間で、前記選択されたフロー定義情報内の、前記モジュール間の接続関係に従ってデータを配信するデータ配信機能とを有する、請求項 1 に記載のデータ処理システム設計装置。

【請求項 3】

さらに、前記インスタンス生成機能により生成された各モジュールのインスタンスに、前記選択されたフロー定義情報内の、それぞれのモジュールのインスタンスのためのオプション又はパラメータ値を配信する機能を有する、請求項 2 に記載のデータ処理システム設計装置。

【請求項 4】

前記モジュールの仕様は、当該モジュールの処理タイプと、当該モジュールを実現するプログラムを特定する情報と、当該モジュールの入力のデータ型の集合及び出力のデータ型の集合を特定するデータ型情報とを含み、

前記フロー定義生成手段は、

前記データ処理システムにおいて必要とされる処理タイプのモジュールをユーザの選択に従って特定するモジュール特定機能と、

前記モジュール特定手段により特定されたモジュールの任意の二つの間の、入力と出力との間の接続をユーザの指示に従って定義する接続特定機能と、

前記接続特定機能により互いに接続される二つのモジュールの一方の入力のデータ型の集合と、他方の出力のデータ型の集合との積集合に属するデータ型のうちの一つをユーザに選択させ、当該接続により表されるモジュール間で移動されるデータの型を特定して前記接続特定機能により特定された接続の属性に追加するデータ型特定機能と、

前記モジュール特定機能により特定されたモジュールに関する情報と、前記データ型特定機能により特定された接続とに基づいて、前記フロー定義を生成する機能とを有する、請求項 1 ~ 請求項 3 のいずれかに記載のデータ処理システム設計装置。

【請求項 5】

コンピュータ上で動作可能なモジュールを組合せることによりデータ処理システムを設計するためのデータ処理システム設計装置としてコンピュータを動作させるコンピュータプログラムであって、

前記コンピュータは、

各モジュールの仕様、及びオプション又はパラメータ情報を含むモジュール情報を記憶するためのモジュール情報記憶手段と、

1 又は複数のプロセスの定義と、定義されたモジュールの入力と出力との間の接続関係と、指定された各モジュールのオプション又はパラメータ値とを含むフロー定義情報を記憶するためのフロー定義情報記憶手段とを含み、

前記 1 又は複数のプロセスの定義の各々は、当該プロセスを実行するホスト名と、当該プロセスを実現するプログラム名と、当該プロセス内で実行されるべきモジュールに関する情報とを含み、

前記コンピュータプログラムは、前記コンピュータを、

前記モジュール情報記憶手段に記憶された各モジュールの仕様、及びオプション又はパラメータ情報に基づいて、前記データ処理システム内の複数のモジュール間の接続関係をユーザが定義すること、及び当該データ処理システム内の各モジュールのオプション又はパラメータ値をユーザが指定することを補助し、前記フロー定義情報を生成して前記フロー定義情報記憶手段に記憶させるためのフロー定義生成手段として機能させる、コンピュータプログラム。

10

20

30

40

50

【発明の詳細な説明】**【技術分野】****【0001】**

この発明は、コンピュータ上で複数のモジュールを組合せることによりデータ処理システムを構築するモジュール型システムの設計及び実行方式に関し、特に、様々なモジュールを所定の目的にあわせて自由に組合せることにより、容易にモジュール型のデータ処理システムを構築し、かつ簡便に動作させて結果を得ることが可能なシステムの設計装置及びプログラムに関する。

【背景技術】

10

【0002】

実時間の音声認識システム等の構築過程においては、音声認識のための処理の設定を様々に変えて、認識率等の性能を確認したいことがある。そのために例えばオプション値を変えて同じプログラムを動作させるだけでなく、あるプログラムを全く別のプログラムに置換したり、処理の流れを自由に変更して実験したりできれば便利である。

【0003】

そのための一つの手段は、それぞれ特有の機能を持つ複数のモジュールを予め準備し、それらを目的にあわせて適切な形で組合せることが可能な、いわゆるモジュール型のシステムである。

【発明の開示】

20

【発明が解決しようとする課題】**【0004】**

しかし、従来のモジュール型システムは、主としてプログラム又は装置の設計段階で採用されている。すなわち、目的にあわせて既存のモジュールを選択し、それらを実際のシステムの条件にあわせて改造することにより、最終製品を製造するもので、エンドユーザが色々なモジュールを自由に組合せてシステムを組立て、テストデータを用いてそうしたシステムの動作を確認するような仕組みは提供されていない。

【0005】

こうした問題は、音声認識のように多量のデータを処理する必要があり、またデータ処理にも多くのバリエーションが存在する場合に特に重要であるが、音声認識にかぎらず種々のデータ処理においても発生し得る。

30

【0006】

それゆえに本発明の目的は、モジュールを自由に組合せ、かつ容易に動作を確認できる、モジュール型システムの設計に適したデータ処理システム設計装置を提供することである。

【課題を解決するための手段】**【0007】**

本発明の第1の局面に係るデータ処理システム設計装置は、コンピュータ上で動作可能なモジュールを組合せることによりデータ処理システムを設計するための装置である。この装置は、各モジュールの仕様、及びオプション又はパラメータ情報を含むモジュール情報を記憶するためのモジュール情報記憶手段と、モジュール情報記憶手段に記憶された各モジュールの仕様、及びオプション又はパラメータ情報に基づいて、データ処理システム内の複数のモジュール間の接続関係をユーザが定義すること、及び当該データ処理システム内の各モジュールのオプション又はパラメータ値をユーザが指定することを補助し、定義されたモジュール間の接続関係と、指定された各モジュールのオプション又はパラメータ値とを含むフロー定義情報を生成するためのフロー定義生成手段と、フロー定義生成手段により生成されたフロー定義情報を記憶するためのフロー定義情報記憶手段とを含む。

40

【0008】

好ましくはこの装置はさらに、フロー定義生成手段により生成されたフロー定義情報のうちの任意のものをユーザに選択させるためのフロー定義情報選択手段と、フロー定義情

50

報選択手段により選択されたフロー定義情報、及びモジュール情報記憶手段に記憶されたモジュール情報に従って、コンピュータシステム上で選択されたフロー定義情報に従うデータ処理システムを実現するために必要なモジュールのインスタンスを生成するためのインスタンス生成手段と、インスタンス生成手段により生成されたモジュールのインスタンス間で、選択されたフロー定義情報に従ってデータを配信し処理させるためのデータ配信手段とを含む。

【0009】

さらに好ましくは、この装置は、インスタンス生成手段により生成された各モジュールのインスタンスに、選択されたフロー定義情報内の、それぞれのモジュールのインスタンスのためのオプション又はパラメータ値を配信するための手段をさらに含む。

10

【0010】

モジュールの仕様は、当該モジュールの処理タイプと、当該モジュールを実現するプログラムを特定する情報と、当該モジュールの入力のデータ型の集合及び出力のデータ型の集合を特定するデータ型情報とを含んでもよい。フロー定義生成手段は、データ処理システムにおいて必要とされる処理タイプのモジュールをユーザの選択に従って特定するためのモジュール特定手段と、モジュール特定手段により特定されたモジュールの任意の二つの間の接続をユーザの指示に従って定義するための接続特定手段と、接続特定手段により互いに接続される二つのモジュールの一方の入力のデータ型の集合と、他方の出力のデータ型の集合との積集合に属するデータ型のうちの一つをユーザに選択させ、当該接続により表されるモジュール間で移動されるデータの型を特定して接続特定手段により特定された接続の属性に追加するためのデータ型特定手段と、モジュール特定手段により特定されたモジュールに関する情報と、データ型特定手段により特定された接続とに基づいて、フロー定義を生成するための手段とを含んでもよい。

20

【0011】

本発明の第2の局面に係るコンピュータプログラムは、コンピュータにより実行されると、上記したいずれかのデータ処理システムとして当該コンピュータを動作させる。

【発明を実施するための最良の形態】

【0012】

- 構成 -

以下、本発明の一実施の形態に係るシステムについて説明する。本システムは、利用可能なプログラムモジュールの中から必要なモジュールを選択し、モジュール間のデータの流に依りてウェブブラウザの画面上でモジュール間を線（矢印）で結ぶことにより、モジュールの組合せによるシステムを設計し、かつ動作させることが可能な、音声認識を行なうためのシステムである。

30

【0013】

本システムでは、音声をリアルタイムで処理することを目的に、音声データを10 msec程度の時間間隔で所定の長さの単位（これをフレームと言う）に分割して、フレーム単位で各モジュールの間でデータを受け渡ししながら処理を実行する。フレーム単位のデータは、あるモジュールに入力され、そのモジュールで当該フレームに対する処理を行なった後に、次のモジュールに渡される。このフレーム単位のデータを本明細書では「イベントメッセージ（以後イベントと略す）」という。

40

【0014】

イベントには、データの種類に応じていくつかの型がある。例えば、生の音声データの場合はWAVE型、特徴パラメータの場合はFEATURE型等、である。従ってモジュールの入出力においては、そこに入力されるデータ、又はそこから出力されるデータがどのような型かを明確に定義しておく必要がある。データの型に応じて決まるイベントの型を、本明細書では「イベントタイプ」と呼ぶ。

【0015】

そして、システムを組む場合には、目的に従ってモジュールを選択し、どのモジュールのどの出力をどのモジュールのどの入力に与えるかを、データの型を考慮しながら決定す

50

る必要がある。このように、モジュール間のデータの流を記述したものを本明細書ではフロー定義と呼び、典型的にはコンピュータ読取可能な形式で記憶装置に格納される。本システムでは、モジュール定義をサーバに格納しておき、ウェブブラウザでこのモジュール定義を読み込んで、ウェブブラウザ上でシステム設計を行なう。システム設計により得られるものは、モジュール間の接続及びそれに伴うオプション/パラメータ値と、データの流とを定義したデータである。本明細書では、このデータをフロー定義と呼び、それをファイルにしたものをフロー定義ファイルと呼ぶ。

【0016】

図1に、本発明の一実施の形態に係る音声認識データ処理システム50の機能的なブロック図を示す。図1を参照して、この音声認識データ処理システム50は、前述したようなモジュール定義ファイル82を予め記憶し、ウェブサーバを通じて、ユーザによるフロー定義の保守・及び実行等の制御を行なうためのフロー定義実行管理装置60と、フロー定義実行管理装置60に記憶されているモジュール定義ファイル82を用いて、ウェブブラウザ経由でフロー定義の作成を行ない、作成したフロー定義をフロー定義実行管理装置60に格納させる処理と、作成したフロー定義に従い、各モジュールを組合せて得られるシステムをフロー定義実行管理装置60に実行させる指示を与えるためのフロー定義作成・実行指示装置62とを含む。なお本実施の形態では、フロー定義実行管理装置60はサーバ装置上で動作し、フロー定義作成・実行指示装置62はクライアント装置上で動作するが、図面ではその詳細は省略する。

【0017】

音声認識データ処理システム50は、1又は複数個のコンピュータシステムから構成されており、それらシステムに備えられた記憶装置には、各モジュールをコンピュータプロセス上で実行させるための一群のプロセス用プログラム66が記憶されている。フロー定義実行管理装置60は、これらコンピュータハードウェアとプロセス用プログラム66とを用いて、フロー定義作成・実行指示装置62により設計された、モジュール100～110からなるデータ処理システム64によるデータ処理を実行させる。フロー定義実行管理装置60によるデータ処理及びその制御の詳細については後述する。

【0018】

フロー定義実行管理装置60は、前述したとおり、フロー定義実行管理装置60で実行可能なモジュールの仕様をそれぞれ定義したモジュール定義ファイル82と、モジュール定義ファイル82を用いてフロー定義作成・実行指示装置62によって定義されたフロー定義を格納するフロー定義ファイル86と、処理のための入力データを格納する入力ファイル90及び出力結果を格納する出力ファイル88とを含む。入力ファイル90及び出力ファイル88はいずれも利用者により選択される。特に出力ファイル88は、フロー定義に従ったデータ処理を実行するとき確保される。

【0019】

フロー定義実行管理装置60はさらに、複数種類の入力ファイルに関する情報を格納する入力ファイル情報記憶部84と、フロー定義作成・実行指示装置62との間でモジュール定義ファイル82及びフロー定義等の情報をウェブ経由で交換するための管理プログラム80と、フロー定義作成・実行指示装置62によってシステム64の動作の実行が指示されると管理プログラム80によって起動され、データ処理システム64をコンピュータ資源を用いて実現し、入力ファイル90からの入力を与えてデータ処理させ、その結果を出力ファイル88に出力させる処理を制御するためのモジュールマネージャ92とを含む。

【0020】

図1においては、データ処理システム64は複数のモジュール100, 102, 104, 106, 108及び110を含んでいる。実際には、モジュール定義はクラス化されており、モジュール100, 102, 104, 106, 108及び110は、モジュールマネージャ92の制御によってデータ処理システム64を実現するプロセス内に生成される、各モジュールのオブジェクトクラスのインスタンスである。従って、同一のモジュール

10

20

30

40

50

定義から、同一のプロセス又は別々のプロセス内に複数のモジュールのインスタンスが生成され動作することもある。

【 0 0 2 1 】

図 2 は、モジュール定義ファイル 8 2 の内容を示す。図 2 を参照して、モジュール定義ファイル 8 2 は、内部的なモジュール名と、フロー定義作成・実行指示装置 6 2 の画面に表示するためのモジュール名（表示用）と、フロー定義作成・実行指示装置 6 2 の画面にモジュールに関する説明を表示するためのテキストからなるモジュール説明（表示用）と、作者名と、バージョン情報とを含む。

【 0 0 2 2 】

モジュール定義ファイル 8 2 はさらに、このモジュールのタイプを内部的に区別するためのモジュールタイプと、フロー定義作成・実行指示装置 6 2 の画面にモジュールタイプを表示するために使用するモジュールタイプ（表示用）と、当該モジュールを使用可能なユーザ情報からなるセキュリティ制限情報とを含む。モジュールタイプは、モジュールの基本的な機能を示すものであり、本実施の形態のように音声認識の場合には、発話区間切出し（これを以下「EPD」と呼ぶ。）、特徴抽出、CMS（平均ケプストラムによる正規化処理）、特徴パラメータ編集、デコーダ、再評価、結果出力等がある。

【 0 0 2 3 】

モジュール定義ファイル 8 2 はさらに、このモジュールが処理できるイベントとして、受取イベントタイプと、出力イベントタイプと、入出力イベント関連情報と、モジュールの枝番とを含む。受取イベントタイプ及び出力イベントタイプは、それぞれ当該モジュールで可能なイベントタイプの集合である。

【 0 0 2 4 】

モジュール定義ファイル 8 2 はさらに、このモジュールのオプション情報を含む。同種のオプションは、同一のセクションにまとめられる。ただし、オプションによってはセクションに属さないものもある。従ってモジュール定義ファイル 8 2 は、0、1、又は複数のセクションを含みうる。セクションに属さないオプションは、セクション内ではなくセクション外に記述されているが、記載の仕方は、後述するセクション内のオプションのものと同様である。

【 0 0 2 5 】

各セクションは、内部的なセクション名と、ユーザに表示するためのセクション名とを含む。各セクションはさらに、1又は複数のオプションを含む。

【 0 0 2 6 】

各オプションは、内部的にオプションを区別するためのオプション名と、ユーザに対してオプションを表示する際の表示用オプション名と、フロー定義作成・実行指示装置 6 2 においてこのオプションに関するヘルプを表示する際のヘルプテキストとなるオプションヘルプ（表示用）とを含む。

【 0 0 2 7 】

オプションはさらに、ユーザによる値の指定がなかった場合のオプション値のデフォルト値と、オプション値がとり得る内部的な値を列挙する、有効値フィールドと、オプション値がとり得る内部的な値に対応する、表示用の値を列挙する表示用有効値フィールドとを含む。実際には表示用のテキストは別ファイルにより記憶されており、モジュール定義ファイル 8 2 に記憶されているのは表示用テキストの識別番号である。

【 0 0 2 8 】

モジュール定義ファイル 8 2 はさらに、オプション値が数値であるときの有効範囲を定める有効範囲フィールドを含む。有効範囲フィールドは、その最小値及び最大値と、それぞれ最小値及び最大値を有効範囲に含むか否かを示す二つのフラグとを含む。

【 0 0 2 9 】

モジュール定義ファイル 8 2 はこれらに加えて、オプション値の型（string、integer、double、boolean、file、modelのいずれか）と、オプションを指定できる回数を示す複数指定回数フィールドと、オプションが必須か否かを示す必須フィールドと、オプション

10

20

30

40

50

値が所定の複数の候補から選択するタイプである場合、フロー定義作成・実行指示装置 62 で行なわれるフロー定義作成処理においてその選択のために使用する GUI (グラフィカル・ユーザ・インタフェース) 部品のタイプを示す表示タイプフィールドとを含む。

【0030】

図 3 には、モジュール定義ファイル 82 の具体的な一例を示す。図 3 に示すように、モジュール定義ファイル 82 は、XML (eXtended Markup Language) で記述されている。開始タグ <module> と終了タグ </module> とで囲まれた部分がモジュール定義である。開始タグ <module-info> と終了タグ </module-info> で囲まれた部分がこのモジュールに関するオプション以外の情報を示す。開始タグ <options> と終了タグ </options> に囲まれた部分がオプション全体に関する情報を示す。開始タグ <section> と終了タグ </section> とに 10
囲まれた部分がセクションに関する情報を示し、開始タグ <option> と終了タグ </option> とに囲まれた部分が個々のオプションに関する情報を示す。

【0031】

フロー定義作成・実行指示装置 62 は、各モジュールに関するモジュール定義情報を用いて、利用者がフロー定義を作成する際の画面情報を生成したり、接続関係の制約を課したり、オプションの値の設定を行ったりする際の利用者の補助を行なう。

【0032】

図 4 に、フロー定義ファイル 86 の構成の詳細を示す。フロー定義ファイル 86 は、ユーザが作成したフロー定義及びオプション値等を格納するための XML 形式のファイルであり、一つのフロー定義ファイル 86 が一つのデータ処理を定義する。このフロー定義 20
ファイルは、図 1 に示すモジュールマネージャ 92 の起動時にフロー定義作成・実行指示装置 62 を介して、利用者により選択され指定される。モジュールマネージャ 92 はフロー定義ファイル 86 を読み込み、フロー定義ファイル 86 内に記述されたプロセスを起動し、フロー定義ファイル 86 内に記述されたモジュールのインスタンスを生成し、各モジュールインスタンスに、フロー定義ファイル 86 に記述されたオプション値を配信して設定した後、データ処理システムを起動する。図 5 にはフロー定義ファイル 86 の具体的な一例を示す。以後、図 4 について説明するが図 5 についても適宜参照されたい。

【0033】

図 4 を参照して、フロー定義ファイル 86 は、コンピュータ上でのデータ処理を実行する一つの単位を形成するプロセスに関する情報を含む。プロセスに関する情報は、データ 30
処理システムに含むプロセスの数だけ繰返される。プロセスに関する情報は、プロセス名と、当該プロセスを実行するホスト名と、当該プロセスのためのプログラム名と、当該プロセス内で実行すべきモジュールに関する情報とを含む。

【0034】

モジュールに関する情報は、一つのプロセスに含まれるモジュールの数だけ繰返される。各モジュールに関する情報は、モジュール名と、枝番付のモジュール名と、モジュール識別子と、枝番とを含む。一つのプロセス内において同じモジュールが複数個動作する場合があるため、それらを区別するために枝番が設けられる。

【0035】

フロー定義ファイル 86 はさらに、ファイルに関する情報を含む。ファイルに関する情報 40
は、データ処理システムの入力ファイルに関する情報と、出力ファイルに関する情報とを含む。各ファイルに関する情報は、入出力名、ファイル名、フォーマット、イベントタイプ、データタイプ、及びサイズを含む。これらは、ファイルが複数個ある場合にはその数だけ繰返される。

【0036】

フロー定義ファイル 86 はさらに、システム内のモジュール間の接続関係を示す矢印に関する情報を含む。矢印に関する情報は、データ処理システム内に存在する矢印の数だけ 50
繰返される。

【0037】

矢印に関する情報は、矢印の出所に関する情報と、矢印の行き先に関する情報と、矢印

10

20

30

40

50

が示すイベントに関する情報とを含む。出所に関する情報は、矢印の出所のモジュール名と、この矢印が当該モジュールの何番目の出力からのものかを示す番号とを含む。同様に、行き先に関する情報は、矢印の行き先のモジュール名と、当該矢印の行き先が、行き先モジュールの何番目の入力であるかを示す番号とを含む。イベントに関する情報は、当該イベントのデータの型を示すイベントタイプを含む。

【 0 0 3 8 】

フロー定義ファイル 8 6 はさらに、各モジュールのオプションに関する情報を含む。モジュールオプションに関する情報は、モジュール数だけ繰返される。

【 0 0 3 9 】

モジュールオプションに関する情報は、もしセクションにまとめられているオプションがあれば、セクションに関する情報を含み、各セクションに関する情報は、当該セクションに属するオプションの値に関する情報を含む。モジュールオプションに関する情報はまた、セクションにまとめられていないオプションの値に関する情報も含む。

【 0 0 4 0 】

なお本実施の形態では、モジュールに関する情報のうち、枝番が付されたものに関しては、同じモジュールであって枝番がないもののオプション値との差分のみを記述することにより、データ量を削減している。

【 0 0 4 1 】

図 6 に、フロー定義作成・実行指示装置 6 2 のブロック図を示す。図 6 を参照して、フロー定義作成・実行指示装置 6 2 は、フロー定義実行管理装置 6 0 から受けるモジュール定義ファイル 8 2 (図 1 を参照されたい。) の内容に基づいて、GUI によりフロー定義を作成するとともに、作成されたフロー定義を指定してフロー定義実行管理装置 6 0 に対してデータ処理の実行を指示するためのフロー定義作成・実行指示プログラム 1 4 0 と、モジュール定義ファイル 8 2 の内容から抽出される、モジュールタイプに関する情報を格納するためのモジュールタイプテーブル 1 4 6 と、フロー定義実行管理装置 6 0 から受取ったモジュール定義ファイル 8 2 の内容を格納するためのモジュールテーブル 1 4 8 と、フロー定義作成・実行指示装置 6 2 が接続されているコンピュータシステムに属するホストのリストを格納するためのホストテーブル 1 5 0 とを含む。これらテーブルの詳細については後述する。

【 0 0 4 2 】

フロー定義作成・実行指示装置 6 2 はさらに、ユーザとの間のインタフェースを実現するためのモニタ 1 4 2 並びにキーボード及びポインティングデバイスを含む入力装置 1 4 4 と、モニタ 1 4 2 及び入力装置 1 4 4 を用いた GUI によりユーザにより画面上に作成されるフロー定義のグラフに関する情報を格納するための、ノードテーブル 1 5 2 及びアークテーブル 1 5 4 とを含む。各モジュールは、フロー定義のグラフ中のノードに相当し、矢印はアークに相当する。矢印に関しては、以後「アーク」と呼ぶ。これらテーブルの詳細についても後述する。

【 0 0 4 3 】

図 7 を参照して、フロー定義作成・実行指示プログラム 1 4 0 によるフロー定義の画面及びその手法について説明する。図 7 を参照して、フロー定義作成・実行指示プログラム 1 4 0 によってモニタ 1 4 2 上に表示されるフロー定義画面 1 7 0 は、フロー定義作成領域 1 7 8 と、フロー定義を新規に作成する際に操作する新規ボタン 1 8 0、作成済みのフロー定義を保存する際に操作する保存ボタン 1 8 2、作成済みのフロー定義を削除する際に操作する削除ボタン 1 8 4、作成済みのフロー定義から所定の条件に合致するものを検索する際に操作する検索ボタン 1 8 6、及び作成済みのフロー定義の実行を指示する際に操作する実行ボタン 1 8 8 とを含む。

【 0 0 4 4 】

画面 1 7 0 はさらに、フロー定義中に使用可能なモジュールタイプをアイコンで表示するモジュールパレット 1 7 2 と、アークを作成する際に操作するアークボタン 1 7 6 とを含む。

10

20

30

40

50

【 0 0 4 5 】

画面 170 を用いたフロー定義の作成は、次のような手順で行なう。まず、マウスポインタ 174 で示すように、使用したいモジュールのアイコンをモジュールパレット 172 からフロー定義作成領域 178 上に、ドラッグ・アンド・ドロップして配置する。次に、各モジュールについて、利用可能なモジュールの中でどのモジュールを使用するかを決定し、オプションを設定する。さらに、モジュール間にアークを配置し、モジュール間の入出力関係を定義する。最後に、各アークのイベントタイプを設定する。

【 0 0 4 6 】

例えば画面上で配置されたモジュールの矩形を選択（マウスでクリック）することにより、そのモジュールタイプのモジュールとして利用可能なものの選択画面が表示される。モジュールが選択されると当該モジュールのオプションを設定する画面が表示される。オプションを設定すると、当該モジュールのモジュール定義ファイル 82 に記述されたオプション情報に従ったチェックが行なわれ、誤りがあればエラーメッセージが表示され再入力が促される。

10

【 0 0 4 7 】

アークを選択（アークをマウスでダブルクリック）すると、当該アークのイベントタイプを選択する画面が表示される。ここで表示されるイベントタイプは、アークの出所のモジュールの出カイベントタイプとしてリストされているものからなる集合と、アークの行き先のモジュールの入カイベントタイプとしてリストされているものからなる集合との共通集合（積集合）の要素である。モジュールの組合せからアークのイベントタイプが一意に定まる場合、そのイベントタイプが自動的に設定される。

20

【 0 0 4 8 】

こうした操作には、モジュールタイプテーブル 146、モジュールテーブル 148、及びホストテーブル 150 の情報が使用され、作成されるフロー定義に関する情報はノードテーブル 152 及びアークテーブル 154 に保持される。

【 0 0 4 9 】

画面 170 を用いた操作は、例えば画面上にオブジェクトを配置することでアプリケーションプログラムの骨格を作成するアプリケーション作成ツールにおいて行なわれるものと本質的に変わらない。

【 0 0 5 0 】

図 8 にモジュールタイプテーブル 146 の構成を示す。図 8 を参照して、モジュールタイプテーブル 146 の各エントリは、モジュールタイプと、画面 170 上に当該モジュールタイプを表示する際の表示文字列とを含む。これら情報は、フロー定義実行管理装置 60 から与えられるモジュール定義ファイル 82 から抽出できる。

30

【 0 0 5 1 】

図 9 に、モジュールテーブル 148 の構成を示す。図 9 を参照して、モジュールテーブル 148 の各エントリは、モジュール識別番号（ID）と、当該モジュールのモジュールタイプと、そのモジュールを実行するプロセスのプログラムへのフルパスとを含む。このモジュールテーブル 148 の情報も、モジュール定義ファイル 82 から抽出できる。

【 0 0 5 2 】

図 10 に、ホストテーブル 150 の構成を示す。図 10 に示すように、ホストテーブル 150 は、システム内で利用可能なホスト名をリストしたものである。本実施の形態では、ホストテーブル 150 の内容はフロー定義実行管理装置 60 が管理し、フロー定義作成・実行指示装置 62 に通知する。

40

【 0 0 5 3 】

図 11 に、ノードテーブル 152 の構成を示す。図 11 を参照して、ノードテーブル 152 の各エントリは、ノード ID と、当該ノードを構成するモジュールタイプと、当該モジュールの、画面表示の矩形の左上座標及び右下座標と、このモジュールを実行するプログラムのフルパス名と、実行される実行モジュール ID と、このモジュールが実行されるホストを示す実行マシン名とを含む。

50

【 0 0 5 4 】

図 1 2 に、アークテーブル 1 5 4 の構成を示す。図 1 2 を参照して、アークテーブル 1 5 4 の各エントリは、開始ノード ID と、終了ノード ID と、当該アークのイベントタイプとを含む。

【 0 0 5 5 】

フロー定義の新規作成では、当初、図 1 1 のノードテーブル 1 5 2 及び図 1 2 のアークテーブル 1 5 4 は空であるが、図 7 に示すフロー定義作成領域 1 7 8 上にモジュール及びアークが配置され、モジュールのパラメータ、及びアークのイベントタイプが決定されていくと、これらテーブルの内容が少しずつ特定される。最終的にフローが完成すると、ノードテーブル 1 5 2 及びアークテーブル 1 5 4 の内容が確定し、それらに基づいてフロー定義ファイルの内容を作成できる。

10

【 0 0 5 6 】

フロー定義の修正では、図 1 に示すフロー定義実行管理装置 6 0 から任意のフロー定義を受取り、それらに基づいてノードテーブル 1 5 2 及びアークテーブル 1 5 4 の内容を作成する。以後の動作はフロー定義の新規作成と同様である。

【 0 0 5 7 】

フロー定義によって定められるモジュール間のグラフの例を図 1 3 に示す。これらは、図 1 に示すデータ処理システム 6 4 に相当する。図 1 3 に示すデータ処理システム 2 0 0 は、第 1 のプロセス 2 2 2 と第 2 のプロセス 2 2 4 とを含む。第 1 のプロセス 2 2 2 は、第 1 のモジュール 2 3 0 と、第 2 のモジュール 2 3 2 と、第 3 のモジュール 2 3 4 とを含む。第 2 のプロセス 2 2 4 は、第 4 のモジュール 2 3 6 と、第 5 のモジュール 2 3 8 と、第 6 のモジュール 2 4 0 とを含む。

20

【 0 0 5 8 】

例えば第 1 のプロセス 2 2 2 をあるコンピュータ上で実行し、第 2 のプロセス 2 2 4 を別のコンピュータで実行することで、全体的な処理速度の向上を図ることができる。もちろん、第 1 のプロセス 2 2 2 と第 2 のプロセス 2 2 4 とを同一のコンピュータ上で動作させても問題はない。しかし、これらシステムをフロー定義に基づいてどのように構成し、効率よく動作させるかという問題がある。本実施の形態では、こうしたデータ処理システム 2 0 0 をフロー定義に基づいて構成し、効率よく動作させるため、以下に述べるようなモジュール実行の機構を用いている。

30

【 0 0 5 9 】

図 1 4 に本実施の形態に係るフロー定義実行管理装置 6 0 により実現されるデータ処理システム 2 5 0 のブロック図を示す。このシステム 2 5 0 は、例えば図 1 に示すフロー定義ファイル 8 6 の一つに従って実現されるものであり、図 1 3 に示すシステム 2 0 0 と同様、第 1 のプロセス 2 5 2 及び第 2 のプロセス 2 5 4 を含む。フロー定義ファイル 8 6 は、本例では第 1 のプロセス 2 5 2 及び第 2 のプロセス 2 5 4 をそれぞれ構成するモジュール間のデータの流れを定義している。

【 0 0 6 0 】

本システム 2 5 0 はさらに、第 1 のプロセス 2 5 2 及び第 2 のプロセス 2 5 4、並びに本システム 2 5 0 の外部とに接続され、第 1 のプロセス 2 5 2 及び第 2 のプロセス 2 5 4 内のモジュール間、及びこれらモジュールと本システム 2 5 0 の外部との間のイベントの流れをフロー定義ファイル 8 6 の記述に従って制御するためのモジュールマネージャ 9 2 とを含む。

40

【 0 0 6 1 】

第 1 のプロセス 2 5 2 は、第 1 のモジュール 2 6 0 と、第 2 のモジュール 2 6 2 と、第 3 のモジュール 2 6 4 と、これら 3 つのモジュール 2 6 0、2 6 2 及び 2 6 4 の間、ならびにこれらモジュールとモジュールマネージャ 9 2 との間のイベントの流れをフロー定義ファイル 8 6 の記述に従って制御し、プロセス内のメッセージ通信を管理するためのメッセージマネージャ 2 6 6 とを含む。

【 0 0 6 2 】

50

第2のプロセス254も第1のプロセス252と同様、第4のモジュール270と、第5のモジュール272と、第6のモジュール274と、これら3つのモジュール270、272及び274の間、ならびにこれらモジュールとモジュールマネージャ92との間のイベントの流れをフロー定義ファイル86の記述に従って制御するためのメッセージマネージャ276とを含む。

【0063】

モジュール260～274は、いずれも入力と出力とを一個以上有する。各モジュールは、入力されたデータをそのモジュール特有の機能に従って内部で処理し、結果を出力する機能を持つ。例えば、音声の特徴抽出モジュールは、音声データを入力とし、その特徴パラメータを出力とする。

10

【0064】

モジュールマネージャ92は、本システム250の外部から与えられる入力を受けるための入力モジュール282と、本システム250からの出力を外部に与えるための出力モジュール284と、入力モジュール282、出力モジュール284、メッセージマネージャ266及びメッセージマネージャ276に接続され、フロー定義ファイル86の記述に基づいて、各プロセス252及び254、並びに入力モジュール282及び出力モジュール284の間のイベントの流れを制御するためのメッセージマネージャ280と、図1に示すモジュールマネージャ92からデータ処理システム250の起動指示を受け、データ処理システム250の実行に必要な処理と起動処理、終了処理を実行するための起動・終了制御部286とを含む。メッセージマネージャ280は、イベントの流れを制御することにより、プロセス間のメッセージ通信を管理するものといえることができる。

20

【0065】

図15に、モジュール間で通信されるイベント320の構成を示す。図15を参照して、イベント320は、送信元モジュールIDと、送信先モジュールIDと、イベントタイプと、データの内容とを含む。イベントタイプとは、イベントに含まれるデータの型のことであり、前述したWAVE型、FEATURE型等がその一例である。

【0066】

図16にメッセージマネージャ280の構成をブロック図形式で示す。図16を参照して、メッセージマネージャ280は、メッセージマネージャ280に入力されるイベントの送信先モジュールを決定するため、イベントの送信元モジュールIDとイベントタイプとをキーとして、送信先モジュールのモジュールIDを検索できるような構成を有するフロー管理テーブル292と、どのプロセスにどのモジュールが含まれているかを検索できる構成を持つモジュール管理テーブル294と、起動・終了制御部286から与えられる起動指示に応答し、フロー定義ファイル86の記述に基づいてフロー管理テーブル292及びモジュール管理テーブル294を作成するためのテーブル作成処理部296とを含む。

30

【0067】

メッセージマネージャ280はさらに、第1のプロセス252、第2のプロセス254、入力モジュール282及び出力モジュール284に接続され、第1のプロセス252、第2のプロセス254及び入力モジュール282から与えられるイベントを、フロー管理テーブル292及びモジュール管理テーブル294の記述に基づいて第1のプロセス252、第2のプロセス254、及び出力モジュール284のいずれかに出力するためのメッセージ振分処理部290を含む。

40

【0068】

図17にフロー管理テーブル292の構成の一例を、図18にモジュール管理テーブル294の構成の一例を、それぞれ示す。図17を参照して、フロー管理テーブル292は複数のエントリを含み、各エントリは、送信元モジュールIDと、イベントタイプと、送信先モジュールIDとを含む。これらは、図4に示すフロー定義ファイル86のアーキの記述に相当する。受信したイベントに含まれる送信元モジュールID及びイベントタイプ(図15を参照されたい。)をキーにフロー管理テーブル292を検索することにより、

50

当該イベントをどのモジュールに送信すべきかが、送信先モジュールIDの欄を見ることで判定できる。なお図17に示すフロー管理テーブル292は、図14に示す構成に必要な情報の一部のみを示す。

【0069】

図18を参照して、モジュール管理テーブル294も複数のエントリを含む。各エントリはモジュールに対応するものであり、そのモジュールが属するプロセス名と、そのモジュールのモジュールIDとを含む。図18に示すモジュール管理テーブル294も、図14に示す本システム250の構成に必要な情報の一部のみを示す。

【0070】

次に図14の第1のプロセス252及び第2のプロセス254を構成する各部について説明する。各モジュールの構成については本発明とは直接関係ないので、それらについての詳細な説明は省略する。またメッセージマネージャ266とメッセージマネージャ276とは同じ構成を有しているので、以下、メッセージマネージャ266について説明する。

【0071】

図19を参照して、メッセージマネージャ266は、メッセージマネージャ266に入力されるイベントの送信先モジュールを決定するため、イベントの送信元モジュールIDとイベントタイプとをキーとして、送信先モジュールのモジュールIDを検索できるような構成を有する、図16に示すものと同様形式のフロー管理テーブル302と、このメッセージマネージャ266が管理するプロセス（本実施の形態では第1のプロセス252）にどのモジュールが含まれているかを記述したモジュール管理テーブル300と、図16に示すフロー管理テーブル292を複写してフロー管理テーブル302を作成するためのフロー管理テーブル作成処理部306と、フロー定義ファイル86の記述に基づき、第1のプロセス252に含まれるモジュールIDを含むモジュール管理テーブル300を作成するためのモジュール管理テーブル作成処理部304とを含む。

【0072】

メッセージマネージャ266はさらに、第1～第3のモジュール260、262及び264とモジュールマネージャ92とからイベントを受取り、フロー管理テーブル302とモジュール管理テーブル300とを参照して、当該イベントを第1～第3のモジュール260、262及び264、並びにモジュールマネージャ92のいずれに送信すべきかを判定し判定結果に従って当該イベントを振り分けるためのメッセージ振分処理部308と、メッセージ振分処理部308によってメッセージ振分処理部308の管理するプロセス（本実施の形態では第1のプロセス252）に送信するよう判定されたイベントを一時的にFIFO形式で格納するためのイベントキュー310と、イベントキュー310からイベントを一つずつ読み出し、当該イベントに含まれる送信先モジュールIDに基づいて、当該イベントを第1～第3のモジュール260、262及び264のいずれかに配信するためのプロセス内メッセージ配信部312とを含む。

【0073】

図20にモジュール管理テーブル300の形式を模式的に示す。メッセージマネージャ266が管理する第1のプロセス252の場合、そこに含まれるモジュールは第1のモジュール260、第2のモジュール262、及び第3のモジュール264である。従ってこの場合のモジュール管理テーブル300の内容は図20に示した通りとなる。

【0074】

図21に、モジュールマネージャ92によるデータ処理システム250の制御処理のフローチャートを示す。図21を参照して、モジュールマネージャ92の起動・終了制御部286は、フロー定義ファイル86の指定を伴う実行指示をフロー定義作成・実行指示装置62から受けると、ステップ340で、指定されたフロー定義ファイルを読み込む。そして、ステップ342～346の処理をフロー定義ファイル内に記述された全てのプロセスについて繰り返す。

【0075】

10

20

30

40

50

すなわち、ステップ 3 4 2 では、フロー定義ファイル内に記述されたプロセスを、同じくフロー定義ファイル内の当該プロセスについて記述されたホストに起動する。そして、このプロセスの定義内に記述された全てのモジュールについてステップ 3 4 4 及び 3 4 6 の処理を繰り返す。

【 0 0 7 6 】

ステップ 3 4 4 では、プロセスの定義内に記述されたモジュールのインスタンスを生成する。そしてステップ 3 4 6 で、生成されたモジュールインスタンスに、同じモジュールについてプロセスの定義内に記述されたオプション/パラメータ値を配信する。

【 0 0 7 7 】

以上の処理を、全プロセス及び全モジュールに対し繰り返すことで、データ処理システム 2 5 0 を構成する各モジュールを含むプロセスが、指定されたホスト上に立ち上がる。

10

【 0 0 7 8 】

ステップ 3 4 8 で、図 1 6 に示すテーブル作成処理部 2 9 6 及び図 1 9 に示すフロー管理テーブル作成処理部 3 0 6 に指示を与えて、ステップ 3 4 0 で読込んだフロー管理ファイルの内容に従い、フロー管理テーブル 2 9 2 及び 3 0 2 を作成する。さらにステップ 3 5 0 で、図 1 6 に示すテーブル作成処理部 2 9 6 及び図 1 9 に示すモジュール管理テーブル作成処理部 3 0 4 に指示を与え、フロー定義ファイルの内容に従ってモジュール管理テーブル 2 9 4 及び 3 0 0 を作成する。ステップ 3 5 2 で、先頭モジュール（入力モジュールがある場合には入力モジュール）に開始指示を与え、最初のイベントを発生させこのデータ処理システムの入力に与える。以後は、メッセージマネージャ 2 8 0 及び各プロセス内のメッセージマネージャ 2 6 6 , 2 7 6 によるイベント配信によってシステムが動作し、システム内の各モジュールで処理されたデータが出力ファイルに出力される。

20

【 0 0 7 9 】

ステップ 3 5 4 では、データ処理システム 2 5 0 内に未処理のイベントがあるか否かを判定し、イベントがなくなるまで待ち、未処理のイベントがなくなったタイミングで、ステップ 3 4 2 で起動した全てのプログラムをステップ 3 5 6 で終了させ、自分も終了する。

【 0 0 8 0 】

以上がモジュールマネージャ 9 2 の起動・終了制御部 2 8 6 によるデータ処理システム 2 5 0 の起動・終了制御の処理の概略である。

30

【 0 0 8 1 】

メッセージマネージャ 2 8 0 のメッセージ振分処理部 2 9 0 及びメッセージマネージャ 2 6 6 のメッセージ振分処理部 3 0 8 も、後述するようにコンピュータハードウェア上で実行されるプログラムにより実現される。本実施の形態の場合、メッセージ振分処理部 2 9 0 及び 3 0 8 が受信したイベントの送信元がプロセス内のものか、プロセス外のものか、によって異なる処理を行なう。それぞれを実現するコンピュータプログラムの制御構造について、特にメッセージ振分処理部 3 0 8 について図 2 2 及び図 2 3 を参照して説明する。

【 0 0 8 2 】

図 2 2 は、プロセス内のモジュールからイベントを受信した場合に、図 1 9 に示すメッセージマネージャ 2 6 6 のメッセージ振分処理部 3 0 8 が実行する処理プログラムのフローチャートである。図 1 6 に示すメッセージマネージャ 2 8 0 のメッセージ振分処理部 2 9 0 についても同様である。図 2 2 を参照して、プロセス内モジュールからイベントを受信すると、ステップ 4 3 0 で、当該イベントの送信元モジュール ID とイベントタイプとをキーにフロー管理テーブル 3 0 2 を検索し、該当する全てのアークを取出す。

40

【 0 0 8 3 】

続いて、ステップ 4 3 0 の処理の結果見つかった全てのアークについて、以下の処理を行なう。すなわち、ステップ 4 3 2 で、当該アークの送信先モジュール ID が当該プロセス内のモジュールのものか否かを判定する。送信先モジュール ID が当該プロセス内のモジュールのものであればステップ 4 3 4 に進み、さもなければステップ 4 4 0 に進む。

50

【 0 0 8 4 】

ステップ 4 3 4 では、イベントの送信先に、検索されたアークに含まれる送信先モジュール ID を設定する。送信先モジュールが同じプロセス内なので、ステップ 4 3 6 でこのイベントをイベントキュー 3 1 0 に出力する。

【 0 0 8 5 】

一方ステップ 4 4 0 では、イベントの送信先に、同じく検索されたアークに含まれる送信先モジュール ID を設定する。送信先モジュールがプロセス外に存在しているので、ステップ 4 4 2 でこのイベントをモジュールマネージャ 9 2 に出力する。

【 0 0 8 6 】

ステップ 4 3 4、4 3 6、4 4 0 及び 4 4 2 の処理を、ステップ 4 3 0 で見出された全てのアークに対して行なうことにより、一つのイベントに対し、見つかった数だけのイベントが生成され、送信先モジュールの所在に応じて、それぞれイベントキュー 3 1 0 又はモジュールマネージャ 9 2 に送信されることになる。

【 0 0 8 7 】

図 2 3 は、プロセス外からイベントを受信した場合に、図 1 6 に示すメッセージマネージャ 2 8 0 のメッセージ振分処理部 2 9 0 が実行する処理プログラムのフローチャートである。図 2 3 を参照して、プロセス外からイベントを受信すると、ステップ 4 6 0 で、モジュール管理テーブル 2 9 4 中でイベント中の送信先モジュール ID を検索し、当該送信先モジュールを含むプロセスを決定する。ステップ 4 6 2 では、決定されたプロセスが自分自身が否かを判定する。もし自分自身が送信先であればステップ 4 6 4 で自己のイベントキューに当該イベントを格納する。もし自分自身が送信先でなければ、ステップ 4 6 6 で送信先モジュールを含むプロセスに当該イベントを送信する。

【 0 0 8 8 】

一方、図 1 9 に示すメッセージマネージャ 2 6 6 がプロセス外からイベントを受信した場合、当該イベントの送信先については、プロセス内であることが保証されている。従って当該イベントを図 1 9 に示すメッセージ振分処理部 3 0 8 に格納する。

【 0 0 8 9 】

- 動作 -

図 1 ~ 図 2 3 に示すシステムは、以下のように動作する。動作は、フロー定義の作成と、作成したフロー定義に基づいてデータ処理システム 2 5 0 を動作させる処理との、大きく分けて二つの局面に分かれる。最初に、フロー定義について説明する。

【 0 0 9 0 】

利用者が図 6 に示すフロー定義作成・実行指示プログラム 1 4 0 を起動すると、利用可能なモジュール定義ファイルの内容及び使用可能なホストのリストがフロー定義実行管理装置 6 0 からフロー定義作成・実行指示装置 6 2 に送信される。フロー定義作成・実行指示装置 6 2 は、受信した情報に従ってモジュールタイプテーブル 1 4 6、モジュールテーブル 1 4 8 及びホストテーブル 1 5 0 を作成する。画面には図 7 に示すような画面が表示される。以下では、フロー定義を新規に作成する場合を例にしてフロー定義作成・実行指示装置 6 2 の動作を説明する。

【 0 0 9 1 】

図 7 を参照して、利用者が画面 1 7 0 上で新規ボタン 1 8 0 を押すことで、フロー定義作成領域 1 7 8 が空白となる。モジュールパレット 1 7 2 には使用可能なモジュールタイプがアイコンとモジュールタイプ名により表示される。利用者がモジュールパレット 1 7 2 上のいずれかのアイコンを選び、ドラッグ・アンド・ドロップでフロー定義作成領域 1 7 8 上に配置することにより、図 6 に示すノードテーブル 1 5 2 中に、そのモジュールに対応するエントリが作成される。利用者が新たなモジュールを配置するたびに、ノードテーブル 1 5 2 に新たなエントリが作成される。なお、図 1 1 を参照して、この時点では各エントリのうち、ノード ID、モジュールタイプ、及び座標のみが設定され、他の項目は空白である。

【 0 0 9 2 】

10

20

30

40

50

利用者はさらに、各モジュールの間を矢印（アーク）で接続していく。アークが一つ配置されるたびに、アークテーブル 154 内に対応のエントリが作成される。ただしこの時点では、図 12 に示す情報のうち、開始ノード ID と終了ノード ID のみが格納され、イベントタイプは未定義である。

【 0093 】

モジュールの一つの矩形をクリックすると、モジュール設定フレームが表示される。モジュール設定フレームでは、当該モジュールタイプのうちで使用可能なモジュールの一覧が表示される。この一覧は、図 9 に示すモジュールテーブル 148 から作成できる。表示されるモジュールのうちの一つを選択することで、どのモジュールを使用するかが確定する。さらに、このモジュールをどのホスト上で実行させるかについても指定する。

10

【 0094 】

さらに、使用するモジュールが確定すると、当該モジュールで指定できるオプション/パラメータの入力フレームが画面左下に表示される。各オプション/パラメータを設定すると、モジュール定義情報に従ったチェックが行なわれ、エラーがあればエラー表示が行なわれ、再入力促される。

【 0095 】

ユーザがアークの一つをダブルクリックすると、イベントタイプ指定のためのリストボックスが表示される。リストボックスには、出所側のモジュールの出力イベントの型の集合と、行き先側のモジュールの入力イベントの型の集合との共通集合（積集合）の要素がリストされる。いずれかを選択することにより、当該アークのイベントタイプが定まり、図 12 に示すアークテーブル 154 のイベントの領域に選択されたイベントタイプを示す情報が格納される。なお、使用可能なイベントが一種類しかない場合には、アークのイベントタイプは自動的に設定される。

20

【 0096 】

こうして、全てのモジュールについて使用モジュール、ホスト、及びオプション/パラメータが決定され、各アークのイベントタイプが決定すると、それらに基づいてフロー定義を作成できる。フロー定義はフロー定義作成・実行指示装置 62 からフロー定義実行管理装置 60 に転送され、フロー定義ファイル 86 のうちの一つとしてフロー定義実行管理装置 60 に格納される。

【 0097 】

フロー定義の修正の場合も、最初にフロー定義ファイル 86 のうちの一つを選択して当該ファイルの内容をフロー定義作成・実行指示装置 62 に転送することを除いて、作成の場合と同様である。

30

【 0098 】

作成したフロー定義に従ってデータ処理システム 64 を動作させたい場合には、次のような操作を行なう。図 7 を参照して、「実行」ボタン 188 を押すと、フロー定義ファイルの選択ダイアログが表示される。そのうちの一つを選択することで、フロー定義作成・実行指示装置 62 からフロー定義実行管理装置 60 に対して当該フローファイルに基づくデータ処理の開始が指示される。

【 0099 】

図 1 に示す管理プログラム 80 は、その指示に従いモジュールマネージャ 92 に対して選択されたフロー定義ファイルを特定する情報を与えるとともに、データ処理の開始を指示する。

40

【 0100 】

図 14 を参照して起動・終了制御部 286 は、この指示に回答して、指定されたフロー定義ファイルを読み込み（図 21 のステップ 340）、データ処理システム 250 の各プロセスを、指定されたホスト上に起動する（ステップ 342）。さらに、フロー定義ファイルに記述された各モジュールについて、指定されたプロセス内にインスタンスを生成し（ステップ 344）、フロー定義ファイル内に記述されたオプション/パラメータを配信する（ステップ 346）。その後、図 16 に示すテーブル作成処理部 296、及び図 19 に

50

示すフロー管理テーブル作成処理部306、モジュール管理テーブル作成処理部304に指示を与えて各フロー管理テーブル292及び302、並びにモジュール管理テーブル294及び300を作成させる(図21のステップ348, 350)。以上でデータ処理システム250のセットアップが完了する。

【0101】

次に起動・終了制御部286は、図14に示す入力モジュールに対し、入力ファイルからデータを読み出しイベントとしてメッセージマネージャ280に与えるよう、始動の指示を与える(図21のステップ352)。入力モジュール282は入力ファイルからフレームデータを作成し、イベントとしてメッセージマネージャ280に与える。

【0102】

図16を参照して、メッセージマネージャ280のメッセージ振分処理部290は、フロー管理テーブル292及びモジュール管理テーブル294を参照して、当該イベントの送信先のモジュールを含むプロセスを決定し(図23のステップ460)、イベントを当該プロセスに送信する(ステップ464又は466)。本例では、例えば第1のプロセス252にこのイベントが送信されるものとする。

【0103】

図19を参照して、メッセージマネージャ266のメッセージ振分処理部308は、このイベントがプロセス外からのものなので、当該イベントをイベントキュー310に格納する。プロセス内メッセージ配信部312が、イベントキュー310からイベントを読み出し、送信先モジュールIDにより指定されるモジュールにイベントを配信する。

【0104】

このイベントを受けたモジュールは、結果のイベントをメッセージマネージャ266のメッセージ振分処理部308に返信してくる。このとき、イベントの送信元モジュールID及びイベントタイプには、それぞれ当該モジュールにより値が設定されている。送信先モジュールIDについては設定されていない。

【0105】

メッセージ振分処理部308は、受けたイベントの送信元モジュールIDとイベントタイプとをキーにフロー管理テーブル302を検索し(図22のステップ430)、一致するアークを得る。メッセージ振分処理部308はさらに、モジュール管理テーブル300を参照して、当該送信先モジュールが第1のプロセス252内か否かを判定する(ステップ432)。第1のプロセス252内であれば(すなわち当該モジュールIDがモジュール管理テーブル300内に存在すれば)、当該アークに含まれている送信先モジュールIDをイベントの送信先モジュールIDに設定し(ステップ434)、メッセージ振分処理部308はこのイベントをイベントキュー310に書出す(ステップ436)。

【0106】

送信先が第1のプロセス252外であれば、アークに含まれている送信先モジュールIDをイベントの送信先モジュールIDに設定し(ステップ440)、モジュールマネージャ92にこのイベントを送信する(ステップ442)。

【0107】

モジュールマネージャ92のメッセージマネージャ280では、最初の入力データを受けた場合と同様、フロー管理テーブル292とモジュール管理テーブル294とに基づいてこのイベントの送信先プロセスを決定し、そのプロセスのメッセージマネージャにイベントを送信する。フロー管理テーブル292により、システム外に出力することが示されている場合、メッセージ振分処理部290は出力モジュール284にイベントを出力する。出力モジュール284は、このイベントをさらに外部に出力する。

【0108】

以下、メッセージマネージャ280のメッセージ振分処理部290、並びに第1のプロセス252及び第2のプロセス254のメッセージマネージャ266及び276は、上記したと同様の処理を繰返す。その結果、外部から次々に入力モジュール282に入力されたデータは、それぞれ所定のモジュールによって処理され、結果のイベントが出力モジ

10

20

30

40

50

ルール 284 から出力される。

【0109】

上に説明したシステム 250 では、各モジュールは自己の処理した結果のイベントをどのモジュールに送信するかについては全く意識する必要がない。どのプロセスにおいても、モジュールではなくメッセージマネージャが送信先モジュールを決定し、さらに送信先モジュールが同一プロセス内であればイベントキューを用いて当該イベントを配信し、プロセス外であれば、モジュールマネージャに当該イベントを配信する。イベントの配信を受けたモジュールマネージャが、送信先モジュール ID に基づいて送信先のプロセスを決定し、送信する。

【0110】

従って、モジュールの独立性が極めて高くなり、システム構成に従ってモジュールを変更したりする必要はない。さらに、プロセス間通信については、どのプロセスも送信先のプロセスを意識せず、すべてモジュールマネージャに送信するだけでよい。従って、各プロセス間通信のためのオーバヘッドが少なくなる。

【0111】

以上のように本実施の形態のシステムによれば、モジュール定義に従ってフロー定義を作成し、そのフロー定義を用いて複数のモジュールからなるシステムを効率よく設計し、かつ簡略な操作で実際に動作させることができる。モジュール間の接続関係を変えたり、使用するモジュールを、同じモジュールタイプで別のモジュールによって置換してシステムの性能を比較したりする作業が簡単に行なえる。

【0112】

また、上記したように設計され、かつ動作する各モジュール間の通信を、各プロセス内とモジュールマネージャ内に設けたメッセージマネージャによって管理でき、各モジュールはイベントの入力となるモジュール及びイベントの行き先のモジュール等を意識する必要はない。従って各モジュールの独立性を高め、自由にシステムを組むことが可能になる。また、データはプロセス間を転送されながら各モジュールで処理され、かつ各プロセスではイベントキューによって処理時間の調整を行ないながら正しい順序でデータ処理を行なうことができる。その結果、リアルタイムで、音声認識等の多量のデータ処理を行なうことができる。またモジュールの組合せが自由にできるので、異なる構成のシステムの性能実験等を簡単に、かつ短い時間に行なうことが可能になる。

【0113】

- コンピュータによる実現 -

この実施の形態のシステムは、コンピュータハードウェアと、そのコンピュータハードウェアにより実行されるプログラムと、コンピュータハードウェアに格納されるデータとにより実現される。図 24 はこのコンピュータシステム 530 の外観を示し、図 25 はコンピュータシステム 530 の内部構成を示す。なお、前述したとおりこのシステムは、これら複数のコンピュータをネットワーク等で結合したものによって実現することもできる。

【0114】

図 24 を参照して、このコンピュータシステム 530 は、FD (フレキシブルディスク) ドライブ 552 及び CD-ROM (コンパクトディスク読出専用メモリ) ドライブ 550 を有するコンピュータ 540 と、キーボード 546 と、マウス 548 と、モニタ 542 とを含む。

【0115】

図 25 を参照して、コンピュータ 540 は、FD ドライブ 552 及び CD-ROM ドライブ 550 に加えて、CPU (中央処理装置) 556 と、CPU 556、FD ドライブ 552 及び CD-ROM ドライブ 550 に接続されたバス 566 と、ブートアッププログラム等を記憶する読出専用メモリ (ROM) 558 と、バス 566 に接続され、プログラム命令、システムプログラム、及び作業データ等を記憶するランダムアクセスメモリ (RAM) 560 とを含む。コンピュータシステム 530 はさらに、プリンタ 544 を含んでい

10

20

30

40

50

る。

【0116】

コンピュータ540はさらにローカルエリアネットワーク(LAN)への接続を提供するネットワークアダプタボード568を含む。

【0117】

コンピュータシステム530に、この実施の形態に係るフロー定義作成・実行指示装置62、フロー定義実行管理装置60、又はシステム250(又はその一部の第1のプロセス252、第2のプロセス254、モジュールマネージャ92等)としての動作を行なわせるためのコンピュータプログラムは、CD-ROMドライブ550又はFDドライブ552に挿入されるCD-ROM562又はFD564に記憶され、さらにハードディスク554に転送される。又は、プログラムはネットワークを通じてコンピュータ540に送信されハードディスク554に記憶されてもよい。プログラムは実行の際にRAM560にロードされる。CD-ROM562から、FD564から、又はネットワークを介して、直接にRAM560にプログラムをロードしてもよい。

10

【0118】

このプログラムは、コンピュータ540(又はコンピュータ540と通信可能な他のコンピュータ)にこの実施の形態に係るフロー定義作成・実行指示装置62、フロー定義実行管理装置60、並びにシステム250に含まれる各プロセス252、254、モジュールマネージャ92、各プロセス中で実行されるモジュール等としての動作を行なわせる複数の命令を含む。このために必要な基本的機能のいくつかはコンピュータ540上で動作するオペレーティングシステム(OS)又はサードパーティのプログラム、もしくはコンピュータ540にインストールされる各種ツールキットのモジュールにより提供される。従って、このプログラムはこの実施の形態のシステムを実現するのに必要な機能全てを必ずしも含まなくてよい。このプログラムは、命令のうち、所望の結果が得られるように制御されたやり方で適切な機能又は「ツール」を呼出すことにより、上記した装置及びシステムを実現する命令のみを含んでいればよい。コンピュータシステム530の動作は周知であるので、ここでは繰返さない。

20

【0119】

上記した実施の形態では、モジュール管理テーブルについては各プロセスに用意した。しかし本発明はそのような実施の形態には限定されず、モジュール管理テーブルをシステム内の全てのモジュールで同じものを使用するようにしてもよい。また、上記実施の形態では、モジュール識別子はシステム内で一意であるものとした。しかし本発明はそのような実施の形態には限定されず、プロセス内でのみモジュール識別子が一意になるようにしてもよい。ただしその場合、イベント通信ではプロセス名とモジュール識別子とによって送信元及び送信先を特定する必要がある。

30

【0120】

また、上記した実施の形態では、フロー定義作成・実行指示装置62はブラウザプログラム上で動作するプログラムにより実現されるものであることを前提としている。しかし本発明はそのような実施の形態には限定されず、フロー定義作成・実行指示装置62をブラウザと異なる独自のアプリケーションプログラムで実装してもよい。また、フロー定義作成・実行指示装置62としての機能の実質的な部分を全てフロー定義実行管理装置60に持たせ、フロー定義作成・実行指示装置62ではブラウザプログラム上の表示及びフロー定義のための操作のみを行なうようにしてもよい。

40

【0121】

さらに、フロー定義ファイル86、モジュール定義ファイル82、入力ファイル90等がフロー定義実行管理装置60と同一のコンピュータ上に存在しているものに限定する必要はなく、ネットワークを介してアクセス可能な他のシステム内に存在しているものを利用可能なシステムとして設計してもよい。

【0122】

今回開示された実施の形態は単に例示であって、本発明が上記した実施の形態のみに制

50

限されるわけではない。本発明の範囲は、発明の詳細な説明の記載を参酌した上で、特許請求の範囲の各請求項によって示され、そこに記載された文言と均等の意味及び範囲内でのすべての変更を含む。

【図面の簡単な説明】

【0123】

【図1】本発明の一実施の形態に係る音声認識データ処理システム50のブロック図である。

【図2】モジュール定義ファイルの構成を示す図である。

【図3】モジュール定義ファイルの例を示す図である。

【図4】フロー定義ファイルの構成を示す図である

10

【図5】フロー定義ファイルの例を示す図である。

【図6】フロー定義作成・実行指示装置62のブロック図である。

【図7】フロー定義作成・実行指示装置62の表示画面の模式図である。

【図8】モジュールタイプテーブル146の構成を示す模式図である。

【図9】モジュールテーブル148の構成を示す模式図である。

【図10】ホストテーブル150の構成を示す模式図である。

【図11】ノードテーブル152の構成を示す模式図である。

【図12】アークテーブル154の構成を示す模式図である。

【図13】各々複数のモジュールを含む複数のプロセスからなるシステムの構成例を示す図である。

20

【図14】本発明の一実施の形態に係るシステム250のブロック図である。

【図15】イベント320の構成を示す図である。

【図16】図14に示すモジュールマネージャ92のメッセージマネージャ280のブロック図である。

【図17】フロー管理テーブル292の構成を示す図である。

【図18】モジュール管理テーブル294の構成を示す図である。

【図19】図14に示す第1のプロセス252のメッセージマネージャ266のブロック図である。

【図20】モジュール管理テーブル300の構成を示す図である。

【図21】起動・終了制御部286が実行する処理の制御構造を示すフローチャートである。

30

【図22】プロセス内イベントを受信したときの、各メッセージ振分処理部で実行される処理を示すフローチャートである。

【図23】モジュールマネージャ92のメッセージ振分処理部290がプロセス外イベントを受信したときに実行する処理を示すフローチャートである。

【図24】本発明の一実施の形態に係るシステム250又はその一部のプロセスを実現するためのコンピュータシステムの外観図である。

【図25】図24に示すコンピュータシステムの内部構成を示すブロック図である。

【符号の説明】

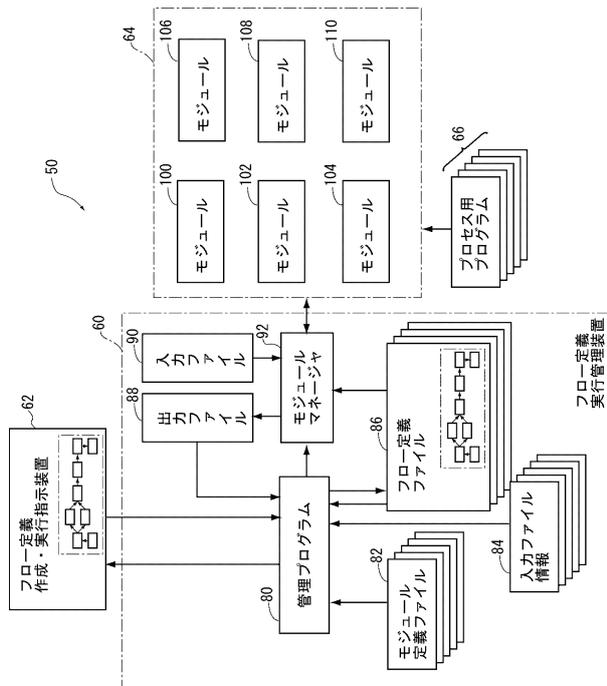
【0124】

40

50 音声認識データ処理システム、60 フロー定義実行管理装置、62 フロー定義作成・実行指示装置、64, 200, 250 データ処理システム、66 プロセス用プログラム、80 管理プログラム、82 モジュール定義ファイル、84 入力ファイル情報、86 フロー定義ファイル、92 モジュールマネージャ、140 フロー定義作成・実行指示プログラム、146 モジュールタイプテーブル、148 モジュールテーブル、150 ホストテーブル、152 ノードテーブル、154 アークテーブル、172 モジュールパレット、178 フロー定義作成領域、222, 252 第1のプロセス、224, 254 第2のプロセス、230, 232, 234, 236, 238, 240, 260, 262, 264, 270, 272, 274 モジュール、266, 276, 280 メッセージマネージャ、286 起動・終了制御部

50

【図1】



【図2】

モジュール定義ファイル

項目	説明
モジュール名	モジュール名
モジュール名 (表示用)	表示用のモジュール名
モジュール説明 (表示用)	-
モジュール説明 (表示用)	表示用のモジュール説明
作者	モジュールの作者
バージョン	モジュールのバージョン
モジュールタイプ	モジュールのタイプ
モジュールタイプ (表示用)	表示用のモジュールのタイプ
セキュリティー制限	-
使用可能ユーザー	使用可能制限
イベント	-
受け取りイベントタイプ	受け取るイベントのタイプ
出力するイベントタイプ	出力するイベントのタイプ
入力イベント関連情報	入力するイベントの関連
枝番	モジュールの枝番
セクション名	モジュール内のセクション名
セクション名 (表示用)	表示用のモジュール内のセクション名と説明
オプション名	オプション名
オプション名 (表示用)	表示用のオプション名
オプションヘルプ (表示用)	表示用のオプションヘルプ
デフォルト値	デフォルト値
有効値	オプション値がとり得る値
有効値 (表示用)	オプション値がとり得る値の表示用ID
最小値	最小値
有効範囲	最小値を範囲に含むかどうかのフラグ
最大値	最大値
有効範囲	最大値を範囲に含むかどうかのフラグ
型	オプション値の型
複数指定回数	複数指定する回数
必須	オプションが必須かどうか
表示タイプ	選択型オプションの表示タイプ

【図3】

```

<module>
  <module-info>
    <name display="モジュール名ID">ATRcep2para</name>
    <description display="モジュール説明ID"/>
    <author>Administrator</author>
    <version>1.0</version>
    <module-type>cep2para</module-type>
    <security-constraint>
      <use>all</use>
    </security-constraint>
    <event>
      <input>ATRD_AAA</input>
      <input>ATRD_BBB</input>
      <output>ATRD_XXX</output>
      <output>ATRD_YYY</output>
      <relation>ATRD_AAA:ATRD_XXX:2</relation>
      <relation>ATRD_BBB:ATRD_YYY</relation>
    </event>
  </module-info>

  <options>
    <option id="CepstrumOrder" display="オプション名ID1" help="ヘルプID1" default="10" min="0" min-include="NO" max="100" max-include="YES" type="integer" plurality="1" is-required="YES" remote-command="NO"/>
    <section id="delta-cepstrum" display="セクション名ID1">
      <option id="window" display="オプション名ID2" help="ヘルプID2" default="1" min="0" min-include="NO" max="10" max-include="YES" type="integer" plurality="1" is-required="YES" remote-command="NO"/>
      <option id="padding" display="オプション名ID3" help="ヘルプID3" default="aaa" effective="aaabbbccc" type="string" plurality="2-4" is-required="NO" remote-command="YES"/>
    </section>
    <option id="rho" display="オプション名ID4" help="ヘルプID4" default="1.0" min="0.0" min-include="YES" max="10.0" max-include="NO" type="double" plurality="1-3" is-required="YES" remote-command="NO"/>
    <option id="OutputParameter" display="オプション名ID5" help="ヘルプID5" max="100" max-include="YES" type="string" plurality="1" is-required="YES" remote-command="NO"/>
    <option id="WindowType" display="オプション名ID6" help="ヘルプID6" max="64" max-include="YES" type="string" plurality="1" is-required="YES" remote-command="NO"/>
  </options>
</module>

```

【図4】

フロー定義ファイル

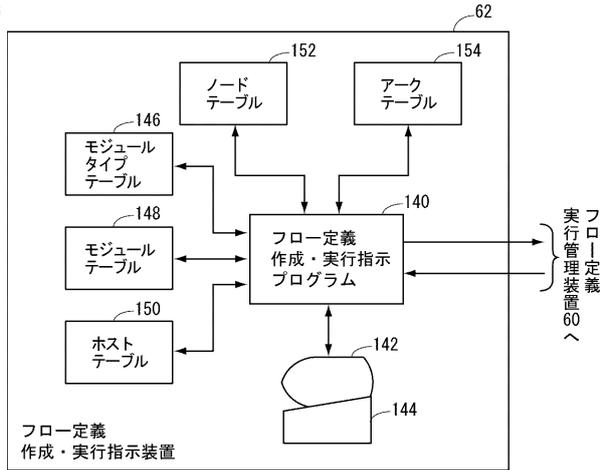
項目	属性	説明	
プロセス	processタグ	processタグ	
プロセス名	name	プロセス名	
ホスト名	host	プロセスを実行するホスト名	
プログラム名	program	プログラム名	
モジュール	moduleタグ	moduleタグ	
モジュール名 (枝番付き)	node name	フロー定義内のalias	
モジュールID	id	モジュールのID	
枝番	number	枝番	
ファイル	fileタグ	fileタグ	
入力	inputタグ	inputタグ	
入力名	name	名称	
入力ファイル名	file	ファイル名	
入力フォーマット	format	フォーマット	
入力イベントタイプ	event-type	イベントタイプ	
入力データタイプ	data-type	データタイプ	
入力サイズ	size	データサイズ	
出力	outputタグ	outputタグ	
出力名	name	名称	
出力ファイル名	file	ファイル名	
出力フォーマット	format	フォーマット	
出力イベントタイプ	event-type	イベントタイプ	
出力データタイプ	data-type	データタイプ	
出力サイズ	size	データサイズ	
矢印	arcタグ	arcタグ	
矢印の出所	fromタグ	fromタグ	
矢印の出所名	name	アークの出所名	
矢印の出所での番号	number	アークの出所での番号	
矢印の行き先	toタグ	toタグ	
矢印の行き先名	name	アークの行き先名	
矢印の行き先での番号	number	アークの行き先での番号	
矢印のイベント	eventタグ	eventタグ	
矢印のイベントタイプ	type	矢印のイベントタイプ	
モジュール数だけ繰返す	モジュール	moduleタグ	
	モジュール名 (枝番付き)	node name	フロー定義内のalias
	モジュールID	id	モジュールのID
	枝番	number	枝番
	セクションID	id	セクションID
	オプション	オプション	オプション値
	オプションID	id	オプションID

【図5】

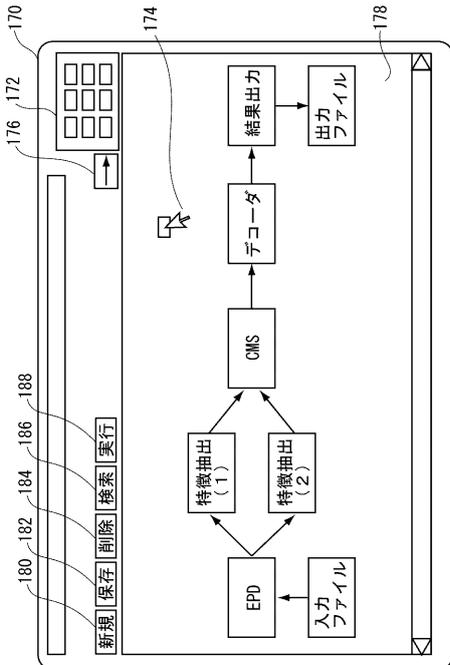
```

<flow-definition>
  <flow name="サンプル">
    <process name="プロセス1" host="pxn160" program="ATRewc">
      <module node_name="EPD" id="EPD" />
      <module node_name="W2C[1]" id="W2C" number="1" />
      <module node_name="W2C[2]" id="W2C" number="2" />
    </process>
    <process name="プロセス2" host="pxn160" program="ATRcdr">
      ...
    </process>
  </flow name>
  <file>
    <input name="音声1" file="sample1.raw" format="NoHeader" event-type="
      ATRD_WAVE" data-type="short" size="160" />
    <input name="TRS" file="sample.trs" format="TRS" data-type="string" />
    <output name="DP" file="sample.dp" format="DP" data-type="string" />
  </file>
  <arc>
    <from name="音声1" number="1" />
    <to name="EPD" number="1" />
    <event type="ATRD_WAVE" />
  </arc>
  ...
  </arc>
</flow-definition>
  <options>
    <common>
      ...
    </common>
    <module node_name="" id="C2P">
      <option id="CepstrumOrder">12</option>
      <section id="delta-cepstrum">
        <option id="window">5</option>
        <option id="padding">zero</option>
      </section>
      <option id="rho">1.0</option>
      <option id="OutputParameter">cep(12)+dpow+dcep(12)</option>
      <option id="WindowType">rectangular</option>
      <section id="message">
        <option id="level">user</option>
        <option id="timing">now_stack</option>
      </section>
    </module>
    <module node_name="" id="W2C">
      <option id="preemphasis">0.98</option>
      <section id="frame">
        <option id="length">20</option>
        <option id="shift">10</option>
      </section>
      <section id="message">
        ...
      </section>
    </module>
    <module node_name="W2C[1]" id="W2C">
      <section id="frame">
        <option id="length">30</option>
      </section>
    </module>
    <module node_name="W2C[2]" id="W2C">
      ...
    </module>
  </options>
</flow-definition>
  
```

【図6】



【図7】



【図8】

モジュールタイプ	表示文字列
FeatureExtraction	特徴抽出
CMS	CMS
Decoder	デコーダ
...	...

【図9】

モジュールID	モジュールタイプ	プログラム
EPD-new	EPD	/usr/local/bin/atriums2
EPD-old	EPD	/usr/local/bin/atriums1
WaveToCep-M	wave2cep	/usr/local/bin/atriums2

【図10】

ホスト名
pxn123
pxn125
host345

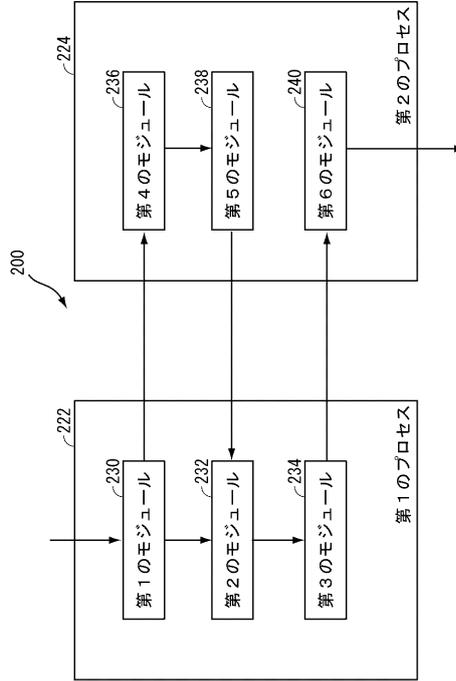
【図 1 1】

ノードID	モジュールタイプ	左上座標	右下座標	実行プログラム	実行モジュールID	実行マシン
1	EPD	10, 10	110, 60	/usr/local/bin/attrs2	EPD-Soong	pxn123
2	FeatureExtraction	200, 10	300, 60			
3	OMS	400, 10	400, 60			
...			

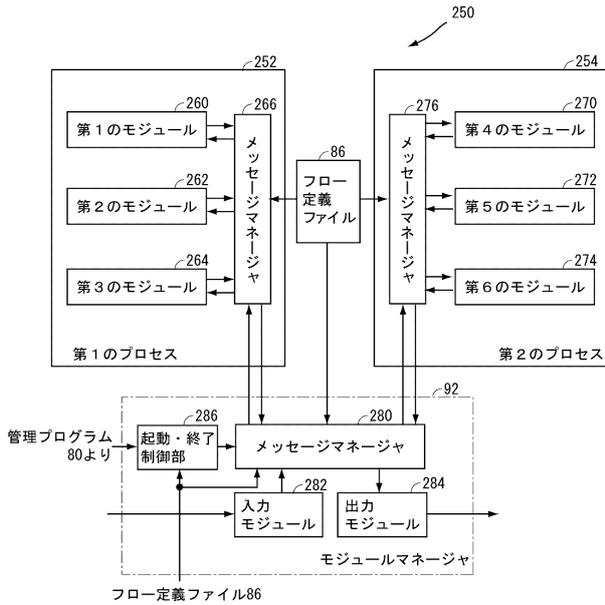
【図 1 2】

開始ノードID	終了ノードID	イベントタイプ
1	2	WAVE
2	3	FEATURE
2	4	FEATURE

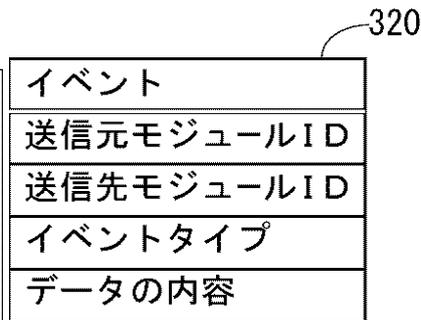
【図 1 3】



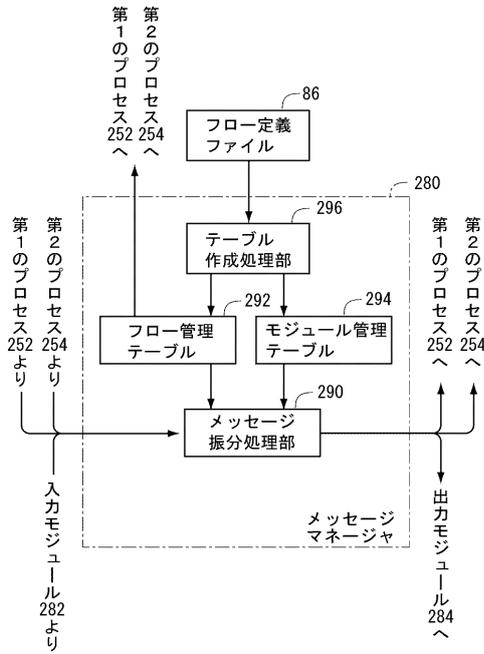
【図 1 4】



【図 1 5】



【図16】



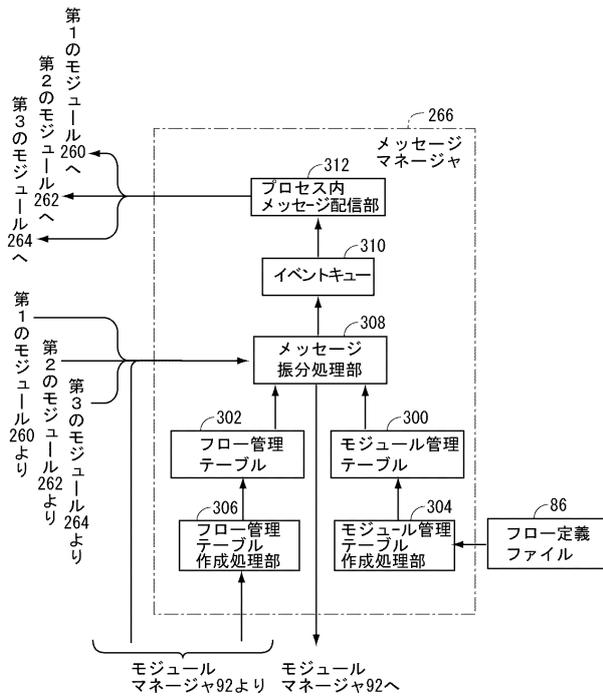
【図17】

送信元 モジュールID	イベント タイプ	送信先 モジュールID
第1のモジュール	WAVE	第4のモジュール
第1のモジュール	WAVE	第2のモジュール
第2のモジュール	FEATURE	第3のモジュール

【図18】

プロセス名	モジュールID
第1のプロセス	第1のモジュール
第1のプロセス	第2のモジュール
第2のプロセス	第4のモジュール
...	...
モジュールマネージャ	出力マネージャ
モジュールマネージャ	入力マネージャ

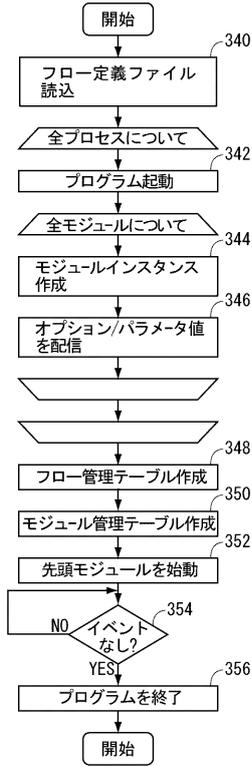
【図19】



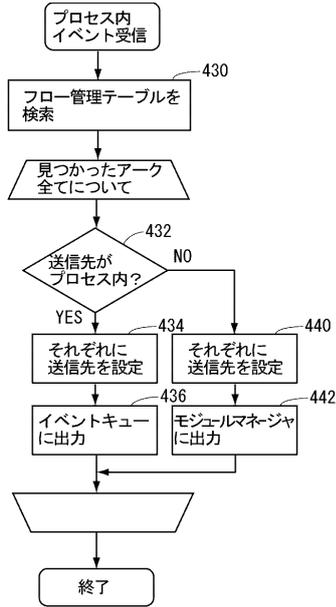
【図20】

モジュールID
第1のモジュール
第2のモジュール
第3のモジュール

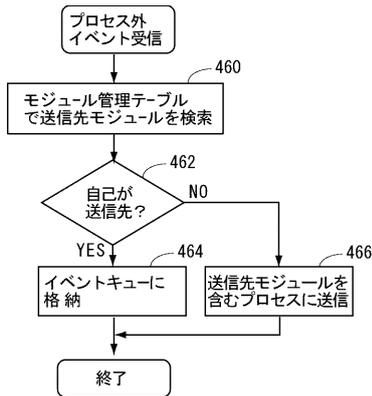
【図 2 1】



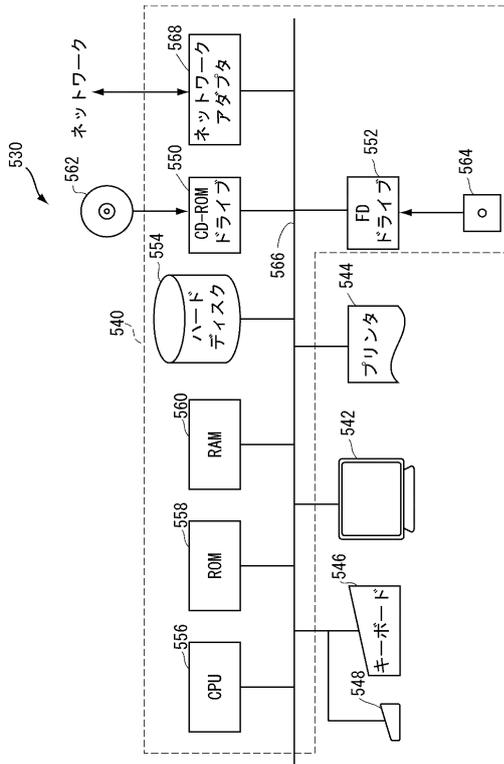
【図 2 2】



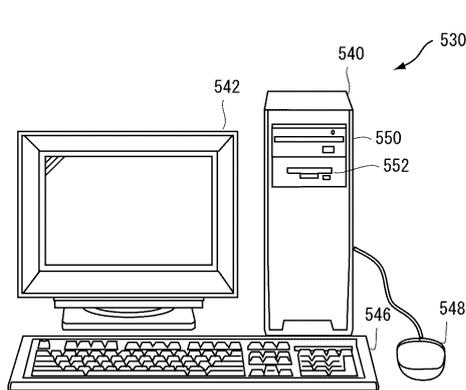
【図 2 3】



【図 2 5】



【図 2 4】



フロントページの続き

審査官 林 毅

- (56)参考文献 特開平08 - 190477 (JP, A)
特開2001 - 216142 (JP, A)
特開2000 - 163262 (JP, A)
特開平07 - 302195 (JP, A)
特開2003 - 223321 (JP, A)
特開平05 - 143309 (JP, A)
特開平11 - 024904 (JP, A)
特開平10 - 083420 (JP, A)

- (58)調査した分野(Int.Cl., DB名)
G06F 9/44