

(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2005-292981

(P2005-292981A)

(43) 公開日 平成17年10月20日(2005. 10. 20)

(51) Int. Cl.⁷

G06F 9/46

F I

G06F 9/46 340B

テーマコード (参考)

5B098

審査請求 未請求 請求項の数 7 O L (全 16 頁)

(21) 出願番号 特願2004-104089 (P2004-104089)
 (22) 出願日 平成16年3月31日 (2004. 3. 31)

(71) 出願人 393031586
 株式会社国際電気通信基礎技術研究所
 京都府相楽郡精華町光台二丁目2番地2
 (74) 代理人 100099933
 弁理士 清水 敏
 (72) 発明者 葦苒 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
 Fターム(参考) 5B098 AA10 GA04 GC16

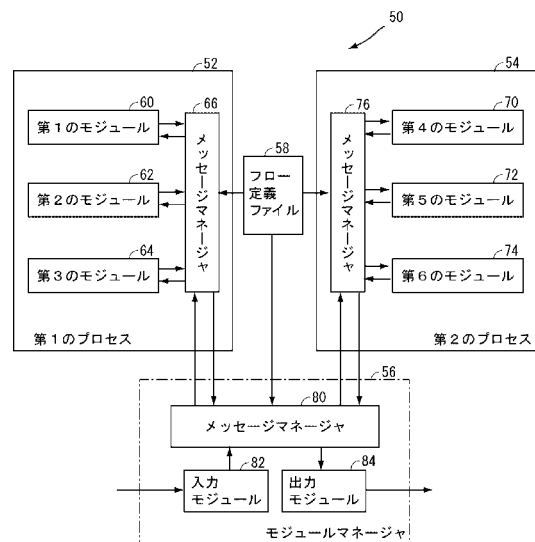
(54) 【発明の名称】 メッセージ管理装置、データ処理システム及びそのためのコンピュータプログラム

(57) 【要約】

【目的】モジュール間でのメッセージの通信を効率的に行なえるようにする。

【解決手段】データ処理システム50は、複数のプロセス52、54と、プロセス間のメッセージ通信を管理するメッセージマネージャ80を含む。プロセス52、54の各々は、1又は複数の入力及び出力を持ち、入力データに所定の処理を行なうモジュール60、62、64、70、72、74と、プロセス内におけるメッセージ通信を管理するメッセージマネージャ66、76を含む。メッセージマネージャ66、76は、プロセス内のモジュールから受けたメッセージに含まれる送信元モジュール識別子及びメッセージの型に基づいて送信先モジュールを決定し、そのモジュールがプロセス内か外かによって、メッセージをプロセス内で配信する処理と、メッセージマネージャ80に送信する処理とを選択的に行なう。

【選択図】 図2



【特許請求の範囲】

【請求項 1】

複数のモジュールを含むプロセス内において、前記複数のモジュール相互の間のメッセージ通信、及び前記複数のモジュールと当該プロセス外のモジュールとの間のメッセージ通信を管理するためのメッセージ管理装置であって、

各モジュールはいずれもモジュール識別子により一意に特定され、

各モジュール間で通信されるメッセージは、送信元モジュール識別子と、送信先モジュール識別子と、送信されるメッセージの型と、送信されるべきデータとを含み、

前記メッセージ管理装置は、

メッセージを一時格納するためのメッセージキューと、

前記メッセージキューに格納されたメッセージを、前記プロセス内の、当該メッセージの送信先モジュール識別子により特定されるモジュールに配信するためのメッセージ配信手段と、

予め定められたデータの流れて、メッセージの送信元モジュール識別子及びメッセージの型に基づいてメッセージの送信先モジュールを決定し、送信先モジュール識別子を出力するためのメッセージフロー管理手段と、

前記プロセス内のモジュールから受けたメッセージに回答して、当該メッセージに含まれる送信元モジュール識別子及びメッセージの型に基づいて、前記メッセージフロー管理手段からメッセージの送信先モジュール識別子を取得して当該メッセージの送信先モジュール識別子に設定し、取得された送信先モジュール識別子により特定されるモジュールが前記プロセス内のモジュールか否かによって、当該メッセージを前記メッセージキューに格納する処理と、前記プロセス外の所定のメッセージ送信先に送信する処理とを選択的に行なうためのメッセージ振分手段とを含む、メッセージ管理装置。

【請求項 2】

前記メッセージ振分手段は、

前記プロセス内のモジュール識別子を記憶するためのモジュール識別子記憶手段と、

前記プロセス内のモジュールから受けたメッセージに回答して、当該メッセージに含まれる送信元モジュール識別子及びメッセージの型に基づいて、前記メッセージフロー管理手段からメッセージの送信先モジュール識別子を取得するための手段と、

前記取得するための手段により取得された送信先モジュール識別子を、前記受けたメッセージの送信先モジュール識別子に設定するための手段と、

前記取得するための手段により取得された送信先モジュール識別子が、前記モジュール識別子記憶手段に記憶されているか否かを判定するための判定手段と、

前記判定手段により、前記取得するための手段により取得された送信先モジュール識別子が前記モジュール識別子記憶手段に記憶されていると判定されたことに回答して、前記受信したメッセージを前記メッセージキューに格納するための手段と、

前記判定手段により、前記取得するための手段により取得された送信先モジュール識別子が前記モジュール識別子記憶手段に記憶されていないと判定されたことに回答して、前記受信したメッセージを前記プロセス外の前記所定のメッセージ送信先に送信するための手段とを含む、請求項 1 に記載のメッセージ管理装置。

【請求項 3】

各々が 1 又は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において、モジュール間のメッセージ通信を管理するためのメッセージ管理装置であって、

前記システム内の各モジュールは、いずれもモジュール識別子により一意に特定され、

各モジュール間で通信されるメッセージは、送信元モジュール識別子と、送信先モジュール識別子と、送信されるメッセージの型と、送信されるべきデータとを含み、

前記メッセージ管理装置は、

予め定められたデータの流れて、メッセージの送信元モジュール識別子及びメッセージの型に基づいてメッセージの送信先モジュールを決定し、送信先モジュール識別子を出力するためのメッセージフロー管理手段と、

前記プロセス内のモジュールから受けたメッセージに回答して、当該メッセージに含まれる送信元モジュール識別子及びメッセージの型に基づいて、前記メッセージフロー管理手段からメッセージの送信先モジュール識別子を取得し、取得された送信先モジュール識別子により特定されるモジュールを含むプロセスに前記受けたメッセージを配信するためのメッセージ振分手段とを含む、メッセージ管理装置。

【請求項 4】

コンピュータにより実行されると、請求項 1～請求項 3 のいずれかに記載のメッセージ管理装置として当該コンピュータを動作させる、コンピュータプログラム。

【請求項 5】

複数のプロセスと、

10

各プロセス間でのデータの流を実現するメッセージ通信を管理するためのプロセス間メッセージ管理手段とを含むデータ処理システムであって、

前記複数のプロセスの各々は、

各々が 1 又は複数の入力及び出力を持ち、入力されるデータに対し所定の処理を行なって出力する複数のモジュールと、

当該プロセス内において、前記複数のモジュール相互の間のメッセージ通信、及び前記複数のモジュールと当該プロセス外のモジュールとの間のメッセージ通信を管理するためのプロセス内メッセージ管理手段とを含み、

各モジュールはいずれもモジュール識別子により一意に特定され、

各モジュール間で通信されるメッセージは、送信元モジュール識別子と、送信先モジュール識別子と、送信されるメッセージの型と、送信されるべきデータとを含み、

20

前記プロセス内メッセージ管理手段は、

メッセージを一時格納するためのメッセージキューと、

前記メッセージキューに格納されたメッセージを、プロセス内の、当該メッセージの送信先モジュール識別子により特定されるモジュールに配信するためのメッセージ配信手段と、

予め定められたデータの流に従って、メッセージの送信元モジュール識別子及びメッセージの型に基づいてメッセージの送信先モジュールを決定し、送信先モジュール識別子を出力するための第 1 のメッセージフロー管理手段と、

プロセス内のモジュールから受けたメッセージに回答して、当該メッセージに含まれる送信元モジュール識別子及びメッセージの型に基づいて、前記第 1 のメッセージフロー管理手段からメッセージの送信先モジュール識別子を取得して当該メッセージの送信先モジュール識別子に設定し、取得された送信先モジュール識別子により特定されるモジュールが前記プロセス内のモジュールが否かに従って、当該メッセージを前記メッセージキューに格納する処理と、前記プロセス間メッセージ管理手段に送信する処理とを選択的に行なうためのメッセージ振分手段とを含む、システム。

30

【請求項 6】

前記メッセージ振分手段は、

前記プロセス内のモジュール識別子を記憶するためのモジュール識別子記憶手段と、

前記プロセス内のモジュールから受けたメッセージに回答して、当該メッセージに含まれる送信元モジュール識別子及びメッセージの型に基づいて、前記第 1 のメッセージフロー管理手段からメッセージの送信先モジュール識別子を取得するための手段と、

40

前記取得するための手段により取得された送信先モジュール識別子を、前記受けたメッセージの送信先モジュール識別子に設定するための手段と、

前記取得するための手段により取得された送信先モジュール識別子が、前記モジュール識別子記憶手段に記憶されているか否かを判定するための判定手段と、

前記判定手段により、前記取得するための手段により取得された送信先モジュール識別子が前記モジュール識別子記憶手段に記憶されていると判定されたことに回答して、前記受信したメッセージを前記メッセージキューに格納するための手段と、

前記判定手段により、前記取得するための手段により取得された送信先モジュール識別

50

子が前記モジュール識別子記憶手段に記憶されていないと判定されたことに応答して、前記受信したメッセージを前記プロセス外の前記所定のメッセージ送信先に送信するための手段とを含む、請求項 5 に記載のシステム。

【請求項 7】

コンピュータにより実行されると、請求項 5 又は請求項 6 に記載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、コンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

この発明は、コンピュータ上で動作する機能モジュールの間のメッセージ通信に関し、特に、複数のモジュールで一つのプロセスを構成し、複数のプロセスで一つのタスクを実現するシステムにおいて、モジュール間のメッセージ通信を高速化するシステムに関する。

【背景技術】

【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等を変えて同じプログラムを動作させるだけでなく、処理の流れそのものも自由に変更して実験できれば便利である。そのため、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、それらモジュールを目的にあわせて適切な形で組合せることができるシステムがあれば好ましい。

【0003】

さらに、実験結果を早く確認するためには、一つのコンピュータ上だけでなく、複数のコンピュータを互いに結合し、それらコンピュータ上で別々に、互いに連携させながらモジュールを動作させることができれば便利である。

【0004】

そのための原理的構成として考えられるものを図 1 に示す。図 1 に示すシステム 20 は、第 1 のプロセス 22 と第 2 のプロセス 24 とを含む。

【0005】

第 1 のプロセス 22 は、第 1 のモジュール 30 と、第 2 のモジュール 32 と、第 3 のモジュール 34 とを含む。第 2 のプロセス 24 は、第 4 のモジュール 36 と、第 5 のモジュール 38 と、第 6 のモジュール 40 とを含む。

【0006】

例えば第 1 のプロセス 22 をあるコンピュータ上で実行し、第 2 のプロセス 24 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 22 と第 2 のプロセス 24 とを同一のコンピュータ上で動作させても問題はない。

【0007】

こうしたシステムを実現するためには、あるモジュールの出力を別のモジュールの入力に与えるため、予め各モジュールの入力と出力とを明確に定義しておかなければならないが、それに加えて各モジュール間及びプロセス間でのデータ（メッセージ）の受け渡しをどのように実現するかを考える必要がある。

【0008】

例えば同じプロセスに含まれるモジュールであれば、プロセス内のメモリ空間を共有しているため、互いのメッセージ交換は高速に行なえる。しかしプロセス間の通信に要するシステムのオーバーヘッドは大きく、プロセス間のメッセージ通信には比較的時間がかかる。

【0009】

そうした問題を解決するために、あるモジュールの出力先が同じプロセス内か別プロセスかで、それぞれ異なる通信方式で通信するようにモジュールを設計することが考えられ

10

20

30

40

50

る。

【発明の開示】

【発明が解決しようとする課題】

【0010】

しかし、モジュールの設計を出力先にあわせて変えるのでは、モジュールの組合せが固定されてしまい、モジュール化の当初の目的を達成できない。従って、モジュールが属するプロセスに関わらず、モジュール化の利点を残しながら、モジュール間の通信を効率的に行なうことができるシステムが望まれている。

【0011】

それゆえに本発明の目的は、自由に組合せたモジュール間でのメッセージの通信を効率的に行なうことが可能なメッセージ管理装置、及びそのようなメッセージ管理装置を備えたデータ処理システムを提供することである。

【課題を解決するための手段】

【0012】

本発明の第1の局面に係るメッセージ管理装置は、複数のモジュールを含むプロセス内において、複数のモジュール相互の間のメッセージ通信、及び複数のモジュールと当該プロセス外のモジュールとの間のメッセージ通信を管理するためのメッセージ管理装置であって、各モジュールはいずれもモジュール識別子により一意に特定され、各モジュール間で通信されるメッセージは、送信元モジュール識別子と、送信先モジュール識別子と、送信されるメッセージの型と、送信されるべきデータとを含み、メッセージ管理装置は、メッセージを一時格納するためのメッセージキューと、メッセージキューに格納されたメッセージを、プロセス内の、当該メッセージの送信先モジュール識別子により特定されるモジュールに配信するためのメッセージ配信手段と、予め定められたデータの流に従って、メッセージの送信元モジュール識別子及びメッセージの型に基づいてメッセージの送信先モジュールを決定し、送信先モジュール識別子を出力するためのメッセージフロー管理手段と、プロセス内のモジュールから受けたメッセージに应答して、当該メッセージに含まれる送信元モジュール識別子及びメッセージの型に基づいて、メッセージフロー管理手段からメッセージの送信先モジュール識別子を取得して当該メッセージの送信先モジュール識別子に設定し、取得された送信先モジュール識別子により特定されるモジュールがプロセス内のモジュールか否かに従って、当該メッセージをメッセージキューに格納する処理と、プロセス外の所定のメッセージ送信先に送信する処理とを選択的に行なうためのメッセージ振分手段とを含む。

【0013】

好ましくは、メッセージ振分手段は、プロセス内のモジュール識別子を記憶するためのモジュール識別子記憶手段と、プロセス内のモジュールから受けたメッセージに应答して、当該メッセージに含まれる送信元モジュール識別子及びメッセージの型に基づいて、メッセージフロー管理手段からメッセージの送信先モジュール識別子を取得するための手段と、取得するための手段により取得された送信先モジュール識別子を、受けたメッセージの送信先モジュール識別子に設定するための手段と、取得するための手段により取得された送信先モジュール識別子が、モジュール識別子記憶手段に記憶されているか否かを判定するための判定手段と、判定手段により、取得するための手段により取得された送信先モジュール識別子がモジュール識別子記憶手段に記憶されていると判定されたことに应答して、受信したメッセージをメッセージキューに格納するための手段と、判定手段により、取得するための手段により取得された送信先モジュール識別子がモジュール識別子記憶手段に記憶されていないと判定されたことに应答して、受信したメッセージをプロセス外の所定のメッセージ送信先に送信するための手段とを含む。

【0014】

本発明の第2の局面に係るメッセージ管理装置は、各々が1又は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において、モジュール間のメッセージ通信を管理するためのメッセージ管理装置であって、システム内の各モジュールは、いずれもモジュール

識別子により一意に特定され、各モジュール間で通信されるメッセージは、送信元モジュール識別子と、送信先モジュール識別子と、送信されるメッセージの型と、送信されるべきデータとを含み、メッセージ管理装置は、予め定められたデータの流に従って、メッセージの送信元モジュール識別子及びメッセージの型に基づいてメッセージの送信先モジュールを決定し、送信先モジュール識別子を出力するためのメッセージフロー管理手段と、プロセス内のモジュールから受けたメッセージに回答して、当該メッセージに含まれる送信元モジュール識別子及びメッセージの型に基づいて、メッセージフロー管理手段からメッセージの送信先モジュール識別子を取得し、取得された送信先モジュール識別子により特定されるモジュールを含むプロセスに受けたメッセージを配信するためのメッセージ振分手段とを含む。

10

【 0 0 1 5 】

本発明の第 3 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メッセージ管理装置として当該コンピュータを動作させる。

【 0 0 1 6 】

本発明の第 4 の局面に係るデータ処理システムは、複数のプロセスと、各モジュール間のデータの流を実現するメッセージ通信を管理するためのプロセス間メッセージ管理手段とを含むデータ処理システムである。複数のプロセスの各々は、各々が 1 又は複数の入力及び出力を持ち、入力されるデータに対し所定の処理を行なって出力する複数のモジュールと、当該プロセス内において、複数のモジュール相互の間のメッセージ通信、及び複数のモジュールと当該プロセス外のモジュールとの間のメッセージ通信を管理するためのプロセス内メッセージ管理手段とを含む。各モジュールはいずれもモジュール識別子により一意に特定される。各モジュール間で通信されるメッセージは、送信元モジュール識別子と、送信先モジュール識別子と、送信されるメッセージの型と、送信されるべきデータとを含む。プロセス内メッセージ管理手段は、メッセージを一時格納するためのメッセージキューと、メッセージキューに格納されたメッセージを、プロセス内の、当該メッセージの送信先モジュール識別子により特定されるモジュールに配信するためのメッセージ配信手段と、予め定められたデータの流に従って、メッセージの送信元モジュール識別子及びメッセージの型に基づいてメッセージの送信先モジュールを決定し、送信先モジュール識別子を出力するための第 1 のメッセージフロー管理手段と、プロセス内のモジュールから受けたメッセージに回答して、当該メッセージに含まれる送信元モジュール識別子及びメッセージの型に基づいて、第 1 のメッセージフロー管理手段からメッセージの送信先モジュール識別子を取得して当該メッセージの送信先モジュール識別子に設定し、取得された送信先モジュール識別子により特定されるモジュールがプロセス内のモジュールか否かに従って、当該メッセージをメッセージキューに格納する処理と、プロセス間メッセージ管理手段に送信する処理とを選択的に行なうためのメッセージ振分手段とを含む。

20

30

【 0 0 1 7 】

好ましくは、メッセージ振分手段は、プロセス内のモジュール識別子を記憶するためのモジュール識別子記憶手段と、プロセス内のモジュールから受けたメッセージに回答して、当該メッセージに含まれる送信元モジュール識別子及びメッセージの型に基づいて、第 1 のメッセージフロー管理手段からメッセージの送信先モジュール識別子を取得するための手段と、取得するための手段により取得された送信先モジュール識別子を、受けたメッセージの送信先モジュール識別子に設定するための手段と、取得するための手段により取得された送信先モジュール識別子が、モジュール識別子記憶手段に記憶されているか否かを判定するための判定手段と、判定手段により、取得するための手段により取得された送信先モジュール識別子がモジュール識別子記憶手段に記憶されていると判定されたことに回答して、受信したメッセージをメッセージキューに格納するための手段と、判定手段により、取得するための手段により取得された送信先モジュール識別子がモジュール識別子記憶手段に記憶されていないと判定されたことに回答して、受信したメッセージをプロセス外の所定のメッセージ送信先に送信するための手段とを含む。

40

【 0 0 1 8 】

50

本発明の第5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】

【0019】

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、複数のモジュールを組合せることにより、音声認識を行なうためのシステムである。

【0020】

音声のような時系列データを処理する場合、発話単位で処理を行なうと、すなわち、話し終わってからまとめて処理を行なうと、少なくとも発話時間分のタイムラグが発生する。その結果、リアルタイムに処理することができない。

10

【0021】

そこで本システムでは、音声データを10msec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「イベントメッセージ（以後イベントと略す）」という。

【0022】

イベントには、データの種別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、又はそこから入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。

20

【0023】

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る必要がある。このように、モジュール間のデータの流れを予め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。

【0024】

図2に本システム50のブロック図を示す。この本システム50は、図1に示すシステム20と同様、第1のプロセス52及び第2のプロセス54を含む。本システム50はさらに、第1のプロセス52及び第2のプロセス54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するフロー定義ファイル58と、第1のプロセス52及び第2のプロセス54、並びに本システム50の外部とに接続され、第1のプロセス52及び第2のプロセス54内のモジュール間、及びこれらモジュールと本システム50の外部との間のイベントの流れをフロー定義ファイル58の記述に従って制御するためのモジュールマネージャ56とを含む。

30

【0025】

第1のプロセス52は、第1のモジュール60と、第2のモジュール62と、第3のモジュール64と、これら3つのモジュール60、62及び64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56との間のイベントの流れをフロー定義ファイル58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66とを含む。

40

【0026】

第2のプロセス54も第1のプロセス52と同様、第4のモジュール70と、第5のモジュール72と、第6のモジュール74と、これら3つのモジュール70、72及び74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56との間のイベントの流れをフロー定義ファイル58の記述に従って制御するためのメッセージマネージャ76とを含む。

【0027】

モジュール60～74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、

50

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。

【0028】

モジュールマネージャ56は、本システム50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82と、本システム50からの出力を外部に与えるための出力モジュール84と、入力モジュール82、出力モジュール84、メッセージマネージャ66及びメッセージマネージャ76に接続され、フロー定義ファイル58の記述に基づいて、各プロセス52及び54、並びに入力モジュール82及び出力モジュール84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80とを含む。すなわちメッセージマネージャ80は、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。

【0029】

図3に、モジュール間で通信されるイベント120の構成を示す。図3を参照して、イベント120は、送信元モジュール識別番号(以下「ID」と呼ぶ。)と、送信先モジュールIDと、イベントの型と、データの内容とを含む。イベントの型とは、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型などがその一例である。

【0030】

図4に、フロー定義ファイル58内に記述される内容を表形式で示す。図4を参照して、フロー定義ファイル58内には、「process」、「arc」及び「file」という三つの項目が定義されている。

【0031】

processは、どのプロセスにどのモジュールが含まれているかを示す情報である。arc(以下「アーク」と呼ぶ。)は、どのモジュールからどのモジュールにどのようなイベントが送られるかを示す情報である。この中にはイベントの型も含まれる。fileは、外部からどのようなイベントがシステム50に送られ、結果がどのようなイベントとして外部に出力されるかを示す。

【0032】

図5にメッセージマネージャ80の構成をブロック図形式で示す。図5を参照して、メッセージマネージャ80は、メッセージマネージャ80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とイベントの型とをキーとして、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92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94と、フロー定義ファイル58の記述に基づいて、フロー管理テーブル92及びモジュール管理テーブル94を作成するためのテーブル作成処理部96とを含む。

【0033】

メッセージマネージャ80はさらに、第1のプロセス52、第2のプロセス54、入力モジュール82及び出力モジュール84に接続され、第1のプロセス52、第2のプロセス54及び入力モジュール82から与えられるイベントを、フロー管理テーブル92及びモジュール管理テーブル94の記述に基づいて第1のプロセス52、第2のプロセス54、及び出力モジュール84のいずれかに出力するためのメッセージ振分処理部90を含む。

【0034】

図6にフロー管理テーブル92の構成の一例を、図7にモジュール管理テーブル94の構成の一例を、それぞれ示す。図6を参照して、フロー管理テーブル92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、イベントの型と、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58のアークの記述に相当する。受信したイベントに含まれる送信元モジュールID及びイベントの型(図3を参照されたい。)をキーにフロー管理テーブル92を検索することにより、当該イベントをどのモ

10

20

30

40

50

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6に示すフロー管理テーブル92は、図2に示す構成に必要な情報の一部のみを示す。

【0035】

図7を参照して、モジュール管理テーブル94も通常は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7に示すモジュール管理テーブル94も、図2に示す本システム50の構成に必要な情報の一部のみを示す。

【0036】

次に図2の第1のプロセス52及び第2のプロセス54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66とメッセージマネージャ76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66について説明する。

【0037】

図8を参照して、メッセージマネージャ66は、メッセージマネージャ66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とイベントの型とをキーとして、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5に示すものと同様形式のフロー管理テーブル102と、このメッセージマネージャ66が管理するプロセス（本実施の形態では第1のプロセス52）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100と、図5に示すフロー管理テーブル92を複写してフロー管理テーブル102を作成するためのフロー管理テーブル作成処理部106と、フロー定義ファイル58の記述に基づき、第1のプロセス52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100を作成するためのモジュール管理テーブル作成処理部104とを含む。

【0038】

メッセージマネージャ66はさらに、第1～第3のモジュール60、62及び64とモジュールマネージャ56とからイベントを受取り、フロー管理テーブル102とモジュール管理テーブル100とを参照して、当該イベントを第1～第3のモジュール60、62及び64、並びにモジュールマネージャ56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108と、メッセージ振分処理部108によってメッセージ振分処理部108の管理するプロセス（本実施の形態では第1のプロセス52）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110と、イベントキュー110からイベントを一つずつ読出し、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60、62及び64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112とを含む。

【0039】

図9にモジュール管理テーブル100の形式を模式的に示す。メッセージマネージャ66が管理する第1のプロセス52の場合、そこに含まれるモジュールは第1のモジュール60、第2のモジュール62、及び第3のモジュール64である。従ってこの場合のモジュール管理テーブル100の内容は図9に示した通りとなる。

【0040】

実際にはメッセージ振分処理部108は、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108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図10及び図11を参照して説明する。

【0041】

図10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8に示すメッセージマネージャ66のメッセージ振分処理部108が実行する処理プログラムのフローチャートである。図5に示すメッセージマネージャ80のメッセージ振分処理部90につい

10

20

30

40

50

ても同様である。図 10 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 130 で、当該イベントの送信元モジュール ID とイベント型とをキーにフロー管理テーブル 102 を検索し、該当する全てのアークを取出す。

【 0 0 4 2 】

続いて、ステップ 130 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 132 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 134 に進み、さもなければステップ 140 に進む。

【 0 0 4 3 】

ステップ 134 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 136 でこのイベントをイベントキュー 110 に出力する。 10

【 0 0 4 4 】

一方ステップ 140 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 142 でこのイベントをモジュールマネージャ 56 に出力する。

【 0 0 4 5 】

ステップ 134、136、140 及び 142 の処理を、ステップ 130 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 110 又はモジュールマネージャ 56 に送信されることになる。 20

【 0 0 4 6 】

図 11 は、プロセス外からイベントを受信した場合に、図 5 に示すメッセージマネージャ 80 のメッセージ振分処理部 90 が実行する処理プログラムのフローチャートである。図 11 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 160 で、モジュール管理テーブル 94 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 162 では、決定されたプロセスが自分自身か否かを判定する。もし自分自身が送信先であればステップ 164 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 166 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。 30

【 0 0 4 7 】

一方、図 8 に示すメッセージマネージャ 66 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 8 に示すメッセージ振分処理部 108 に格納する。

【 0 0 4 8 】

- 動作 -

図 2 ~ 図 11 を参照して、以上のような構成を有する本システム 50 は以下のように動作する。まず、図 2 を参照して、最初にシステムの構成が定められ、その構成に従ってフロー定義ファイル 58 が記述され所定の記憶装置に格納される。図 5 に示すテーブル作成処理部 96 がフロー定義ファイル 58 に基づいて図 5 に示すフロー管理テーブル 92 及びモジュール管理テーブル 94 を作成する。各プロセスでは、図 8 に示すフロー管理テーブル作成処理部 106 及びモジュール管理テーブル作成処理部 104 が、それぞれフロー管理テーブル 92 及びフロー定義ファイル 58 に基づいてフロー管理テーブル 102 及びモジュール管理テーブル 100 を作成する。 40

【 0 0 4 9 】

動作において、入力モジュール 82 に入力データが与えられる。入力データは本システムでは通常は音声データファイルである。このシステムでは、ファイルはプロセスと同様に扱われるので、入力ファイルから作成されたフレームデータがイベントとして入力モジュール 82 に与えられる。本例の場合、図 1 に示すようにモジュールを組合せるものとするれば、入力モジュール 82 に与えられるイベントの送信先モジュール ID は第 1 のモジュ 50

ールに対応するものであり、イベントの型は「WAVE」となる。送信元モジュールIDは入力ファイルに相当するものになる。このイベントは入力モジュール82からメッセージマネージャ80に与えられる。

【0050】

図5を参照して、メッセージマネージャ80のメッセージ振分処理部90は、フロー管理テーブル92及びモジュール管理テーブル94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11のステップ160)、イベントを当該プロセスに送信する(ステップ164又は166)。本例では、例えば第1のプロセス52にこのイベントが送信されるものとする。

【0051】

図8を参照して、メッセージマネージャ66のメッセージ振分処理部108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110に格納する。プロセス内メッセージ配信部112が、イベントキュー110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。

【0052】

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66のメッセージ振分処理部108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及びイベント型には、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。

【0053】

メッセージ振分処理部108は、受けたイベントの送信元モジュールIDとイベント型とをキーにフロー管理テーブル102を検索し(図10のステップ130)、一致するアークを得る。メッセージ振分処理部108はさらに、モジュール管理テーブル100を参照して、当該送信先モジュールが第1のプロセス52内か否かを判定する(ステップ132)。第1のプロセス52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100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134)、メッセージ振分処理部108はこのイベントをイベントキュー110に書出す(ステップ136)。

【0054】

送信先が第1のプロセス52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140)、モジュールマネージャ56にこのイベントを送信する(ステップ142)。

【0055】

モジュールマネージャ56のメッセージマネージャ80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92とモジュール管理テーブル94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92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90は出力モジュール84にイベントを出力する。出力モジュール84は、このイベントをさらに外部に出力する。

【0056】

以下、メッセージマネージャ80のメッセージ振分処理部90、並びに第1のプロセス52及び第2のプロセス54のメッセージマネージャ66及び76は、上記したの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82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ュール84から出力される。

【0057】

上に説明したシステム50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

10

20

30

40

50

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。

【0058】

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ーヘッドが少なくなる。

【0059】

従って、このようなプロセス間通信の仕組みを用いることで、イベントを効率よく配信することが可能となり、さらに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、音声認識など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などを簡単に、かつ短い時間に行なうことが可能になる。

〔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12はこのコンピュータシステム330の外観を示し、図13はコンピュータシステム330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などで結合したものによって実現することもできる。

【0060】

図12を参照して、このコンピュータシステム330は、FD（フレキシブルディスク）ドライブ352及びCD-ROM（コンパクトディスク読出専用メモリ）ドライブ350を有するコンピュータ340と、キーボード346と、マウス348と、モニタ342とを含む。

【0061】

図13を参照して、コンピュータ340は、FDドライブ352及びCD-ROMドライブ350に加えて、CPU（中央処理装置）356と、CPU356、FDドライブ352及びCD-ROMドライブ350に接続されたバス366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）358と、バス366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）360とを含む。コンピュータシステム330はさらに、プリンタ344を含んでいる。

【0062】

コンピュータ340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368を含む。

【0063】

コンピュータシステム330にこの実施の形態に係る本システム50（又はその一部の第1のプロセス52、第2のプロセス54、モジュールマネージャ56など）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350又はFDドライブ352に挿入されるCD-ROM362又はFD364に記憶され、さらにハードディスク354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340に送信されハードディスク354に記憶されてもよい。プログラムは実行の際にRAM360にロードされる。CD-ROM362から、FD364から、又はネットワークを介して、直接にRAM360にプログラムをロードしてもよい。

【0064】

このプログラムは、コンピュータ340にこの実施の形態に係るシステム50に含まれ

10

20

30

40

50

る各プロセス 5 2、5 4、モジュールマネージャ 5 6、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 3 4 0 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 3 4 0 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、上記した換言規則選別装置を実行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 3 3 0 の動作は周知であるので、ここでは繰返さない。

【0065】

10

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。

【0066】

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のすべての変更を含む。

20

【図面の簡単な説明】

【0067】

【図 1】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。

【図 2】本発明の一実施の形態に係るシステム 5 0 のブロック図である。

【図 3】イベント 1 2 0 の構成を示す図である。

【図 4】フロー定義ファイル 5 8 の構成を示す図である。

【図 5】図 2 に示すモジュールマネージャ 5 6 のメッセージマネージャ 8 0 のブロック図である。

30

【図 6】フロー管理テーブル 9 2 の構成を示す図である。

【図 7】モジュール管理テーブル 9 4 の構成を示す図である。

【図 8】図 2 に示す第 1 のプロセス 5 2 のメッセージマネージャ 6 6 のブロック図である。

【図 9】モジュール管理テーブル 1 0 0 の構成を示す図である。

【図 10】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 11】モジュールマネージャ 5 6 のメッセージ振分処理部 9 0 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。

40

【図 12】本発明の一実施の形態に係るシステム 5 0 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觀図である。

【図 13】図 1 2 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】

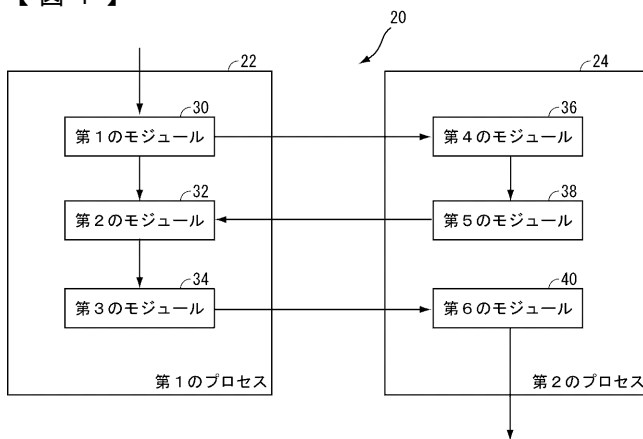
【0068】

5 0 システム、5 2, 5 4 プロセス、5 6 モジュールマネージャ、5 8 フロー定義ファイル、6 0, 6 2, 6 4, 7 0, 7 2, 7 4 モジュール、6 6, 7 6, 8 0 メッセージマネージャ、8 2 入力モジュール、8 4 出力モジュール、9 0, 1 0 8 メッセージ振分処理部、9 2, 1 0 2 フロー管理テーブル、9 4, 1 0 0 モジュール管理テーブル、9 6 テーブル作成処理部、1 0 4 モジュール管理テーブル作成処理部、

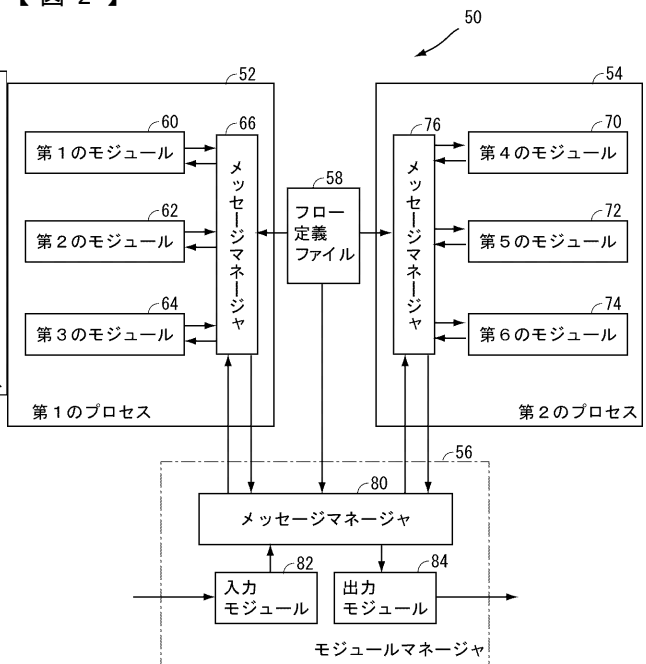
50

106 フロー管理テーブル作成処理部、110 イベントキュー、112 プロセス内
メッセージ配信部

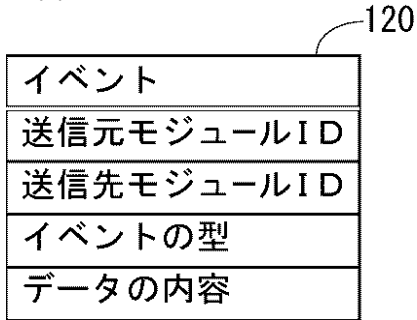
【図1】



【図2】



【図 3】



【図 4】

58

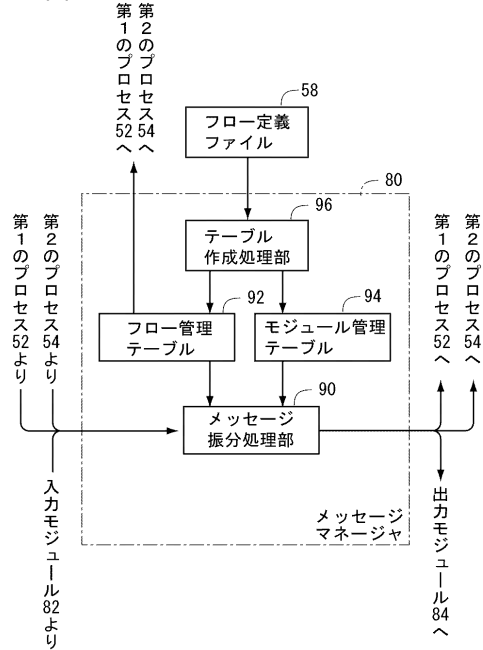
フロー定義ファイル	
process	どのプロセスにどのモジュールが含まれるか
arc	どのモジュールからどのモジュールにどんなイベントが送られるか
file	外部からどのようなデータ（イベント）がシステムに送られ、結果がどのようなデータとして外部に出力されるか

【図 7】

94

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 5】

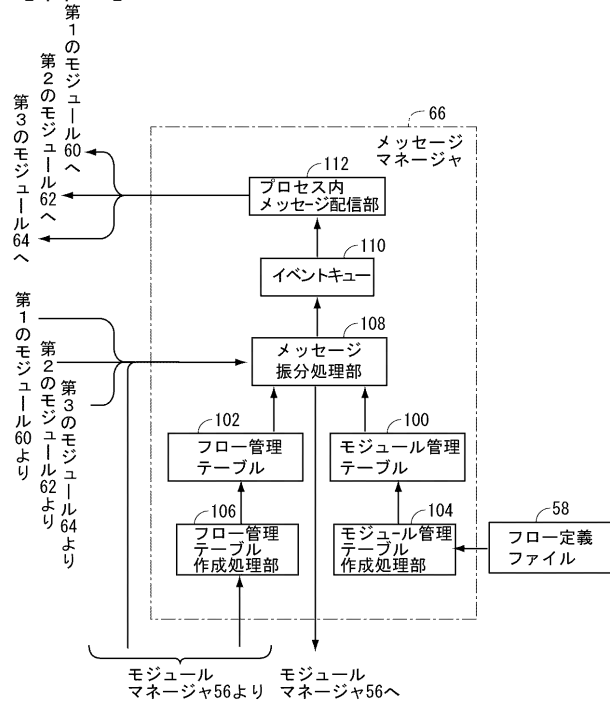


【図 6】

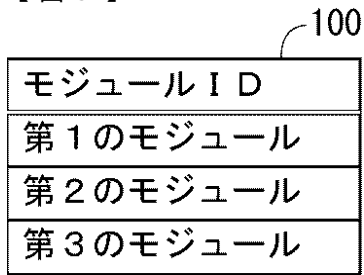
92

送信元モジュールID	イベントの型	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

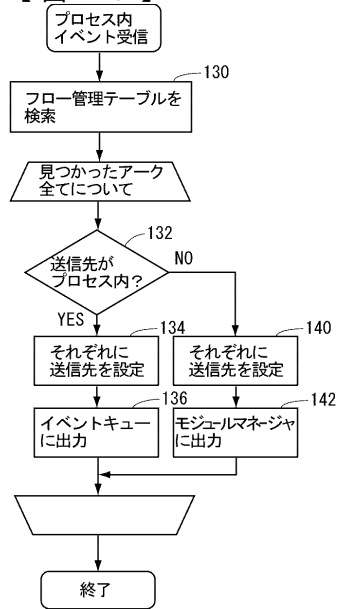
【図 8】



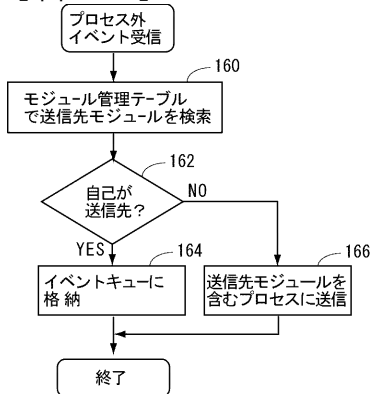
【図 9】



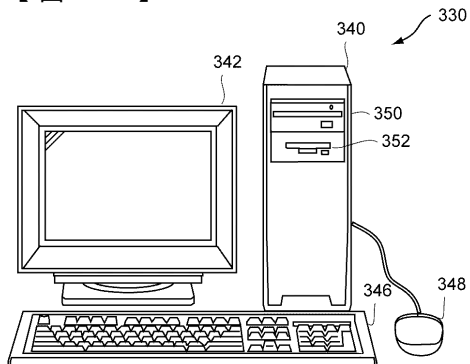
【図 10】



【図 11】



【図 12】



【図 13】

