(12)特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2002 年3 月7 日 (07.03.2002)

PCT

(10) 国際公開番号
WO 02/19104 A1

(51) 国際特許分類[7]: G06F 9/46, 9/44, 13/00, B25J 13/00, G05B 19/02, B25J 5/00, A63H 11/00

(21) 国際出願番号: PCT/JP01/07394

(22) 国際出願日: 2001 年8 月28 日 (28.08.2001)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願2000-258039 2000 年8 月28 日 (28.08.2000) JP

(71) 出願人 (米国を除く全ての指定国について): ソニー株式会社 (SONY CORPORATION) [JP/JP]; 〒141-0001 東京都品川区北品川6丁目7番35号 Tokyo (JP).

(72) 発明者; および
(75) 発明者/出願人 (米国についてのみ): 佐部浩太郎 (SABE, Kohtaro) [JP/JP]. 河本献太 (KAWAMOTO, Kenta) [JP/JP]. コスタ ガブリエル (COSTA, Gabriel) [ES/JP]; 〒141-0001 東京都品川区北品川6丁目7番35号 ソニー株式会社内 Tokyo (JP).

(74) 代理人: 小池 晃, 外(KOIKE, Akira et al.); 〒105-0001 東京都港区虎ノ門二丁目6番4号 第11森ビル Tokyo (JP).

(81) 指定国 (国内): CN, US.

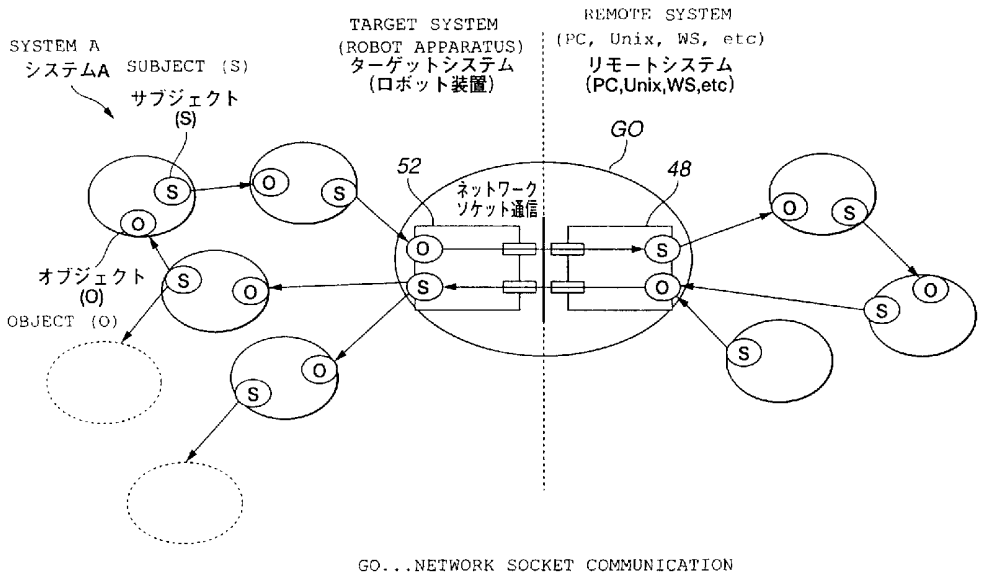
(84) 指定国 (広域): ヨーロッパ特許 (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

添付公開書類:
— 国際調査報告書

２文字コード及び他の略語については、定期発行される各PCTガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

(54) Title: COMMUNICATION DEVICE AND COMMUNICATION METHOD, NETWORK SYSTEM, AND ROBOT APPARATUS

(54) 発明の名称: 通信装置及び通信方法、ネットワークシステム、並びにロボット装置



(57) Abstract: A gateway object (48) for transmission/reception of data to/from an object of a robot apparatus (1) is assigned to a radio LAN PC card (41) of the robot apparatus (1). A gateway object (52) for transmission/reception of data to/from an object on a personal computer (32) is assigned to a network adapter (31) of a remote system (30). When the radio LAN PC card (41) and the network adapter (31) are interconnected by radio or cable, object-to-object communication between the gateway object (48) of the radio LAN PC card (41) and the gateway object (52) of the network adapter (31). Therefore object-to-object communication between an object of the robot apparatus (1) and an object of the personal computer (32) is made. As a result, a program is easily written.

[続葉有]

WO 02/19104 A1

(57) 要約:

ロボット装置（１）の無線ＬＡＮ用ＰＣカード（４１）は、ロボット装置（１）のオブジェクトに対してデータの送受信を行うゲートウェイオブジェクト（４８）が割り当てられ、リモートシステム（３０）のネットワークアダプタ（３１）は、パーソナルコンピュータ（３２）上のオブジェクトに対してデータの送受信を行うゲートウェイオブジェクト（５２）が割り当てられる。無線ＬＡＮ用ＰＣカード（４１）とネットワークアダプタ（３１）とが無線又は有線により接続されたときに、無線ＬＡＮ用ＰＣカード（４１）のゲートウェイオブジェクト（４８）とネットワークアダプタ（３１）のゲートウェイオブジェクト（５２）との間でオブジェクト間通信を行うことにより、ロボット装置（１）のオブジェクトとパーソナルコンピュータ（３２）のオブジェクトとの間でオブジェクト間通信がなされる。これにより、プログラム作成を容易にする。

明細書

通信装置及び通信方法、ネットワークシステム、並びにロボット装置

技術分野

本発明は、電子機器間で通信を行うための通信装置及び通信方法、情報処理を行う情報処理システム、プログラムを作成するためのプログラム作成方法、ネットワークシステム、並びにロボット装置に関する。

背景技術

近年、動物のように行動するペット型のロボット装置が提案されている。例えば、ロボット装置は、複数のプロセス間にてプロセス間通信を行うプログラム等によってその行動が規定されている。

ところで、ロボット装置のような複雑な処理をリアルタイムで行う組み込み機器において、ターゲットボード上で動作するソフトウェアの開発には大変手間がかかる。その理由として、以下のようなことが挙げられる。

(1) ターゲットボード上で動作するプログラムの状態を監視する手段が乏しい。

(2) 通常はターゲットボード上には内部状態を表示するようなデバイスはついていないため、なんらかの方法でリモートマシンと通信をして内部の情報を送り、リモートマシンのコンソールで確認をする手法がとられる。また、このような場合、ターゲットボード上のプログラムに通信用のコードを埋め込む必要がある。

(3) さらに、通信手法が一般化されていないような場合には、ユーザがデバッグしたいプログラム毎に通信用のコード作成しなければならない。このような手法には、最も単純なものがソースコード中に print 分を挿入してシリアル通信を用いてターミナルで確認する手法がある。この手法は、いわゆる print 文デバッグといわれるものであり、多くのシステムにおいて未だこの原始的な手法が使わ

れている。

(4) 多くのリアルタイムＯＳでは、ユーザのリアルタイムタスクはカーネル空間で実行されるためにそのタスクの不正な処理のためにシステムが破壊されてしまう。しかし、そのような破壊の原因を調べようとしても、システムが破壊されるために原因を知ることはできない。また、print 文デバッグを行っているとコンパイルしなおすことで、コードの配置が変わりバグの挙動が変化する可能性があり原因はますますわかりにくいものとなる。

(5) ユーザのリアルタイムタスクはカーネル空間にリンクされるために、ユーザプログラムの変更は、システム全体の再リンク、再実行になる。例えば、print 文を１行を入れただけでも全てやり直すことになる。これでは非効率的である。

発明の開示

そこで、本発明は、上述の実情に鑑みてなされたものであり、ソフトウェア（プロセス）部品間の変更を容易にする通信装置及び通信方法、情報処理システム、プログラム作成方法、ネットワークシステム、並びにロボット装置を提供することを目的としている。

本発明に係る通信装置は、第１の電子機器の情報処理プロセスに対してデータの送受信を行う情報処理プロセスを有する第１の通信手段と、第２の電子機器の情報処理プロセスに対してデータの送受信を行う情報処理プロセスを有する第２の通信手段とを備える。そして、第１の通信手段と第２の通信手段とが無線又は有線により接続されたときに、第１の通信手段のデータの送受信を行う情報処理プロセスと第２の通信手段のデータの送受信を行う情報処理プロセスとの間でプロセス間通信を行うことにより、第１の電子機器の情報処理プロセスと第２の電子機器の情報処理プロセスとの間でプロセス間通信がなされる。

このような構成を備えた通信装置により、第１の通信手段の情報処理オブジェクトと第２の通信手段のオブジェクトとの間のプロセス間通信があたかも同一システム上におけるプロセス間通信のように振る舞うために、第１の電子機器の情

報処理プロセスと第２の電子機器の情報処理プロセスとの間の通信は、同一システム上におけるプロセス間通信として取り扱われる。

また、本発明に係る通信方法は、第１の電子機器の情報処理プロセスに対してデータの送受信を行う情報処理プロセスを有する第１の通信手段と、第２の電子機器の情報処理プロセスに対してデータの送受信を行う情報処理プロセスを有する第２の通信手段とを無線又は有線により接続して、第１の通信手段のデータの送受信を行う情報処理プロセスと第２の通信手段のデータの送受信を行う情報処理プロセスとの間でプロセス間通信を行うことにより、第１の電子機器の情報処理プロセスと第２の電子機器の情報処理プロセスとの間でプロセス間通信を行う。

このような通信方法により、第１の通信手段の情報処理オブジェクトと第２の通信手段のオブジェクトとの間のプロセス間通信があたかも同一システム上におけるプロセス間通信のように振る舞うために、第１の電子機器の情報処理プロセスと第２の電子機器の情報処理プロセスとの間の通信は、同一システム上におけるプロセス間通信として取り扱われる。

また、本発明に係る情報処理システムは、情報処理プロセスにより制御される電子機器と、電子機器の制御に使用する情報処理プロセスが取り込まれて、当該情報処理プロセスを取り扱う情報処理装置と、電子機器と情報処理装置との間で無線又は有線により通信を行う通信手段とを備えておいる。そして、情報処理装置は、電子機器が動作されている際の当該電子器機の情報の情報処理プロセスと取り込んだ情報処理プロセスとの間で通信手段による通信を行い、電子機器の情報処理プロセスから送信されてくるデータを使用し、取り込んだ情報処理プロセスを取り扱う。

この情報処理システムにより、情報処理装置は、実環境にある電子機器が考慮された情報処理プロセスを取り扱う。

また、本発明に係るプログラム作成方法は、情報処理プロセスにより制御される電子機器が動作されている際の当該電子器機の情報処理プロセスと情報処理装置上の情報処理プロセスとの間で無線又は有線の通信手段による通信を行い、情報処理装置にて、電子機器の情報処理プロセスから送信されてくるデータを使用し、電子機器に使用する情報処理プロセスを作成するプログラム作成工程と、プ

ログラム作成工程にて作成した情報処理プロセスを電子機器に組み込むプログラム組み込み工程とを有する。

このようなプログラム作成方法により、情報処理装置は、実環境にある電子機器が考慮して情報処理プロセスを作成し、電子機器は、そのような実環境が考慮されて作成された情報処理プロセスが組み込まれ、当該情報処理プロセスに基づいて動作する。

さらに、本発明に係るネットワークシステムは、動作制御プログラムがオブジェクト間通信で構築されたロボット装置と、オブジェクト間通信で構築されたコンピュータシステムとが有線又は無線を経由してデータ通信を行うネットワークシステムにおいて、上記オブジェクト間通信と、上記ロボット装置と上記コンピュータシステムとの間でのネットワーク通信におけるプロトコル変換を動的に行うゲートウェイオブジェクトを有することを特徴とする。

また、本発明に係る通信方法は、オブジェクト間通信を有して構築されたロボット装置と、オブジェクト間通信を有して構築されたコンピュータシステムとが有線又は無線を経由してデータ通信を行う通信方法において、上記オブジェクト間通信と、上記ロボット装置と上記コンピュータシステムとの間でのネットワーク通信におけるプロトコル変換を行うことを特徴とする。

またさらに、本発明に係るロボット装置は、動作制御プログラムがオブジェクト間通信で構築されたロボット装置であって、オブジェクト間通信で構築されたコンピュータシステムとの間で有線又は無線を経由してデータ通信を行うロボット装置において、上記オブジェクト間通信と、上記コンピュータシステムとの間でのネットワーク通信におけるプロトコル変換を動的に行うゲートウェイオブジェクトを有することを特徴とする。

図面の簡単な説明

図１は、本発明の実施の形態に係るロボット装置の構成を示す斜視図である。

図２は、ロボット装置の構成を示すブロック図である。

図３は、４足型におけるソフトウェア構造を示す略線図である。

図４は、タイヤ型におけるソフトウェア構造を示す略線図である。

図５は、４足型における姿勢の変化を示す状態遷移図である。

図６は、入力と状態の関係を示す図表である。

図７は、デザインロボットの処理手順を示すフローチャートである。

図８は、デザインファイルの記述を示す図表である。

図９は、結合情報を示す図表である。

図１０は、ブート後におけるオブジェクトマネージャ、バーチャルロボット、デザインロボットの動作を示すタイムチャートである。

図１１は、プラグイン、プラグアウト等におけるオブジェクトマネージャ、バーチャルロボット、デザインロボットの動作を示すタイムチャートである。

図１２は、オブザーバとサブジェクトとの関係を示す略線図である。

図１３は、マルチオブザーバにおけるオブザーバとサブジェクトとの関係を示す略線図である。

図１４は、マルチサブジェクトにおけるオブザーバとサブジェクトとの関係を示す略線図である。

図１５は、オブジェクトマネージャとオブザーバ、サブジェクトの関係を示す略線図である。

図１６は、結合情報を受け取った場合におけるオブジェクトマネージャの動作を示すタイムチャートである。

図１７は、ドゥスタート以降のシーケンスを示すタイムチャートである。

図１８は、プラグイン、プラグアウト、状態変更におけるシーケンスを示すタイムチャートである。

図１９は、実施の形態のネットワークシステムの構成を示すブロック図である。

図２０は、実施の形態のネットワークシステムを構成するロボット装置とリモートシステムの要部の構成を示すブロック図である。

図２１は、port.cfg ファイルの記述事項を示す図である。

図２２は、接続前のオブジェクト群間の状態を示す図である。

図２３は、ゲートウェイオブジェクトにより接続されたオブジェクト群を示す図である。

図２４は、ロボット装置のソフトウェアを構成するSoccerDogオブジェクトを含めたオブジェクト群を示すブロック図である。

図２５は、コネクションファイルの前半部分を示す図である。

図２６は、コネクションファイルの中間部分を示す図である。

図２７は、コネクションファイルの後半部分を示す図である。

図２８は、ロボット装置のより具体的な例を示す図である。

発明を実施するための最良の形態

以下、本発明の実施の形態について図面を用いて詳しく説明する。この実施の形態は、本発明を、オブジェクト指向によりソフトウェア（オブジェクト）が構築されたペット型のロボット装置が、有線又は無線による通信により接続されたネットワークシステムに適用したものである。

ペット型のロボット装置は、オブジェクト間通信（プロセス間通信）による種種の処理結果として、例えば動物に似た仕草等を出現するようになされている。なお、ネットワークシステムに接続されるロボット装置は、ペット型のロボット装置に限定されないことはいうまでもなく、本発明の意義に応じて構成されることはいうまでもない。ロボット装置の構成等を先ず説明して、その後、本発明が適用されたネットワークシステムを詳述する。

（１）ロボット装置の構成

（１－１）全体構成

ロボット装置１は、図１に示すように、いわゆるペット型のロボット装置であり、全体が犬に似た形状であり、４本足により歩行できるようになされている。すなわちロボット装置１は、本体２に移動ユニット３が取り付けられ、この本体２が頭部２Ａと胴体部２Ｂとにより構成される。

頭部２Ａは、マイク４、タッチセンサ５、立体視可能なテレビジョンカメラ６、遠隔制御に係る赤外線の受信部７等が配置され、これらにより各種情報を取得できるようになされている。また、頭部２Ａは、画像表示部９、スピーカ１０が配置され、これらにより種々の情報を出力できるようになされている。なお、頭部

２Ａの構成は、このような構成からなることに限定されないことはいうまでもなく、すなわち例えば、画像表示部９を、ＬＥＤ（Light Emitting Diode）により構成して、表情を表現するようにすることもできる。

胴体部２Ｂは、背中の部分に操作子１１が配置され、また尻尾に相当する箇所にアンテナ１２が配置されるようになされ、これらを介してユーザの操作を検出できるようになされている。また、胴体２Ｂは、スロットを有し、このスロットにＩＣカードを装着してソフトウェアのバージョンアップ等を実行できるようになされている。

さらに、胴体部２Ｂは、頭部２Ａを介して入出力する各種情報、操作子１１等の操作情報を処理して全体の動作を制御するコントローラ１５、各部に電源を供給する電源ユニット１４、アンテナ１２を介して各種情報を送受する通信ユニット１６等が配置され、下部にバッテリ１７が配置されるようになされている。

これに対して移動ユニット３は、それぞれ関節に相当する箇所にアクチュエータ、角度検出センサを配置してなる４つの足を配置して形成され、本体２とシリアルバスにより接続され、本体２の制御により動作するようになされている。これによりロボット装置１は、４足歩行により移動できるようになされている。

さらに、この移動ユニット３は、後足が本体２より取り外すことができるようになされている。ロボット装置１は、この後足に代えて、タイヤを備えた移動ユニットを取り付けることができるようになされている。これによりこのロボット装置１は、ユーザの好みに応じて４足により移動する形態（以下、４足型という。）と、タイヤにより移動する形態（以下、タイヤ型という。）とで形態を変更できるようになされている。

図２は、このロボット装置１の構成を示すブロック図である。ロボット装置１においては、処理プログラムを記録したプログラムメディア２０がペリフェラル２１を介して中央処理ユニット２２に接続され、この中央処理ユニット２２によりプログラムメディア２０に記録された処理手順を実行するようになされている。また、中央処理ユニット２２は、ペリフェラル２１を介して移動ユニット３の各アクチュエータ、センサ、さらにはテレビジョンカメラ６、操作子１１等であるロボット装置コンポーネント２４、２５、２６と接続され、これにより全体の動

作を制御できるようになされている。

また、中央処理ユニット２２は、同様に、ペリフェラル２１を介して電源ユニット１４を構成する電源制御用の１チップマイコン２７と接続され、この１チップマイコン２７の動作を制御してバッテリ１７より全体に電源を供給し、さらには省電力モードにおいてバッテリ１７に代えてボタン電池２９より全体の電源を供給するようになされている。

（１－２）ソフトウェア構成

図３は、ロボット装置１の４足型におけるソフトウェア構成をレイヤにより示す略線図であり、図４は図３との対比によりタイヤ型におけるソフトウェア構成を示す略線図である。このソフトウェア構成において、最も下位層はデバイスドライバレイヤであり、各種デバイスドライバが配置される。ここでデバイスドライバは、移動ユニット３のセンサ、アクチュエータ、さらには頭部２Ａ等に配置した各種ロボット装置コンポーネントとの間で各種情報を入出力し、これらの処理のためのソフトウェアオブジェクトを備えるようになされている。

このうちバッテリマネージャデバイスドライバ（バッテリマネージャＤＤ）は、電源ユニット１４の１チップマイコン２７との間のデータ通信によりバッテリ１７の残量を検出して定期的に上位のロボットシステムソフトウェアに通知する。また、中央処理ユニット２２等のクロックを管理し、ロボットシステムソフトウェアの指示によりこのクロックの周波数を低減すると共に、不必要なデバイスの動作を停止し、これにより全体の消費電力を低減する。

また、ロボットシリアルバスデバイスドライバ（ロボットシリアルバスＤＤ）は、本体２に配置されたシリアルバスのマスター制御用のハードウェアとの間のデータ通信により、シリアルバスに接続された移動ユニット３のセンサ情報等を取り込んで上位のロボットシステムソフトウェアに通知し、またこれとは逆にロボットシステムソフトウェアより出力されるアクチュエータの制御データ、オーディオ信号等を各デバイスに送出する。

また、ロボットシリアルバスデバイスドライバは、システムの起動時、シリアルバスに接続されたデバイスの変更（追加、削除である）を検出する。さらに、この変更に係るデバイス固有の情報を取り込むことにより、ロボット装置１の現

在の形態を示す情報をロボットシステムソフトウェアに出力する。

このデバイスドライバレイヤの上位には、ロボットシステムソフトウェアが配置される。ここでロボットシステムソフトウェアは、バーチャルロボット、デザインロボット、パワーマネージャ、オブジェクトマネージャ等のソフトウェアオブジェクトより構成される。

このうちバーチャルロボットは、ロボットシリアルバスデバイスドライバとの間のデータ通信において、各デバイス固有のフォーマットによるデータをロボット装置の一般的なフォーマットに変換して送受する。具体的に、例えばモータ制御用のデータについて、デバイス固有のフォーマットにおいてはポテンションメーターであるセンサ出力をアナログディジタル変換して得られる１０ビットにより表現するのに対し、バーチャルロボットは、この表現によるデータを最下位の１ビットが０．００１度を示すデータに変換してロボット装置１における一般的なフォーマットとして出力する。また、バーチャルロボットは、このようなフォーマット変換によるデータの送受に加えて、例えばテレビジョンカメラ６より取得したイメージデータを上位のソフトウェアに出力する。

また、バーチャルロボットは、ロボットシリアルバスデバイスドライバよりロボット装置１の現在の形態を示す情報を受け取り、この情報を取りまとめる。これによりバーチャルロボットは、ロボット装置１全体でどの様なロボットコンポーネントがどの様な順番で接続されているのかを示す結合情報（CPC（Configurable Physical Component）Connection Information）を管理し、この結合情報（CPC Connection Information）をデザインロボットに通知する。

また、バーチャルロボットは、シリアルバスに接続されたデバイスに変更があった場合、デバイスドライバのロボットシリアルバスデバイスドライバより通知を受け、デザインロボットに通知する。

デザインロボットは、バーチャルロボットより結合情報（CPC Connection Information）を受け取り、事前に記録した結合テンプレート情報と順次比較することにより、現在のロボット装置１の形態に適したテンプレートを選択する。さらに、デザインロボットは、この選択したテンプレートに応じて、上位の形態依存ソフトウェアを現在の形態に適した形態依存ソフトウェアに更新するように、

オブジェクトマネージャに指示する。

なお、この実施の形態において、結合テンプレート情報は、デザインファイルにデザインデータとして記述されるようになされている。

オブジェクトマネージャは、デザインロボットの指示により、デザインロボットより通知されるオブジェクト、各オブジェクトの結合に関する情報を用いて形態依存ソフトウェアを現在の形態に適した形態依存ソフトウェアに更新する。すなわち、オブジェクトマネージャは、形態依存ソフトウェアを構成する全てのオブジェクトについて、動作の停止、オブジェクト間通信の結合解除、オブジェクトの破壊、リソースの開放を指示する。さらに新たに必要なオブジェクトのロード、初期化し、さらにはオブジェクト間通信のための結合構築、これらオブジェクトの起動を指示する。なお、ここで、オブジェクト間の接合とは、オブジェクト間でデータを送受信するための結合を意味する。

パワーマネージャは、バッテリマネージャデバイスドライバとの間でデータ通信し、上位のソフトウェア（形態独立ソフトウェア又は形態依存ソフトウェア）の指示に従って、バッテリマネージャデバイスドライバにクロックの切り換えを指示し、またオブジェクトの動作の停止等を指示する。

このロボットシステムソフトウェアの上位には、形態依存ソフトウェアが配置され、さらにその上位に形態独立ソフトウェアが配置される。形態依存ソフトウェアは、このロボット装置１の形態に応じて変更されるソフトウェアであり、形態独立ソフトウェアは、これとは逆に、ロボット装置の形態に係わらずに固定されたこのロボット装置１の上位のソフトウェアである。このため形態依存ソフトウェアには、ロボット装置１の形態に依存した各種オブジェクトが存在する。この実施の形態においては、形態依存ソフトウェアの変更により４足型とタイヤ型とでそれぞれ形態に適したソフトウェアを全体として簡易に構成できるようになされている。

すなわち、４足型において、この形態依存ソフトウェアは（図３）、移動ユニット３に関連するものとしてコマンドコンバータ、モーションネットワーク、モーションリプレイ、ウォーキングパターンジェネレータ、ビジョン等のオブジェクトにより構成される。

コマンドコンバータは、形態に依存しない形態独立ソフトウェアからのコマンドをロボット装置１の形態に適したコマンドに変換する。すなわち、形態独立ソフトウェアから移動ユニット３に関するコマンドとして、それぞれ睡眠（Sleep）、休憩（Rest）、起床（Wakeup）、移動（Move）の姿勢を指示するコマンドが入力されると、コマンドコンバータは、この４足型において、これらのコマンドをそれぞれ睡眠中（Sleeping）、お座り（Sitting）、起立（Standing）、歩行（Walking）の各姿勢を指示するコマンドに変換する。

モーションネットワークは、コマンドコンバータにより出力されるこれら姿勢を示すコマンドより、図５に示す遷移に従って、この遷移図の矢印に対応するオブジェクトのメソド（エントリ）を起動する。すなわち、睡眠中の状態で歩行のコマンドが入力された場合、お座り、起立、歩行と姿勢を順次変化させるように、モーションリプレイオブジェクトの対応するエントリを順次起動する。なお、この場合、最後の歩行の状態に辿り着いたときには、この状態を示す自己ループに対応するウォーキングパターンジェネレータのエントリを起動する。

モーションリプレイは、モーションネットワークによるエントリの起動により、対応する姿勢変化用の関節角度指令値を出力する。具体的に、モーションリプレイは、登録された離散的な関節角度指令値をキーデータとして保持し、このキーデータを用いた補間演算処理により一連の角度指令値を出力する。

ウォーキングパターンジェネレータは、形態独立ソフトウェアにより指定される方向に移動するように、各関節角度指令値を計算して出力する。

ビジョンは、例えば画像処理用のオブジェクトであり、テレビジョンカメラ６で取得したイメージデータをバーチャルロボットオブジェクトより受け取り処理する。ビジョンは、これにより例えば赤色の物体を識別し、このロボット装置１を基準にしたこの赤色の物体の位置、見かけの大きさを検出して出力する。

これに対してタイヤ型において、形態依存ソフトウェアは（図４）、移動ユニット３に関するオブジェクトとして、４足型と共通のビジョンのオブジェクト、このタイヤ型に特有のコマンドコンバータ、モーションコマンダ、モーションリプレイ２、ホイールパターンジェネレータのオブジェクトを有する。

このうちコマンドコンバータは、４足型におけるコマンドコンバータと同様に、

形態に依存しない形態独立ソフトウェアからのコマンドをロボット装置１の形態に適したコマンドに変換する。この場合、ロボット装置１においてはタイヤ型であることにより、コマンドコンバータは、それぞれ睡眠（Sleep）、休憩（Rest）、起床（Wakeup）、移動（Move）の姿勢を指示するコマンドを、睡眠中（Sleeping）、休憩（Rest）、待機（Ready）、移動（Go）のコマンドに変換する。

モーションコマンダは、コマンドコンバータの出力を受け、タイヤ用の制御コマンドと関連する関節角度指令値を生成する。すなわち、モーションコマンダは、睡眠中（Sleeping）のコマンドが入力された場合、関節角度指令値を中立（neutral）に設定する。ここで、中立（neutral）においては、何らモータを駆動、制動しない状態であり、これにより移動ユニットを構成する両腕にあっては、垂れ下がった状態に保持され、またタイヤにあっては停止状態に保持される。

これに対して休憩（Rest）のコマンドが入力された場合、モーションコマンダは、両腕にあっては肘の部分で９０度折り曲げて前に差し出すように、また、頭部２Ａにあっては顔を正面に向けるように、関節角度指令値を生成する。なおタイヤは停止したままである。

また、待機（Ready）のコマンドが入力された場合、モーションコマンダは、両腕にあっては肘を伸ばして前に差し出すように、また頭部２Ａにあっては顔を正面に向けるように関節角度指令値を生成する。なお、この場合もタイヤは停止したままである。さらに、移動（Go）のコマンドにあっては、待機（Ready）の状態と同様に両腕を保持したまま、タイヤが前進方向に回転するように、指令値を出力する。

モーションリプレイ２は、モーションコマンダの指示によりタイヤ型に固有の動作に関して、タイヤ以外の動作（例えば両腕の動作である）を制御する。ホイールパターンジェネレータは、モーションコマンダの制御によりタイヤの制御情報を生成する。

形態独立ソフトウェアは、ロボット装置１の形態に依存しないソフトウェアであり、形態設定オブジェクト等を有する。ここで形態設定オブジェクトは、形態依存ソフトウェアよりロボット装置１の形態に依存しない情報を受け取り、この

情報によりロボット装置１の形態に依存しない上述したようなロボット装置１の姿勢を示すコマンド（睡眠等である）を形態依存ソフトウェアに出力する。

すなわち例えば、ビジョンオブジェクトより通知される赤色の物体の有無とその位置情報に基づいて、１分以上赤色の物体の存在が検出されなかった場合、図６に示すように、睡眠（Sleep）の状態にあってはその状態を維持するように、休憩（Rest）の状態にあっては睡眠（Sleep）に、起床（Wakeup）の状態にあっては休憩（Rest）に、移動（Move）にあっては起床（Wakeup）の状態に遷移するようにコマンドを発行する。

また、赤色物体が所定値より小さな大きさで検出され（ロボット装置の設定による値１０以下）、これが１０秒以上続いた場合、睡眠（Sleep）の状態にあっては休憩（Rest）の状態に、休憩（Rest）の状態にあっては起床（Wakeup）に、起床（Wakeup）の状態にあっては移動（Move）に遷移するようにコマンドを発行し、また移動（Move）にあっては移動（Move）を継続するようにコマンドを発行する。

さらに、赤色物体が所定値以上の大きさで検出された場合、睡眠（Sleep）の状態にあっては（Wakeup）の状態に遷移するように、休憩（Rest）及び起床（Wakeup）の状態にあっては移動（Move）に遷移するように、また移動（Move）にあっては移動（Move）を継続するようにコマンドを発行する。

これらによりロボット装置１では、最も上位のソフトウェアが、形態に依存しないように形成され、この形態に依存しないソフトウェアとの間では形態に依存しない形式によりデータを送受するようになされている。これによりこのロボット装置１では、形態に応じて、この形態に依存するソフトウェアである形態依存ソフトウェアだけを変更して、形態に応じてソフトウェアを簡易に変更することができるようになされている。

（１－３）形態依存ソフトウェアの変更

ロボット装置１において、この形態依存ソフトウェアは、デザインロボットの指示によるオブジェクトマネージャの処理により更新される。このとき、形態依存ソフトウェアを構成するオブジェクトのロード、アンロード、オブジェクト間通信のための結合再構築により形態依存ソフトウェアを更新する。

このためデザインロボットは、バーチャルロボットよりロボット装置１の形態を特定する結合情報（CPC Connection Information）の通知を受け、この結合情報（CPC Connection Information）に基づいてオブジェクトマネージャに形態依存ソフトウェアの更新を指示すると共に必要な情報を提供する。

また、バーチャルロボットにおいては、下位層であるロボットシリアルバスデバイスドライバを介してシリアルバスに接続された各デバイスの情報を取得し、この情報より結合情報（CPC Connection Information）を作成してデザインロボットに通知する。

このためこのロボット装置１においては、各ロボットコンポーネントのメモリに所定の情報が記録され、ロボットシリアルバスデバイスドライバを介して実行されるデータ通信により、バーチャルロボット１においてシリアルバスに接続されたデバイスの情報をその位置情報と共に取得できるようになされている。

ここでロボット装置１においては、シリアルバスに接続されるテレビジョンカメラ、スピーカ、マイク、各種アクチュエータ、センサ等が基本的な構成要素（CPC Primitive）であり、リンク状態に設定されたこれら基本的な構成要素の複数の組み合わせにより各部品（CPC Model）が構成される。すなわち例えば、部品である１つの足は、３個のモータ、１個のスイッチによる複数の基本的な構成要素の特定なリンク状態により構成される。

ロボット装置１においては、これら各部品に固有の識別データが割り当てられ、この識別データによりシリアルバスに接続された各構成要素を特定できるようになされている。なお、この識別データは、部品を作成した製造工場のＩＤと製品番号に対応するＩＤとにより構成される。また、この識別データに対応するように、各構成要素の情報、部品における各構成要素の位置情報が部品中における各構成要素のリンク情報、各構成要素の属性情報等により設定され、これらの情報が各部品の識別データと共に各部品のメモリに保持されるようになされている。なお、この実施の形態では、部品側にてメモリ容量が少ない場合には、各部品側に識別データのみ記録され、構成要素の情報、部品における構成要素の位置情報は、本体２側に保持するようになされている。

バーチャルロボットは、システム起動時、デバイスの変更時、ロボットシリア

ルバスデバイスドライバより通知されるこれら識別データ、位置情報等により、ツリー構造によるシリアルバスを順次辿って、各基本的な構成要素（CPC Primitive）と各構成要素の結合の順番を示すデータ構造とにより結合情報（CPC Connection Information）を作成してデザインロボットに通知する。かくするにつきバーチャルロボットより通知される結合情報（CPC Connection Information）にあっては、形態に対応する構成要素情報（CPC Primitive Location Information ）により構成される。

デザインロボットは、バーチャルロボットより通知される結合情報（CPC Connection Information）を基準にしてデザインファイルを参照することにより現在のロボット装置１の形態に対応する結合テンプレート（後述するラベルである）を選択する。さらに、デザインロボットは、この選択したテンプレートに応じて、現在の形態に適した形態依存ソフトウェアへの更新をオブジェクトマネージャに指示する。

ここで、デザインファイルは、テキストデータ形式の記述であり、ロボット装置１の形態毎に、各部品の構成要素情報（CPC Primitive Location Information ）群に対してラベルを付加して記述される。また、構成要素情報（CPC Primitive Location Information）は、各部品の基本的な構成要素（CPC Primitive）と各構成要素の位置情報（CPC Coordinate Locator）により構成される。このうち位置情報（CPC Coordinate Locator）は、ロボット装置１に設定された基準位置からの座標であり、一般的なリンク座標系により表記され、座標系変換のための回転行列と位置ベクトルの縦続接続により構成されるようになされている。

また、ラベルは、対応するオブジェクトの特定、このオブジェクトのロード、オブジェクト間通信の構築に必要なデータを特定するものであり、デザインラベル（Design Label）、又はデザインラベルに比して汎用的な理解容易な記述によるバーチャルラベル（Virtual Label）、デザインラベル（Design Label）とバーチャルラベル（Virtual Label）との複合型のラベル（Composite Label）が適用される。なお、ビジュアルラベルにあっては、デザインロボットによりデザインラベルに置き換えられて利用に供されるようになされている。

次式は、このうちのデザインラベルの記述を示すものである。デザインラベルは、１つ以上の構成要素情報（CPC Primitive Location Information ）、又はデザインラベル（Design Label）により構成される。

```
Design Label (
        CPC Primitive Location Information 又は Design Label ;
        .....
)
```

・・・ (1)

これに対して次式は、バーチャルラベル（Virtual Label）の記述を示すものである。バーチャルラベルは、デザインラベル（Design Label）、構成要素（CPC Primitive）、構成要素（CPC Primitive）の位置情報（CPC Coordinate Locator）により構成される。

```
Virtual Label (
        Design Label :
        CPC Primitive :
        CPC Coordinate Locator :
)
```

・・・ (2)

これに対して次式は、複合型のラベル（Composite Label）の記述を示すものである。

```
Composite Label (
            Composite Label ;
            or/and Design Label :
            or/and Virtual Label ;
)
```

・・・ (3)

かくするにつきデザインロボットは、通知された結合情報（CPC Connection Information）により特定される構成要素情報（CPC Primitive Location Information）において、特定される位置情報（CPC Coordinate Locator）よりツリー構造の先端側にバーチャルラベル（Virtual Label）内で定義された位置情報（CPC Coordinate Locator）が存在する場合、このバーチャルラベル（Virtual Label）をデザインラベル（Design Label）に置き換え、また、このようにして定義したデザインラベル（Design Label）を有効なものと設定する。

これらにより、結合情報（CPC Connection Information）によりこのデザインファイルをアクセスして現在のロボット装置を形態（４足型、タイヤ型等の形態）を特定できるようになされ、また、形態依存ソフトウェアの更新に必要な各種データを取得できるようになされている。

図７は、このデザインロボットの処理手順を具体的に示すフローチャートである。デザインロボットは、バーチャルロボットより結合情報（CPC Connection Information）が通知されると、ステップＳＰ１からステップＳＰ２に移り、この結合情報（CPC Connection Information）によりデザインファイルをアクセスし、これにより現在のロボット装置１の形態を特定する。

続いてデザインロボットは、ステップＳＰ３に移り、バーチャルロボットより通知された結合情報（CPC Connection Information）による構成要素情報（CPC Primitive Location Information）とデザインファイルに保持された結合情報（CPC Connection Information）による構成要素情報（CPC Primitive Location Information）とを比較する。

続いてデザインロボットは、ステップＳＰ４に移り、この比較結果に基づいて、デザインファイルをアクセスし、現在の形態に対応するオブジェクト、オブジェクト間通信の再構築に必要なデータを特定するデザインラベルを検出する。

続いてデザインロボットは、ステップＳＰ５に移り、検出したデザインラベルより結合ファイル（Connection File）をアクセスし、ラベルに対応する結合情報（Connection Data）を検出することにより、対応するオブジェクトの特定、オブジェクト間通信の再構築に必要なデータを取得する。かくするにつき結合ファイル（Connection File）は、ラベルとの対応により結合情報（Connection Data）

を記録したファイルであり、結合情報（Connection Data）は、対応するオブジェクトの特定、オブジェクト間通信の再構築に必要なデータである。

続いてデザインロボットは、ステップＳＰ６に移り、この結合情報（Connection Data）をオブジェクトマネージャに送出し、これにより形態依存ソフトウェアの更新を指示した後、ステップＳＰ７に移ってこの処理手順を終了する。

図８は、デザインファイルの一部の記述を示す図表である。このデザインファイルにおいて、DRX は、このロボット装置１の形態を示し、４ Legged は、４足型の形態であることを示す記述である。また Head、RightFrontLeg、LeftFrontLeg、RightRearLeg、LeftRearLeg は、それぞれ頭部、右前足、左前足、右後足、左後足を示し、これにより第１文において４足型が頭部、右前足、左前足、右後足、左後足により構成されることが特定される。

また、第２文における Wheel はタイヤ型を示し、第１文との対比によりこの第２文においてタイヤ型が頭部、右前足、左前足、右後輪、左後輪により構成されることが特定される。また、第３文においては、頭部にカメラリンクの配置が記述され、第４文にはこのカメラリンクのバーチャルラベル（Virtual Label）が記述されるようになされ、第５文以降においては、右前足、左前足等について、位置情報（CPC Coordinate Locator）がツリー形式により記述されるようになされている。

これに対して図９は、結合情報（CPC Connection Information）の一部の記述を示す図表である。結合情報は、デザインファイルと同様にテキストデータ形式により記述される。この記述において、joint はインターフェースを示し、この部分が他の構成要素との結合箇所であることを示すものである。この場合、第２行目から第５行目までの記載により、CO、CO、CO、CO、CO による５つのハブ等を介してテレビジョンカメラが接続されていることが判る。

これによりこの図９の例による結合情報（CPC Connection Information）にあっては、図８について上述した記述である４足型と一致することになる。

これに対して次式は、結合ファイル（Connection File）の一部の記述を示す図表である。結合ファイルは、テキストデータ形式により記述され、この記述にお

いて、第１文及び第２文は、それぞれ４足型及びタイヤ型における足及びタイヤに対応するオブジェクトと、このオブジェクトにおけるオブジェクト間通信の構築に必要なデータが記述される。

すなわち、第１文にあっては、サービス名としてオブジェクト名である MONet とデータ形式 MONetOutData、サブジェクトを示すＳの記述が記述される。さらに、同様のサービス名としてオブジェクト名である MONetReplay、データ形式 MONetOutData 等が記述され、最後にオブザーバを示すＯの記述が割り当てられる。また、続く行において、同様のサブジェクト及びオブザーバ、データ形式等が記述される。

```
DRX 4Legged (
   MoNet.MoNetOutData..S,MoNetReplay.MoNetOutData...0
   MoNet.MoNetOutData..S,WalkingPatternGenerator.MoNetOutData...0
)
DRX Whell (
   MotionConverter.MCData..S,MotionReplay2.MCData...0
   MotionConverter.MCData..S,Whell.MCData...0
)
```

・・・ (4)

これによりこの図８の例の場合、デザインロボットは、この DRX 4Legged による第１文の記述より結合情報（Connection Data）（DRX 4Legged 以降の括弧書の記述）を抽出してオブジェクトマネージャに通知し、これにより形態依存ソフトウェアを更新するようになされている。

図１０は、オブジェクトマネージャ、バーチャルロボット、デザインロボットにおけるブート時の処理を示すタイムチャートである。ロボット装置１においては、電源が投入されると、又はリセットがされると、ブートの処理を開始する。このとき、ロボット装置１においては、各オブジェクトがファイルよりロードさ

れて形態依存ソフトウェアが構築され、このとき各オブジェクトマネージャの内部変数が初期化される。

続いてオブジェクトマネージャよりドゥイニシャライズ（DoInit）が各オブジェクトに送出され、このドゥイニシャライズ（DoInit）により各オブジェクトの問い合わせ用エントリ（Entry）がオブジェクトマネージャ等に登録される。これによりロボットシステムソフトウェアレイヤにおいて、各オブジェクト間通信が構築される。

続いてオブジェクトマネージャよりドゥスタート（DoStart）がバーチャルロボット、デザインロボットに順次送出され、バーチャルロボット、デザインロボットが動作を開始する。これによりバーチャルロボットにより結合情報（CPC Connection Information）が生成され、デザインロボットの要求によりこの結合情報（CPC Connection Information）がデザインロボットに渡され、結合情報（Connection Data）が生成される。さらに、この結合情報（Connection Data）がオブジェクトマネージャに渡され、この結合情報（Connection Data）により形態依存ソフトウェアが構築される。

これに対してシャットダウンにおいては、オブジェクトマネージャよりドゥストップ（DoStop）が各オブジェクトに送出され、このドゥストップ（DoStop）により各オブジェクトにおいては、全てのリクエストに対してエラーを通知して動作を停止する。統いてオブジェクトマネージャよリドゥデストロイ（DoDestroy）が各オブジェクトに送出され、これにより各オブジェクトにおいては、使用していたソースを開放し、またオブジェクトマネージャに登録したエントリを抹消して待機する。

これによりロボット装置１においては、形態が変更されて立ち上げられた場合でも、この変更された形態に適したソフトウェアにより動作できるようになされている。

これに対していわゆるプラグイン（Plug-In）、プラグアウト（Plug-Out）においては、図１１に示すように、オブジェクトマネージャよりドゥストップ（DoStop）が各オブジェクトに送出され、このドゥストップ（DoStop）により各オブジェクトにおいては、全てのリクエストに対してエラーを通知して動作を

停止する。

続いてオブジェクトマネージャよりブート時と同様にして順次ドゥスタート（DoStart）がバーチャルロボット、デザインロボットに順次送出され、バーチャルロボットにより結合情報（CPC Connection Information）が生成され、この結合情報（CPC Connection Information）がデザインロボットに渡されて結合情報（Connection Data）が生成され、この結合情報（Connection Data）により形態依存ソフトウェアが構築される。

これによりロボット装置１においては、プラグイン（Plug-In）、プラグアウト（Plug-０ ut）により動的に形態が変更された場合でも、この変更された形態に適したソフトウェアにより動作できるようになされている。

また、バッテリ１７が消耗した場合等にあっては、バッテリマネージャからの状態変更要求により、プラグイン（Plug-In）、プラグアウト（Plug-Out）の場合と同様に、ドゥストップ（DoStop）、ドゥスタート（DoStart）の処理が繰り返され、これによりバッテリ１７が消耗した場合にあっては、クロック周波数を低減すると共に不必要なデバイスの動作を停止してボタン電池２９により動作するように状態が変更され、これとは逆に充電されたバッテリ１７が装着された場合にあっては、クロック周波数を増大すると共に各デバイスの動作が開始されてバッテリ１７により動作するようになされている。

（１－４）オブジェクトマネージャの処理

オブジェクトマネージャは、このようにして実行されるデザインロボットからの通知により形態依存ソフトウェアを構成するオブジェクトを再構成し、これにより形態依存ソフトウェアを更新する。具体的に、オブジェクトマネージャは、デザインロボットからの通知によりオブジェクトをロード、アンロードすると共に、これらアンロード及びロードに対応するようにオブジェクト間通信を再構築し、これにより形態依存ソフトウェアを変更する。

このオブジェクト間通信を構築する際に、対応するオブジェクト名等がオブジェクトに登録されていなければならないとすると、各オブジェクトの独立性が害され、自由な形態に対応するようにオブジェクトを複数種類用意することが必要になる。このためこの実施の形態において、オブジェクトマネージャは、デザイ

ンロボットより出力される結合情報（Connection Data）に基づいてオブジェクト間通信を構築し、これによりオブジェクトの独立性を担保する。

そして、本発明は、オブジェクト間通信により構築されたロボット装置１（ターゲットボード或いはターゲットシステム）のオブジェクト群と、パーソナルコンピュータにプログラム作成等のために取り込んだロボット装置１のオブジェクトとをオブジェクト間通信によりデータ通信を可能とすることに特徴がある。後で詳述するが、ロボット装置１とパーソナルコンピュータとが無線又は有線による通信手段により接続された際に、それぞれの有するゲートウェイオブジェクトによるオブジェクト間通信が可能とされることにより、ロボット装置１上のオブジェクトとパーソナルコンピュータ上のオブジェクトとのオブジェクト間通信が構築されるといったようにである。

以下、オブジェクト間通信について具体的な説明をする。なお、以下の説明では、オブジェクト名の特定を避けて説明を簡略化し、すなわち例えば以下の説明はロボット装置１内のオブジェクト間通信について適用可能であり、また、ロボット装置１とパーソナルコンピュータの備えるオブジェクト（後述するゲートウェイオブジェクト）間のオブジェクト間通信について適用可能である。また、オブジェクト間通信は、非同期により実行するものとする。

図１２は、この実施の形態に係る２つのオブジェクト間におけるオブジェクト間通信の基本構成を示す略線図である。この実施の形態においては、レディ（Ready）とノーティファイ（Notify）とを用いてオブジェクト間通信を実行し、これによりオブジェクト（この例では ObjectB）の処理能力以上のデータを送出しないようになされている。

すなわち、図１２において、オブジェクトＡは、オブジェクトＡのメンバーであるクラス０のサブジェクトからオブジェクトＢのメンバーである０オブザーバクラスに対してデータを送出し、これによりオブジェクトＢのメソドがインヴォウク（invoke）される。ここでいう、サブジェクト及びオブザーバは、その属するオブジェクトにおいていわゆる代理プログラムを構成している。

この関係において、サブジェクトは、オブザーバよりレディ（Ready）の信号が送出されている場合にのみデータを送出する。なお、レディ（Ready）の信号

は、１回のデータにつき１回送る必要がある。これによりオブザーバ側であるオブジェクトＢにおいては、受け取ったデータの処理の完了を待って続くデータを受信することができるようになされている。

また、図１３は、マルチオブザーバにおけるオブジェクト間通信の基本構成を示す略線図である。この場合、サブジェクトであるオブジェクトＡは、全てのオブザーバに対してデータを配信することができ、また、オブジェクトＩＤにより特定される特定のオブザーバに対してのみにデータを配信することもできる。なお、この実施の形態においては、このようなマルチオブザーバの状態においても、レディ（Ready）の信号により対応するオブザーバに対してデータを送出する。

各オブジェクトにおいては、オブジェクトを特定するオブジェクトＩＤ、メソド（method）を特定するセレクタ番号（selector）（メソド番号である）を指定することにより、対応するメソドを起動（エントリ）し、オブザーバを特定して所望のデータを送出する。

図１４は、いわゆるマルチサブジェクトにおけるオブジェクト間通信の基本構成を示す略線図である。この場合、オブジェクトＡ及びＢがサブジェクトであり、オブジェクトＣがオブザーバである。オブザーバにおいては、このように複数のサブジェクトよりそれぞれデータを受信することができ、また、データを受信する毎にデータを処理するメソドがインヴォウク（invoke）されるようになされている。また、オブザーバは、サブジェクトＩＤにより特定される特定のサブジェクトに対してのみにレディ（Ready）の信号を送出して所望のデータを選択的に入力できるようになされている。

この実施の形態において、少なくとも形態依存ソフトウェアレイヤの属するオブジェクトにおいては、これら図１２～図１４を用いて説明したように、レディ（Ready）とノーティファイ（Notify）とを用いてオブジェクト間通信を実行し、また、必要に応じてマルチオブザーバ、マルチサブジェクトの結合形態により他のオブジェクトを結合できるようになされている。すなわち、複数のオブザーバを有するオブジェクトにあっては、これら複数のオブザーバに対応する数のコネクトエントリを有するようになされている。

図１５は、オブジェクトマネージャと各オブジェクトとの関係を示す略線図で

ある。各オブジェクトにおいては、それぞれオブジェクトＩＤ、セレクタ番号（Selector）によりメソドを特定してデータ交換し、それぞれ対応するサブジェクト、オブザーバとは何ら無関係に、基本的なメソドについてセレクタ番号０～３がそれぞれ割り当てられるようになされている。

ここでセレクタ番号０には、ドゥイニシャライズ（DoInit）が割り当てられ、各オブジェクトは、このドゥイニシャライズ（DoInit）の起動により初期化されるようになされている。また、セレクタ番号１には、ドゥスタート（DoStart）が割り当てられ、各オブジェクトは、このドゥスタート（DoStart）の起動により動作を開始するようになされている。さらに、セレクタ番号２には、ドゥストップ（DoStop）が割り当てられ、各オブジェクトは、このドゥストップ（DoStop）の起動により動作を停止するようになされている。また、セレクタ番号２には、ドゥデストロイ（DoDestroy）が割り当てられ、各オブジェクトは、このドゥデストロイ（DoDestroy）によりリソースを開放するようになされている。

さらに、各オブジェクトにおいては、これらオブジェクトマネージャからの要求、また他のオブジェクトからの要求に対して、適宜、リターン値によりオブザーバによるサービス名、セレクタ番号等を通知するようになされている。

これにより、デザインロボットより通知される結合情報（Connection Data）に基づいて、オブジェクトマネージャによりオブジェクトをロード、アンロードした後、オブジェクト間通信を再構築できるようになされている。

次式は、このデザインロボットより通知される結合情報（Connection Data）を示すものであり、サブジェクトのサービス名とオブザーバのサービス名をコロンで区切ってテキストデータ形式により記述される。この場合、サブジェクト側においては、オブジェクト名である FooB、データ形式 Image、対応する構成要素名による名称 RightCamera、サブジェクトを示すＳの記述がサービス名として記述される。また、オブザーバ側にあっては、オブジェクト名である FooA、データ形式 Image、対応する構成要素名による名称 RightCamera、オブザーバを示すＯの記述がサービス名として記述される。

FooB.Image.RightCamera.S,FooA.Image.RightCamera.0 ・・・ (5)

オブジェクトマネージャにおいては、オブジェクトをロードする際に、この結合情報（Connection Data）の記述よりロードするオブジェクト名を検出する。

さらに、オブジェクトマネージャは、スタックメモリ（Stack）、ヒープメモリ（heap）を確保してこの検出したオブジェクト名のオブジェクトを所定のファイルよりロードする。このときオブジェクトマネージャは、オペレーションシステム（ＯＳ，Operation System）よりオブジェクトＩＤを取得し、このオブジェクトＩＤを結合情報（Connection Data）に記述されたオブジェクト名と共に記録する。

これによりオブジェクトマネージャにおいては、このようにして登録したオブジェクトＩＤを用いて以下のドゥイニシャライズ（DoInit）、ドゥコネクト（DoConnect）、ドゥスタート（DoStart）等の処理を実行するようになされている。

すなわち、ドゥイニシャライズ（DoInit）において、取得したオブジェクトＩＤと、上述したセレクタ番号に基づいて、オブジェクトマネージャは、オブジェクトのロードにより取得した全てのオブジェクトに対してドゥイニシャライズ（DoInit）をコールする。各オブジェクトにおいては、このドゥイニシャライズ（DoInit）により内部変数等を初期化し、これによりオブジェクトマネージャにおいては、各オブジェクトを初期化する。

この初期化の処理において、オブジェクトマネージャは、各オブジェクトからの通知により、それぞれ各オブジェクトのサブジェクトとしてのエントリ（Control）、オブジェクトとしてのエントリ（Connect）を登録する。なお、この登録は、結合情報（Connection Data）におけるサブジェクト及びオブジェクトの記述に対応して、オブジェクトの名前とサブジェクト、オブジェクトの名前により構成される。

これに対してドゥコネクト（DoConnect）において、オブジェクトマネージャは、ドゥイニシャライズ（DoInit）における登録に基づいて、オブザーバを有す

る各オブジェクトのエントリ（Connect）に対して、それぞれ登録対象であるサブジェクトのサブジェクトＩＤ（オブジェクトＩＤ）、エントリ（Control）を通知する。これによりこの通知を受けたオブジェクトにおいて、通知されたサブジェクトＩＤ、エントリ（Control）を用いて対応するサブジェクトを呼び出し、ここにエントリ（Notify）が接続登録される。また、接続登録が要求されたサブジェクトよりレディ（Ready）がオブザーバに返され、これによりオブザーバと対応するサブジェクトとが結合される。

このようにしてオブザーバ及びサブジェクト間でオブジェクト間通信を構築するにつきオブジェクトマネージャにおいては、結合情報（Connection Data）で通知されたオブザーバに対して対応するサブジェクトのサブジェクトＩＤ（オブジェクトＩＤ）、エントリ（Control）を通知する。これにより、各オブジェクトにおいては、何ら結合対象のオブジェクトを明示的に示すことなく開発できるようになされ、また必要に応じてオブジェクトマネージャの指示により種々のオブジェクトと結合できるようになされ、これにより従来に比して高い独立性を確保できるようになされている。

また、このときオブジェクトマネージャにより結合情報（Connection Data）に従ってオブジェクト間通信を構築することにより、簡易かつ自在に形態依存ソフトウェアを更新して、形態に適してソフトウェアを簡易に変更できるようになされている。

これに対してドゥスタート（DoStart）において、オブジェクトマネージャは、上述したセレクタ番号１により各オブジェクトにドゥスタート（DoStart）を指示する。各オブジェクトにおいては、オブザーバを有する場合、ドゥコネクト（DoConnect）により取得したサブジェクトＩＤ、エントリ（Ready）を用いてサブジェクトにレディ（Ready）を送出し、これにより対応するサブジェクトよりデータ受信可能な状態が形成され、形態依存ソフトウェアにおいては、動作を開始するようになされている。

すなわち、マルチオブザーバによるサブジェクトにおいては、登録したオブザーバのうちのレディ（Ready）を送出しているオブザーバに例えばセンサ出力であるデータをノーティファイ（Notify）し、またオブザーバにおいては、受信し

たデータの処理を完了して次のデータを受信可能となるとレディ（Ready）を送出する。

これに対して上述したシャットダウン、プラグイン、プラグアウト、状態変更の場合、デザインロボットより通知される結合情報（Connection Data）が、以前デザインロボットより送出されて登録した内容と異なることにより、この場合オブジェクトマネージャは、上述したセレクタ番号２により各オブジェクトにドゥストップ（DoStop）を送出する。この場合オブザーバにあっては、エントリ（Ready）を解除する。

また、ドゥディスコネクト（DoDisconnect）において、オブジェクトマネージャは、サブジェクトとオブザーバとの間の通信接続を遮断する。この場合、オブジェクトマネージャは、オブザーバのエントリ（Connect）に対してドゥディスコネクト（DoDisconnect）のメッセージを送出することにより、オブザーバより対応するサブジェクトにエントリ（Control）の切断要求（Remove Observer）を発行させ、これにより通信接続を遮断する。

これに対してドゥデストロイ（DoDestroy）において、オブジェクトマネージャは、上述したセレクタ番号３により対応するオブジェクトにドゥデストロイ（DoDestroy）を送出し、これによりこのオブジェクトを破壊する。このときこのオブジェクトにおいては、ドゥイニシャライズ（DoInit）で実行した登録の解除を実行する。

オブジェクトアンロード（Object Unload）において、オブジェクトマネージャは、ドゥデストロイ（DoDestroy）により破壊したオブジェクトについて、スタックメモリ、ヒープメモリの領域を開放し、このオブジェクトをアンロードする。また、ロード時に登録したサブジェクトＩＤ、サブジェクト名を抹消する。

オブジェクトマネージャは、これらの制御により、例えば結合情報（Connection Data）をデザインロボットより受け取ると、図１６に示すシーケンスによりドゥスタート（DoStart）までの制御を実行する。

すなわち、オブジェクトマネージャは、メッセージにより結合情報（Connection Data）が送られると起動し、例えばこの結合情報に記述されたオブジェクトＡ及びオブジェクトＢをロードする。なお、ここではオペレーション

システムのシステム命令によりオブジェクトＡ及びオブジェクトＢをロードする。このときオブジェクトマネージャは、オブジェクトＡ及びＢのオブジェクトＩＤを取得して登録する。

続いてオブジェクトマネージャは、上述したようにしてこの取得したオブザーバＩＤ、セレクタ番号０によりオブジェクトＡ及びＢのドゥイニシャライズ（DoInit）を起動し、これによりオブジェクトＡ及びＢより、サブジェクトとしてのエントリ（Control）、オブジェクトとしてのエントリ（Connect）を取得して登録する。

続いて登録したエントリ（Connect）を用いてオブザーバであるオブジェクトＡのドゥコネクト（DoConnect）を起動し、サブジェクトであるオブジェクトＢとこのオブジェクトＡとを結合する。このようにして結合情報（Connection Data）に基づいてオブジェクト間通信が構築され、オブジェクトマネージャは、続いてオブジェクトＡ及びＢのドゥスタート（DoStart）を起動する。

図１７は、このドゥスタート（DoStart）以降のシーケンスを示すタイムチャートである。オブジェクトマネージャのドゥスタート（DoStart）によりオブジェクトＡ、オブジェクトＢ、他のオブジェクト等が起動すると、これらのオブジェクトにおいては、上述したレディ（Ready）とノーティファイ（Notify）とを用いてオブジェクト間通信を実行する。

すなわち、この場合、起動によりオブジェクトＡがオブジェクトＢのレディ（Ready）エントリにレディ（Ready）メッセージを送出し、オブジェクトＢによるオブジェクトＡのノーティファイ（Notify）エントリによりオブジェクトＢからオブジェクトＡにデータ等が送出される。また、このデータの処理中に、オブジェクトＡよりオブジェクトＢのレディ（Ready）エントリに未だメッセージが送出されていない場合には、オブジェクトＢによるオブジェクトＡのノーティファイ（Notify）エントリが登録され、オブジェクトＡによるオブジェクトＢのレディ（Ready）エントリによりデータ等が送出される。これによりこのオブジェクト間通信においては、オブジェクトＡのデータ処理能力を越えたデータの伝送が防止される。

図１８は、プラグイン、プラグアウト、状態変更におけるシーケンスを示すタ

イムチャートである。オブジェクトマネージャは、デザインロボットより通知される結合情報（Connection Data）が、以前デザインロボットより送出されて登録した内容と異なっていると、全てのオブジェクトにドゥストップ（DoStop）を送出し、全てのオブジェクトの動作を停止する。この場合、各オブジェクトA及びBにおいては、エントリ（Ready）を解除することにより、それ以後、自己のノーティファイエントリが起動されないようにする。

全てのオブジェクトのドゥストップ（DoStop）が完了すると、オブジェクトマネージャは、この場合、オブジェクトAにドゥディスコネクト（DoDisconnect）を通知してこのオブジェクトAを他のオブジェクトと切り離した後、このオブジェクトAのドゥデストロイ（DoDestroy）を起動する。これによりオブジェクトAのリソースを開放し、またドゥイニシャライズ（DoInit）で実行したオブジェクトBの登録を解除する。

必要なオブジェクトに対するドゥデストロイ（DoDestroy）が終了するとオブジェクトマネージャは、アンロードをオペレーションシステムのシステム命令により実行する。なお、ここで必要なオブジェクトの destructor が呼び出され、この場合はオブジェクトAの destructor が呼び出されることにより、ロード時に実行したオブジェクトAの登録を解除する。さらに、オブジェクトマネージャは、スタックメモリ、ヒープメモリを開放し、これによりオブジェクトAのアンロードを完了する。

統いてオブジェクトマネージャは、結合情報（Connection Data）に従ってオブジェクトCのロードを指示し、図１６について上述したと同様にして、順次ドゥイニシャライズ（DoInit）、ドゥコネクト（DoConnect）、ドゥスタート（DoStart）を起動する。これによりこの場合、動作中に、コンパイルすることなくオブジェクトA及びBによる構成がオブジェクトB及びCによる構成に変更されるようになされている。

以上、図１２乃至図１８を用いて、オブジェクト間通信等について説明をした。次に、実施の形態における要部である本発明に関するネットワークシステムについて説明する。本発明が適用されるネットワークシステムは、ネットワーク上においてリモートシステム（例えば、端末をパーソナルコンピュータとするシステ

ム）に接続されているロボット装置の当該ソフトウェアを変更すること等を可能とするものであって、ロボット装置１がリモートシステムに接続されることが前提となるネットワークシステムである。

図１９には、本発明が適用されるネットワークシステムの構成例を示している。このネットワークシステムは、ローカルエリアネットワークとして構築されており、このネットワークシステムにおいて、リモートシステム３０は、パーソナルコンピュータ３２やワークステーション３３を端末として有するシステムであり、ロボット装置１は、ネットワークへの通信手段を備えたロボット装置である。そして、ロボット装置１は、ネットワーク通信用のアダプタとして無線ＬＡＮ用ＰＣカード４１を備え、リモートシステム３０は、ロボット装置１との通信のためのネットワークアダプタ３１を備えている。また、ネットワークシステムでは、ネットワーク通信のプロトコルとされるＴＣＰ／ＩＰ（Transmission Control Protocol/Internet Protocol）が採用されて、リモートシステム３０とロボット装置１との間の通信がなされる。これにより、ロボット装置１にＩＰアドレスが与えられることにより、ロボット装置１は、ネットワークシステム上のパーソナルコンピュータ３２との通信が可能になる。このようにロボット装置１との通信を可能としてネットワークシステムが構築されることにより、ロボット装置１は、無線ＬＡＮ（ローカルエリアネットワーク）上のマシン（電子機器）として扱われるようになる。

なお、ロボット装置１は、例えば図１９に示すように、少なくとも、ＰＣカードＩ／Ｆ４２、フラッシュＲＯＭ４３、ＤＲＡＭ４４及びＣＰＵ４５を備えている。ＰＣカードＩ／Ｆ４２は、無線ＬＡＮ用ＰＣカード４１のインターフェースであり、フラッシュＲＯＭ４３及びＤＲＡＭ４４は、ロボット装置１が取り扱う各種情報が記憶される記憶手段であり、ＣＰＵ４５は、ロボット装置１の各部を制御する制御手段である。例えば、ＣＰＵ４５は、図２に示した中央処理ユニット２２に対応される。図１９には、３台のロボット装置１を示しているが、各ロボット装置１ともにこのような構成を有している。

後で詳述するが、このようなネットワークシステムにより、ロボット装置１の開発サイドでは、ネットワークにロボット装置１を接続した状態で当該ロボット

装置１の有するオブジェクトを開発することができる。また、ユーザサイドでは、パーソナルコンピュータ３２等によってネットワーク上のロボット装置１の内部状態等を監視することができるようになる。

なお、上述の例では、無線ＬＡＮによりネットワークシステムにロボット装置１が接続されているが、これは例えばロボット装置１のように動き回る計算機には物理的な結線が存在しないほうが望ましい、という理由からくるものであり、よって、無線により通信がなされることに限定されるものではなく、例えば、ロボット装置１とパーソナルコンピュータ３２とを有線より通信することも可能である。

図２０には、上述したようなネットワークシステムにおいて通信を可能とするロボット装置１及びリモートシステム３０の構成例を示している。

ロボット装置１は、通信部分として、無線ＬＡＮ用ＰＣカード４１、デバイスドライバ４６、ＴＣＰ／ＩＰプロトコルスタック４７及び及びＴＣＰゲートウェイ４８を備え、リモートシステム３０は、通信部分として、ネットワークアダプタ３１、デバイスドライバ５１及びＴＣＰゲートウェイ５２を備えている。

ロボット装置１及びリモートシステム３０がともに最下層にネットワーク用のアダプタを備えており、ロボット装置１におけるハードウェアとしての無線ＬＡＮ用ＰＣカード４１及びリモートシステム３０におけるネットワークアダプタ３１はそのネットワーク用のアダプタを構成している。

ここで、第１の電子機器とされるロボット装置１の無線ＬＡＮ用ＰＣカード４１（例えば、ＰＣカードＩ／Ｆも含む。）は、ロボット装置１の情報処理プロセス（オブジェクト）に対してデータの送受信を行う情報処理プロセス（ゲートウェイオブジェクト４８）を有する（割り当てられた）第１の通信手段とされ、第２の電子機器とされるパーソナルコンピュータ３２（リモートシステム３０）に接続されるネットワークアダプタ３１は、パーソナルコンピュータ３２上の情報処理プロセス（オブジェクト）に対してデータの送受信を行う情報処理プロセス（ゲートウェイオブジェクト５２）を有する（割り当てられた）第２の通信手段とされる。そして、ネットワークシステムにおいて、無線ＬＡＮ用ＰＣカード４１とネットワークアダプタ３１とが無線又は有線により接続されたときに、無線

ＬＡＮ用ＰＣカード４１のデータの送受信を行う情報処理プロセス（ゲートウェイオブジェクト４８）とネットワークアダプタ３１のデータの送受信を行う情報処理プロセス（ゲートウェイオブジェクト５２）との間でプロセス間通信（オブジェクト間通信）を行うことにより、ロボット装置１の情報処理プロセスとパーソナルコンピュータ３２の情報処理プロセスとの間でプロセス間通信（オブジェクト間通信）がなされる。

また、無線ＬＡＮ用ＰＣカード４１のデータの送受信を行う情報処理プロセスとパーソナルコンピュータ３２のデータの送受信を行う情報処理プロセスとの間で、ネットワーク通信で良く用いられるＴＣＰ／ＩＰのデータ通信方式を採用したデータ通信を行っている。

ロボット装置１及びリモートシステム３０はともに、このような無線ＬＡＮ用ＰＣカード４１及びネットワークアダプタ３１の上位にデバイスドライバ４６，５１を有している。

デバイスドライバ４６，５１は、各ネットワーク用のアダプタとされる無線ＬＡＮ用ＰＣカード４１及びネットワークアダプタ３１を制御するソフトウェアである。

そして、ロボット装置１及びリモートシステム３０では、デバイスドライバ４６，５１の上位に、図示しないＴＣＰ／ＩＰのソケットインターフェースを有している。ロボット装置１及びリモートシステム３０では、ＴＣＰ／ＩＰのソケットインターフェースを介して、データの入出力を行う。例えば、ロボット装置１側ではソケットライブラリーに相当するプロトコルスタックオブジェクトを有しており、これによりアクセスする。

そして、ロボット装置１及びリモートシステム３０が図示しないＴＣＰ／ＩＰのソケットインターフェースの上位層としてゲートウェイオブジェクト（ＴＣＰゲートウェイ）４８，５２がある。ゲートウェイオブジェクト４８，５２は、ネットワークへの出入り口を構成するものとなる。ゲートウェイオブジェクト４８，５２は、上述したオブジェクト間通信とネットワーク通信のプロトコルとされるＴＣＰ／ＩＰプロトコルとの変換を行う。

各システム内の２つのゲートウェイオブジェクト４８，５２について、一方の

ゲートウェイオブジェクトはサーバ（以下、サーバゲートウェイオブジェクトという。）としての役割を果たし、他方のゲートウェイオブジェクトはクライアント（以下、クライアントゲートウェイオブジェクトという。）として役割を果たす。例えば、システムの性質を考慮し、ユーザとのやり取りを仲介するリモートシステム側のゲートウェイオブジェクトがクライアントゲートウェイオブジェクトとなり、ターゲットシステム（ロボット装置）のゲートウェイオブジェクトをサーバゲートウェイオブジェクトとなる。ゲートウェイオブジェクト４８，５２の動作は、具体的には次のようになる。

ゲートウェイオブジェクト４８，５２は、port．cfg ファイルという通信方式の設定が記述された設定ファイルを有しており、これによってオブジェクト間通信で定義される通信のサービス名とＴＣＰ／ＩＰの通信で用いられるＩＰアドレスとポート番号の対応を取る。

図２１には、port．cfg ファイル（設定ファイル）の一例を示している。例えば、port．cfg ファイルには、テキストデータ形式として各種情報が記述されており、具体的には、オブジェクト名、データ名、サブジェクト名、ポート番号、ＩＰアドレス等が記述されている。また、この port．cfg ファイルは書き換えが自由になされるようになっている。

各ゲートウェイオブジェクト４８，５２は、上述の図１２或いは図１５に示すようなオブジェクトＡとオブジェクトＢとの関係とされていることから、所定のシーケンスに沿って所定の処理を行うことができる。具体的には、各ゲートウェイオブジェクト４８，５２は、代理プログラムとされるオブザーバ及びサブジェクトを動的に生成して、それぞれが有する port．cfg ファイルに記述されたサービス名とそのオブザーバ又はサブジェクトのＩＤを結びつけて、各々のシステム上のサービスマネージャ（オブジェクトマネージャ）に登録を行う。

また、上述したように、port．cfg ファイルへのサービス名が自由に記述することができるようになっており、各ゲートウェイオブジェクト４８，５２は、port．cfg ファイルにそのように自由に記述されている全てのサービスを実行する。このように、port．cfg ファイルへのサービス名の記述の制限をなくすことにより、このゲートウェイオブジェクト４８，５２を介してあらゆるオブジェク

ト間（ロボット装置１のオブジェクトとパーソナルコンピュータ３２のオブジェクト）のオブジェクト間通信ができるようになる。

ここで、サービス名は port．cfg ファイルに記述されているネットワークポート（ＩＰアドレスとポート番号）に対応するために結局はオブザーバ又はサブジェクトのＩＤとポート番号が対応することになる。また、ネットワーク上で異なる場所に存在するゲートウェイオブジェクトにおいて、port．cfg ファイルの記述を対応させる、すなわち対応するオブジェクト間通信のサービス名に対して同じポート番号を割り付けることにより、ネットワーク上において、ロボット装置１とリモートシステム３０とのそれぞれのゲートウェイオブジェクト４８，５２間でのオブジェクト間通信の対応が取られる。

そして、リモートシステム３０のクライアントゲートウエイオブジェクト５２内のオブザーバ又はサブジェクトに接続要求がきた場合、対応するポート番号に対して接続（connect 命令）を行う。ここで、ターゲットシステム（ターゲットボード、ロボット装置１）側のサーバゲートウェイオブジェクト４８が既に起動していれば接続が成立する。一方、ターゲットシステムの準備できていなければ数秒後に再接続を行い準備ができるまで待ち、接続を成立させる。

接続が成立すると、以後、ゲートウェイオブジェクトのオブザーバ又はサブジェクトは、データ（ノーティファイ（Notify）やレディ（Ready）メッセージ）を受信するたびに、対応するポートにそのメッセージが送りこまれる。また、逆にネットワークポートがデータを受信すれば対応するオブザーバはノーティファイ（Notify）メッセージを送り出し、また、サブジェクトはレディ（Ready）メッセージを送り出す。

上述したようにシステム間（ロボット装置１とリモートシステム３０との間）の接続が確立されると、本来ネットワークにより接続されるロボット装置１とリモートシステム３０とを、複数のオブジェクトからなる一つのシステムとして取り扱うことができるようになる。すなわち例えば、図２２に示すように、接続前においては、一群のオブジェクト（例えば、ソフトウェアの一部）からなるシステムＡ（例えば、ロボット装置１）と、他の一群のオブジェクト（例えば、ソフトウェアの一部）からなるシステムＢ（例えば、リモートシステム３０）とが図

中点線で示すような境界を境にそれぞれ独立して構築された存在とされるが、接続後においては、図中点線をまたぐコネクションが切断されて、図２３に示すような仮想的なゲートウェイオブジェクト（例えば、ゲートウェイオブジェクト４８，５２の組）ＧＯを一つおいて各システムのオブジェクト群が接続されたことと等価になる。これにより、ロボット装置１のオブジェクトとリモートシステム３０のオブジェクトとが一つのシステムを構成するものと把握されるようになり、ロボット装置１は、一つの仮想的なゲートウェイオブジェクトＧＯを介してネットワーク越しのリモートシステム３０で動作されるようになる。

よって、仮想的なゲートウェイオブジェクト（仮想ゲートウェイオブジェクト）ＧＯによりシステム間（ロボット装置１とリモートシステム３０）が接続され、図２０に示したようなデータの受け渡しが全て行わることになるので、各システム（ロボット装置１及びリモートシステム３０）のゲートウェイオブジェクトに接続されたオブジェクトは、ネットワーク越しの通信を全く意識することなく、他のシステム上に存在するオブジェクトとの間でオブジェクト間通信を行うことができるようになる。

また、図２２に示すようなネットワーク構成から図２３に示すようなネットワーク構成への変更をオブジェクト間の接続の変更と等価なものとして取り扱うことができるので、図２３に示すようなネットワーク構成は、新たにオプジェクトのコードに変更を加える必要がなく、所望のオブジェクト同士を接合するためのコネクションファイルの記述を変えるだけで実現される。

また、リモートシステム３０とターゲットシステム（ロボット装置１）にてＯＳやプロセッサの違いなどにより同じコードが実行できない場合でも、上述のオブジェクト間通信のフレームワークはＯＳやプロセッサに依存しないために、実行するシステム上で実装されたフレームワークに基づいて再コンパイルしなおすことで当該実行するシステム上にて実行可能オブジェクトを生成することができる。

このようなネットワークシステムにより、ターゲットシステム（ロボット装置１）上のオブジェクトに接続（通信状態）されたリモートシステム３０上のオブジェクトを、リモートシステム３０にて変更することができるようになる。これ

により、ターゲットシステムの実際の状態（実環境）が考慮されて、或いはリアルタイムにて、オブジェクトをリモートシステム３０上にて開発（設計）することができるようになる。

また、開発後は、リモートシステム３０上で開発したオブジェクトが再コンパイルしてロボット装置１に導入される。よって、ロボット装置１に導入されたオブジェクトは、開発時点でロボット装置１に搭載された状態（実環境）での検討がなされているので、何ら問題なく、例えば他のオブジェクトと不都合を生じることなく、ロボット装置１にて処理を実行するようになる。また、単に仮想的なゲートウェイオブジェクトの接続によりオブジェクトの開発が実現されるので、オブジェクトの開発は容易なものとなる。

ここで、以上説明した内容について、所定の規格、例えばソニー社の提唱するエンターテインメントロボットのためのハードウェア及びソフトウェアのプラットフォーム技術であるＯＰＥＮ－Ｒに適用したネットワーク拡張技術について説明する。このプラットフォーム技術においては、ソフトウェアがコンポーネント化され、オブジェクト（プロセスに相当）という単位で実行される。

オブジェクト指向におけるデザインパターンのオブザーバパターンを基に設計されたオブジェクト間の通信規約が決められており、ユーザは、オブジェクトに入力されるデータ構造とそのサービス名、それを処理するメソッド、出力データ構造とそのサービス名の３つを定義する必要がある。これらが個々に定義されたオブジェクトを接続することでイベントが伝搬し、認識から行動までロボットの一連のシステムが実現される。

オブジェクト間通信は、以下のような特徴を持つ。

・オブジェクト設計時には受信するデータと送信するデータだけを定義してメッセージの送信先と受信元は決めない。

・送受信する口をサービスといい、それぞれにはユーザが定義したユニークな名前が与えられる。

・メッセージの送信と受信は、OSubject と OObserber　という代理クラスが行う。それらのインスタンスが前述の個々のサービスに１対１で対応する。

・サービスとサービスの接続はコネクションファイルに定義され、システムの起

動時に OSubject と OObserber のインスタンスに通知される。

・サービス間の通信は非同期通信で、接続は多対多通信をサポートしている。

・受信可能であるという Ready メッセージによってデータのフロー制御を行っている。

・データのスケーラビリティの確保と受け渡しの高速化のために共有メモリを使用している。

次に、ゲートウェイオブジェクトについて説明する。ロボットにおいて無線ネットワークはその動きが制限されないために非常に有用であり、ロボットのセンサ情報を送ったり、処理結果を送ったり、リモートホスト上のツールを使ってロボットを動かしたりなど、種々の利用方法が考えられる。ネットワーク上の分散オブジェクトに対するアクセスをクライアント側のプロキシに隠蔽することで意識させないフレームワークの提供も知られているが、こうしたアプローチは新たなサービスを提供する度に通信用のオブジェクトを設計したり、既存のオブジェクトやプロキシに通信用のコードを書き加えたりすることになり開発の効率が悪い。ここで、ロボット内部のソフトウェアは上記所定のプラットフォーム技術のプログラミングフレームワークに基づいて記述されているために、オブジェクト間のデータの定義、通信方法、通信相手の依存関係が全く分離されている。そこで、上記図２０、図２１と共に説明したように、ネットワークの出入り口に相当するＴＣＰゲートウェイというオブジェクトを導入した。このオブジェクト（ＴＣＰゲートウェイ）は、オブジェクト間通信のサービス名とＴＣＰ／ＩＰのポート番号の対応が記述された設定ファイル（図２１）を読み込み、サービスとポートを動的に生成してそこに流されるデータのプロトコル変換を行う（図２０）。そこで決められたデータ用のポートに対してサービスを繋ぐことで通信相手を選択することができ、これにより通信方法と通信相手はＴＣＰゲートウェイが解決してくれるために、オブジェクトの設計者はデータの定義だけを決定すればよく、簡便にネットワークを利用することができる。

上述した所定のプラットフォーム技術のフレームワークは、Ｕｎｉｘの Fork(),exec()や、ＩＰＣのメッセージ通信と共有メモリに相当する機能、すなわち、文献「Unix ネットワークプログラミング，第２版，Vol.2，IPC：プロセス間通

信」W．Richard Stevens，ピアソン・エデュケーション，2000 を利用すれば、現在ロボット上で利用されているＯＳ以外でも実装が可能である。ここで、Linux 上に用意された実行可能環境下では、オブジェクトはロボット内部のコードとソースコードレベルで互換であるので、両方のシステムで実行可能なコード記述が可能である。

このエミュレーション環境上で上述のＴＣＰゲートウェイオブジェクトを同様に実装し、それぞれのＴＣＰゲートウェイのポート番号を一致させると、上記図２３のように、２つのシステムを対称に接続することができる。これによりロボット内のオブジェクトとリモートホスト上のオブジェクトがリアルタイムで通信できるために、リモートホストのオブジェクト開発者は例えば Linux 上の便利なツールを使いながら、ロボット上のセンサ情報やロボットを動作させるコマンドを自由に駆使できる。実際に複雑な画像認識をリモートホスト上で行いその判断結果でロボットを動作させたり、商品化されたペットプログラムの上位層のオブジェクトをコンパイルし直してリモートホスト上でロボットと通信させながら実行したりして、スタンドアローンシステムと実時間性で全く遜色のないことを確認している。

勿論、組み込み機器のソフトウェア開発より、Linux 等の汎用コンピュータ上でのソフトウェア開発の方が圧倒的に効率が良い。また、世界中で利用されている開発環境でロボットのプログラムをすることができることは普及面でも大きなメリットとなる。

ここで開発されたコードを最終的に組込用の開発環境でコンパイルし直し、スタンドアローンシステム用のソフトウェアとして商品パッケージ化することも可能である。

次に、マルチＣＰＵタイプのロボットへの適用について説明する。上述のようなオブジェクト間通信の拡張の仕組みは、組込用のＣＰＵを２つ以上持ったロボットシステムに対しても応用することができる。現在２つのＣＰＵは、上述した所定のロボット用のプラットフォーム技術のいわゆるＯＰＥＮ－Ｒバスを介して通信することができるが、このバス通信のプロトコル変換オブジェクトを導入することで、一方にあるモーション制御用のオブジェクト群ともう一方にある思考

系オブジェクト群の通信が自由に定義できる。

以上説明したように、ＴＣＰオブジェクトを導入することでオブジェクト間通信でやり取りするデータをネットワーク上から利用可能にし、Unix，Linux 等の汎用コンピュータシステム上に上記ロボット用の所定のプラットフォーム技術のプログラミングフレームワークを実現し、該プラットフォーム技術のアーキテクチャの移植性を示すことができ、また、ゲートウェイとエミュレータを組み合わせることでロボットとリモートホスト間でシームレスなオブジェクト実効、開発環境を構築可能としたものである。

次に、ゲートウェイオブジェクトを使用したネットワークシステムの具体例を説明する。図２４には、上述したオブジェクト間通信のフレームワークに基づいて、いわゆるサッカードッグを実行するためのロボットソフトウェア（複数のオブジェクト群からなるソフトウェア）の構成例を示す。

サッカードッグのプログラムとは、ボールを探して、ボールに近づいて蹴るといったサッカープレイヤのような行動をロボット装置に実行させるプログラムである。そして、このサッカードッグのプログラムは、そのような行動を実行するため記述がなされており、その記述に基づいてタスクの実行のために必要なソフトウェア部品（オブジェクト）が選択され、必要であれば全体のシステムが新規に生成してかたちづくられる。

図２４に示す各オブジェクトの機能の概略は次のようになる。VRComm ６１はロボット装置１の各種デバイスにアクセスするオブジェクトであり、OMPsd ６２は距離センサの情報を処理するオブジェクトであり、OMCdt ６３は色の検出によりボールの位置を計算するオブジェクトであり、OMTE ６４は首の姿勢管理をしてボールの方向に視線を制御するオブジェクトであり、OMLE2_S ６５は歩行動作を制御するオブジェクトであり、OMNE ６６は転倒復帰を制御するオブジェクトであり、OMSE ６７は尻尾を制御するオブジェクトであり、OMoNET ６８は動作制御オブジェクト間の排他制御及び姿勢情報管理を行うオブジェクトであり、OMGsensor ６９はロボット装置のボディの姿勢を検出するオブジェクトであり、OMBC ７０はバッテリ情報を管理するオブジェクトであり、SoccerDog ７１はサッカーにおける行動計画を行うオプジェクトとなってい

る。また、これらのオブジェクトの接続は図２５乃至図２７に示すように記述されている各オブジェクト接合のためのコネクションファイルによって実現される。なお、図２５乃至図２７に示すコネクションファイルは、紙面の都合上分割されたのであって、実際は図２５、図２６、図２７の順で連続して記述されている。

このような構成からなる既に存在するオブジェクト群における SoccerDog オブジェクト７０を新規に開発する場合（ソフトウェアの開発環境の場合）について説明する。以下のように処理がなされることにより、SoccerDog オブジェクト７０が新規に開発される。

SoccerDog オブジェクト７０以外の既存のオブジェクトは全てすべてロボット装置１側に残して SoccerDog オブジェクト７０のみをリモートシステム３０上に取り入れる。

そして、ロボット装置１側では、SoccerDog オブジェクトの代わりにゲートウェイオブジェクト（図２０に示すＴＣＰゲートウェイオブジェクト４８）を置き、SoccerDog オブジェクトに接続すべきサービスを全て当該ゲートウェイオブジェクトに接続する。一方、リモートシステム３０側では、既存のオブジェクトの代わりにゲートウェイオブジェクト（図２０に示すＴＣＰゲートウェイオブジェクト５２）を導入して SoccerDog オブジェクトの全ての接続を当該ゲートウェイオブジェクトに対して行う。

このように接続されることにより、ロボット装置１側のゲートウェイオブジェクトをサーバとするリモートシステム３０側からは接続が随時可能になる。よって、ロボット装置１側のシステムを予め立ち上げておき、SoccerDog オブジェクトのコンパイルが終わったら実行（起動）することでシステム全体を動作させることができる。これにより、リモートシステム３０上における SoccerDog オブジェクトの開発が可能な環境が設定される。すなわち、リモートシステム３０（パーソナルコンピュータ３２）上の SoccerDog オブジェクトとロボット装置１上のオブジェクトとのオブジェクト間通信が構築される。

リモートシステム３０における SoccerDog オブジェクトの開発は、例えば SoccerDog オブジェクトに不具合があったらリモートシステム３０側の SoccerDog オブジェクトを停止して再度修正をかけてコンパイルして実行すると

いったようにして行う。例えば、SoccerDog オブジェクトの不具合とは、SoccerDog オブジェクトにより新たな動作（例えば、ボールを蹴る動作についての新たな動作）を開発する場合に、その新たな動作をロボット装置１が正常に実行できないような場合であり、このような場合には、リモートシステム３０にて、SoccerDog オブジェクトを停止して修正をかけて再コンパイルして実行しなおしたりする。

そして、不具合を調整等して最終的に開発が終わったら、リモートシステム３０上における SoccerDog オブジェクトのソースコードをターゲットシステムの環境用（ロボット装置１用）にコンパイルしなおして、既存のオブジェクトとリンクしてロボット装置１用の実行プログラムを作成してロボット装置１に導入する。ロボット装置１は、このように導入された実行プログラムにより動作が制御される。

以上のように、本発明が適用されたネットワークシステムによりサッカードッグを実行するためのロボットソフトウェアの開発がなされる。

よって、リモートシステム３０とロボット装置１とが接続された状態において、リモートシステム３０上でソフトウェア（SoccerDog オブジェクト）の開発がなされるので、ロボット装置１側のシステム全体を立ち上げなおす必要がなく、SoccerDog オブジェクトに修正をかける（加工する）ことができるようになる。これにより、SoccerDog オブジェクトの開発を効率よく行うことができるようになる。

また、ロボット装置１へのプログラムのダウンロードを要することなくソフトウェアの開発がなされるので、効率よく開発を行うことができる。そして、端末とされるリモートシステム３０（パーソナルコンピュータ３２）では、従前のリモートシステム（パーソナルコンピュータ３２）として使用されているデバッガー、プロファイラーなどの汎用プログラムを利用することができるので、資源を有効利用してソフトウェアを開発することができる。

以上の例は、開発環境としての例であり、次にユーザサイドの使用例について、例えばユーザがパーソナルコンピュータ等を介してロボット装置１と通信してそのロボット装置１の内部状態を観測したり、或いはロボット装置１に対して命令

を送ったりして楽しみようなプログラムを作る場合を説明する。

通常、そのような機能を実現させるためには、パーソナルコンピュータ側のプログラムを作成する他に、ロボット装置１側の例えばロボット装置１の感情を計算するモジュール（オブジェクト）に対してパーソナルコンピュータ３２に感情データを送信するようなコードを付加したり、動作命令を受け取り実行するオブジェクト（ロボット装置１側のオブジェクト）にパーソナルコンピュータ３２からのデータを受信するコードを付加したりする必要がある。

これに対して、本発明が適用されたネットワークシステムでは、上述のように各ソフトウェアモジュール（オブジェクト）がオブジェクト間通信のフレームワークに基づいて動作されており、ネットワーク通信を中継するゲートウェイオブジェクトを導入することができれば、コネクションファイルを書き換えて、既に存在するロボット装置１の感情出力や動作命令受信サービスをゲートウェイオブジェクトに接続するだけで、パーソナルコンピュータ３２とロボット装置１とのオブジェクト間でのオブジェクト間通信が構築され、各オブジェクトでの処理がなされるようになる。

これにより、既存のロボット装置１に組み込まれているプログラムを変更することなく、パーソナルコンピュータ３２上で新たにプログラムを開発することができるようになる。例えば、これにより、ユーザがパーソナルコンピュータ３２等を介してロボット装置１と通信してそのロボット装置１の内部状態を観測したり（オブジェクトの処理状態を外部出力したり）、ロボット装置１に対して命令を送ったりすることができ、また、ロボット装置１に対して新機能を付け加えることができるようになる。

以上のように本発明を適用してネットワークを構築することにより次のような効果が得られる。

ロボットシステム（ロボット装置）のプログラム（ソフトウェア）の開発者は、より利便性の高いリモートシステム上で開発が可能になる。

また、シミュレータ等と違い実際にロボット装置１を動作させて開発が可能となるので、開発者は、実環境とシミュレータ環境の違いに悩まされる必要がなくなった。すなわち、実環境に真似たシミュレータ環境の設定は困難であるが、開

発者は、そのようなシミュレータ環境の設定をすることなく、プログラムを開発することができる。

また、ターゲットシステム（ターゲットボード、ロボット装置）のソースコードとリモートシステムのソースコードは同じものが使われるためにシステム変更に伴う導入作業（移植作業）は簡便となり、また、ミスの混入も減らすことができる。

また、オブジェクト間通信のサービスの接続をゲートウェイオブジェクトに対して行うだけで、既存のロボットプログラムを変更することなく、ネットワークを使った機能拡張が行えるようになる。

また、ロボット装置上のプログラムとリモートシステムのプログラムとの融合が容易とされ、リモートシステム上でのプログラムの開発が可能になるために、ロボット装置１上の計算機資源では処理しきれないような処理も強力なリモートシステムの処理能力で実現可能になる。これにより、より複雑なプログラムの開発が可能になる。

また、本発明の実施の形態とされるロボット装置は、上述の図１に示したような外観となされることに限定されるものではない。例えば、図２８に示すように、ロボット装置１は、より動物に近いような外観形状を有して構成することもできる。

以上の説明からも明らかなように、本発明に係る通信装置は、第１の電子機器の情報処理プロセスに対してデータの送受信を行う情報処理プロセスを有する第１の通信手段と、第２の電子機器の情報処理プロセスに対してデータの送受信を行う情報処理プロセスを有する第２の通信手段とを備える。そして、第１の通信手段と第２の通信手段とが無線又は有線により接続されたときに、第１の通信手段のデータの送受信を行う情報処理プロセスと第２の通信手段のデータの送受信を行う情報処理プロセスとの間でプロセス間通信を行うことにより、第１の電子機器の情報処理プロセスと第２の電子機器の情報処理プロセスとの間でプロセス間通信がなされることにより、第１の通信手段の情報処理オブジェクトと第２の通信手段のオブジェクトとの間のプロセス間通信があたかも同一システム上におけるプロセス間通信のように振る舞うために、第１の電子機器の情報処理プロセ

スと第２の電子機器の情報処理プロセスとの間の通信は、同一システム上におけるプロセス間通信として取り扱うことができるようになる。

また、本発明に係る通信方法は、第１の電子機器の情報処理プロセスに対してデータの送受信を行う情報処理プロセスを有する第１の通信手段と、第２の電子機器の情報処理プロセスに対してデータの送受信を行う情報処理プロセスを有する第２の通信手段とを無線又は有線により接続して、第１の通信手段のデータの送受信を行う情報処理プロセスと第２の通信手段のデータの送受信を行う情報処理プロセスとの間でプロセス間通信を行うことにより、第１の電子機器の情報処理プロセスと第２の電子機器の情報処理プロセスとの間でプロセス間通信を行うことにより、第１の通信手段の情報処理オブジェクトと第２の通信手段のオブジェクトとの間のプロセス間通信があたかも同一システム上におけるプロセス間通信のように振る舞うために、第１の電子機器の情報処理プロセスと第２の電子機器の情報処理プロセスとの間の通信は、同一システム上におけるプロセス間通信として取り扱うことができるようになる。

また、本発明に係る情報処理システムは、情報処理プロセスにより制御される電子機器と、電子機器の制御に使用する情報処理プロセスが取り込まれて、当該情報処理プロセスを取り扱う情報処理装置と、電子機器と情報処理装置との間で無線又は有線により通信を行う通信手段とを備えている。そして、情報処理装置は、電子機器が動作されている際の当該電子器機の情報の情報処理プロセスと取り込んだ情報処理プロセスとの間で通信手段による通信を行い、電子機器の情報処理プロセスから送信されてくるデータを使用し、取り込んだ情報処理プロセスを取り扱うことにより、情報処理装置は、実環境にある電子機器が考慮された情報処理プロセスを取り扱う（例えば加工する）ことがきるようになる。

また、本発明に係るプログラム作成方法は、情報処理プロセスにより制御される電子機器が動作されている際の当該電子器機の情報処理プロセスと情報処理装置上の情報処理プロセスとの間で無線又は有線の通信手段による通信を行い、情報処理装置にて、電子機器の情報処理プロセスから送信されてくるデータを使用し、電子機器に使用する情報処理プロセスを作成するプログラム作成工程と、プログラム作成工程にて作成した情報処理プロセスを電子機器に組み込むプログラ

ム組み込み工程とを有することにより、情報処理装置は、実環境にある電子機器が考慮して情報処理プロセスを作成し、電子機器は、そのような実環境が考慮された作成された情報処理プロセスが組み込まれ、当該情報処理プロセスに基づいて動作するようになる。実環境が考慮されて情報処理プロセスが作成されているので、電子機器は何ら不具合なく動作する。

請求の範囲

１．第１の電子機器の情報処理プロセスに対してデータの送受信を行う情報処理プロセスを有する第１の通信手段と、

第２の電子機器の情報処理プロセスに対してデータの送受信を行う情報処理プロセスを有する第２の通信手段とを備え、

上記第１の通信手段と上記第２の通信手段とが無線又は有線により接続されたときに、当該第１の通信手段の上記データの送受信を行う情報処理プロセスと当該第２の通信手段の上記データの送受信を行う情報処理プロセスとの間でプロセス間通信を行うことにより、上記第１の電子機器の情報処理プロセスと上記第２の電子機器の情報処理プロセスとの間でプロセス間通信がなされること

を特徴とする通信装置。

２．上記情報処理プロセスがオブジェクト指向により作成されたプログラムからなること

を特徴とする請求の範囲第１項記載の通信装置。

３．上記第１の電子機器が情報処理プロセスにより動作が制御されるロボット装置であり、上記第２の電子機器が情報処理装置であること

を特徴とする請求の範囲第１項記載の通信装置。

４．上記第１の通信手段の上記データの送受信を行う情報処理プロセスと上記第２の通信手段の上記データの送受信を行う情報処理プロセスとの間のプロセス間通信にてネットワーク通信へのプロトコル変換をすることにより、上記第１の電子機器の情報処理プロセスと上記第２の電子機器の情報処理プロセスとの間でプロセス間通信がなされること

を特徴とする請求の範囲第１項記載の通信装置。

５．上記ネットワーク通信へのプロトコル変換によりＴＣＰ／ＩＰ（Transmission Control Protocol/Internet Protocol）への変換がなされること

を特徴とする請求の範囲第４項記載の通信装置。

６．上記第１及び第２の通信手段の上記データの送受信を行う情報処理プロセス

は、通信方式の設定が記述される設定ファイルを有し、この設定ファイルの記述事項に基づいて、上記第１の通信手段の上記データの送受信を行う情報処理プロセスと上記第２の通信手段の上記データの送受信を行う情報処理プロセスとの間でプロセス間通信を行うこと

を特徴とする請求の範囲第１項記載の通信装置。

７．上記設定ファイルには、上記第１の電子機器の情報処理プロセスと上記第２の電子機器の情報処理プロセスとの間のプロセス間通信のサービス名が記述されており、

上記第１及び第２の通信手段の上記データの送受信を行う情報処理プロセスが上記設定ファイルに記述されているサービス名に基づいて上記第１の通信手段の上記データの送受信を行う情報処理プロセスと上記第２の通信手段の上記データの送受信を行う情報処理プロセスとの間でプロセス間通信を行うことにより、上記第１の電子機器の情報処理プロセスと上記第２の電子機器の情報処理プロセスとの間で上記サービス名に対応されるプロセス間通信がなされること

を特徴とする請求の範囲第６項記載の通信装置。

８．上記設定ファイルには、上記第１の通信手段の上記データの送受信を行う情報処理プロセスと上記第２の通信手段の上記データの送受信を行う情報処理プロセスとの接続の対応事項が記述されており、

上記第１の通信手段の上記データの送受信を行う情報処理プロセスと第２の通信手段の上記データの送受信を行う情報処理プロセスとは、上記設定ファイルに記述されている上記接続の対応事項に基づいて、プロセス間通信が可能な状態に接続されること

を特徴とする請求の範囲第６項記載の通信装置。

９．第１の電子機器の情報処理プロセスに対してデータの送受信を行う情報処理プロセスを有する第１の通信手段と、第２の電子機器の情報処理プロセスに対してデータの送受信を行う情報処理プロセスを有する第２の通信手段とを無線又は有線により接続して、当該第１の通信手段の上記データの送受信を行う情報処理プロセスと当該第２の通信手段の上記データの送受信を行う情報処理プロセスとの間でプロセス間通信を行うことにより、上記第１の電子機器の情報処理プロセ

スと上記第２の電子機器の情報処理プロセスとの間でプロセス間通信を行うこと
を特徴とする通信方法。

１０．情報処理プロセスにより制御される電子機器と、

上記電子機器の制御に使用する上記情報処理プロセスが取り込まれて、当該情報処理プロセスを取り扱う情報処理装置と、

上記電子機器と上記情報処理装置との間で無線又は有線により通信を行う通信手段とを備え、

上記情報処理装置は、上記電子機器が動作されている際の当該電子器機の情報の情報処理プロセスと上記取り込んだ情報処理プロセスとの間で上記通信手段による通信を行い、上記電子機器の情報処理プロセスから送信されてくるデータを使用し、上記取り込んだ情報処理プロセスを取り扱うこと
を特徴とする情報処理システム。

１１．上記情報処理プロセスがオブジェクト指向により作成されたプログラムからなること
を特徴とする請求の範囲第１０項記載の情報処理システム。

１２．上記情報処理装置は、上記情報処理プロセスを取り扱うこととして、当該情報処理プロセスを加工すること
を特徴とする請求の範囲第１０項記載の情報処理システム。

１３．上記情報処理装置は、上記情報処理プロセスを取り扱うこととして、当該情報処理プロセスによる処理状態を外部出力すること
を特徴とする請求の範囲第１０項記載の情報処理システム。

１４．上記通信手段は、電子機器の情報処理プロセスに対してデータの送受信を行う情報処理プロセスを有する第１の通信手段と、上記情報処理装置の情報処理プロセスに対してデータの送受信を行う情報処理プロセスを有する第２の通信手段とを備え、

上記第１の通信手段と上記第２の通信手段とが無線又は有線により接続されたときに、当該第１の通信手段の上記データの送受信を行う情報処理プロセスと当該第２の通信手段の上記データの送受信を行う情報処理プロセスとの間でプロセス間通信を行うことにより、上記電子機器の情報処理プロセスと上記情報処理装

置の情報処理プロセスとの間でプロセス間通信がなされること

を特徴とする請求の範囲第１０項記載の情報処理システム。

１５．上記第１の通信手段の上記データの送受信を行う情報処理プロセスと上記第２の通信手段の上記データの送受信を行う情報処理プロセスとの間のプロセス間通信にてネットワーク通信へのプロトコル変換をすることにより、上記電子機器の情報処理プロセスと上記情報処理装置の情報処理プロセスとの間でプロセス間通信がなされること

を特徴とする請求の範囲第１４項記載の情報処理システム。

１６．上記ネットワーク通信へのプロトコル変換によりＴＣＰ／ＩＰ（Transmission Control Protocol/Internet Protocol）への変換がなされること

を特徴とする請求の範囲第１５項記載の情報処理システム。

１７．上記第１及び第２の通信手段の上記データの送受信を行う情報処理プロセスは、通信方式の設定が記述される設定ファイルを有し、この設定ファイルの記述事項に基づいて、上記第１の通信手段の上記データの送受信を行う情報処理プロセスと上記第２の通信手段の上記データの送受信を行う情報処理プロセスとの間でプロセス間通信を行うこと

を特徴とする請求の範囲第１４項記載の情報処理システム。

１８．上記設定ファイルには、上記電子機器の情報処理プロセスと上記情報処理装置の情報処理プロセスとの間のプロセス間通信のサービス名が記述されており、

上記第１及び第２の通信手段の上記データの送受信を行う情報処理プロセスが上記設定ファイルに記述されているサービス名に基づいて上記第１の通信手段の上記データの送受信を行う情報処理プロセスと上記第２の通信手段の上記データの送受信を行う情報処理プロセスとの間でプロセス間通信を行うことにより、上記電子機器の情報処理プロセスと上記情報処理装置の情報処理プロセスとの間で上記サービス名に対応されるプロセス間通信がなされること

を特徴とする請求の範囲第１７項記載の情報処理システム。

１９．上記設定ファイルには、上記第１の通信手段の上記データの送受信を行う情報処理プロセスと上記第２の通信手段の上記データの送受信を行う情報処理プロセスとの接続の対応事項が記述されており、

上記第１の通信手段の上記データの送受信を行う情報処理プロセスと第２の通信手段の上記データの送受信を行う情報処理プロセスとは、上記設定ファイルに記述されている上記接続の対応事項に基づいて、プロセス間通信が可能な状態に接続されること

を特徴とする請求の範囲第１７項記載の情報処理システム。

２０．情報処理プロセスにより制御される電子機器が動作されている際の当該電子器機の情報処理プロセスと情報処理装置上の情報処理プロセスとの間で無線又は有線の通信手段による通信を行い、上記情報処理装置にて、上記電子機器の情報処理プロセスから送信されてくるデータを使用し、上記電子機器に使用する情報処理プロセスを作成するプログラム作成工程と、

上記プログラム作成工程にて作成した上記情報処理プロセスを上記電子機器に組み込むプログラム組み込み工程と

を有することを特徴とするプログラム作成方法。

２１．上記情報処理プロセスがオブジェクト指向により作成されたプログラムからなること

を特徴とする請求の範囲第２０項記載のプログラム作成方法。

２２．上記電子機器の情報処理プロセスに対してデータの送受信を行う情報処理プロセスを有する第１の通信手段と、上記情報処理装置の情報処理プロセスに対してデータの送受信を行う情報処理プロセスを有する第２の通信手段とを無線又は有線により接続して、当該第１の通信手段の上記データの送受信を行う情報処理プロセスと当該第２の通信手段の上記データの送受信を行う情報処理プロセスとの間でプロセス間通信を行うことにより、上記電子機器の情報処理プロセスと上記情報処理装置の情報処理プロセスとの間でプロセス間通信を行うこと

を特徴とする請求の範囲第２０項記載のプログラム作成方法。

２３．上記第１の通信手段の上記データの送受信を行う情報処理プロセスと上記第２の通信手段の上記データの送受信を行う情報処理プロセスとの間でプロセス間通信にてネットワーク通信へのプロトコル変換をすることにより、上記電子機器の情報処理プロセスと上記情報処理装置の情報処理プロセスとの間でプロセス間通信がなされること

を特徴とする請求の範囲第２２項記載のプログラム作成方法。

２４．上記ネットワーク通信へのプロトコル変換によりＴＣＰ／ＩＰ（Transmission Control Protocol/Internet Protocol）への変換がなされること

を特徴とする請求の範囲第２３項記載のプログラム作成方法。

２５．上記第１及び第２の通信手段の上記データの送受信を行う情報処理プロセスは、通信方式の設定が記述される設定ファイルを有し、この設定ファイルの記述事項に基づいて、上記第１の通信手段の上記データの送受信を行う情報処理プロセスと上記第２の通信手段の上記データの送受信を行う情報処理プロセスとの間でプロセス間通信を行うこと

を特徴とする請求の範囲第２２項記載のプログラム作成方法。

２６．上記設定ファイルには、上記電子機器の情報処理プロセスと上記情報処理装置の情報処理プロセスとの間のプロセス間通信のサービス名が記述されており、

上記第１及び第２の通信手段の上記データの送受信を行う情報処理プロセスが上記設定ファイルに記述されているサービス名に基づいて上記第１の通信手段の上記データの送受信を行う情報処理プロセスと上記第２の通信手段の上記データの送受信を行う情報処理プロセスとの間でプロセス間通信を行うことにより、上記電子機器の情報処理プロセスと上記情報処理装置の情報処理プロセスとの間で上記サービス名に対応されるプロセス間通信がなされること

を特徴とする請求の範囲第２５項記載のプログラム作成方法。

２７．上記設定ファイルには、上記第１の通信手段の上記データの送受信を行う情報処理プロセスと上記第２の通信手段の上記データの送受信を行う情報処理プロセスとの接続の対応事項が記述されており、

上記第１の通信手段の上記データの送受信を行う情報処理プロセスと第２の通信手段の上記データの送受信を行う情報処理プロセスとは、上記設定ファイルに記述されている上記接続の対応事項に基づいて、プロセス間通信が可能な状態に接続されること

を特徴とする請求の範囲第２５項記載のプログラム作成方法。

２８．動作制御プログラムがオブジェクト間通信で構築されたロボット装置と、オブジェクト間通信で構築されたコンピュータシステムとが有線又は無線を経由

してデータ通信を行うネットワークシステムにおいて、

上記オブジェクト間通信と、上記ロボット装置と上記コンピュータシステムとの間でのネットワーク通信におけるプロトコル変換を動的に行うゲートウェイオブジェクトを有すること

を特徴とするネットワークシステム。

２９．上記ゲートウェイオブジェクトは、上記オブジェクト間通信で定義される通信のサービス名と上記ネットワーク通信で用いられるアドレスとポート番号の対応を取ることを特徴とする請求の範囲第２８項記載のネットワークシステム。

３０．上記ネットワーク通信のプロトコル変換によりＴＣＰ／ＩＰ（Transmission Control Protocol/Internet Protocol）への変換がなされることを特徴とする請求の範囲第２８項記載のネットワークシステム。

３１．オブジェクト間通信を有して構築されたロボット装置と、オブジェクト間通信を有して構築されたコンピュータシステムとが有線又は無線を経由してデータ通信を行う通信方法において、

上記オブジェクト間通信と、上記ロボット装置と上記コンピュータシステムとの間でのネットワーク通信におけるプロトコル変換を行うこと

を特徴とする通信方法。

３２．上記プロトコル変換は、ゲートウェイオブジェクトにより行われ、上記ゲートウェイオブジェクトは、上記オブジェクト間通信で定義される通信のサービス名と上記ネットワーク通信で用いられるアドレスとポート番号の対応を取ることを特徴とする請求の範囲第３１項記載の通信方法。

３３．上記ネットワーク通信のプロトコル変換によりＴＣＰ／ＩＰ（Transmission Control Protocol/Internet Protocol）への変換がなされることを特徴とする請求の範囲第３１項記載の通信方法。

３４．動作制御プログラムがオブジェクト間通信で構築されたロボット装置であって、オブジェクト間通信で構築されたコンピュータシステムとの間で有線又は無線を経由してデータ通信を行うロボット装置において、

上記オブジェクト間通信と、上記コンピュータシステムとの間でのネットワーク通信におけるプロトコル変換を動的に行うゲートウェイオブジェクトを有する

こと

を特徴とするロボット装置。

３５．上記ゲートウェイオブジェクトは、上記オブジェクト間通信で定義される通信のサービス名と上記ネットワーク通信で用いられるアドレスとポート番号の対応を取ることを特徴とする請求の範囲第３４項記載のロボット装置。

３６．上記ネットワーク通信のプロトコル変換によりＴＣＰ／ＩＰ（Transmission Control Protocol/Internet Protocol）への変換がなされることを特徴とする請求の範囲第３４項記載のロボット装置。

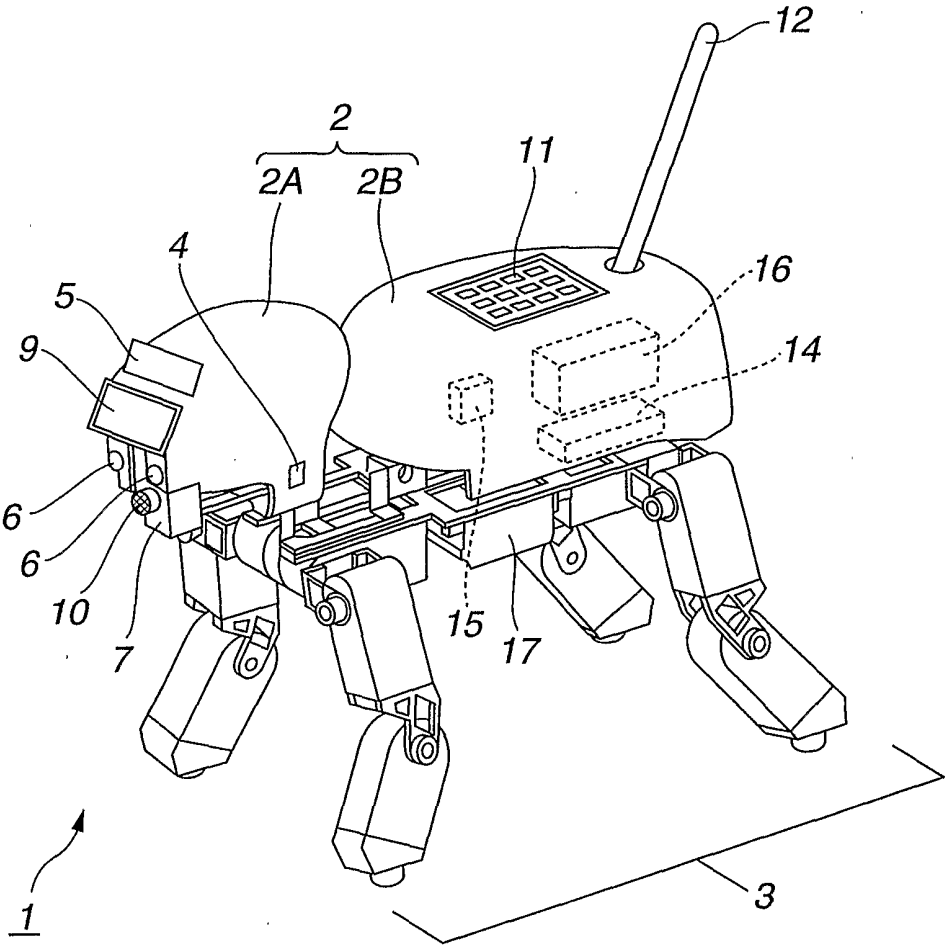


FIG.1

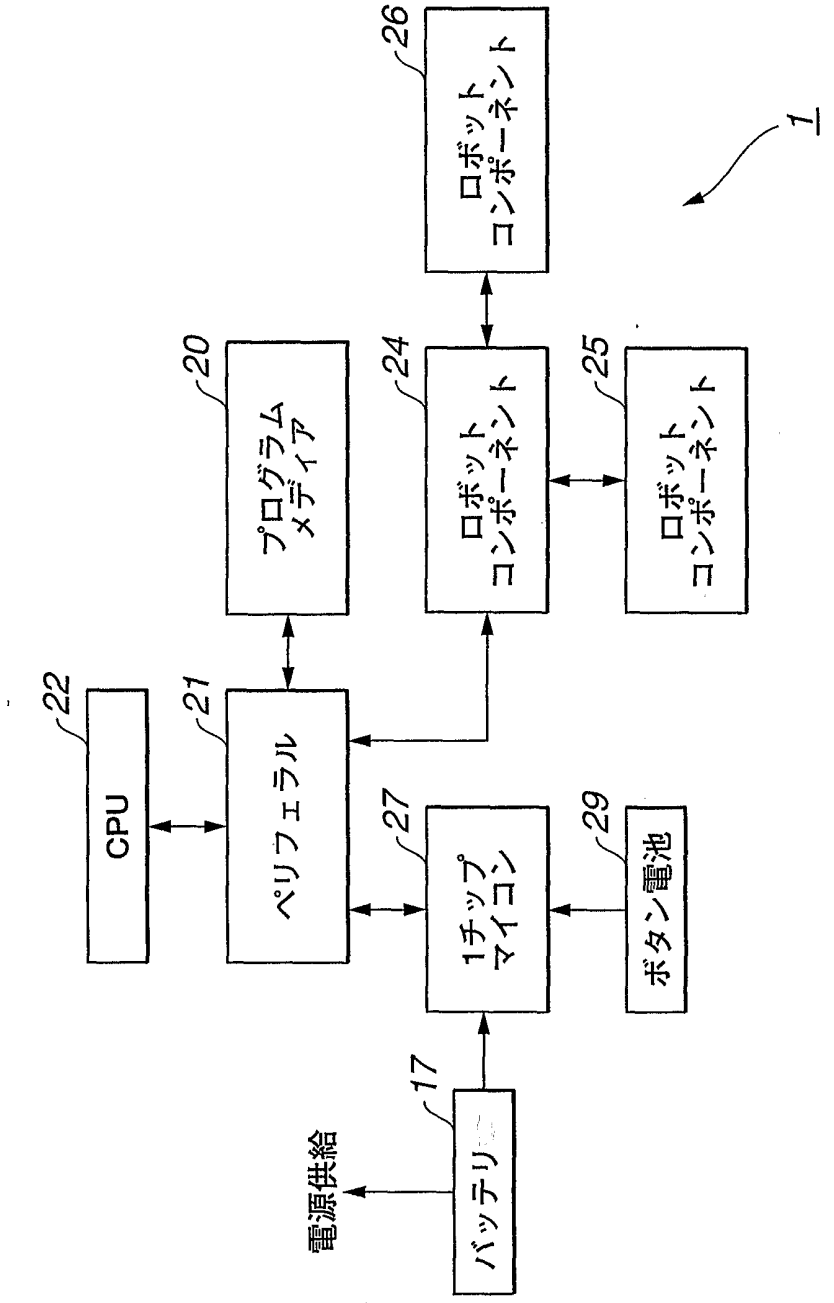

1
26
ロボット
コンポーネント
20
プログラム
メディア
24
ロボット
コンポーネント
25
ロボット
コンポーネント
22
CPU
21
ペリフェラル
27
1チップ
マイコン
29
ボタン電池
17
バッテリー
電源供給

FIG.2

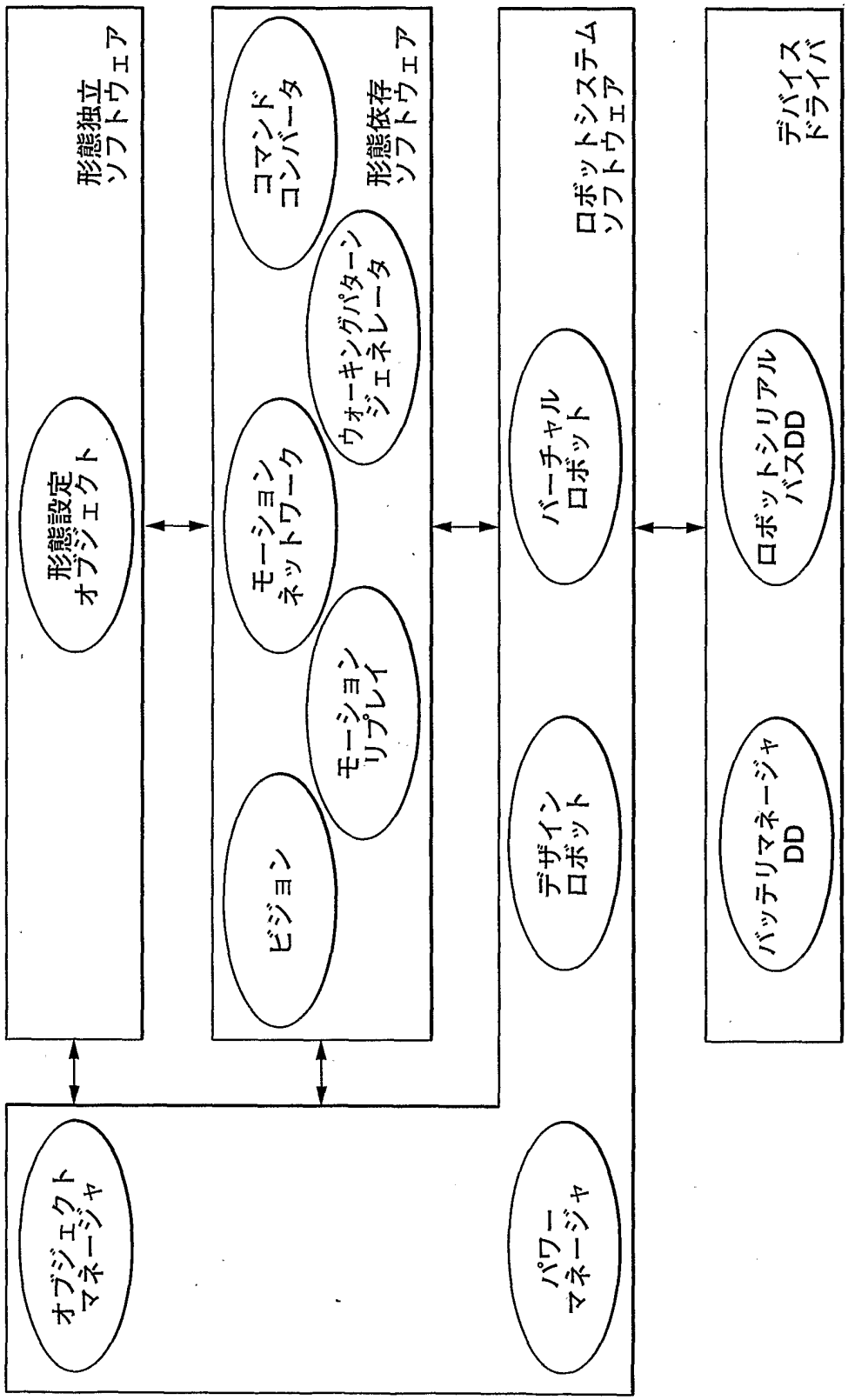

形態独立
ソフトウェア
形態設定
オブジェクト
形態依存
ソフトウェア
コマンド
コンバータ
ウォーキングパターン
ジェネレータ
モーション
ネットワーク
モーション
リプレイ
ビジョン
ロボットシステム
ソフトウェア
バーチャル
ロボット
デザイン
ロボット
パワー
マネージャ
オブジェクト
マネージャ
デバイス
ドライバ
ロボットシリアル
バスDD
バッテリマネージャ
DD

FIG.3

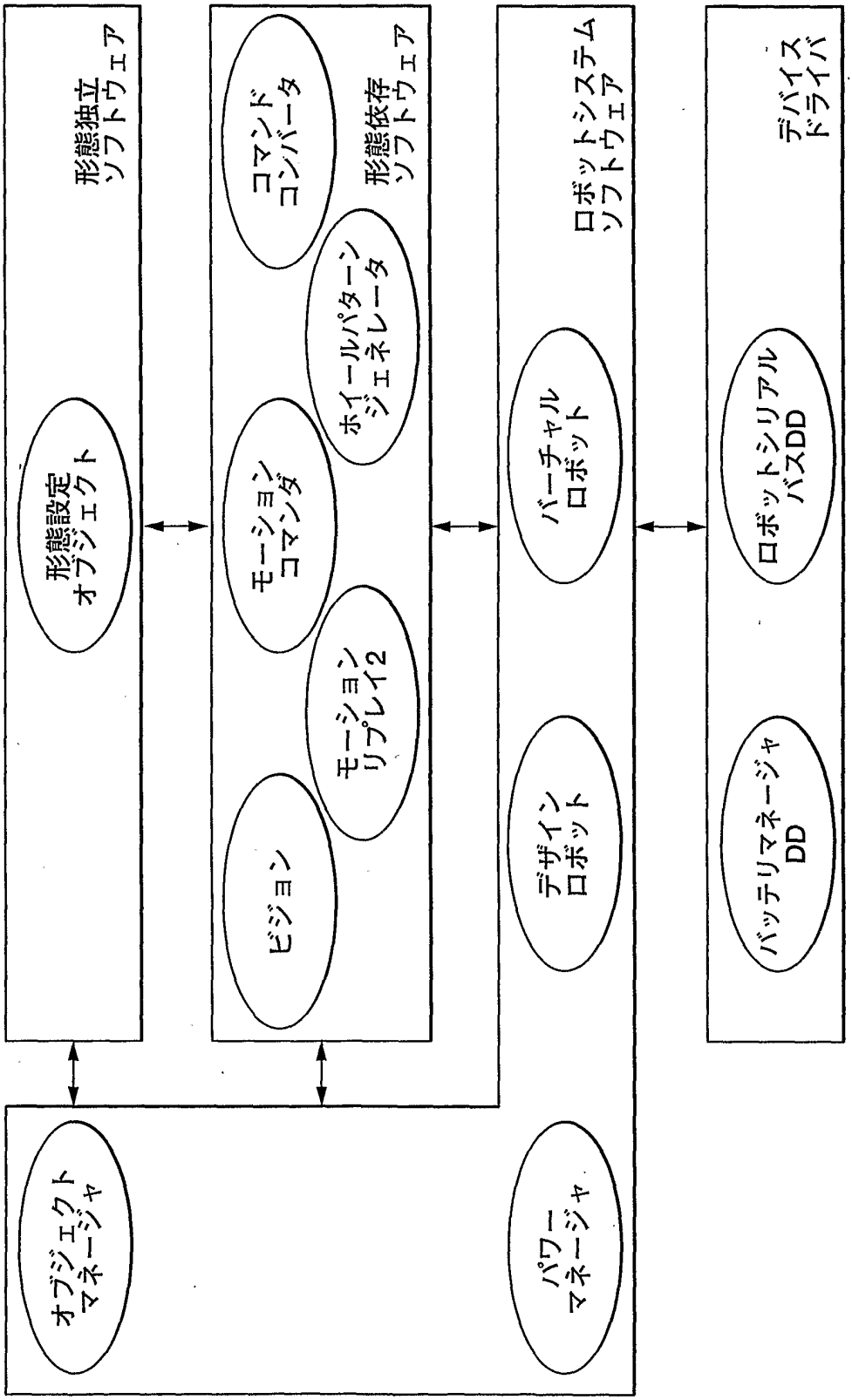

形態独立
ソフトウェア
形態設定
オブジェクト
形態依存
ソフトウェア
コマンド
コンバータ
ホイールパターン
ジェネレータ
モーション
コマンダ
モーション
リプレイ2
ビジョン
ロボットシステム
ソフトウェア
バーチャル
ロボット
デザイン
ロボット
オブジェクト
マネージャ
パワー
マネージャ
デバイス
ドライバ
ロボットシリアル
バスDD
バッテリマネージャ
DD

FIG.4

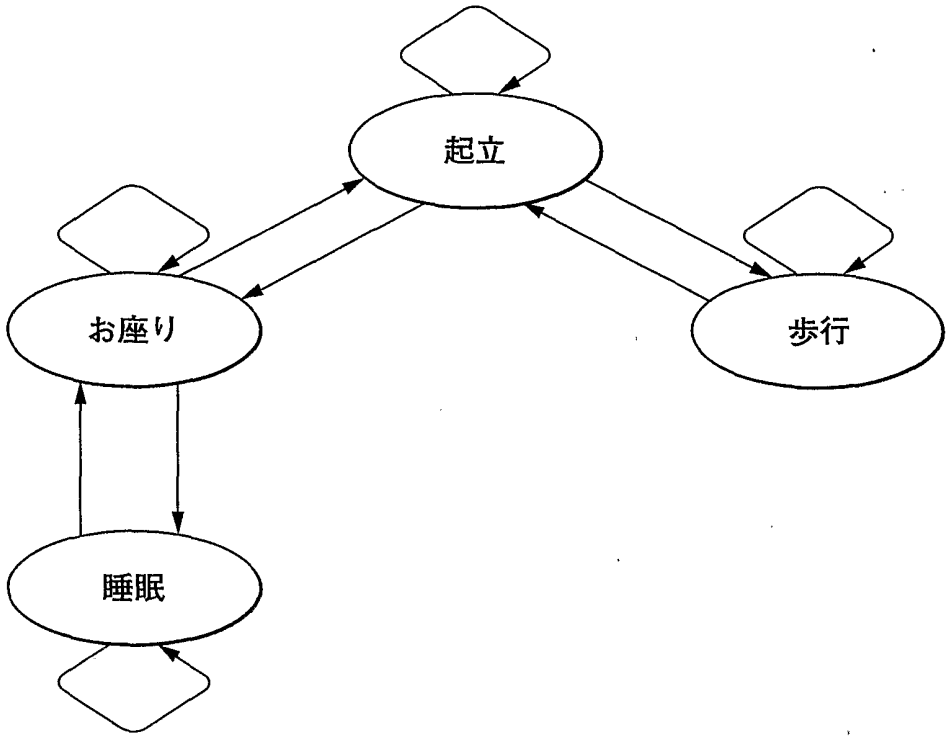


FIG.5

状態 / 入力	Sleep	Rest	Wakeup	Move
1分何も検出されない	Sleep	Sleep	Rest	Wakeup
赤、大きさ＜10が10秒以上続いた	Rest	Wakeup	Move	Move
赤、大きさ＞＝10	Wakeup	Move	Move	Move

FIG.6

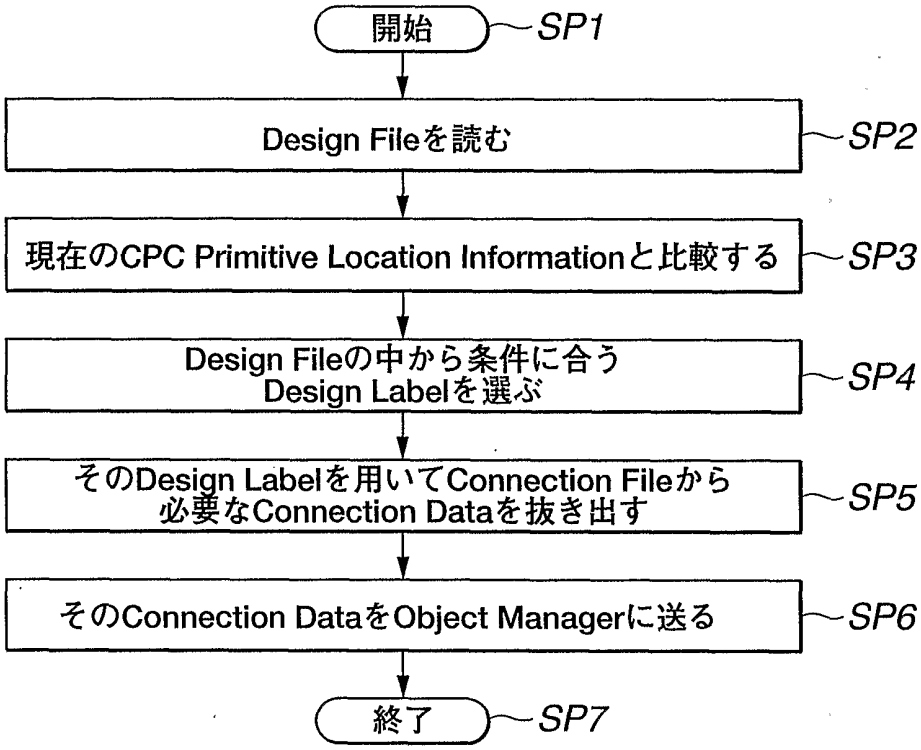

開始
SP1
Design Fileを読む
SP2
現在のCPC Primitive Location Informationと比較する
SP3
Design Fileの中から条件に合う
Design Labelを選ぶ
SP4
そのDesign Labelを用いてConnection Fileから
必要なConnection Dataを抜き出す
SP5
そのConnection DataをObject Managerに送る
SP6
終了
SP7

FIG.7

```
DRX-4Legged {
        DRX Head ;
        DRX RightFrontLeg ;
        DRX LeftFrontLeg ;
        DRX RightRearLeg ;
        DRX LeftRearLeg ;
}
DRX Wheel {
        DRX Head ;
        DRX RightFrontLeg ;
        DRX LeftFrontLeg ;
        DRX RightRearWheel ;
        DRX LeftRearWheel ;
}
DRX Head {
        DRX CameraLink ;
}
Virtual CameraLink {
        DRX CameraLink ;
        Camera ;
        Coordinate : /
}
DRX RightFrontLeg {
        Coordinate : /C1/C0-Joint ;
        Coordinate : /C1/C0/C0-Joint ;
        Coordinate : /C1/C0/C0/C0-Joint ;
        Coordinate : /C1/C0/C0/C0/C0-TouchSensor :
}
DRX LeftFrontLeg {
        Coordinate : /C2/C0-Joint ;
        Coordinate : /C2/C0/C0-Joint ;
        Coordinate : /C2/C0/C0/C0-Joint ;
        Coordinate : /C2/C0/C0/C0/C0-TouchSensor :
}
DRX RightRearLeg {
        Coordinate : /C3/C0-Joint ;
        Coordinate : /C3/C0/C0-Joint ;
        Coordinate : /C3/C0/C0/C0-Joint ;
        Coordinate : /C3/C0/C0/C0/C0-TouchSensor :
}
DRX LeftRearLeg {
        Coordinate : /C4/C0-Joint ;
        Coordinate : /C4/C0/C0-Joint ;
        Coordinate : /C4/C0/C0/C0-Joint ;
        Coordinate : /C4/C0/C0/C0/C0-TouchSensor :
}
DRX RightRearWheel {
        Coordinate : /C3/C0-Wheel ;
}
DRX LeftRearWheel {
        Coordinate : /C4/C0-Wheel ;
}
```

FIG.8

```
CPC Current Information {
        Coordinate : /C0/C0-Joint ;
        Coordinate : /C0/C0/C0-Joint ;
        Coordinate : /C0/C0/C0/C0-Joint ;
        Coordinate : /C0/C0/C0/C0/C0-Camera ;
        Coordinate : /C1/C0-Joint ;
        Coordinate : /C1/C0/C0-Joint ;
        Coordinate : /C1/C0/C0/C0-Joint ;
        Coordinate : /C1/C0/C0/C0/C0-TouchSensor :
        Coordinate : /C2/C0-Joint ;
        Coordinate : /C2/C0/C0-Joint ;
        Coordinate : /C2/C0/C0/C0-Joint ;
        Coordinate : /C2/C0/C0/C0/C0-TouchSensor :
        Coordinate : /C3/C0-Joint ;
        Coordinate : /C3/C0/C0-Joint ;
        Coordinate : /C3/C0/C0/C0-Joint ;
        Coordinate : /C3/C0/C0/C0/C0-TouchSensor :
        Coordinate : /C4/C0-Joint ;
        Coordinate : /C4/C0/C0-Joint ;
        Coordinate : /C4/C0/C0/C0-Joint ;
        Coordinate : /C4/C0/C0/C0/C0-TouchSensor :
}
```

FIG.9

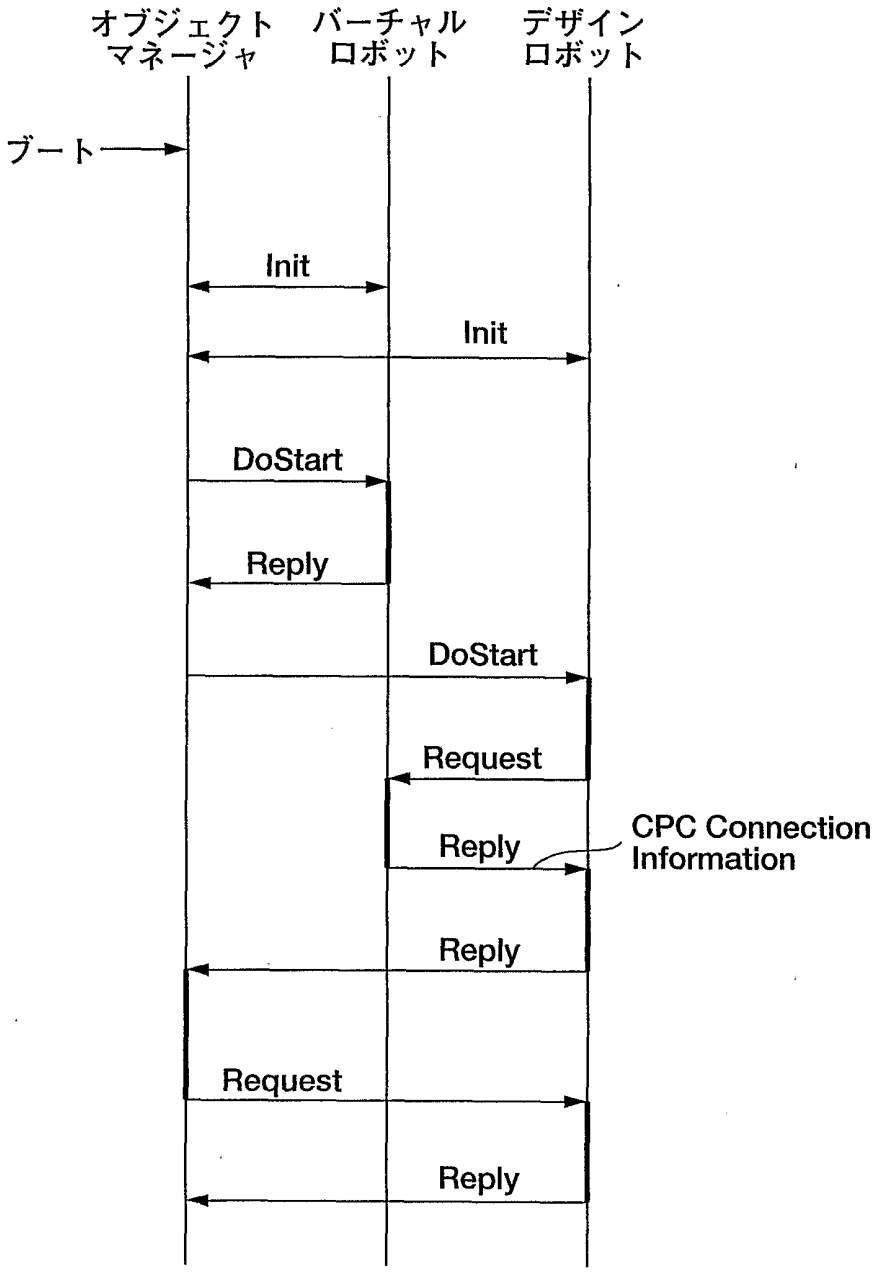

オブジェクト
マネージャ
バーチャル
ロボット
デザイン
ロボット
ブート
Init
Init
DoStart
Reply
DoStart
Request
Reply
CPC Connection
Information
Reply
Request
Reply

FIG.10

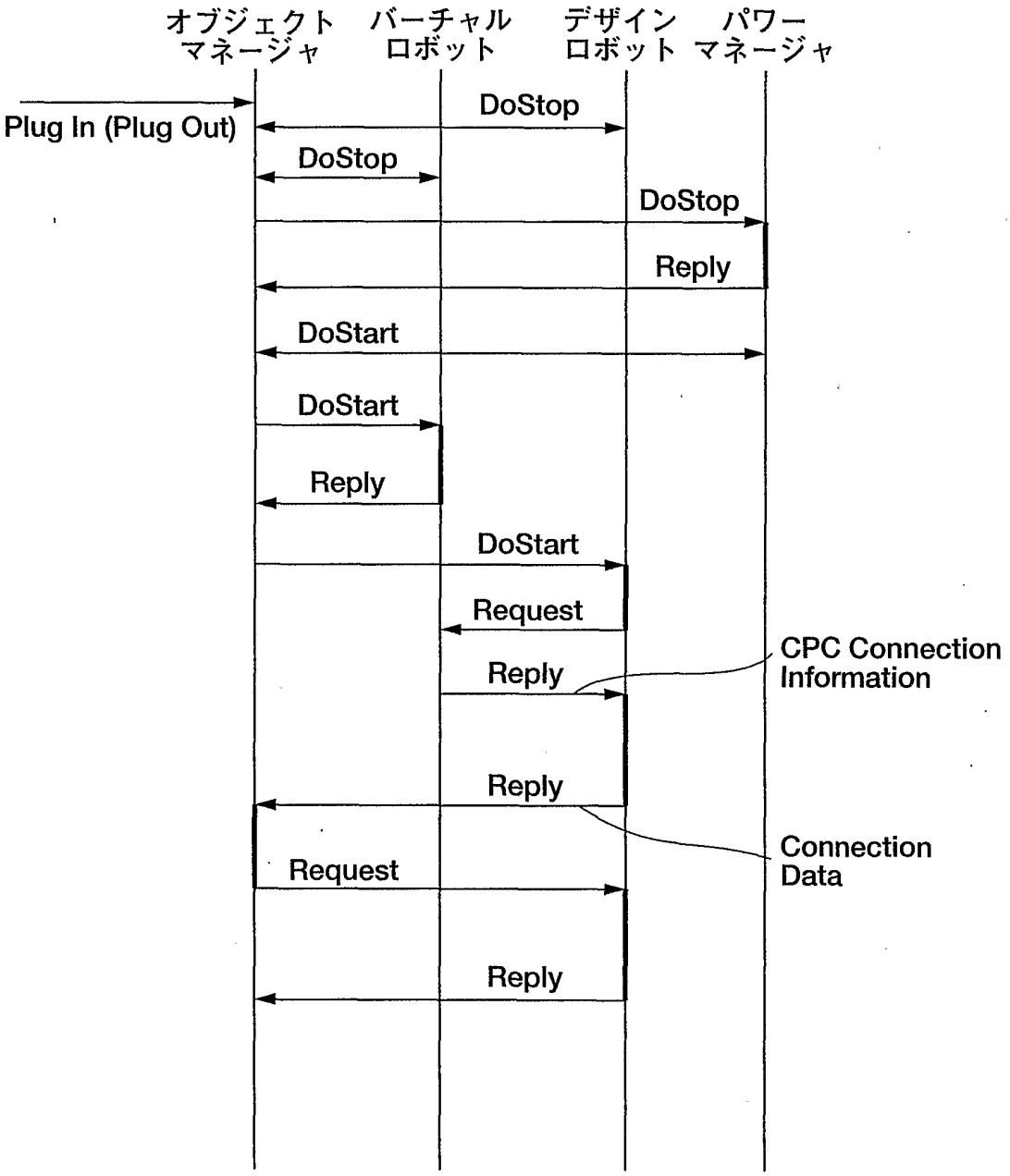

オブジェクト マネージャ
バーチャル ロボット
デザイン ロボット
パワー マネージャ
Plug In (Plug Out)
DoStop
DoStop
DoStop
Reply
DoStart
DoStart
Reply
DoStart
Request
Reply
CPC Connection Information
Reply
Connection Data
Request
Reply

FIG.11

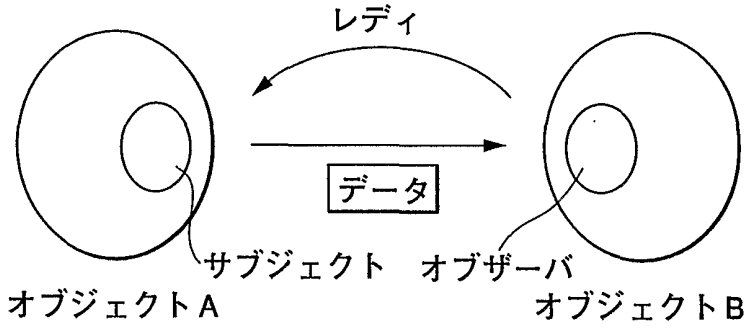

レディ
データ
サブジェクト
オブザーバ
オブジェクトA
オブジェクトB

FIG.12

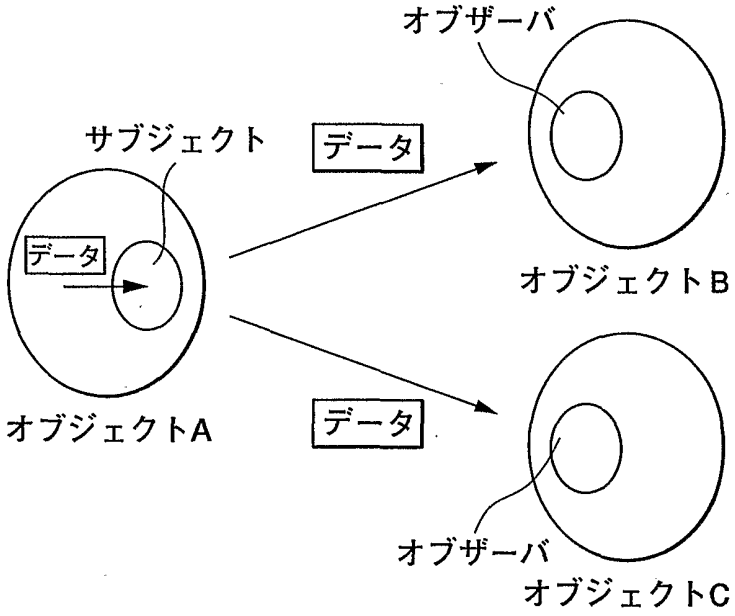

オブザーバ
サブジェクト
データ
データ
オブジェクトB
オブジェクトA
データ
オブザーバ
オブジェクトC

FIG.13

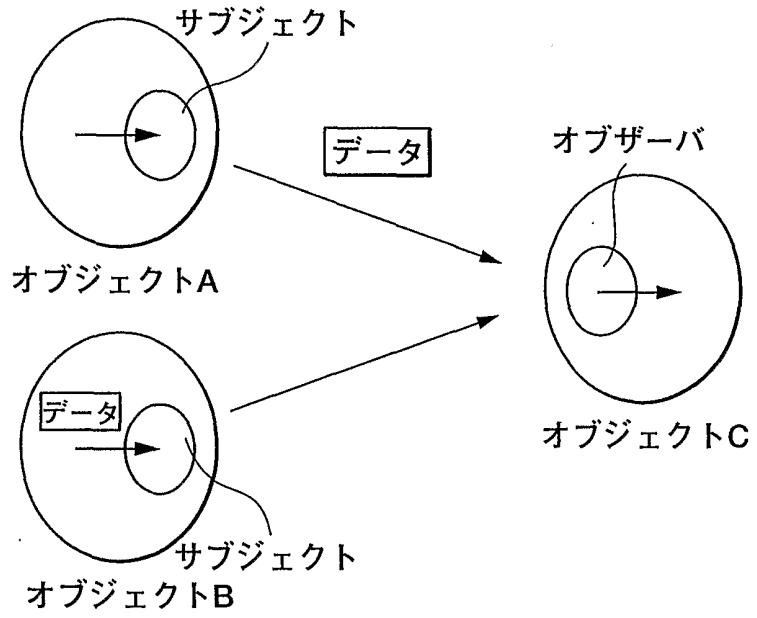


FIG.14

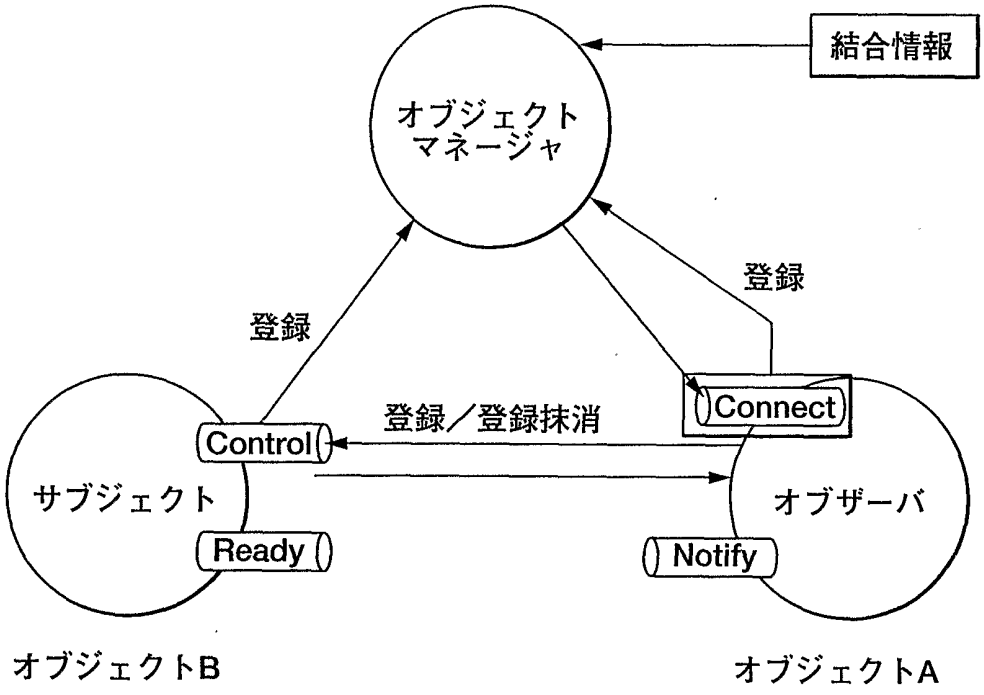

結合情報
オブジェクト
マネージャ
登録
登録
登録／登録抹消
Connect
Control
サブジェクト
オブザーバ
Ready
Notify
オブジェクトB
オブジェクトA

FIG.15

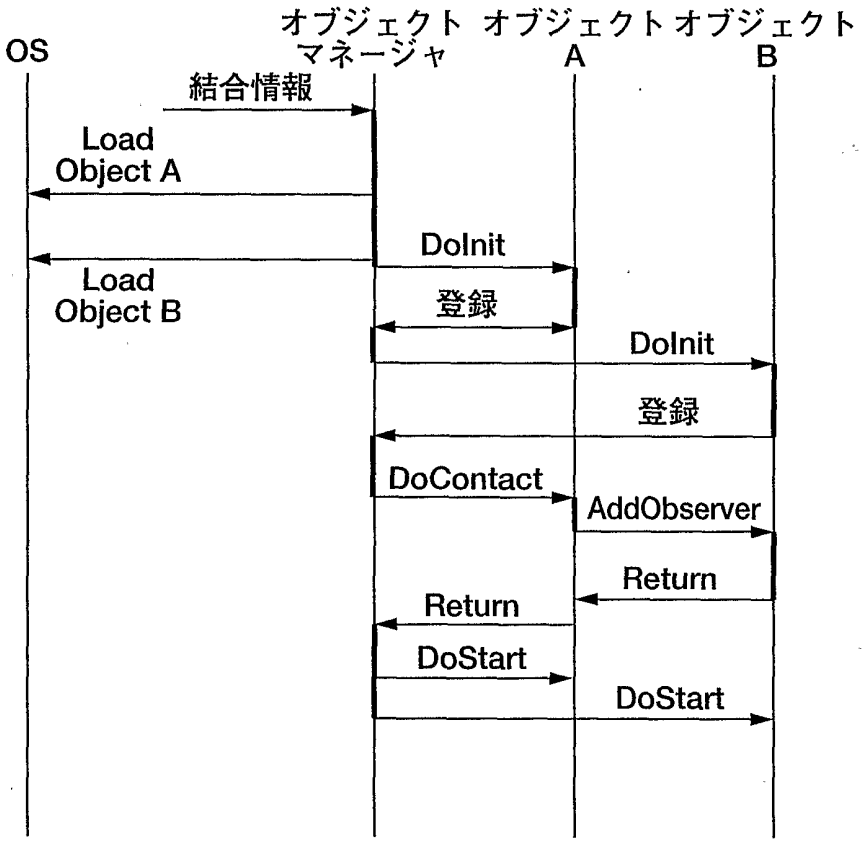

OS
オブジェクト マネージャ
オブジェクト A
オブジェクト B
結合情報
Load Object A
DoInit
Load Object B
登録
DoInit
登録
DoContact
AddObserver
Return
Return
DoStart
DoStart

FIG.16

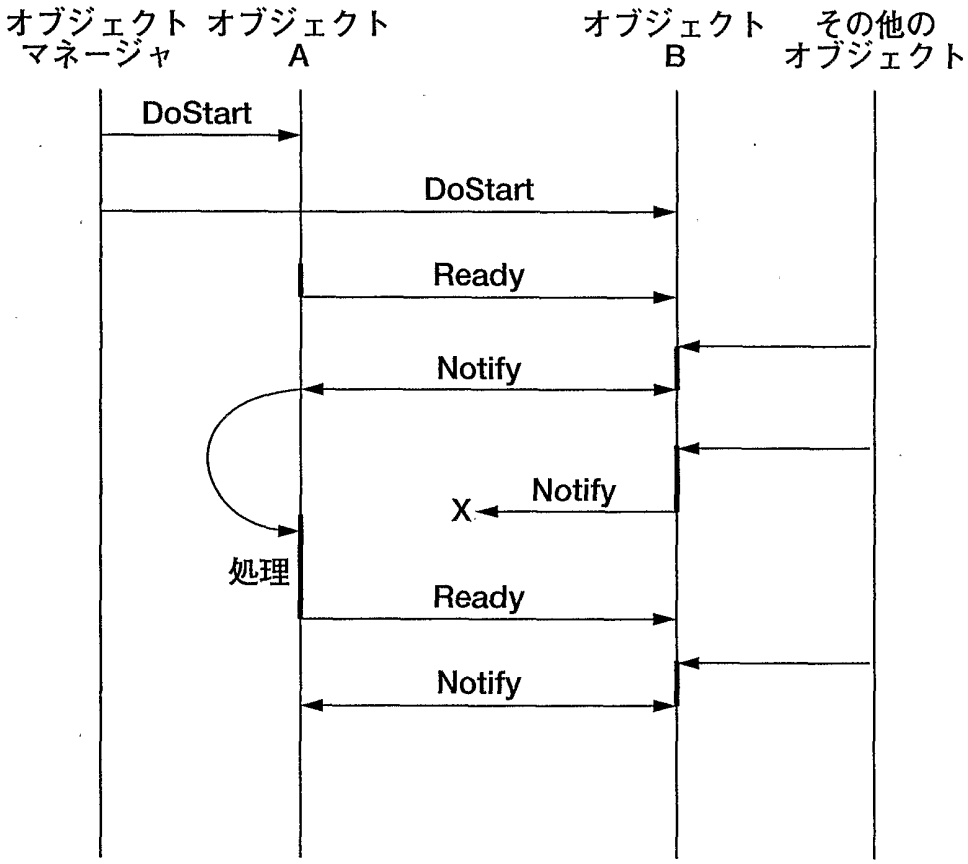

オブジェクト マネージャ
オブジェクト A
オブジェクト B
その他の オブジェクト
DoStart
DoStart
Ready
Notify
Notify
X
処理
Ready
Notify

FIG.17

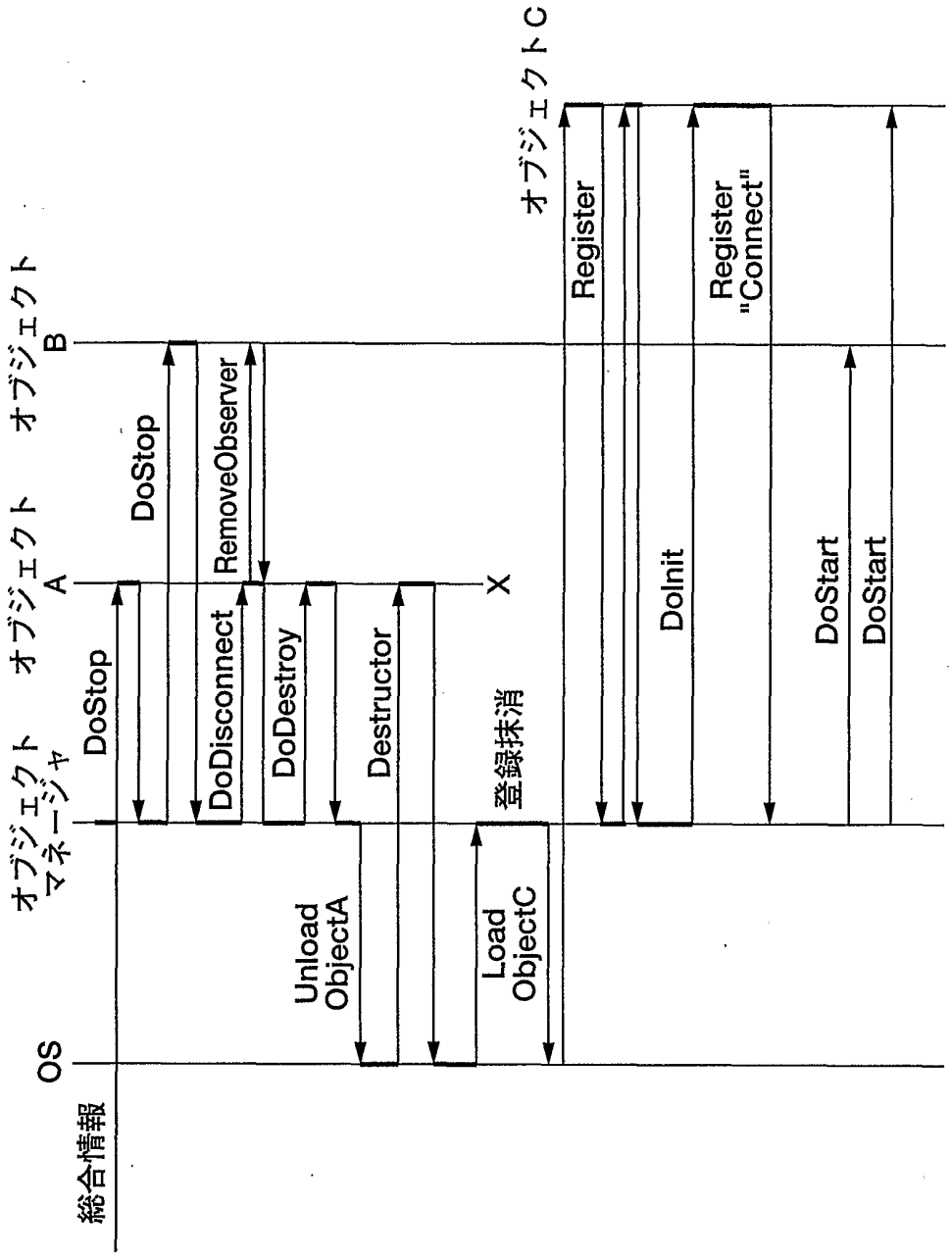

総合情報
OS
オブジェクト
マネージャ
オブジェクト
A
オブジェクト
B
オブジェクトC
DoStop
DoStop
DoDisconnect
RemoveObserver
DoDestroy
Destructor
X
登録抹消
Unload
ObjectA
Load
ObjectC
Register
Register
"Connect"
DoInit
DoStart
DoStart

FIG.18

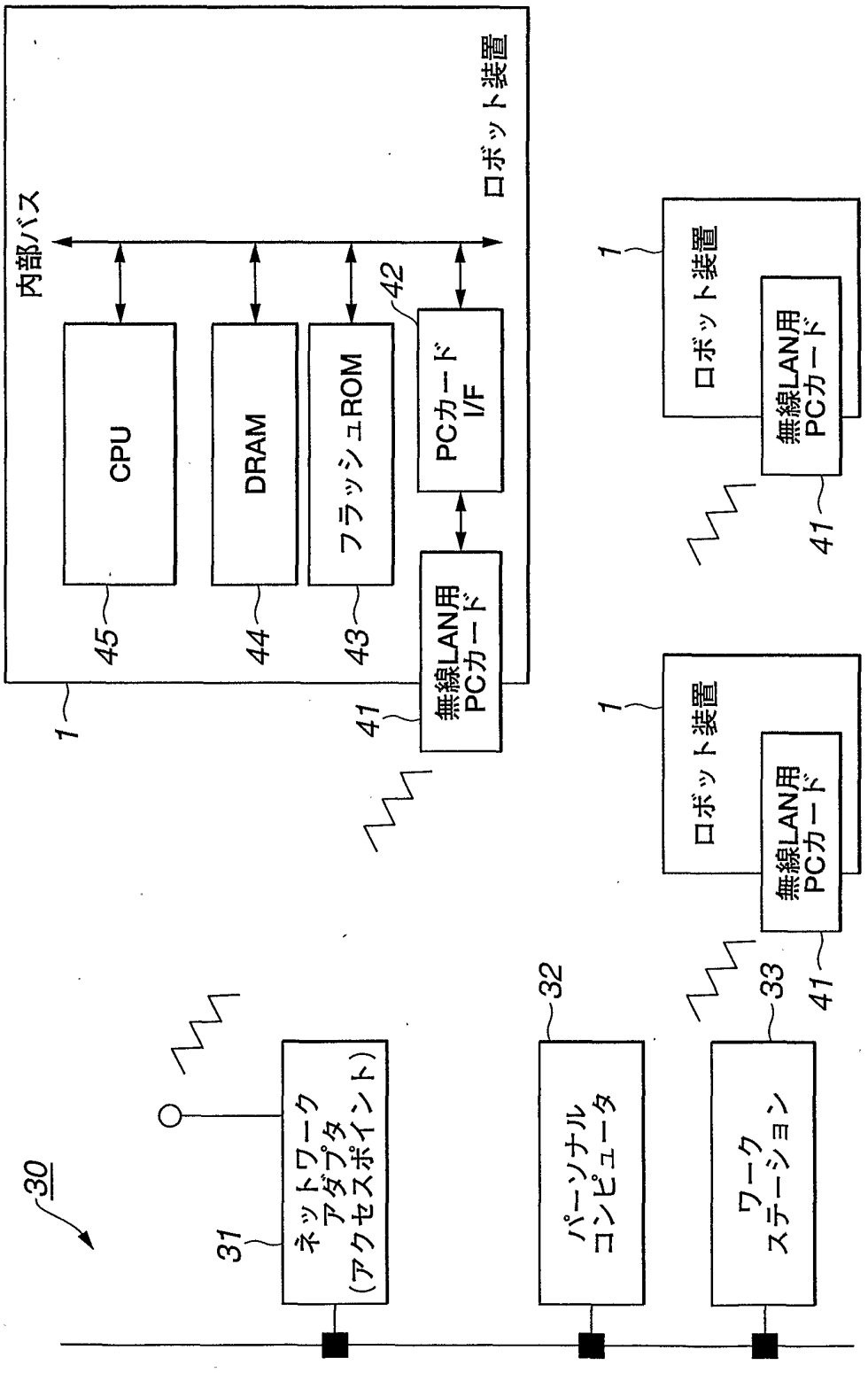

ロボット装置
内部バス
CPU
DRAM
フラッシュROM
PCカード I/F
無線LAN用 PCカード
ロボット装置
無線LAN用 PCカード
ロボット装置
無線LAN用 PCカード
ネットワーク アダプタ (アクセスポイント)
パーソナル コンピュータ
ワーク ステーション
30
31
32
33
41
42
43
44
45
1

FIG.19

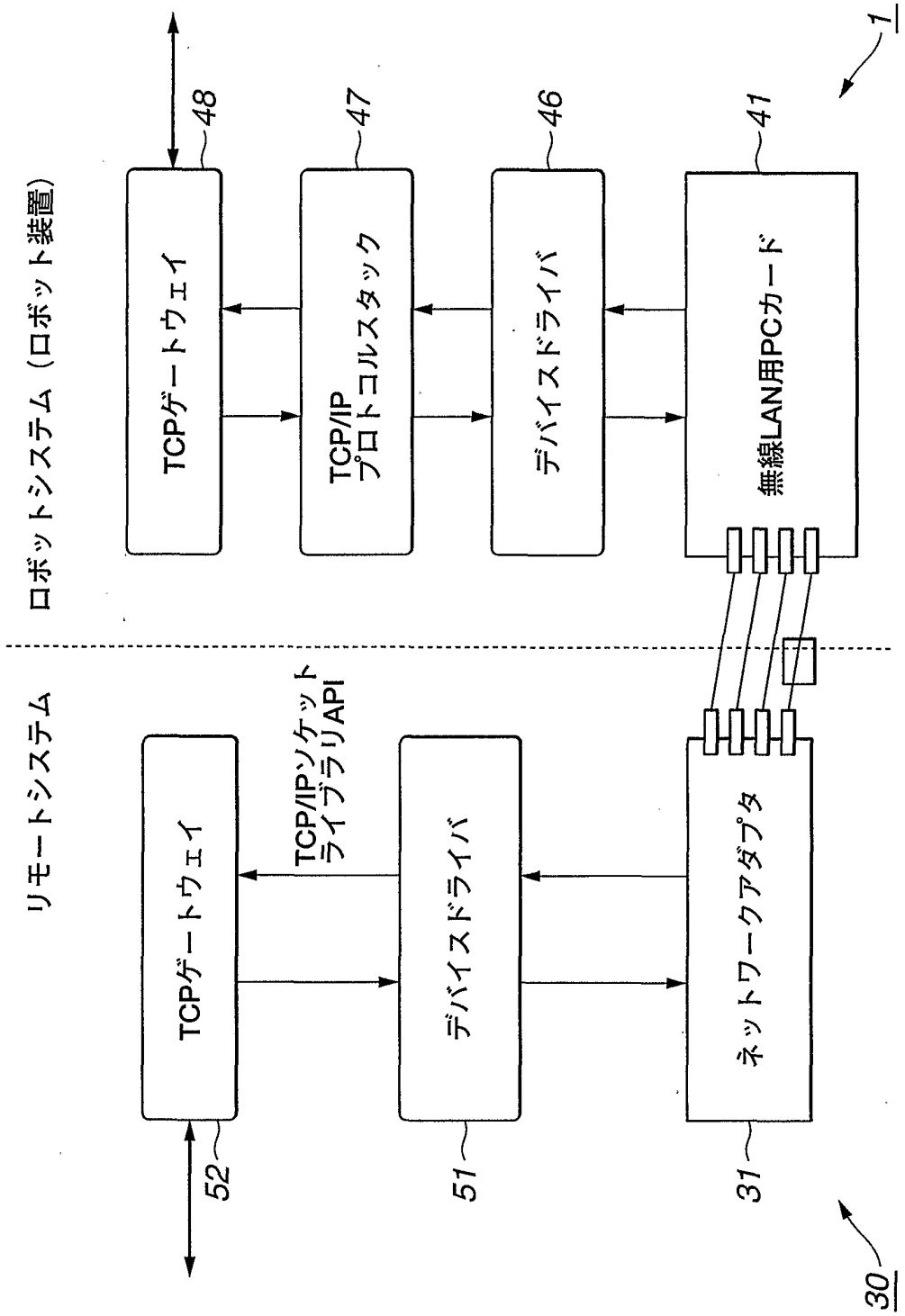

ロボットシステム（ロボット装置）
リモートシステム
TCPゲートウェイ
TCP/IP
プロトコルスタック
デバイスドライバ
無線LAN用PCカード
TCP/IPソケット
ライブラリAPI
TCPゲートウェイ
デバイスドライバ
ネットワークアダプタ
48
47
46
41
1
52
51
31
30

FIG.20

```
#OPEN-R Service <---> TCP/IP port
#(format)
#ServiceName1 port host
#ServiceName2 port host
#   :
#

#  BeNet < = PostureSC, SoundSC, VisionSC, ObstacleSC    9009  43.13.50.106
TCPGateway.visonSC_Result.OSemanticsInputData.S          9007  43.13.50.106
TCPGateway.ObstacleSC_visonSC.OSemanticsInputData.S      9001  43.13.50.106
TCPGateway.postureSC_ControlKeep.OSemanticsInputData.S   9002  43.13.50.106
TCPGateway.soundSC_ResultKeep.OSemanticsInputData.S      9003  43.13.50.106
```

FIG.21

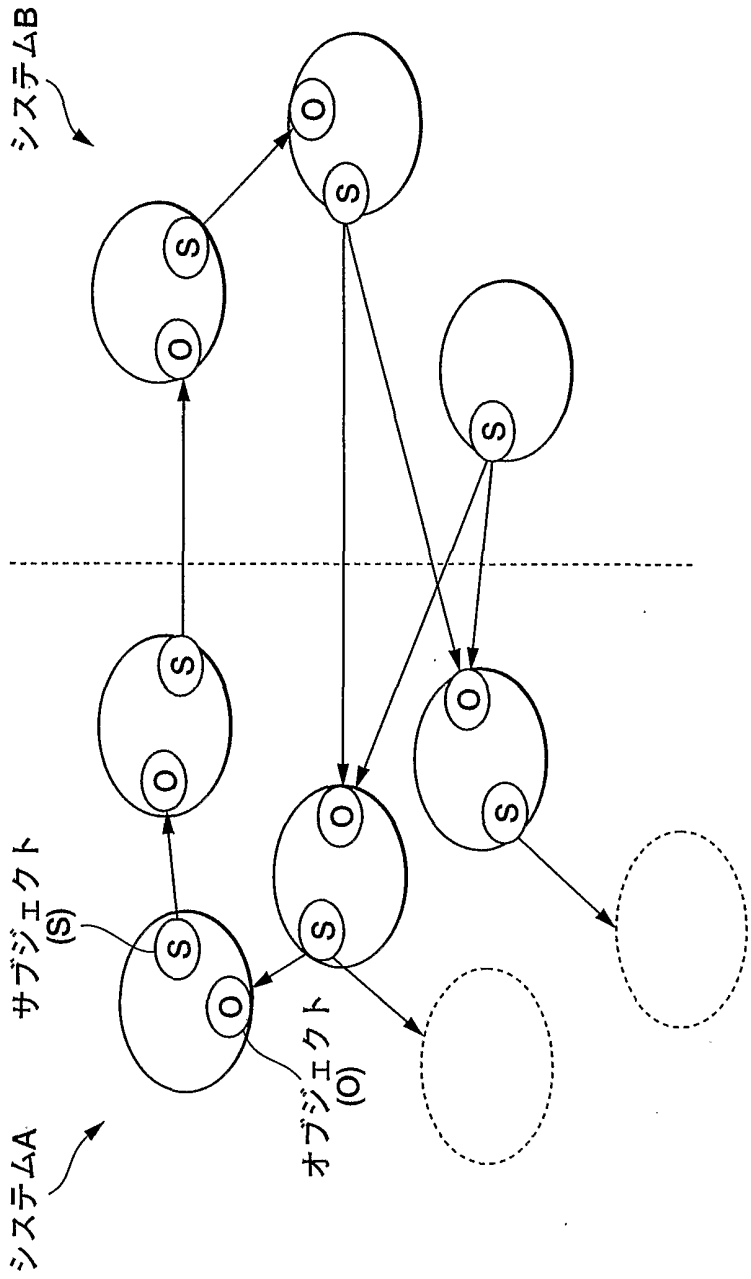


FIG.22

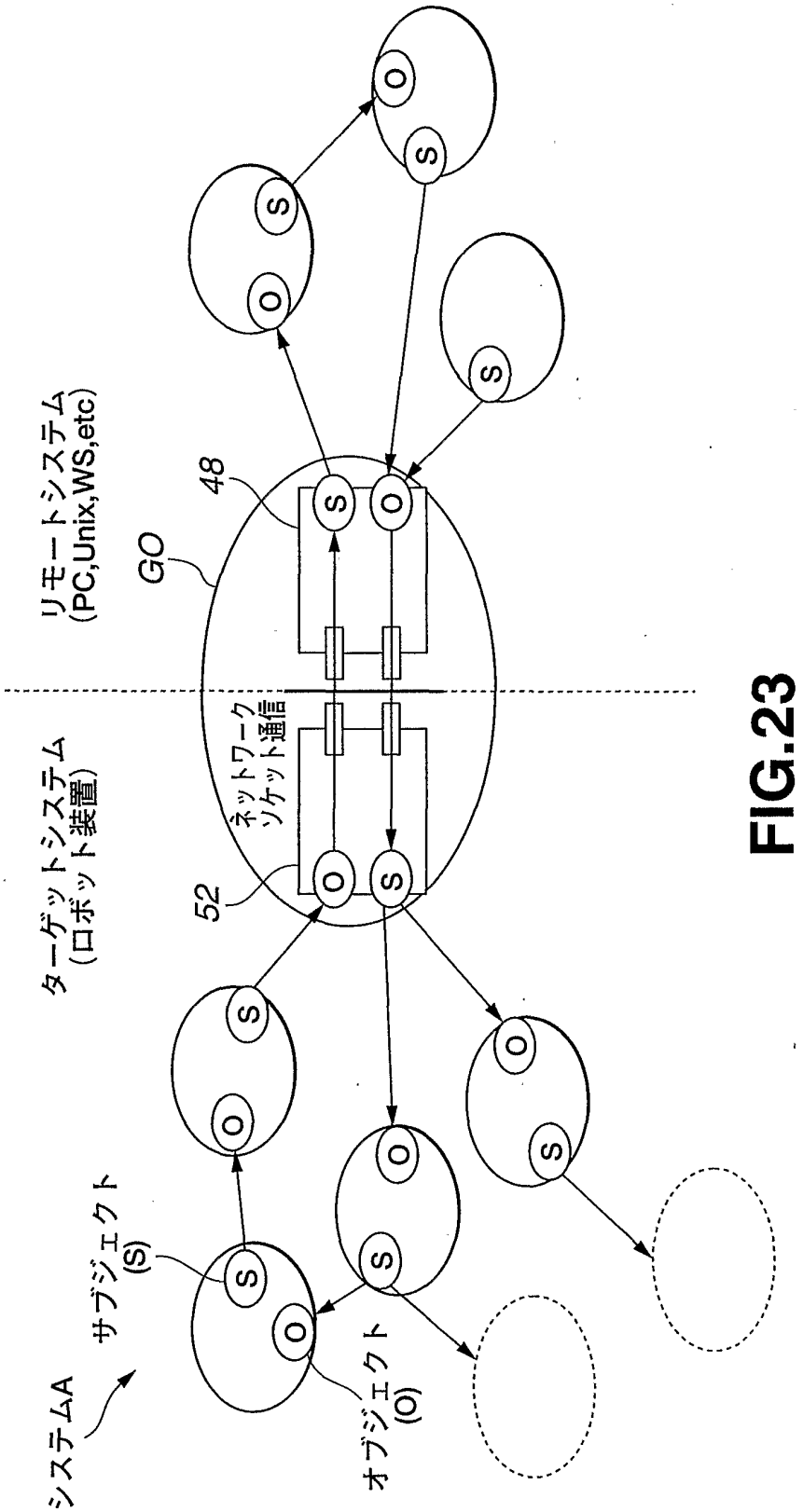

リモートシステム
(PC,Unix,WS,etc)
ターゲットシステム
(ロボット装置)
GO
48
52
ネットワーク
ソケット通信
サブジェクト
(S)
オブジェクト
(O)
システムA
S
O

FIG.23

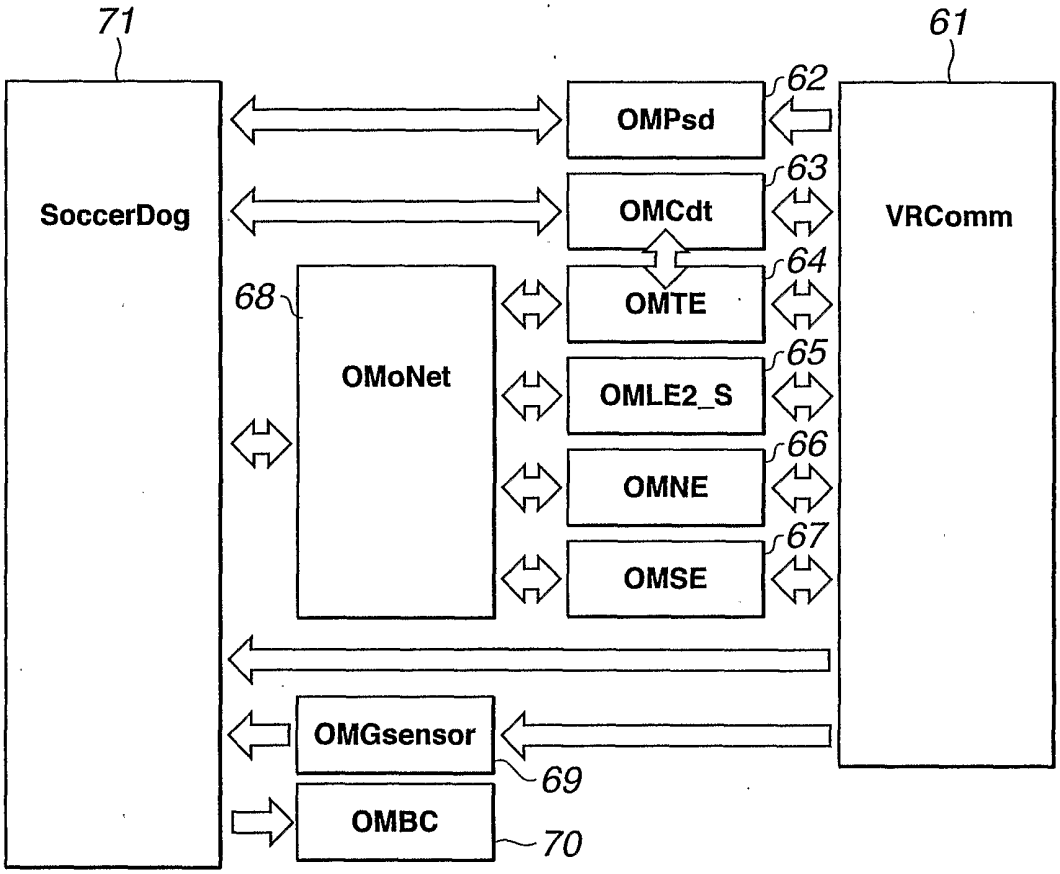

71
61
SoccerDog
VRComm
62
OMPsd
63
OMCdt
64
OMTE
65
OMLE2_S
66
OMNE
67
OMSE
68
OMoNet
OMGsensor
69
OMBC
70

FIG.24

```
//
// connect.  ctg
//
// Copyright (C) 1999 ABCD Corporation
//  All Rights Reserved.
//

//  LE2_S
OMLE2_S.Effector.OCommandVectorData.S,OVirtualRobotComm.Effector.OCommandVectorData.0
OVirualRobotComm.Sensor.OSensorFrameVectorData.S,OMLE2_S.Sensor.OSensorFrameVectorData.0
OMoNet.StartMotion.OMoNetMessage.S,OMLE2_S.Cntrl.OMoNetMessage.0
OMLE2_S.Results.OMoNetMessage.S,OMoNet.ReturnStatus.OMoNetMessage.0
OMLE2_S.Register.OMoNetMessage.S,OMoNet.RegisterMotion.OMoNetMessage.0
//  OMTE
OMTE.Effector.OCommandVectorData.S,OVirtualRobotComm.Effector.OCommandVectorData.0
OVirtualRobotComm.Sensor.OSensorFrameVectorData.S,OMTE.Sensor.OSensorFrameVectorData.0
OMoNet.StartMotion.OMoNetMessage.S,OMTE.Contrl.OMoNetMessage.0
OMTE.Results.OMoNetMessage.S,OMoNet.ReturnStatus.OMoNetMessage.0
OMTE.Register.OMoNetMessage.S,OMoNet.RegisterMotion.OMoNetMessage.0
OMCdt.ChannelO.CdtResults.S,OMTE.Image.CdtResults.0
//  OMSE
```

FIG.25

```
OMSE.Effector.OCommandVectorData.S,OVirtualRobotComm.Effector.OCommandVectorData.0
OVirualRobotComm.Sensor.OSensorFrameVectorData.S,OMSE.Sensor.OSensorFrameVectorData.0
OMoNet.StartMotion.OMoNetMessage.S,OMSE.Cntrl.OMoNetMessage.0
OMSE.Results.OMoNetMessage.S,OMoNet.ReturnStatus.OMoNetMessage.0
OMSE.Register.OMoNetMessage.S,OMoNet.RegisterMotion.OMoNetMessage.0
//  OMNE
OMNE.Effector.OCommandVectorData.S,OVirtualRobotComm.Effector.OCommandVectorData.0
OVirtualRobotComm.Sensor.OSensorFrameVectorData.S,OMNE.Sensor.OSensorFrameVectorData.0
OMoNet.StartMotion.OMoNetMessage.S,OMNE.Contrl.OMoNetMessage.0
OMNE.Results.OMoNetMessage.S,OMoNet.ReturnStatus.OMoNetMessage.0
OMNE.Register.OMoNetMessage.S,OMoNet.RegisterMotion.OMoNetMessage.0
//  OMPsd
OVirtualRobotComm.Sensor.OSensorFrameVectorData.S,OMPsd.Sensor.OSensorFrameVectorData.0
//  OMCDT
OVitualRobotComm.FBK_M_YUVC.OImageVector.S,OMCdt.Image.OImage.Vector.0
```

FIG.26

OMCdt.CdtTable,OCdtVectorData,S,OVirtualRobotComm.SetCdtVector.OCdtVectorData.0
// OMGsensor
OVirtualRobotComm.Sensor.OSensorFrameVectorData.S,omG.UpdateData.OSensorFrameVectorData.0
// SoccerDog
SoccerDog.SetBCParam.OMBatteryClockParam.S,omBC.Parameter.OMBatteryClockParam.0
SoccerDog.SetMotion.OMoNetMessage.S,OMoNet.Move.OMoNetMessage.0
SoccerDog.SetPSDParam.PsdCntrl.S,OMPsd.ParamCntrl.PsdCntrl.0
SoccerDog.SetCDTTable.CdtCntrl.S,OMCdt.Command.CdtCntrl.0
OMoNet.FinishCommand.ONoNetMessage.S,SoccerDog.GetMotionResult.OMoNetMessage.0
OVirtualRobotComm.Sensor.OSensorFrameVectorData.S,SoccerDog.Sensor.OSensorFrameVectorData.0
OMPsd.Result,PsdResults.S,SoccerDog.GetPSDResult.PsdResults.0
OMCdt.ChannelO.CdtResults.S,SoccerDog.GetCDTResult.CdtResults.0
omG.Control_E.OMGsensorMessage.S,SoccerDog.GetGResult.OMGsensorMessage.0

FIG.27

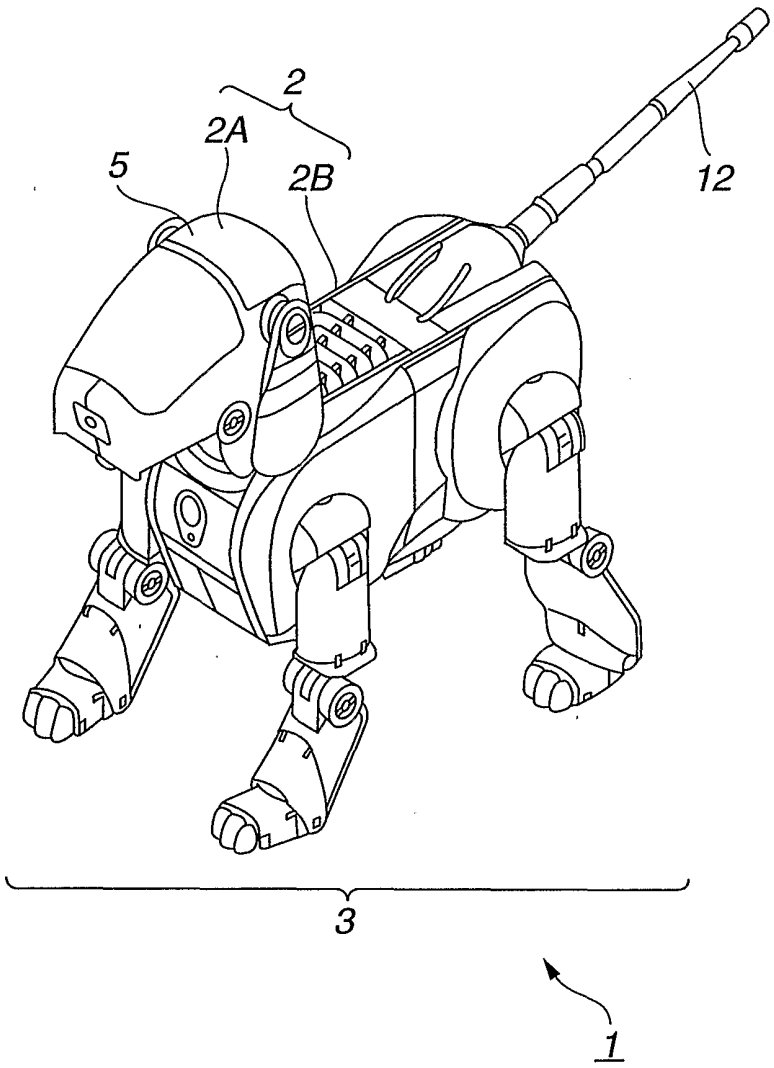


FIG.28

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP01/07394

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl[7] G06F9/46, G06F9/44, G06F13/00, B25J13/00, G05B19/02, B25J5/00, A63H11/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
Int.Cl[7] G06F9/46-9/54, G06F9/44, G06F13/00, G06F9/06, G06F11/28-11/34, B25J1/00-21/02, G05B19/00-19/16, G05B24/00-24/04, A63H1/00-37/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Jitsuyo Shinan Koho 1922-1996 Toroku Jitsuyo Shinan Koho 1994-2001
Kokai Jitsuyo Shinan Koho 1971-2001 Jitsuyo Shinan Toroku Koho 1996-2001

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 940748 A2 (International Business Machines Corporation), 08 September, 1999 (08.09.99), Par. No. [0011]; Figs. 1(a) to 1(c)	1-3,9-14
Y	Par. No. [0011]; Figs. 1(a) to 1(c) & JP 11-312088 A Par. No. [0007]; Figs. 1(a) to 1(c)	4-8,15-36
Y	EP 877320 A1 (International Business Machines Corporation), 11 November, 1998 (11.11.98), column 8, line 25 to column 11, line 53; Fig. 1 & US 5909569 A & JP 10-320322 A, Par. Nos. [0023] to [0036]; Fig. 1	4-8,15-19, 23-36
Y	JP 10-240504 A (Digital Electronics Corporation), 11 September, 1998 (11.09.98), Par. Nos. [0020], [0029] to [0037]; Figs. 3, 6 (Family: none)	20-27

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 October, 2001 (30.10.01)	06 November, 2001 (06.11.01)
Name and mailing address of the ISA/ Japanese Patent Office	Authorized officer
Facsimile No.	Telephone No.

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP01/07394

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2000-210886 A (Sony Corporation), 02 August, 2000 (02.08.00), Full text & WO 00/43168 A1 & EP 1120205 A1	1-36
A	JP 10-320367 A (Fujitsu Limited), 04 December, 1998 (04.12.98), Figs. 11 to 12 (Family: none)	6-8,17-19, 25-27
A	JP 4-74345 U (Mitsubishi Electric Corporation), 29 June, 1992 (29.06.92), page 4, line 5 to page 6, line 6; Figs. 1 to 2 (Family: none)	20-27

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ０１／０７３９４

Ａ．　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Ｉｎｔ　Ｃｌ[7]　　Ｇ０６Ｆ９／４６，Ｇ０６Ｆ９／４４，Ｇ０６Ｆ１３／００，Ｂ２５Ｊ１３／００，Ｇ０５Ｂ１９／０２，Ｂ２５Ｊ５／００，Ａ６３Ｈ１１／００

Ｂ．　調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

Ｉｎｔ　Ｃｌ[7] Ｇ０６Ｆ９／４６－９／５４，Ｇ０６Ｆ９／４４，Ｇ０６Ｆ１３／００，Ｇ０６Ｆ９／０６，Ｇ０６Ｆ１１／２８－１１／３４，Ｂ２５Ｊ１／００－２１／０２，Ｇ０５Ｂ１９／００－１９／１６，Ｇ０５Ｂ２４／００－２４／０４，Ａ６３Ｈ１／００－３７／００

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	１９２２－１９９６年
日本国公開実用新案公報	１９７１－２００１年
日本国登録実用新案公報	１９９４－２００１年
日本国実用新案登録公報	１９９６－２００１年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

Ｃ．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
	ＥＰ　９４０７４８　Ａ２（ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＣＨＩＮＥＳ　ＣＯ．）８．９月．１９９９（０８．０９．９９），第１１段落，第１ａ－１ｃ図	
Ｘ		１－３，９－１４
Ｙ	第１１段落，第１ａ－１ｃ図	４－８，１５－３６
	＆　ＪＰ　１１－３１２０８８　Ａ，第７段落，第１ａ－１ｃ図	

[x] Ｃ欄の続きにも文献が列挙されている。　　[ ] パテントファミリーに関する別紙を参照。

＊　引用文献のカテゴリー

「Ａ」特に関連のある文献ではなく、一般的技術水準を示すもの

「Ｅ」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「Ｌ」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）

「Ｏ」口頭による開示、使用、展示等に言及する文献

「Ｐ」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献

「Ｔ」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「Ｘ」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Ｙ」特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「＆」同一パテントファミリー文献

国際調査を完了した日　３０．１０．０１	国際調査報告の発送日　06.11.01
国際調査機関の名称及びあて先 日本国特許庁（ＩＳＡ／ＪＰ） 郵便番号１００－８９１５ 東京都千代田区霞が関三丁目４番３号	特許庁審査官（権限のある職員）　清木　泰　５Ｂ　２９４４ 電話番号　０３－３５８１－１１０１　内線　３５４５

様式ＰＣＴ／ＩＳＡ／２１０（第２ページ）（１９９８年７月）

C（続き）．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
Y	EP　877320　A1 （INTERNATIONAL　BUSINESS MACHINES　CO.） 11．11月．1998（11．11．98） 第8欄第25行ー第11欄第53行，第1図 ＆　US　5909569　A ＆　JP　10－320322　A，第23－36段落，第1図	4－8， 15－19， 23－36
Y	JP　10－240504　A（株式会社デジタル） 11．9月．1998（11．09．98）， 第20段落，第29－37段落，第3図，第6図 （ファミリーなし）	20－27
A	JP　2000－210886　A（ソニー株式会社） 2．8月．2000（02．08．00），全文 ＆　WO　00／43168　A1 ＆　EP　1120205　A1	1－36
A	JP　10－320367　A（富士通株式会社） 4．12月．1998（04．12．98）， 第11－12図（ファミリーなし）	6－8， 17－19， 25－27
A	JP　4－74345　U（三菱電機株式会社） 29．6月．1992（29．06．92）， 第4頁第5行ー第6頁第6行，第1－2図（ファミリーなし）	20－27