

(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2005-228183

(P2005-228183A)

(43) 公開日 平成17年8月25日(2005.8.25)

(51) Int. Cl. ⁷	F I	テーマコード (参考)
G06F 9/46	G06F 9/46 360F	5B033
G06F 9/40	G06F 9/40 310B	5B076
G06F 11/00	H04M 11/00 302	5B098
H04M 11/00	G06F 9/06 630A	5K101

審査請求 未請求 請求項の数 11 O L (全 19 頁)

(21) 出願番号	特願2004-37804 (P2004-37804)	(71) 出願人	000005108 株式会社日立製作所 東京都千代田区丸の内一丁目6番6号
(22) 出願日	平成16年2月16日 (2004.2.16)	(74) 代理人	100068504 弁理士 小川 勝男
		(74) 代理人	100086656 弁理士 田中 恭助
		(74) 代理人	100094352 弁理士 佐々木 孝
		(72) 発明者	森本 義章 神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所 内

最終頁に続く

(54) 【発明の名称】 プログラム実行方法、および、プログラム実行のための計算機システム

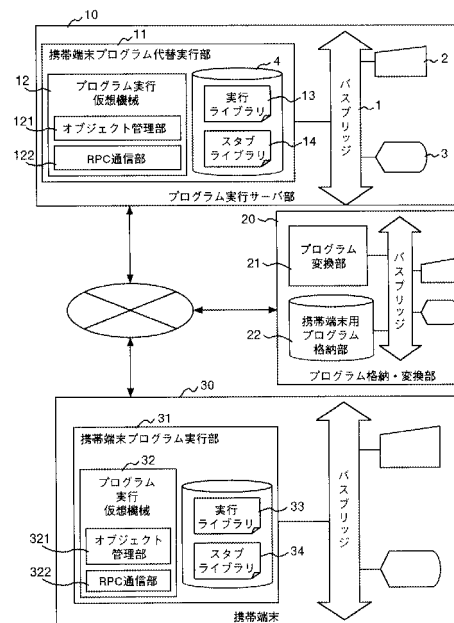
(57) 【要約】

【課題】サーバと携帯端末からなる計算機システムにおいて、既に広く流布している携帯端末向けのプログラムを修正することなく、携帯端末上での実行の際に課せられる処理速度、メモリ容量などの携帯端末の特性から生ずる様々な制限を取り除きつつ、従来と同様な機能を携帯端末で実行する。

【解決手段】プログラム格納・変換部は、関数呼び出しの変更情報により、該当する関数呼び出しをスタブ関数の呼び出しに変更し、プログラム実行サーバに配布する。携帯端末にも、実行するプログラムを配布し、サーバから、スタブ関数の呼び出しにより、スタブ関数を呼び出して、そのスタブ関数から携帯端末上のプログラムを、通信路を介して実行する。

【選択図】 図1

図 1



【特許請求の範囲】

【請求項 1】

第一のプログラム実行装置と第二のプログラム実行装置と記憶装置とからなる計算機システムのプログラム実行方法において、

前記記憶装置から第一のプログラムと第二のプログラムとを読み出すステップと、

前記記憶装置から前記第一のプログラムと前記第二のプログラムのそれぞれに記述されている関数呼び出しの置換情報を読み出すステップと、

読み出された前記第一のプログラムと前記第二のプログラムのそれぞれに記述されている関数呼び出しを、それぞれ前記置換情報に基づき別の関数呼び出しに置換するステップと、

前記関数呼び出しを置換された第一のプログラムと第二のプログラムとを、前記第一のプログラム実行装置と前記第二のプログラム実行装置に、それぞれ配布するステップと、

前記第一のプログラム実行装置において前記第一のプログラムを実行し、前記第一のプログラムの置換された別の関数呼び出しにより、前記第二のプログラムの置換された関数を呼び出して、前記第二のプログラム実行装置において前記第二のプログラムを実行するステップとを有することを特徴とするプログラム実行方法。

10

【請求項 2】

前記計算機システムは、さらに、プログラムを配布する装置と、前記プログラム実行装置と前記プログラムを配布する装置とを接続する通信手段とを有し、

前記プログラムを配布する装置は、前記通信手段によって前記第一のプログラム実行装置と前記第二のプログラム実行装置とに関数呼び出しを置換された前記第一のプログラムと前記第二のプログラムとを、それぞれ配布することを特徴とする請求項 1 記載のプログラム実行方法。

20

【請求項 3】

前記関数呼び出しの置換情報は、プログラム実行環境情報、要求性能の諸元を含み、これらの情報を参照して、別の関数呼び出しの置換の処理をおこなうことを特徴とする請求項 1 記載のプログラム実行方法。

【請求項 4】

プログラムライブラリを有する計算機システムにおいて、

プログラム実行サーバと、

プログラム格納・変換部とを有し、

前記プログラム格納・変換部は、関数呼び出しの置換情報を保持し、前記プログラムライブラリよりプログラムを読み出して、前記関数呼び出しの置換情報により、該当する関数呼び出しを仲介関数の呼び出しに置換し、前記プログラム実行サーバに配布することを特徴とする計算機システム。

30

【請求項 5】

さらに、プログラム実行端末を有し、

前記プログラム格納・変換部は、このプログラム実行端末に実行するプログラムを配布し、

前記プログラム実行サーバでは、前記関数呼び出しを置換した仲介関数の呼び出しにより、仲介関数を呼び出して、

前記仲介関数から前記プログラム実行端末上のプログラムを、通信路を介して実行することを特徴とする請求項 4 記載の計算機システム。

40

【請求項 6】

前記プログラム実行サーバと前記プログラム実行端末のプログラムの実行は、オブジェクトのメソッド実行によりおこなわれることを特徴とする請求項 5 記載の計算機システム。

【請求項 7】

前記プログラム実行サーバと前記プログラム実行端末のプログラムの実行は、前記プログラム実行端末において、オブジェクトがイベントを検知することをトリガーとしておこ

50

なわれることを特徴とする請求項 5 記載の計算機システム。

【請求項 8】

前記プログラム実行サーバとプログラム実行端末は、一方にオブジェクト、他方にそれと対となる仲介オブジェクトを有し、

前記プログラム実行サーバから前記プログラム端末の関数の実行は、それらのオブジェクト間の通信によりおこなわれることを特徴とする請求項 5 記載の計算機システム。

【請求項 9】

前記プログラム実行端末は、そのプログラム実行端末上のイベント発生を検知するリスナ代替オブジェクトを有し、

前記プログラム実行サーバは、イベントを処理するイベント関数を呼び出すリスナオブジェクトを有し、

前記プログラム実行端末で、イベント発生したときに、前記リスナ代替オブジェクトは、前記プログラム実行サーバのリスナオブジェクトと通信することにより、前記プログラム実行サーバのイベント処理関数を実行することを特徴とする請求項 5 記載の計算機システム。

【請求項 10】

携帯端末のプログラムを実行するプログラム実行サーバにおいて、

このプログラム実行サーバは、携帯端末において実行されるプログラムの関数呼び出しの部分の仲介関数呼び出しに置き換えたプログラムを実行し、

前記仲介関数呼び出しにより、通信路を介して携帯端末上のプログラムの関数を呼び出して実行させることを特徴とするプログラム実行サーバ。

【請求項 11】

前記携帯端末において実行されるプログラムは、その携帯端末のユーザインタフェースに関係するプログラムであることを特徴とする請求項 10 記載のプログラム実行サーバ。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、プログラム実行方法に係り、コンピュータプログラムの実行をサーバと携帯端末の双方でおこなえるようにしたプログラム実行方法に関する。

【背景技術】

【0002】

近年、携帯電話や P D A 等をはじめとする携帯端末の普及が目覚ましい。特に、携帯電話は、初期の通話以外の用途のほかに、様々な機能を持つ携帯機器として進化してきており、インタプリタの実行環境が搭載され、様々な処理がおこなえるようになってきている。例えば、ゲームや、予定管理等のアプリケーションをネットワーク経由で携帯電話にダウンロードして使用することが一般的におこなわれている。(以下の非特許文献 1 に記載)。

【0003】

【非特許文献 1】NTT DoCoMo 社, i アプリコンテンツ作成について, [Online], , [参照:平成 15 年 5 月 15 日], インターネット <<http://www.nttdocomo.co.jp/mc-user/i/java/index.html>>

【発明の開示】

【発明が解決しようとする課題】

【0004】

携帯端末がプログラムを実行できるようになり、ユーザの利便性は飛躍的に向上した。

【0005】

しかしながら、携帯端末はその特性上、プログラムを実行するためのリソース、例えば、C P U の処理性能、メモリ量、二次記憶装置容量、ネットワーク性能等の性能が低く、またその機能が制限されていることが多い。このため、規模の大きなプログラムが実行できない場合や、たとえ実行できても処理に時間がかかることが多い。また、無線ネットワークを前提したものが多いため、通信にかかる費用も一般に高く、ネットワークを多用す

10

20

30

40

50

る用途に利用しにくいというコスト的な問題も存在する。

【0006】

本発明は、上記問題点を解決するためになされもので、その目的は、既に広く流布している携帯端末向けのプログラムを修正することなく、これらの携帯端末上での実行の際に課せられる携帯端末の特性から生ずる様々な制限を取り除きつつ、従来と同様な機能を携帯端末で利用できるようにすることにある。

【課題を解決するための手段】

【0007】

本発明に係る計算機システムのプログラム実行方法は、第一のプログラム実行装置と第二のプログラム実行装置と記憶装置とからなる計算機システムのプログラム実行方法において、前記記憶装置から第一のプログラムと第二のプログラムとを読み出すステップと、前記記憶装置から前記第一のプログラムと前記第二のプログラムのそれぞれに記述されている関数呼び出しの置換情報を読み出すステップと、読み出された前記第一のプログラムと前記第二のプログラムのそれぞれに記述されている関数呼び出しを、それぞれ前記置換情報に基づき別の関数呼び出しに置換するステップと、前記関数呼び出しを置換された第一のプログラムと第二のプログラムとを、前記第一のプログラム実行装置と前記第二のプログラム実行装置に、それぞれ配布するステップと、前記第一のプログラム実行装置において前記第一のプログラムを実行し、前記第一のプログラムの置換された別の関数呼び出しにより、前記第二のプログラムの置換された関数を呼び出して、前記第二のプログラム実行装置において前記第二のプログラムを実行するステップとを有するようにしたものである。 10 20

【0008】

すなわち、本発明では、上記の課題を解決するために、携帯端末向けのプログラムを格納しているプログラム保存部、携帯端末向けのプログラムの全て、あるいは一部を実行可能なプログラム実行サーバ、および、携帯向けプログラムをプログラム実行サーバで実行可能に変換するプログラム変換部を備える。

【0009】

携帯端末でプログラムを実行する場合の従来手順は、プログラム格納部から目的のプログラムをネットワーク経由でダウンロードして、その携帯端末上で実行するものであった。 30

【0010】

一方、本方式では携帯端末からのプログラムロード要求に対して、プログラム保存部に格納されたプログラムをプログラム変換部に、いったん読み込む。次にプログラム変換部では、プログラムを解析し、プログラム実行サーバで実行するのが妥当と判断できる部分と、携帯端末側で実行するのが妥当と判断できる部分に分離する。

【0011】

プログラム実行サーバで実行するのが妥当な部分とは、例えば、単純な数値計算や、データベースにアクセスする部分、あるいは、ネットワークにアクセスする部分が考えられる。また携帯端末上で実行するのが妥当な部分としては、例えば、ユーザインターフェースに関係した部分などが考えられる。このうち携帯端末側で実行すべき部分については、プログラム中で当該機能を読み出している箇所を検出し、スタブ関数（仲介関数）の呼出しに変換し、プログラム実行サーバにロードする。携帯端末側には、プログラム実行サーバのスタブ関数からの要求を実行する関数の呼出し手続きを含むプログラムをロードする。 40

【0012】

このような構成にしてプログラムを実行するとサーバで実行すべき部分はサーバ上で高速実行される。スタブ関数呼び出しに置き換えられたUI等の処理は、これを呼び出すことでサーバ上のスタブ関数が呼び出され、そのスタブ関数からネットワーク経由の関数呼び出し、すなわち、遠隔関数呼び出し（以下、RPC、Remote Procedure Call）を利用して、携帯端末上の関数呼出し手続きを含むプログラムから、目的とする関数を呼び出し、携帯端末上で処理がおこなわれる。 50

【0013】

ユーザの端末操作等の処理についても、スタブ関数とRPCを利用して同様な方法で、携帯端末からスタブ関数を呼び出し、プログラム実行サーバ上のプログラムを実行することによりおこなうことができる。

【0014】

このような構成と動作によって、既存のプログラムを、その処理の大部分を高速なプログラム実行サーバ部で実行しながら実行し、UI等は従来通り携帯端末を使用することが可能することで、プログラムの利便性を変えることなく高速な処理が可能となる。

【発明の効果】

【0015】

本発明によれば、既に広く流布している携帯端末向けのプログラムを修正することなく、これらの携帯端末上での実行の際に課せられる携帯端末の特性から生ずる様々な制限を取り除きつつ、従来と同様な機能を携帯端末で利用できる。

【発明を実施するための最良の形態】

【0016】

以下、図1ないし図18を用いて本発明に係る一実施形態を説明する。

【0017】

〔関数呼出しの置換処理をおこなう計算機システム〕

先ず、図1ないし図3を用いて本発明の計算機システムの構成について説明する。

図1は、本発明の計算機システムの構成図である。

図2は、RPC通信部の構成図である。

図3は、オブジェクト管理表を示す図である。

【0018】

本発明の計算機システムは、プログラム実行サーバ10と、プログラム格納変換部20と、携帯端末30がネットワーク40を介して接続される構成である。携帯端末プログラム代替実行部11は、従来、携帯端末上で実行されていたプログラムをプログラム実行サーバ10で実行されるようにした機構である。このプログラム実行サーバ10は、標準的なコンピュータが備えるように、入力機器2、出力機器3、プログラム、ライブラリファイルからなる。プログラムは、外部記憶装置4等がバスブリッジ1で接続されたコンピュータにおけるメモリ上で実行される。また、ライブラリファイルは、外部記憶装置4に記録されている。

【0019】

携帯端末プログラム代替実行部11は、プログラム実行仮想機械12、スタブライブラリ14および実行ライブラリ13からなる。プログラム実行仮想機械12は、オブジェクト管理部121とRPC通信部122を持つ。

【0020】

オブジェクト管理部121は、システムで扱うオブジェクトを管理する部分であり、後に説明するオブジェクト管理表を有する。ここで、オブジェクトとは、関連するデータの集合と、それに対する手続き(メソッド)をまとまりとして管理したものであり、本実施形態のシステムは、いわゆる「オブジェクト指向」のシステムである。

【0021】

オブジェクト管理表は、データとプログラムの実行に伴い生成するオブジェクトに一意的なオブジェクトIDを振り、これを管理するためのテーブルである。テーブルは、オブジェクトID(1211)とオブジェクトが記録されているアドレス1212を対にして記録する。オブジェクトID(1211)からアドレス1212を検索したり、逆にアドレス1212からオブジェクトID(1211)を検索することが可能であり、高速化のためにハッシュ表をつかった実装も考えられる。

【0022】

なお、携帯端末30中のオブジェクト管理部321も同様なテーブルを有する。ここではアドレス1212に格納する情報としてオブジェクトのアドレスを使用した。プロ

10

20

30

40

50

ラム実行環境に存在するオブジェクトを一意に識別できる情報であれば、その他のものでもよい。

【0023】

R P C 通信部 1 2 2 の構成は、図 2 に示す如くであり、符号化部 1 2 2 1 , オブジェクト I D 取得部 1 2 2 2 、複合化部 1 2 2 3 、オブジェクト呼出部 1 2 2 4 、およびネットワーク通信部 1 2 2 5 からなる。

【0024】

R P C 通信部 1 2 2 は、呼出し続き情報を受信すると、オブジェクト I D 取得部 1 2 2 2 から、オブジェクト I D を得て、引数などと合わせて、符号化部 1 2 2 1 によって、ネットワーク 4 0 で交換可能な形態に変換し、ネットワーク通信部 1 2 2 5 から R P C として送信される。

10

【0025】

また、R P C として送信された情報は、別コンピュータ上のネットワーク通信部 1 2 2 5 で受信され、復号化部 1 2 2 3 で、元のオブジェクト I D と引数に復元されて、オブジェクト呼出部 1 2 2 4 に渡される。オブジェクト呼出部 1 2 2 4 は、オブジェクト I D を参照して要求されたオブジェクトに対し、呼び出して引数を渡す。

【0026】

なお、リモート関数の呼出は上記の方式に限定されるものではなく、既存のリモート関数呼び出し方式を利用して良く、引数と実行する関数名が指定でき、その結果を返せるものであればいずれの方式も利用可能である。

20

【0027】

プログラム格納変換部 2 0 は、この計算機システムで動作できるように、プログラムを変換する部分であり、プログラム変換部 2 1 と、携帯端末用プログラム格納部 2 2 を備える。プログラム変換部これも、プログラム実行サーバ 1 0 のように入出力機器がバスブリッジで接続されたコンピュータ上で動作する。

【0028】

携帯端末 3 0 は、プログラム実行サーバ 1 0 上のものと同じ携帯端末プログラム実行部 3 1 があり、これはプログラム実行仮想機械 3 2 、実行ライブラリ 3 3 、およびスタブライブラリ 3 4 からなる。プログラム実行仮想機械 3 2 は、さらにオブジェクト管理部 3 2 1 、および、R P C 通信部 3 2 2 を備える。R P C 通信部 3 2 2 の構成は、図 2 で示したものと同様である。

30

【0029】

携帯端末プログラム実行部 3 1 も、プログラム実行サーバ 1 0 のように入出力機器がバスブリッジで接続された携帯端末上で動作する。

【0030】

また、図 1 ではプログラム実行サーバ 1 0 とプログラム格納・変換部 2 0 を別のコンピュータ上で構成する実施形態を示したが、これらが同一のコンピュータ上で動作することも考えられる。

【0031】

プログラム実行サーバ 1 0 上の実行ライブラリ 1 3 やスタブライブラリ 1 4 、あるいは携帯端末 3 0 上の実行ライブラリ 3 3 やスタブライブラリ 3 4 はストレージ上に存在せず、ネットワークからダウンロードする形態も考えられる。

40

【0032】

次に、図 4 ないし図 6 を用いて本発明のプログラム実行方法の原理について説明する。

図 4 は、プログラムあるいはライブラリの構成の例を示した図である。

図 5 は、関数呼出手続置換設定データの構成を示した図である。

図 6 は、関数呼出手続の置換の例を示した図である。

【0033】

本システムのプログラム実行は、関数を呼び出す際に関数の呼出しを別の関数に変更して呼び出す処理を含んでいる。

50

【0034】

一般に、図4に示されるように、プログラムあるいはライブラリ5は、一つまたは複数の関数(51~53)と、関数情報54を持っている。関数情報は、プログラムモジュールに特有な情報を保持している。例えば、呼出規約、関数で利用する局所変数、モジュール名、関数に関連した情報のテーブルなどが考えられる。またオブジェクト指向言語ではクラス名なども考えられる。図4では関数情報をプログラムが内部に保有している構造を記したが、プログラム情報が外部に、例えば、別ファイルとして保持される形式も考えられる。この例としては、コンパイラから出力されるMAPファイルなどが考えられる。

【0035】

関数呼出手続置換設定データ6は、置き換えられる関数である置換対象関数61と、これを置き換える関数である置換関数62の組からなる。一つの置換関数62に対して複数の置換対象関数61が対応することも考えられる。置換対象関数61および置換関数62の保持する情報は、関数を指定する情報であり、関数名、関数を格納しているライブラリ名またはクラス名、関数へのアドレス、関数のIDなどが考えられる。

10

【0036】

また、関数呼出手続置換設定データ6は、適用するプログラム毎、あるいはプログラム実行環境毎に予め定められている場合や、プログラムをロードした際に、プログラムを解析して利用できるリソースや要求される性能などを考慮して作成する方法が考えられる。また、関数置換をおこなった後に実行した結果を考慮することも考えられる。

【0037】

さて、上記のプログラムあるいはライブラリと関数呼出手続置換設定6データがあるときに、本発明の関数呼出手続の置換は、以下のようにおこなわれる。

20

【0038】

ユーザプログラム610は、関数A(611)と関数情報614を持っているとする。関数A(611)からは、関数Bの呼出手続612と関数Cの呼出手続613があるものとする。そして、関数B(621)と関数C(622)はライブラリA(620)内に存在する。

【0039】

これに対して関数呼出手続置換設定データ6に、置換対象関数61として関数B(641)が、置換関数62として関数D(643)が登録されている。この場合、関数B呼出手続612はライブラリB(630)の関数D(631)への呼出手続に書き換える。以下同様にして関数C呼出613の呼出手続も書き換える。置換される前のプログラムがライブラリA(620)内の関数だけ呼び出していた場合は、実行にはライブラリA(620)だけが必要であったが、当該変換によってライブラリB(630)の関数も呼び出すようになると、実行時にはライブラリB(630)も必要となる。

30

【0040】

このような手法により、プログラムから呼び出す関数を置き換えることが可能となる。また、本説明ではプログラムの呼出手続を直接書き換えたが、関数の呼出手続が関数テーブルを参照しておこなうものであれば、このテーブルを書き換えることでも同様な関数呼び出しの置換が可能である。

40

【0041】

関数をソースレベルで置き換えたときには、インタープリタ環境ならそのコードを実行すればよい。コンパイラ環境のときには、コンパイルが必要になる。

【0042】

次に、図7ないし図11を用いて本発明のプログラム実行をおこなうための計算機システムの動作について説明する。

図7は、本発明の計算機システムで仮想計算機がプログラムを実行するまでの流れを示した図である。

図8は、プログラム実行までの各サイトにおける処理の概要を示したフローチャートで

50

ある。

図 9 は、関数呼び出しの置換処理を示したフローチャートである。

図 10 は、RPC 通信部 (1 2 2 , 3 2 2) がネットワークを使用して遠隔の関数を呼び出す手順を示したフローチャートである。

【 0 0 4 3 】

先ず、図 7 により本発明で関数呼び出しの置換処理した場合のプログラム実行の流れについて説明する。

【 0 0 4 4 】

本実施形態では、システム上で関数 と関数 が実行する場合であって、関数 がプログラム実行サーバ 1 0 で実行され、関数 が携帯端末上で実行されるものとする。

10

【 0 0 4 5 】

携帯端末プログラム代替実行部 1 1 のプログラム実行仮想機械 A (1 2) 上で動作しているオブジェクト 1 2 0 1 が関数 の呼び出し 1 2 0 2 を実行した場合には、実行ライブラリ 1 3 中の関数 (1 3 1) が実行される。

【 0 0 4 6 】

一方、本来の関数 (1 3 2) への呼び出しは、スタブライブラリ 1 4 中のスタブ関数 (1 4 1) への呼び出しに置換されている。このため本来の関数 (1 3 2) への呼び出しは、スタブ関数 (1 4 1) への呼び出しに変更される。スタブ関数 (1 4 1) は、RPC 通信部 1 2 2、ネットワーク 4 0 を経由して、オブジェクト 1 0 2 1 のオブジェクト ID (1 2 1 1) とスタブ関数 (1 4 1) へ渡された引数を渡す。

20

【 0 0 4 7 】

携帯端末 3 0 上のプログラム実行仮想機械 B (3 2) には、関数 呼出手続 3 2 0 2 を含んだプログラムがロードされる。そして、プログラム仮想機械 B (3 2) では、渡されたオブジェクト ID (3 2 1 1) から対象となるオブジェクト (3 2 0 1) を得い、その中の関数 への関数 呼出手続 3 2 0 2 を実行することで、実行ライブラリ 3 3 中の関数 (3 3 1) が実行される。

【 0 0 4 8 】

このような仕組みで、従来では一つのプログラムから呼出されていた機能を、二つの実行環境に任意に分割し、動作させることが可能となる。

【 0 0 4 9 】

次に、図 8 を用いてプログラム実行までの各サイトにおける処理の概要について説明する。

30

【 0 0 5 0 】

携帯端末 3 0 のイベント発生などのトリガーなどにより、プログラム格納・置換部 2 0 は、実行の対象となるプログラムを選択する (ステップ 9 0 1 0) 。

【 0 0 5 1 】

次に、読み込んだプログラムの関数呼び出しを、プログラム変換部 2 1 が置換し (ステップ 9 0 2 0) する。

【 0 0 5 2 】

そして、プログラム変換部 2 1 は、プログラム実行サーバ 1 0 向けには関数呼出手続を置換したプログラムを配布して、プログラム実行サーバ 1 0 に実行させる (ステップ 9 0 3 0) 。

40

【 0 0 5 3 】

また、プログラム変換部 2 1 は、携帯端末 3 0 向けにも関数呼出手続を置換したプログラムを配布して、実行させる (ステップ 9 0 4 0) 。

【 0 0 5 4 】

次に、図 9 を用いて関数呼び出しの置換処理について説明する。

【 0 0 5 5 】

最初に、プログラム変換部 1 0 は、関数置換データ 6 を読み込む (ステップ 9 1 1 0) 。次に、置換対象のプログラムを解析し (ステップ 9 1 2 0) 、設定データ 6 に指定され

50

た関数を呼び出している箇所があるか否かを判定して(ステップ9130)、呼び出している場合には設定データ6に基づき関数呼び出しを変更する(ステップ9140)。これを全てのプログラムに対しておこなう(ステップ9150)。

【0056】

次に、図10を用いてRPC通信部(122, 322)がネットワークを使用して遠隔の関数を呼び出す手順について説明する。

【0057】

RPCの処理では、最初に対象とするオブジェクトのID(1211)をオブジェクト管理部121から取得する(ステップ9210)。

【0058】

次に、関数呼び出しの引数、および、オブジェクトIDを通信可能な形式に符号化する(ステップ9220)。そして、この情報と共に携帯端末30のRPC通信部322にデータを送信する(ステップ9230)。

【0059】

携帯端末30上のRPC通信部322では、引数とオブジェクトIDを復号し(ステップ9240)、得たオブジェクトIDからオブジェクトを取得し(ステップ9250)、オブジェクトの関数を引数と共に呼び出す(ステップ9260)。次に関数が実行されることによって得られた戻り値を符号化し(ステップ9270)、プログラム実行サーバ10のRPC通信部122に送信する(ステップ9280)。RPC通信部12通信部122は戻り値を復号して(ステップ9290)、その戻り値をオブジェクトに設定して(ステップ9295)、処理を終了する。

【0060】

〔イベント処理とオブジェクト動作〕

次に、図11ないし図15を用いて本発明のプログラム実行方法におけるイベント処理と関連するオブジェクトの動作について説明する。

【0061】

本発明のプログラム実行方法は、プログラム実行サーバ10と携帯端末30で機能分担し、両方でプログラムを実行するものであった。そのため、プログラム実行サーバ10の携帯端末プログラム代替実行部11のプログラム仮想実行部12で関数呼び出しをスタブ関数呼び出しの手続きに置換して実行するものであった。本システムでのイベント処理も両方のサイトのオブジェクトが以下に説明するようにメッセージを送受信しながらおこなう。

【0062】

まず、図11により一般的なボタンなどからのイベントを処理するイベントリスナ方式を説明する。

図11は、一般的なボタンなどからのイベントを処理するイベントリスナ方式を説明する図である。

【0063】

イベントリスナ方式はイベントの授受に広く利用されている方法である。まず、イベントを発生するオブジェクト700はリスナテーブル710を持ち、このオブジェクトの発生するイベントを受け取るオブジェクトを登録する。この図ではオブジェクトA(720)と、オブジェクトB(730)が登録されている。例えば、ユーザがボタンを押下するなどしてイベントが発生すると、イベント発生オブジェクト700は、リスナテーブル710に登録されたオブジェクトのイベント処理関数を順番に、あるいは一斉に呼び出す。このような方法でユーザの操作をイベントとしてオブジェクトが受け取ることができる。このようなオブジェクトをリスナオブジェクトと呼ぶ。

【0064】

またこれとは別の方法として、イベントを処理する専用の関数を設け、その中で必要な処理をおこなうものも存在する。

【0065】

10

20

30

40

50

次に、図 1 2 により本発明の計算機システムのオブジェクトの実装とイベント発生時の処理について説明する。

【 0 0 6 6 】

図 1 2 は、本発明の計算機システムのオブジェクトの実装状態の一例を示す図である。

【 0 0 6 7 】

先ず、携帯端末 3 0 上には、ボタンオブジェクト 9 0 0 とリスナスタブオブジェクト 9 2 0 (仲介オブジェクト) が配置され、一方のプログラム実行サーバ 1 0 には、ボタンオブジェクト 9 4 0 とリスナオブジェクトが配置される

ボタンオブジェクト 9 0 0 は、携帯端末 3 0 上のボタンをオブジェクト指向により具現化したオブジェクトであり、ボタン押下などのイベントを発生させる。ボタンの表示変更をするときには、メソッドとして表示変更関数が呼び出される。内部には、上記で説明したイベントリスナテーブル 7 1 0 を有しており、リスナスタブオブジェクト A 9 2 0 が登録されている。リスナスタブオブジェクト A 9 2 0 は、ボタンオブジェクトのイベントを処理するためのオブジェクトであり、プログラム実行サーバ 1 0 上のリスナオブジェクト A 9 3 0 のスタブ (仲介) となるオブジェクトである。

10

【 0 0 6 8 】

携帯端末上でボタン押下などのイベントが発生したとする。ボタンオブジェクト 9 0 0 は、上記のイベント発生オブジェクト 7 0 0 に該当し、リスナテーブル 9 1 0 のリスナスタブオブジェクト A 9 2 0 のスタブイベント処理関数 A 9 2 1 を呼び出す。リスナスタブオブジェクト A 9 2 0 では、スタブイベント処理関数 A 9 2 1 から、ネットワーク経由で R P C によりプログラム実行サーバ 1 0 のリスナオブジェクト A 9 3 0 のイベント処理関数 A 9 3 1 を呼び出してイベント処理が実行される。

20

【 0 0 6 9 】

ボタンオブジェクト 9 0 0 の表示を変更する場合には、ボタンスタブオブジェクト 9 4 0 の表示変更スタブ関数 9 4 1 から、ネットワーク経由で R P C によりボタンオブジェクト 9 0 0 の表示変更関数 9 0 1 が呼び出される。

【 0 0 7 0 】

このように、プログラム実行サーバ 1 0 と携帯端末 3 0 の両者にオブジェクトとスタブオブジェクトを配置して処理の機能分担をおこなうことができる。

【 0 0 7 1 】

次に、図 1 3 により本発明の計算機システムのオブジェクトの生成の処理の一例について説明する。

30

図 1 3 は、プログラム実行サーバ 1 0 と携帯端末 3 0 とに、オブジェクトとそれに対応するスタブオブジェクトを生成する処理を示すフローチャートである。

【 0 0 7 2 】

従来のオブジェクト生成関数は、一つのオブジェクトを生成する関数であったとすると、本発明のプログラム実行方法では、一つのサイトにオブジェクトを生成し、他のサイトにはそれに対応するスタブオブジェクトを生成するようにプログラム変換部 2 1 が変換する。

【 0 0 7 3 】

図 1 3 に示される例は、携帯端末 3 0 上にオブジェクトを生成し、プログラム実行サーバ 1 0 にそれに対応するスタブオブジェクトを生成するものである。したがって、プログラム変換部 2 1 は、プログラム上の通常のオブジェクト生成関数の呼び出しを、スタブオブジェクトの生成関数の呼び出しに置換する。

40

【 0 0 7 4 】

このため、プログラム実行サーバ 1 0 が、スタブオブジェクトの生成関数の呼び出しをおこなうと (ステップ 9 4 1 0) 、 R P C による携帯端末 3 0 への通信が発生させ (ステップ 9 4 2 0) 、携帯端末 3 0 上にオブジェクトを生成するトリガーをかける。

【 0 0 7 5 】

携帯端末 3 0 では、この呼び出しを受けて、プログラム実行サーバ 1 0 側から指定され

50

たオブジェクトを生成し（ステップ 9 4 3 0）、初期化し（ステップ 9 4 4 0）、一意なオブジェクト ID を決定し（ステップ 9 4 4 5）、オブジェクト管理部 3 2 1 に登録する（ステップ 9 4 5 0）。

【 0 0 7 6 】

そして、この生成結果とオブジェクト ID を R P C によりプログラム実行サーバ 1 0 に返し（ステップ 9 4 6 0）、オブジェクトの生成に成功しているならば、プログラム実行サーバ 1 0 上でスタブオブジェクトを生成し（ステップ 9 4 7 0）、生成されたスタブオブジェクトのオブジェクト ID をオブジェクト管理部 1 2 1 に登録し、そのオブジェクト ID を戻り値として返して、オブジェクト生成関数の呼び出しが終了する。

【 0 0 7 7 】

ここで、スタブオブジェクトは、プログラム変換部 2 1 により元の関数のスタブとして置かれる関数から呼び出されることを意図している。図 1 3 の例で言えば、プログラム実行サーバ 1 0 には、ボタンスタブオブジェクト 9 4 0 が置かれ、表示変更スタブ関数 9 4 1 をメソッドとして有している。携帯端末 3 0 上には、その本体として、ボタンオブジェクト 9 0 0 と、表示変更関数 9 0 1 がある。このように、異なったスタブオブジェクトとオブジェクトが両方のサイトに置かれる構成は、本発明のプログラム実行方法の関数とスタブ関数により機能の実行を分散させることをサポートする形態であると言える。

【 0 0 7 8 】

次に、図 1 4 により携帯端末 3 0 上でおこなわれたイベントを処理するための登録処理について説明する。

図 1 4 は、プログラム実行サーバ 1 0 にリスナオブジェクトを、携帯端末 3 0 上にリスナスタブオブジェクトを登録するための処理を示すフローチャートである。

【 0 0 7 9 】

ここでは、図 1 2 のモデルを例にとり説明する。

【 0 0 8 0 】

通常のリスナオブジェクトの登録関数の呼び出しは、プログラム変換部 2 1 により、予めリスナオブジェクトの登録関数のスタブ関数の呼び出しに置換される。

【 0 0 8 1 】

通常のリスナオブジェクトの登録関数は、オブジェクトをリスナテーブルに登録する関数である。本発明のプログラム実行方法では、リスナオブジェクトの登録関数の呼び出しを、スタブ関数の呼び出しに置換しておく。

【 0 0 8 2 】

プログラム実行サーバ 1 0 から置換されたスタブ関数が呼び出されると、まず、リスナオブジェクト A 9 3 0 を生成する（ステップ 9 5 1 0）。

【 0 0 8 3 】

次に、R P C 呼出 9 5 2 0 により、携帯端末 3 0 上にリスナスタブオブジェクト A 9 2 0 を生成する（ステップ 9 5 3 0）。

【 0 0 8 4 】

そして、リスナスタブオブジェクト A 9 2 0 を、ボタンオブジェクト 9 0 0 のリスナテーブル 9 1 0 に登録する（ステップ 9 5 4 0）。

【 0 0 8 5 】

次に、図 1 5 により携帯端末 3 0 上でおこなわれたイベントが発生したときの動作について説明する。

図 1 5 は、携帯端末 3 0 上でおこなわれたイベントが発生したときの動作について説明するフローチャートである。

【 0 0 8 6 】

これは、イベントを処理するための登録処理が完了した後にのプログラム実行サーバ 1 0 と携帯端末 3 0 上でおこなわれる動作である。

【 0 0 8 7 】

10

20

30

40

50

ここでも、図13を例にとり説明する。すなわち、携帯端末30上で画面上に表示されているボタンを操作することによりイベントが発生した場合の処理である。

【0088】

携帯端末30上で画面上に表示されているボタンが操作されると、イベントが発生し、ボタンオブジェクト900が、イベントリスナ方式によりリスナテーブル910を参照し、リスナスタブオブジェクトA920を呼び出す(ステップ9610)。そして、RPC呼出9620により、プログラム実行サーバ10上のリスナオブジェクトA930が呼び出され(ステップ9630)、イベント処理関数931が実行される。

【0089】

このようにすることで、携帯端末側で生じたイベントをプログラム実行サーバ側で処理することが可能となる。 10

【0090】

〔実行環境・性能を考慮した関数呼出しの置換処理の切り分け〕

次に、図16を用いて本発明のプログラム実行方法で実行環境・性能を考慮した関数呼出しの置換処理の切り分けについて説明する。

図16は、発明のプログラム実行方法で実行環境・性能を考慮した関数呼出しの置換処理の切り分けを説明する図である。

【0091】

図16の計算機システムには、プログラム実行サーバ10と携帯端末30に、それぞれ実行性能管理部15および35が追加されている。 20

【0092】

この実行性能管理部15, 35はそれぞれの実行環境情報、要求性能などの情報、例えば、CPU能力、メモリ容量、利用できるハードウェア等のプログラム実行に影響する情報が記録されている。

【0093】

プログラム格納・変換部20には、プログラム変換切分部23が追加されている。プログラム変換切分部23は、実行性能管理部15, 35の情報を利用して、プログラムの関数の置換処理を決定する。

【0094】

例えば、実行するプログラムに対して携帯端末30のメモリが極端に少ない場合は、ほとんど全ての処理をプログラム実行サーバ10で実行するようにする。メモリに余裕がある場合は、関数呼出しの置換処理をおこなって、サーバ側のスタブ関数から携帯端末30のプログラムを実行するようにして、携帯端末30側での処理を増やすなどが考えられる。このように決定した方針によってプログラム変換をプログラム変換部21でおこない、変換したプログラムをプログラム実行サーバ10と、携帯端末30に配布する。 30

【0095】

〔他の計算機システム構成〕

次に、図17および図18を用いて本発明のプログラム実行方法を適用する場合の他の計算機システム構成について説明する。

図17および図18は、本発明のプログラム実行方法を適用する場合の他の計算機システム構成について説明する図である。 40

【0096】

図1に示した実施形態は、プログラム実行サーバ10と携帯端末30という二つの異なった種別のプログラム実行装置に適用したものであった。これ以外にも同等の方式で同種のプログラム実行装置を二台利用することも同様に可能である。また二台のプログラム実行装置による処理の分担だけではなく、二台以上のプログラム実行装置に対して適用することも可能である。

【0097】

図17は、その一例を示したものであり、複数のコンピュータにより、振り分けテーブルを利用してスタブ関数からの呼び出しプログラムが実行される場合である。 50

【 0 0 9 8 】

この図 1 7 の構成では、スタブ関数呼び出し 8 1 0 が呼び出されることにより、スタブライブラリ 8 1 1 中のスタブ関数 8 1 2 が呼び出される。そして、それぞれの関数と処理をおこなうコンピュータの振り分けをするための情報である振り分けテーブル 8 1 4 を参照して、このスタブ関数は R P C 通信部 8 1 3 を通じてリモートのコンピュータの関数を呼び出してプログラムを実行する。

【 0 0 9 9 】

図 1 8 は、他の一例を示したものであり、リモートコンピュータの関数を呼び出すスタブライブラリをコンピュータ毎に準備する方法である。

【 0 1 0 0 】

この図 1 8 の構成では、プログラム実行前に関数呼出をスタブ関数呼び出しに置換する際に、どのコンピュータの関数を呼び出すかによってそれぞれ別のスタブライブラリ上のスタブ関数の呼び出しに変更する。このようにすることで関数呼び出しの置換をおこなった後は、特別な処理をすることなく適切なリモートコンピュータ上の関数呼び出しをおこなうことができる。

【 図面の簡単な説明 】

【 0 1 0 1 】

【 図 1 】 本発明の計算機システムの構成図である。

【 図 2 】 R P C 通信部の構成図である。

【 図 3 】 オブジェクト管理表を示す図である。

【 図 4 】 プログラムあるいはライブラリの構成の例を示した図である。

【 図 5 】 関数呼出手続置換設定データの構成を示した図である。

【 図 6 】 関数呼出手続の置換の例を示した図である。

【 図 7 】 本発明の計算機システムで仮想計算機がプログラムを実行するまでの流れを示した図である。

【 図 8 】 プログラム実行までの各サイトにおける処理の概要を示したフローチャートである。

【 図 9 】 関数呼び出しの置換処理を示したフローチャートである。

【 図 1 0 】 R P C 通信部 (1 2 2 , 3 2 2) がネットワークを使用して遠隔の関数を呼び出す手順を示したフローチャートである。

【 図 1 1 】 一般的なボタンなどからのイベントを処理するイベントリスナ方式を説明する図である。

【 図 1 2 】 本発明の計算機システムのオブジェクトの実装状態の一例を示す図である。

【 図 1 3 】 プログラム実行サーバ 1 0 と携帯端末 3 0 とに、オブジェクトとそれに対応するスタブオブジェクトを生成する処理を示すフローチャートである。

【 図 1 4 】 プログラム実行サーバ 1 0 にリスナオブジェクトを、携帯端末 3 0 上にリスナスタブオブジェクトを登録するための処理を示すフローチャートである。

【 図 1 5 】 携帯端末 3 0 上でおこなわれたイベントが発生したときの動作について説明するフローチャートである。

【 図 1 6 】 発明のプログラム実行方法で実行環境・性能を考慮した関数呼び出しの置換処理の切り分けを説明する図である。

【 図 1 7 】 本発明のプログラム実行方法を適用する場合の他の計算機システム構成について説明する図である (その一) 。

【 図 1 8 】 本発明のプログラム実行方法を適用する場合の他の計算機システム構成について説明する図である (その二) 。

【 符号の説明 】

【 0 1 0 2 】

1 0 ... プログラム実行サーバ部, 1 1 ... 携帯端末プログラム仮想実行部, 1 2 ... プログラム実行仮想機械, 1 3 ... 実行ライブラリ, 1 4 ... スタブライブラリ, 1 2 1 ... オブジェクト管理部, 1 2 2 ... R P C 通信部, 2 0 ... プログラム格納・変換部, 3 0 ... 携帯端末,

10

20

30

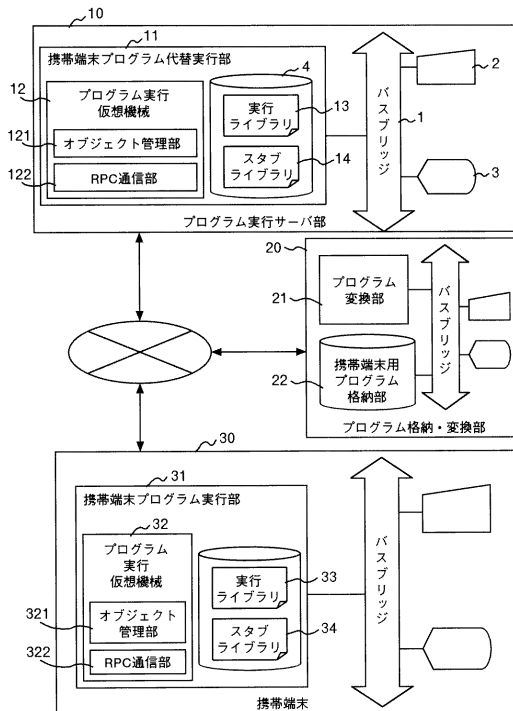
40

50

3 1 ... 携帯端末プログラム実行部, 3 2 ... プログラム実行仮想機械, 3 3 ... 実行ライブラリ、3 4 ... スタブライブラリ, 4 0 ... ネットワーク。

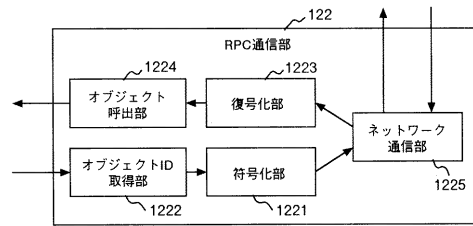
【 図 1 】

図 1



【 図 2 】

図 2



【 図 3 】

図 3

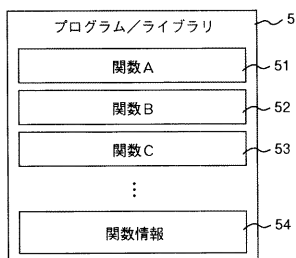
オブジェクト管理表

オブジェクトID	アドレス
1	0x000f0000
2	0x001f0000
⋮	
n	0x00ff0000

1211 1212

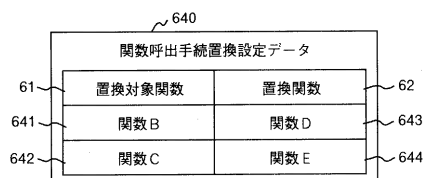
【 図 4 】

図 4



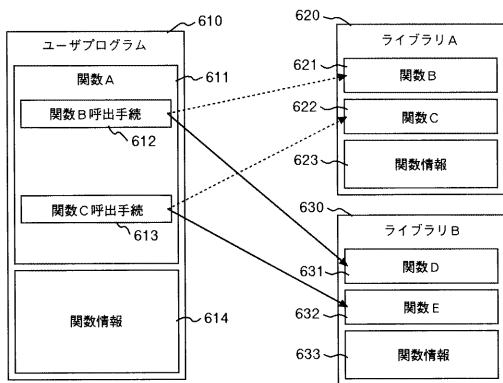
【 図 6 】

図 6



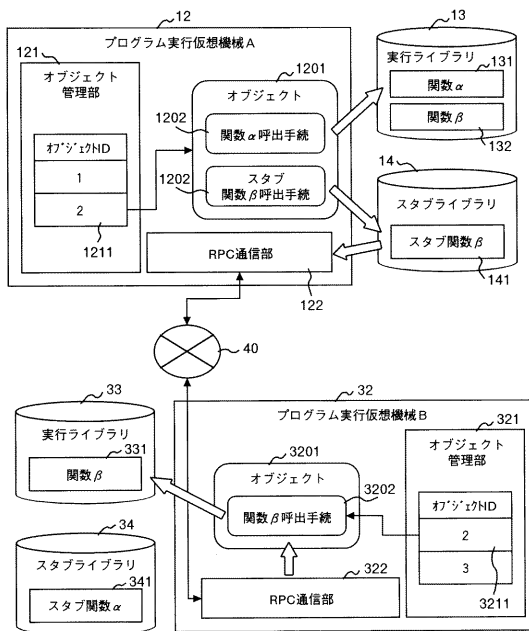
【 図 5 】

図 5



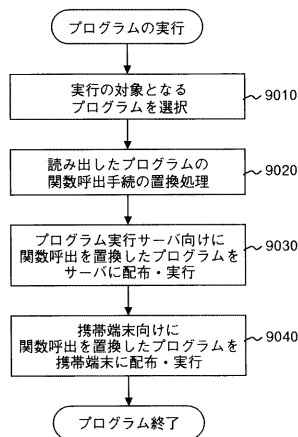
【 図 7 】

図 7

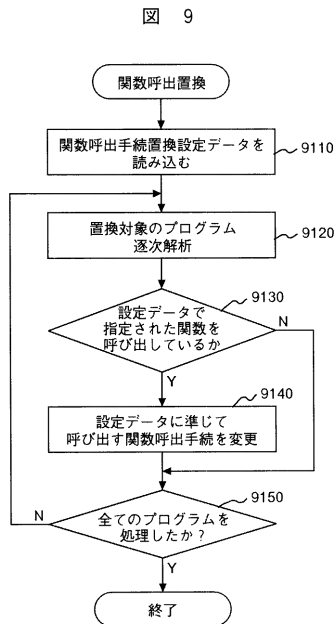


【 図 8 】

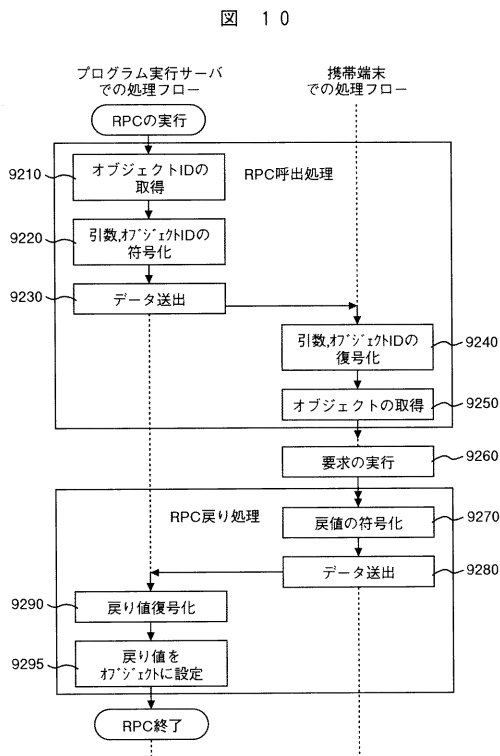
図 8



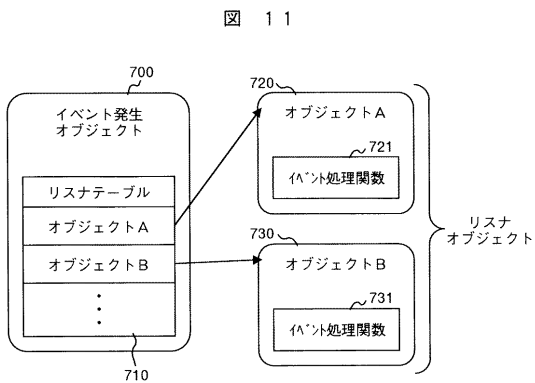
【 図 9 】



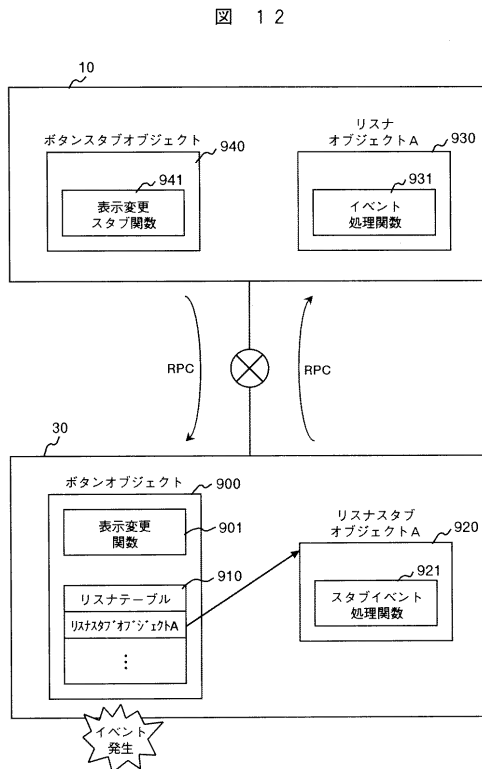
【 図 10 】



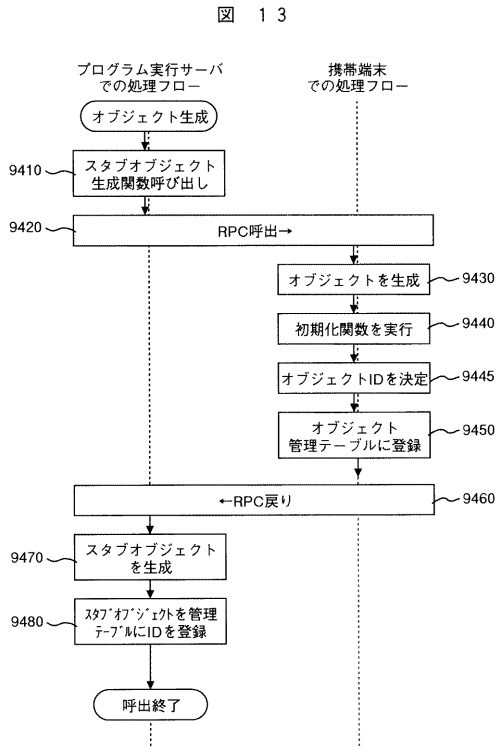
【 図 11 】



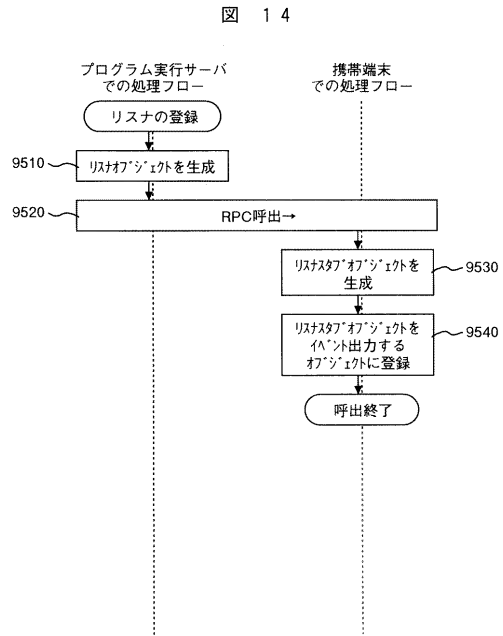
【 図 12 】



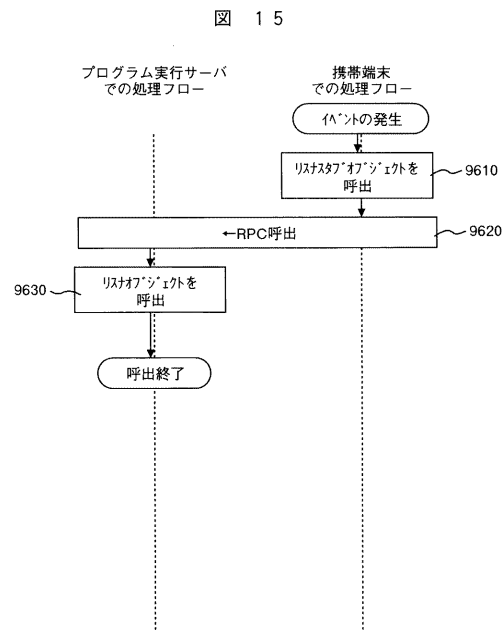
【 図 1 3 】



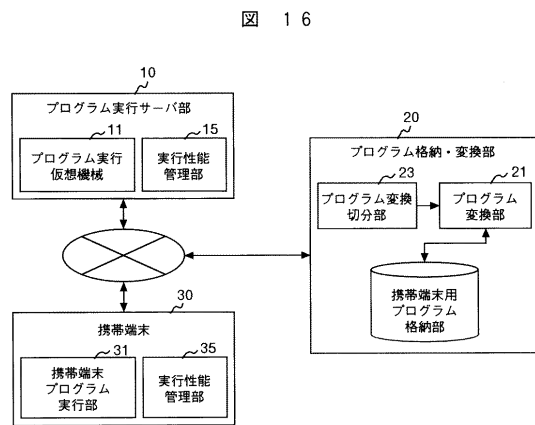
【 図 1 4 】



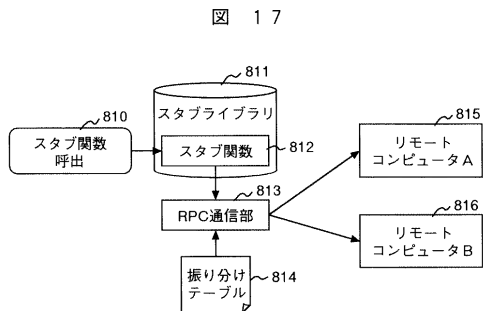
【 図 1 5 】



【 図 1 6 】

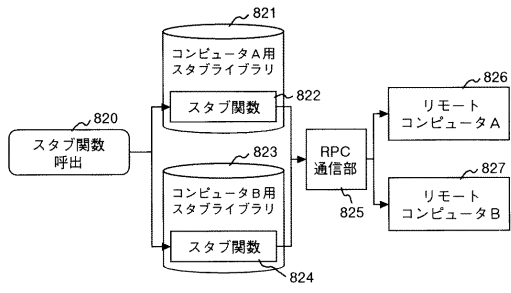


【 図 1 7 】



【 図 1 8 】

図 1 8



フロントページの続き

(72)発明者 里山 元章

神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所システム開発研究所内

Fターム(参考) 5B033 AA11 AA14 AA15 AA19 AA20 EA01

5B076 AB04 AB10

5B098 AA10 GA06

5K101 KK15 LL12 MM07 NN21 RR12