

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4597464号
(P4597464)

(45) 発行日 平成22年12月15日(2010.12.15)

(24) 登録日 平成22年10月1日(2010.10.1)

(51) Int.Cl. F I
H O 4 L 29/06 (2006.01) H O 4 L 13/00 3 O 5 A

請求項の数 11 (全 13 頁)

(21) 出願番号	特願2001-576591 (P2001-576591)	(73) 特許権者	597149146
(86) (22) 出願日	平成13年3月17日(2001.3.17)		ドイッチェ テレコム アーゲー
(65) 公表番号	特表2004-518310 (P2004-518310A)		ドイツ国. 5 3 1 1 3 ボン, フリードリ
(43) 公表日	平成16年6月17日(2004.6.17)		ヒ - エベルト - アレエ 1 4 O
(86) 国際出願番号	PCT/EP2001/003076	(74) 代理人	100064447
(87) 国際公開番号	W02001/079991		弁理士 岡部 正夫
(87) 国際公開日	平成13年10月25日(2001.10.25)	(74) 代理人	100085176
審査請求日	平成20年3月13日(2008.3.13)		弁理士 加藤 伸晃
(31) 優先権主張番号	100 18 288.7	(74) 代理人	100106703
(32) 優先日	平成12年4月13日(2000.4.13)		弁理士 産形 和央
(33) 優先権主張国	ドイツ(DE)	(74) 代理人	100096943
			弁理士 臼井 伸一
		(74) 代理人	100091889
			弁理士 藤野 育男

最終頁に続く

(54) 【発明の名称】 送信されたプロトコル・データ単位を解析する方法

(57) 【特許請求の範囲】

【請求項 1】

2つのオブジェクト・リクエスト・ブローカ(4)間で通信のためのプロトコルの送信済みプロトコル・データ単位(PDU)を解析して、含まれるメッセージの可能な限り完全な表現を得る方法であって、分散CORBAアプリケーション(14)の2つのコンポーネント(1、1')が互いに通信し、一方のコンポーネント(1、1')がクライアント役として動作し、他方がサーバ役として動作するものであり、そして各コンポーネントがどちらも異なるホスト上にインストールされており、

個々のプロトコル・データ単位(PDU)から導出される不完全な情報の集合を、単一オブジェクトごとに個別に収集して関連させ、

復号中に、基礎とするインターフェース定義用の十分な選択基準を複数の必要条件から得、そして、

復号を3つの段階で実施するものであり、

第1の段階が、すべての可能なインターフェース定義のリストが作成される初期化段階であり、

第2の段階が、可能性の数を減らすための排他基準が導入される学習段階であり、どのインターフェース定義がオブジェクト下にあるかを一意的に決定し、そして、

復号段階である第3の段階で、オブジェクトと送受信するメッセージを復号することを特徴とする方法。

【請求項 2】

10

20

請求項 1 に記載の方法において、

最初に、交換する情報の集合を収集し、それをロギングすることによって排他的受動的に分析し、

交換した情報の集合 (P D U) を、分散アプリケーション (1 4) で使用される I D L 定義に基づいて復号し、

既存のあいまいさの一部を、復号したメソッド名の一意性またはその固有パラメータ表示によって直ちに除外し、

単一プロトコル・データ単位 (P D U) で復号された情報を使用して除外することができないあいまいさを、一般 O R B 間プロトコル (G I O P P D U) の複数のプロトコル・データ単位を介して反復的系統的に解決することの特徴とする方法。

10

【請求項 3】

請求項 1 に記載の方法において、

最初に、すべての記録したプロトコル・データ単位 (P D U) を用いて段階 2 を実施し、

その後で、これらプロトコル・データ単位 (P D U) のすべてを、段階 3 で再び復号することを特徴とする方法。

【請求項 4】

請求項 1 乃至 3 のいずれかに記載の方法において、

使用するインターフェース定義を確立するために、特定のエージェントについての予備段階で、初期可能性の制限を構成し、前記特定のエージェントが、インターネット O R B 間プロトコル (8) では、 I P アドレスおよび T C P ポート (1 3) によって定義されることを特徴とする方法。

20

【請求項 5】

請求項 1 乃至 3 のいずれかに記載の方法において、

どのインターフェース定義が使用可能であるかに関する初期可能性を制限するために、特定のホストについての予備段階で構成し、前記特定のホストが、インターネット O R B 間プロトコル (8) では、 I P アドレスによって定義されることを特徴とする方法。

【請求項 6】

請求項 1 乃至 5 のいずれかに記載の方法において、

情報の不足により復号することができないプロトコル・データ単位 (P D U) をキュー内に置き、

30

後で、この情報を他の復号済みプロトコル・データ単位 (P D U) に基づいて得たときに、これらのプロトコル・データ単位を復号することを特徴とする方法。

【請求項 7】

請求項 1 乃至 6 のいずれかに記載の方法において、

ブロック化リストに基づいて、含まれるメソッド名を介して定義することができるある要求がインターフェース定義を除外する目的でアクティブとならないことを特徴とする方法。

【請求項 8】

請求項 1 乃至 7 のいずれかに記載の方法において、

40

G I O P デコーダ (3) の任意選択設定のために、プロトコル・データ単位 (P D U) の不規則な復号により、必ずしも下にあるインターフェース定義が除外されず、

正しく復号したメソッドが、時折コーディング・エラーとなる場合であっても依然として復号可能であることを特徴とする方法。

【請求項 9】

請求項 1 乃至 8 のいずれかに記載の方法において、

オブジェクトの下にあるインターフェース定義に関する情報を供給する C O R B A 標準で定義されるメソッドの内容を、アドレス指定したオブジェクトに対する可能なインターフェース定義の数を制限するために評価されることを特徴とする方法。

【請求項 10】

50

請求項 1 乃至 9 のいずれかに記載の方法において、

一般 O R B 間プロトコル (G I O P) のログインしたプロトコル・データ単位 (P D U) が、インターネット O R B 間プロトコル (I I O P) を介して送られることを特徴とする方法。

【請求項 1 1】

請求項 1 乃至 1 0 のいずれかに記載の方法において、

一般 O R B 間プロトコル (G I O P) の復号メッセージによって得られる情報が、アクティブ・サーバを介して使用され、非システム・クライアント・アプリケーションを使用してこれらのサーバへの接続が確立され、これらのサーバに対する要求、例えば C O R B A 要求が送信されることを特徴とする方法。

【発明の詳細な説明】

【 0 0 0 1】

本発明は、請求項 1 の種概念の定義に従って送信されたプロトコル・データ単位 (Protocol Data Unit : P D U) を解析する方法に関する。

【 0 0 0 2】

含まれるメッセージを可能な限り完全に表すために、オブジェクト・リクエスト・ブローカ (Object Request Brokers : O R B) 間で通信するための一般プロトコルの送信されたプロトコル・データ単位を解析する方法が一般に知られている。

【 0 0 0 3】

C O R B A は、Uwe Seimet による「Wissen」第 1 0 8 頁乃至第 1 1 3 頁の論文「C O R B A , ubernehmen Sie」、Services : Basisdienste fur den ORB」に記載されている。C O R B A はまた、「NOMS 2000, 2000 IEEE/IFIP, Network and Management Symposium」「The Networked Planet: Management Beyond 2000」のフレームワークにおける Guido Carls、Gerd Aschemann、Ying Lu の「Validating Interaction Patterns of CORBA Based Network Management Systems」(2000 年 4 月 10 日乃至 14 日、Hilton Hawaiian village、Honolulu、Hawaii) の文献にも記載されている。C O R B A は、いくつかのオペレーティング・システム・プロセスにわたってアプリケーションを分散するために使用される。C O R B A により、分散アプリケーションの非独占ホスト間通信 (non-proprietary, interhost communication) が可能となる。この状況では、クライアントの役割とサーバの役割とが区別される。クライアント役のアプリケーションは、サーバ役で動作する別のアプリケーションに要求を送信する。この要求は、サーバ役のアプリケーションで処理され、対応する応答がクライアントに返される。2 つのオブジェクト・リクエスト・ブローカ間の通信用の一般プロトコルは、これらのメッセージを搬送するためのプロトコルとして使用される。T C P / I P ベースのネットワークでは、その特殊な形式、いわゆる「インターネット O R B 間プロトコル (Internet Inter-ORB Protocol : I I O P) 」が使用される。どの内容をクライアントとサーバとの間で交換することができるかについての情報が、オブジェクト指向インターフェース記述言語 (I D L) を使用して確立される。

【 0 0 0 4】

分散アプリケーションが予期せぬ挙動をした場合、または分散アプリケーションの受諾テスト中、個々のアプリケーション・コンポーネントの挙動に関する結論を得ることができるよう、クライアントとサーバとの間の通信を監視することが必要となることがある。例えば、上記で引用した参考文献では、以下の手順を記載している。

【 0 0 0 5】

分散アプリケーションをクライアント側またはサーバ側で拡張し、それによって交換する G I O P メッセージも完全に、または部分的に記録用および評価用の分散アプリケーションのコンポーネントに経路指定するようにできる。

クライアントとサーバとの間で交換される P D U をトレース・ツールによって記録することができ、かつ交換するメッセージのタイプと、使用する製品とに関する追加の知識を使用して復号することができる。メッセージの復号は、手動または自動で行うことができる。

10

20

30

40

50

テスト目的専用に開発されたアプリケーションを、クライアントとサーバとの間の通信に組み込む。このアプリケーションは、サーバとしてクライアントからの要求を受け取り、クライアントとしてこの要求をサーバに経路指定する。応答がそれに応じて返される。この結果、このテスト・アプリケーションによって、すべてのPDUを監視し、評価することができる。

【0006】

これらの周知の方法は、クライアントとサーバとの間で交換される一般ORB間プロトコル（General Inter ORB Protocol：GIOP）PDUの受動記録だけでは、メッセージを完全に復号するのに十分ではないという欠点を有する。これはいくつかの理由のために生じるが、それを以下で示す。

10

【0007】

規則として、値、例えば123だけしかGIOP PDUで符号化されず、そのタイプ・コード、例えば整数は符号化されない。したがって、PDUだけでは、PDU内に含まれるパラメータの数、または値のタイプを決定することができない。

起動されるメソッドの名前は、呼出しではそのフル・コンテキスト内に符号化されない。このメソッドが宣言されるインターフェース定義の固有の識別が欠けている。同一のメソッドを異なるシグニチャで異なるインターフェース定義内に定義すべき場合、同一の値符号化長が与えられると、実際に符号化されるメソッドを識別することができない。

サーバ側のエージェントが、自動生成し、発行した「ターゲット・アドレス」を介して着信要求を特定のCORBAオブジェクトと関連付けることができる間、CORBA標準によるトレース・ツールは実際に、索引に従ってターゲット・アドレスの内容が同等かをチェックすることができるが、具象オブジェクト、およびオブジェクトの下にあるインターフェース定義を関連付けることはできない。CORBA標準のプロプラエタリ拡張を用いた場合にのみ、トレース・ツールを使用してターゲット・アドレスの内容から情報を得ることが可能である。

20

CORBAでは、サーバからクライアントに転送されるオブジェクト参照が提供され、例えばディスク上のテキスト・ファイルのようなGIOPプロトコルが回避される。IIOPプロトコルに基づいて一意的にCORBAオブジェクトをアドレス指定する働きをする相互運用オブジェクト参照（IOR）として交換されるこのオブジェクト参照が、関連するインターフェース定義に関する情報も含むとき、この情報の項目は、一般に決して再度クライアントとサーバとの間で交換されない。したがって、通信に関するアプリケーション・コンポーネントに能動的に照会せずに、CORBAトレース・ツールがこの情報を得る方法はない。しかし、これは、トレース・ツールは受動的に監視することしか可能ではないという枠組みと相容れない。

30

【0008】

したがって、クライアントとサーバとに周知の状況で単一のGIOP PDUを一意的に相関させ、復号することは確かに可能であるが、記録されたPDUをこの周知の状況の外部で固有成準に準拠して復号する可能性はないという問題が存在する。

【0009】

したがって、本発明の目的は、CORBAクライアントとCORBAサーバとの間の通信を制御し、またはそれを可能にする2つのオブジェクト・リクエスト・ブローカ間で通信するため、すなわち含まれるメッセージを可能な限り完全に表すための一般的プロトコルの送信されたプロトコル・データ単位を解析する方法であって、交換するGIOP PDUを十分な期間受動的に監視するだけで、送信されたメッセージをクライアントおよびサーバの実装を変更することなく完全に復号することを可能にするように意図した方法を提供することである。

40

【0010】

本発明の目的を達成する手段は、請求項1に記載されている。

【0011】

本発明の別の設計手法または実施例は、請求項2から11に記載されている。

50

【 0 0 1 2 】

本発明による方法により、交換する G I O P P D U を十分な期間の間、受動的に監視するだけで、クライアントおよびサーバの実装を変更することなく送信されたメッセージを完全に復号することが可能となるので有利である。

これは、一種の学習段階で、交換したメッセージを系統的に評価し、相関させ、その結果個々のメッセージの復号中に発生する可能性のあるあいまいさを除外することによって達成される。

【 0 0 1 3 】

このようにして、記録されたプロトコル・データ単位をこの周知の状況の外部で復号することを可能にする、固有かつ標準準拠の解決策が生成される。

10

【 0 0 1 4 】

含まれるメッセージを可能な限り完全に表す目的で送信された G I O P P D U を解析するためのこの方法の別の利点、特徴、および可能な使用法を、図面に示す例示的实施例と共に以下で説明する。

【 0 0 1 5 】

以下では、図面に示す例示的实施例に照らして、本発明をより詳細に説明する。最後に示した参照符号のリストで使用される用語および関連する参照符号を、特許請求の範囲、要約、および図面中で使用する。

【 0 0 1 6 】

以下は、この後で使用する用語の簡単な説明または定義である。

20

C O R B A : 共通オブジェクト・リクエスト・ブローカ・アーキテクチャ。各ホストにアプリケーションを分散するためのフレームワークの定義。

G I O P : 一般 O R B 間プロトコル。2つの O R B 間通信用一般プロトコル。

O R B : オブジェクト・リクエスト・ブローカ。C O R B A クライアントと C O R B A サーバとの間の通信を許可する。

I I O P : インターネット O R B 間プロトコル。T C P / I P ベースのネットワーク用の G I O P のアプリケーション。

I I O P P D U : インターネット O R B 間プロトコル・データ単位。

P D U : プロトコル・データ単位。送信機と受信機との間で交換されるバイナリ・コード・メッセージ。

30

メッセージ : P D U として送られる情報の項目。符号化方式ではなく、情報の意味が先頭にある。

T C P / I P : 伝送制御プロトコル / インターネット・プロトコル (一般に使用されるインターネット・プロトコル) 。

U M L : 統合モデル化言語。

ホスト : 分散アプリケーションの一部が動作し、固有アドレスによってアクセスすることができるコンピュータ。

ポート : ホスト内の複数の異なるエージェントに対する接続を確立することが可能なアドレスの論理サブディビジョン。

エージェント : G I O P メッセージを送受信することを担当するエンティティ。

40

オブジェクト・キー : エージェントにより供給され、I O R を生成するために使用される特定の値。

I O R : 相互運用オブジェクト参照。I I O P プロトコルに基づいて C O R B A オブジェクトを一意的にアドレス指定するために使用される。

ターゲット・アドレス : G I O P リリース 1 . 2 現在、オブジェクトに関する複数の異なるアドレス指定モードを記述する用語「ターゲット・アドレス」が導入されている。基本機能に関しては、以前は排他的に使用されていた「ターゲット・アドレス」とオブジェクト・キーは、ほぼ同一と見なすことができる。

タイプ・コード : O R B 内のデータ構造の記述。

クライアント : プロセスが分散アプリケーション内で請け負うことができる役割。クライ

50

アントは、サーバに対する接続をオープンするエージェントである。

サーバ：プロセスが分散アプリケーション内で請け負うことができる役割。サーバは、クライアントがそれに対する接続を確立するエージェントである。

アドレス：例示的な実施例でのホストの固有識別。

分散C O R B Aアプリケーション：異なるホスト上に分散し、I D L内で定義されたインターフェースを介してG I O Pメッセージを交換する複数のプロセスから構成される。

I D L：オブジェクト指向インターフェース記述言語。

C D R：共通データ表現。

要求：クライアントからサーバにメソッド呼出しを送る特定のG I O Pメッセージ。

応答：先行する要求に応答して、サーバからクライアントに送られる特定のG I O Pメッセージ。

プロセス：ホスト上のプログラムの記憶エンティティ。

インターフェース：メソッド呼出しの形でサーバ内のエージェントを介してそのエンティティにアクセスすることができるC O R B Aオブジェクトの、I D L内に定義される記述。

デコーダ：P D U中に符号化される情報を、含まれるメッセージの可読表現に変換するシステム。この例示的な実施例の状況では、これは、本発明による方法を使用するシステムである。

シグニチャ：プログラミング言語によって作り出された用語。同名のメソッドは、パラメータが異なるために相異なる可能性がある。この識別基準がシグニチャと呼ばれる。I D Lは、同じコンテキスト内の同名のメソッドの定義を除外し、それによってメソッドのシグニチャも符号化する必要がなくなる。

I F R：インターフェース・リポジトリ。後にアクセスする目的で、I D L定義を介して情報を利用可能にするために、C O R B A標準で定義されるオプション。

【 0 0 1 7 】

図1に、本発明による方法の基本的実施例、具体的には、その実用的な応用例の基本的実施例を示す。分散C O R B Aアプリケーション14の2つのコンポーネント1および1'が通信し、一方のコンポーネント1または1'がクライアント役として動作し、他方がサーバ役として動作する。この2つのコンポーネント1および1'は、異なるホスト・システム上で動作し、その結果インターネットORB間プロトコルI I O P 2を介してメッセージを送らなければならないと仮定する。このプロセスでは、2つの関係するオブジェクト・リクエスト・ブローカ6間にT C P / I P接続13が確立され、この接続を介して、インターネットORB間プロトコル8データ単位(I I O P - P D U)が送られる。インターフェース8によって値6の厳密な記述がORB 4内のデータ構造の記述を表すいわゆる「タイプ・コード」5の形でオブジェクト・リクエスト・ブローカ4内に存在する間、プロトコル・データ単位の場合にタイプ・コード5の伝送が不要となる。それぞれの他のオブジェクト・リクエスト・ブローカ4中の値が、インターフェース8のC D Rデコーダ7を介して復号される。すべての必要な情報、例えば、Request_id、「ターゲット・アドレス」、および呼び出されるメソッドの名前がそこで利用可能であるからである。この情報により、プロトコル・データ単位(P D U)をオブジェクトに関連付けることが可能となる。G I O Pデコーダ3で復号するのに必要なタイプ・コード5は、オブジェクトに関連するインターフェース定義によって決定することができる。G I O Pデコーダ3は、受動C O R B Aアナライザ11の一部を形成し、C O R B Aインターフェース・リポジトリ10を介してI D Lファイル9からも情報を受け取り、G I O Pメッセージ12を送達する。

【 0 0 1 8 】

図2に、このようなトレース・ツールが、監視するプロトコル・データ単位(P D U)を別の評価のために転送するようにどのように意図されるかを示す。システムの様々な地点、例えば各ホストでのプロトコル・データ単位(P D U)の監視は同時に行われる。

【 0 0 1 9 】

10

20

30

40

50

図3は、以下で例示によってさらに詳細に説明する方法により、監視するプロトコル・データ単位(PDU)、接続パラメータ、および図4による関連するIDL定義だけに基いて、一般ORB間プロトコル・データ単位(GIOP PDU)を図5に示すように完全に復号することが可能となることを示す。

【0020】

使用する接続指向プロトコル特有の情報により、記録された一般ORB間プロトコル・メッセージを、2つの通信するエージェントまたはコンポーネント1および1'と関連付けることが可能となる。さらに、要求で符号化されるターゲット・アドレスを介して、アドレス指定したCORBAオブジェクトを一意的に識別することも可能となる。やはり送られる「応答id(Reply id)」により、応答が同じ接続を介して関連する要求として送られることによって、応答を一意的に関連付けることができる。

10

【0021】

受動GIOPデコーダ3によるプロトコル・データ単位(PDU)の分析中に問題が存在する。その解決策を以下で説明する。

【0022】

ロギングした要求プロトコル・データ単位(要求PDU)は、それ自体で呼び出すメソッドの名前だけを含み、そのIDL定義や、アドレス指定したオブジェクトの下にあるインターフェースの名前を含まない。したがって、プロトコル・データ単位(PDU)を復号するために必要な情報が欠けている。図6に示す例示の実施例は、受け取ったプロトコル・データ単位(PDU)をGIOPデコーダ3が常に復号することが可能となる方法を表す。

20

【0023】

図6は、UML表現のインターフェース定義を含む。

【0024】

既に説明したように、メソッド「Count」を呼び出したという情報は、インターフェースA1、インターフェースA2、またはこれらの定義から継承するインターフェース定義のうち1つがアドレス指定したオブジェクト下にあるかどうかを判定するのに十分ではない。メソッドgetElemの呼出しも、どのインターフェース定義がオブジェクト下にあるかを一意的に判定するのに十分ではない。しかし、他のインターフェース定義は、この方法の範囲内で完全に可能である。したがって、この方法の基本的概念は、プロトコル・データ単位(PDU)から導出される個々の不完全な情報の集合を単一オブジェクトごとに個別に収集し、相関させることである。このようにして、復号中に基礎とするインターフェース定義用の十分な選択基準が、複数の必要条件から得られる。

30

【0025】

この状況では、単一オブジェクトに関連して、GIOPデコーダ3で以下の段階を識別することができる。

1. 初期化

すべての可能なインターフェース定義のリストを作成する。このリストは、具体的には、特定のホスト、特定のエージェント、すなわちGIOPメッセージを送受信することを担当するエンティティ、またはエージェント特有のターゲット・アドレスに関して記憶される。追加の情報が周知でない場合、最も一般的なケースでは、システム全体で使用するすべてのインターフェース定義のリストだけを初期化すべきである。

40

2. 学習

排除基準(exclusion criteria)を使用して、評価されたメッセージをまだ一意的に復号することができない可能性の数を減らす。このような場合、特定のオブジェクト下にある可能性のある複数のインターフェース定義が存在する。

3. 復号

どのインターフェース定義がそのオブジェクト下にあるかを一意的に確認した。したがって、オブジェクトと送受信するメッセージを復号することができる。

【0026】

50

原理上は、特に段階 2 と段階 3 との間に明確な区分はない。段階 3 から段階 2 に部分的に戻る必要がある可能性がある。段階 3 にまだ到達していないために復号することができないすべてのメッセージは、単に非復号形式で出力するか、または復号プロセスを後に完了しなければならない可能性がある。特に単純かつ有利な実施例は、始めにすべての記録されたプロトコル・データ単位 (P D U) と共に段階 2 を実施し、その後ですべてのこれらのプロトコル・データ単位 (P D U) を段階 3 で再度復号することである。

【 0 0 2 7 】

I I O P プロトコルに基づいて C O R B A オブジェクトを一意的にアドレス指定するために使用する相互運用オブジェクト参照 (I O R) の交換中、情報がサーバからクライアントに送られる。この情報を得ることにより、G I O P メッセージの後の復号をずっと容易にすることができる。しかし、相互運用オブジェクト参照 (I O R) の交換は、実際に後続のメッセージ交換の前提条件ではあるが、その一部を形成しない。このため、実際のクライアント・サーバ接続に加えて、別の I I O P 接続を選択的にロギング/トレースし、それを後続の復号に含めることが助けになる。しかし、これにより、解析すべき追加の知識がシステム上に必要となる。これに関して、命名サーバまたは情報リポジトリへの接続が特に興味深い。

【 0 0 2 8 】

請求項に記載の方法の、別の有利な実施例は、アクティブなサーバに関して得られる情報、下にあるインターフェース定義、ならびに選択的なロギング/記録および復号による関連するアドレスが、非システム・クライアント実装に渡されるものである。この手順により、あらかじめオブジェクト参照をロードし、または要求したりせずに、分散アプリケーションの通信で仲裁することが可能となる。これにより、多くのアクティブ・テスト・アプリケーションで共通の問題、すなわちオブジェクト参照が分散アプリケーションの外部で周知でないときにオブジェクト参照を得る問題が回避される。

【 0 0 2 9 】

次に、以下ではある条件におけるいくつかの特殊なケースを説明する。

【 0 0 3 0 】

第 1 のケースでは、呼出しメソッドおよび関連するインターフェース定義が、インターフェース・リポジトリ (I n t e r F a c e R e p o s i t o r y : I F R) 内に格納されないと仮定する。この状況では、分散アプリケーションで使用されるインターフェース定義のすべてが、G I O P デコーダ 3 のインターフェース・リポジトリ (I F R) 内に格納されるわけではないと仮定する。このケースでは、インターフェース・リポジトリ (I F R) で、メソッドが適切な名前前で全く定義されていないと仮定する。したがって、メソッド A N D は要求プロトコル・データ単位 (P D U) で呼び出され、したがってインターフェース・リポジトリ (I F R) (例えばインターフェース A、インターフェース A 1 および A 2) に存在するインターフェース定義に基づいて、アドレス指定したオブジェクトに対して可能なインターフェース定義の数を制限することができず、このプロトコル・データ単位 (P D U) の復号に関して情報を得ることもできない。したがって、このプロトコル・データ単位 (P D U) は、どちらの復号形式でも表すことができない。

【 0 0 3 1 】

第 2 の特殊ケースでは、A I S u b で定義されるメソッド g e t E l e m が要求プロトコル・データ単位で符号化されると仮定する。さらに、メソッド g e t E l e m が同時に A 2 にも定義されると仮定する。議論する要求に関する可能なインターフェース定義の集合内にインターフェース定義 A 2 が依然として存在する場合、G I O P デコーダ 3 は、符号化 G I O P メッセージを定義 A 2 : g e t E l e m を使用して復号できないことを確立することになる。このために、G I O P デコーダ 3 はインターフェース A 2 を除外し、それから継承したすべての定義は、アドレス指定したオブジェクトに対して可能なインターフェース定義の集合に属する。したがってこの例では、定義 A 2 および A 2 S u b は、A 2 S u b は G I O P デコーダ 3 ではいずれにしても注目されないことを除けば、除外することができる。

【 0 0 3 2 】

第3の特殊ケースA3では、呼び出すメソッドおよび関連するインターフェース定義がインターフェース・リポジトリ（IFR）内に格納されると仮定する。しかし、それと両立する定義は、別のインターフェース定義内のインターフェース・リポジトリIFRの外部に存在する。メソッドA2「Sub::ismember」を起動する要求は、このメソッドによってインターフェース定義A1に間違って関連付けられる。インターフェース定義A1が、このメソッドを含むインターフェース・リポジトリ（IFR）内の唯一の定義であるからである。したがって、正しくプロトコル・データ単位（PDU）を復号することが実際に可能であるが、将来の要求は、オブジェクトに対して使用されるインターフェース定義に関する偽データのために復号することができない。この不規則な挙動を除外するため、2つの基本的手順を実装することができる。

10

1. G I O Pデコーダ3のオペレータは、分散アプリケーションで使用されるすべてのIDL定義がインターフェース・リポジトリ（IFR）内に格納されることを保証しなければならない。この例で、A2Subがインターフェース・リポジトリIFR内でも周知であった場合、ここで説明した不規則な判定は生じない。

2. ブロック化リストに基づいて、含まれるメソッド名を介して識別されるある要求を使用してインターフェース定義を除外しないように、G I O Pデコーダを手動で構成する。

【0033】

2つの別の特殊ケースB1およびB2では、要求プロトコル・データ単位（PDU）を単一インターフェース定義に一意的に関連付けることができないが、同一のメソッド定義だけが異なるインターフェース定義内で選択用に残されているので、プロトコル・データ単位を事前に復号することができると仮定する。それでも、特殊ケースB1では、能力が共通祖先を介して継承されるので復号が可能である。したがって、メソッド定義は等しいだけでなく、同一であり、より厳密なステートメントが可能となるまで定義Aがオブジェクト下にあると仮定することができる。

20

【0034】

特殊ケースB2では、共通祖先を用いずに複数のインターフェース定義が可能であると仮定する。したがって、特定のメソッドが実際には共通祖先を介して継承されたのではなく、何回かの定義が存在し、それによってA1:Xを表したのか、それともA2:Xを表したのかに関する判定を行うことさえせずに要求を復号することが可能となるケースが生ずる。

30

【0035】

最後に、要求を複数の異なるメソッド定義に基づいて復号することができる特殊ケースCを説明する。定義は互いに異なるので、プロトコル・データ単位（PDU）の復号は、各ケースで、結果として異なるメッセージを生成することになる。したがって、プロトコル・データ単位（PDU）は、非両立な定義を他のプロトコル・データ単位（PDU）に基づいて除外することができる場合にのみ復号することができる。

【0036】

疑似コードとして単純化形式で表されるメソッドの基本シーケンスを以下で説明する。

【0037】

話を簡単にするために、TCPを移送プロトコルとして使用すると仮定する。やはり話を簡単にするために、G I O Pメッセージ・タイプの要求および応答だけが存在すると仮定する。しかし、このことは、メソッドのオペレーションの基本モードについて何ら変更を行うものではない。

40

1. 次のTCPメッセージを読み取る

2. メッセージ・タイプを判定する

2. 1 要求の場合、3に進む

2. 2 応答の場合、4に進む

3. 要求を復号する

3. 1 フィールドiAddrDestおよびDestPortからC O R B Aエージェントを判定する

3. 2 G I O Pコーディングから呼び出すメソッドの名前を判定する

50

3.3 使用するobject_key下の、このエージェントに対するすべてのインターフェース定義のリストを確立する

3.3.1 リストが格納されていない場合、I F R内に格納されているすべてのインターフェース定義を含む新しいリストを生成する

3.3.2 リストをロードする

3.4 メソッド名がブロック化メソッドのリスト中にある場合、3.5をスキップし、3.6に進む

3.5 リストを更新する

3.5.1 インターフェース定義で現メソッドが許可されていないすべての要素をリストから除去する

10

3.5.1.1 メソッドがインターフェース定義のいずれにも格納されていない場合、特殊ケースA1が当てはまる。このステップをやり直し、1に進む

3.5.1.2 現メッセージを復号することができるようにメソッドがインターフェース定義のいずれにも格納されていない場合、特殊ケースA2が当てはまる。このケースでは、非両立のメソッドを定義する要素だけを除去し、1に進む

3.5.1.3 このメソッドを非両立に定義するか、または全く定義しないすべての要素を除去する

3.5.2 更新したリストを格納する

3.6 メッセージを復号する

3.6.1 1つのインターフェース定義だけがリスト内に格納されている場合、このインターフェース定義を取り、メッセージを復号する。3.7に進む

20

3.6.2 固有の階層的継承関係がリスト内に含まれるインターフェース定義間に存在する場合、特殊ケースB1が当てはまる。この場合、リスト内に含まれる共通祖先のIDL定義を取る。3.7に進む

3.6.3 それぞれのメソッドがリスト内に含まれるインターフェース定義内に等しく定義された場合、特殊ケースB2が当てはまる。この場合、復号のために含まれる任意のIDL定義を取る。3.7に進む

3.6.4 特殊ケースCが当てはまる。メッセージは現在、一意的に復号可能ではない。1に進む

3.7 応答を復号するための情報を、「iAddrSource」、「SourcePort」、「iAddrDest」、および「DestPort」から構成されるラベルの下に格納する。

30

3.8 1に進む

4. 応答を復号する

4.1 フィールドiAddrSourceおよびSourcePortからC O B R Aエージェントを決定する

4.2 関連する要求がiAddrDest、DestPort、iAddrSource、およびSourcePortから構成されるラベルの下に格納されていない場合、1に進む

4.3 関連する要求をロードする

4.4 要求のIDL定義を取り、応答を復号する

4.5 1に進む

40

【図面の簡単な説明】

【図1】 インターネットO R B間プロトコル(I I O P)の例による、受動G I O Pデコードの使用法を示す図である。

【図2】 T C P / I Pを介して送られるG I O Pメッセージ用のトレース・インターフェースを表す図である。

【図3】 記録されたG I O Pメッセージの表現である。

【図4】 インターフェースのIDL定義を示す図である。

【図5】 手動で復号されるG I O Pメッセージを示す図である。

【図6】 階層インターフェース定義のU M L表現を示す図である。

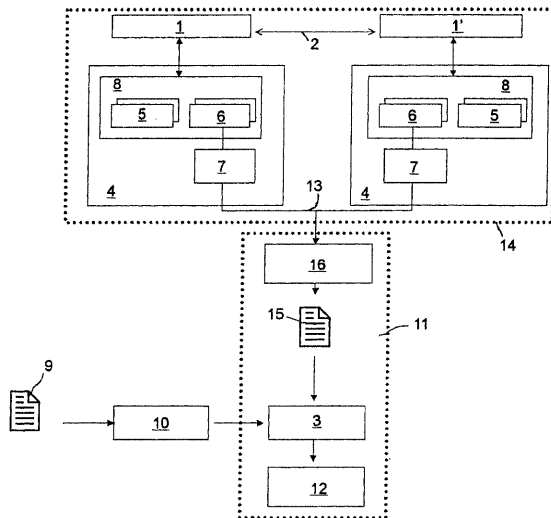
【符号の説明】

50

- 1, 1' I I O Pを介するクライアント/サーバ分散型C O R B Aアプリケーションのコンポーネント
- 2 I I O PインターネットO R B間プロトコル
- 3 G I O Pデコーダ
- 4 オブジェクト・リクエスト・ブローカ
- 5 タイプ名またはコード
- 6 値
- 7 C D Rコーダ
- 8 インターフェース
- 9 I D Lファイル
- 10 C O R B A I F R
- 11 受動C O R B Aアナライザ
- 12 G I O Pメッセージ
- 13 T C P接続
- 14 分散C O R B Aアプリケーション
- 15 G I O P P D Uを用いたT C P / I Pメモリ・ダンプ
- 16 トレース・ツール

10

【図1】



【図2】

2 byte	year
1 byte	month (0-11)
1 byte	date
1 byte	hour
1 byte	minute
1 byte	second
2 byte	millisek
4 bit (MSB)	0000 TCP (RFC: 793)
4 bit (LSB)	interfaceNr
4 byte	iAddrSource
4 byte	iAddrDest
2 byte	sourcePort
2 byte	destPort
4 byte	seqNumber
4 byte	ackNumber
1 byte	dataOffset
1 byte	flags
2 byte	windowSize
2 byte	checksum
2 byte	urgentPointer
2 byte	messageSize
[messageSize] byte	GIOP Message

【図3】

```

00000000h: 47 49 4F 50 01 00 00 00 00 00 00 00 00 00 00 00 : GIOP
00000010h: 00 00 00 04 01 00 00 00 00 00 33 3A 5C 31 39      : 3:19
00000020h: 32 2E 31 36 38 2E 31 39 39 2E 31 30 33 3A 4E 53 : 2.160.199.103:NS
00000030h: 3A 3A 3A 49 46 52 3A 43 6F 79 4E 61 6D 69 6E 67 : IFR:CodeName
00000040h: 5F 4E 61 6D 69 6E 67 43 6F 6E 74 65 78 74 00 00 : NamingContext
00000050h: 00 00 00 05 6C 69 73 74 00 00 00 00 00 00 01    : list
00000060h: 00 00 00 00 7F FF FF FF

```

【 図 4 】

```

interface NamingContext {
    void list (
        in unsigned long how_many,
        out BindingList bl,
        out BindingIterator bi
    );
    Object resolve (
        in Name n
    ) raises( NotFound, CannotProceed, InvalidName);
};

```

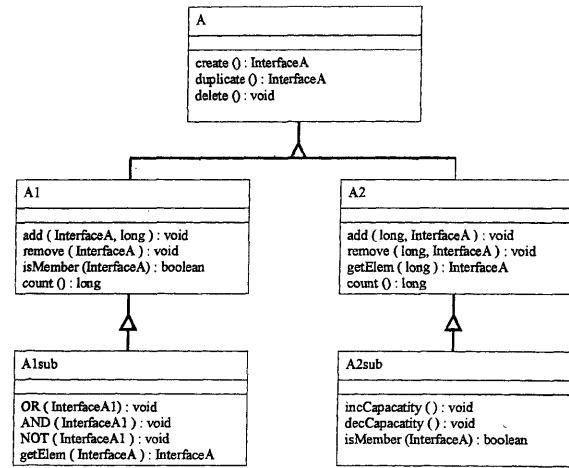
【 図 5 】

```

Message {
    MessageHeader {
        magic {'G','T','O','P'},
        GIOP_version {major 1, minor 0},
        flags 0,
        message_type 0,
        message_size 92
    },
    RequestHeader {
        service_context {},
        request_id 4,
        response_expected true,
        reserved {46,49,57},
        object_key {58, 92, 49, 57, 50, 46, 49, 54, 56, 46, 49, 57, 57, 46, 49, 48, 51, 58, 78, 83,
                    58, 73, 82, 58, 67, 111, 115, 78, 97, 109, 105, 110, 103, 95, 78, 97, 109, 10,
                    103, 67, 111, 110, 116, 101, 120, 116, 0},
        operation "list",
        requesting_principal {}
    },
    CosNaming/NamingContext/list{ how_many 2147483647}
}

```

【 図 6 】



フロントページの続き

- (74)代理人 100101498
弁理士 越智 隆夫
- (74)代理人 100096688
弁理士 本宮 照久
- (74)代理人 100102808
弁理士 高梨 憲通
- (74)代理人 100104352
弁理士 朝日 伸光
- (74)代理人 100107401
弁理士 高橋 誠一郎
- (74)代理人 100106183
弁理士 吉澤 弘司
- (72)発明者 ダッソウ, ヘイコ
ドイツ 6 4 3 4 7 グリースハイム, ガルテンシュトラーク 4
- (72)発明者 コーペ, グイド
ドイツ 6 4 3 4 7 グリースハイム, フルッグハーフェンシュトラーク 3 ツェー

審査官 阿部 弘

- (56)参考文献 特開 2 0 0 0 - 0 5 6 9 9 7 (J P , A)
特開平 1 1 - 1 1 9 9 8 6 (J P , A)

- (58)調査した分野(Int.Cl., D B 名)
H04L 29/06