(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4818251号
(P4818251)
(45)発行日　平成23年11月16日(2011.11.16)　(24)登録日　平成23年9月9日(2011.9.9)

(51)Int.Cl.　F I
H04L 29/06 (2006.01)　H04L 13/00　305C

請求項の数 5　(全 9 頁)

(21)出願番号	特願2007-500201 (P2007-500201)
(86)(22)出願日	平成17年1月24日 (2005.1.24)
(65)公表番号	特表2007-525133 (P2007-525133A)
(43)公表日	平成19年8月30日 (2007.8.30)
(86)国際出願番号	PCT/EP2005/050299
(87)国際公開番号	WO2005/083967
(87)国際公開日	平成17年9月9日 (2005.9.9)
審査請求日	平成19年11月2日 (2007.11.2)
(31)優先権主張番号	10/789,573
(32)優先日	平成16年2月27日 (2004.2.27)
(33)優先権主張国	米国(US)

前置審査

(73)特許権者　390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国１０５０４　ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
(74)代理人　100108501
弁理士　上野　剛史
(74)代理人　100112690
弁理士　太佐　種一
(74)代理人　100091568
弁理士　市位　嘉宏

最終頁に続く

(54)【発明の名称】アクセスするクライアントのサーバー側プロトコル構成

(57)【特許請求の範囲】

【請求項１】

クライアント側通信プロトコルスタックを構成する方法で、前記方法が、サーバー側プロトコルスタックインスタンスにまとめられるプロトコルスタックコンポーネントに対応する必要なクライアント側プロトコルスタックコンポーネントのリスティングにアクセスするステップと、前記リスティングから、前記必要なクライアント側プロトコルスタックコンポーネントを含むクライアント側プロトコルスタックインスタンスがすでに存在しているかどうかを判別するステップと、前記クライアント側プロトコルスタックインスタンスがすでに存在している場合、前記クライアント側プロトコルスタックインスタンスを再利用して前記サーバー側プロトコルスタックと通信し、そうでなければ前記必要なクライアント側プロトコルスタックコンポーネントを使ってクライアント側プロトコルスタックの新たなインスタンスを作成するステップとを含む方法。

【請求項２】

前記アクセスするステップが、ネイティブディレクトリインターフェースから前記リスティングにアクセスするステップを含む請求項１に記載の方法。

【請求項３】

さらに、前記必要なクライアント側プロトコルスタックコンポーネントに適用される属性を識別するステップと、前記識別した属性を前記新たなインスタンスの前記クライアント側プロトコルスタックコンポーネントに適用するステップとを含む請求項１に記載の方法。

【請求項４】
前記再利用するステップが、前記クライアント側プロトコルスタックインスタンスが前記識別した属性を含む場合にのみ、前記クライアント側プロトコルスタックインスタンスを再利用して、前記サーバー側プロトコルスタックと通信するステップと、そうでなければ、前記必要なクライアント側プロトコルスタックコンポーネントを使用し、前記識別した属性を含むクライアント側プロトコルスタックの新たなインスタンスを作成するステップとを含む請求項６に記載の方法。
【請求項５】
請求項１から４のいずれかの方法の各ステップをコンピューターに実行させるプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明はプロセス間通信分野に関係し、より具体的にはサーバープロセスとの通信のためのクライアントプロセスの構成に関係する。
【背景技術】
【０００２】
プロセス間通信は、プロセスが１台のコンピューティングデバイス内にまとめて配設されているか、又はプロセスがコンピュータ通信ネットワークの周りに互いに離れて配置されているかどうかに関係なく、２つ以上のコンピューティングプロセス間の情報の交換に関係する。従来、個別のコンピューティングプロセスが、情報交換のために確立された通信リンクを介して互いに通信できる。ほとんどの場合、通信リンクを確立する前、又は通信リンクを確立する過程で取り決められる共通の通信プロトコルを使って情報を交換できる。
【０００３】
多くの場合、複数のコンピューティングプロセスを利用して情報を交換する共通の通信プロトコルは、通信リンクを確立しようと試みる前に予め決めることができる。例えば、インターネットの場合、ウエブのコンテンツは、ＴＣＰ／ＩＰ上でハイパーテキスト転送プロトコル（ＨＴＴＰ）を使って、コンテンツサーバーとコンテンツブラウザとの間で交換できる。さらに、通信プロトコルは必ずしも物理的なネットワークレイヤに基づくデータ交換に限られたものではない。むしろ、通信プロトコルは二以上のコンピューティングプロセス間の情報交換を可能にするために必要な動作および通信構成のあらゆる集合体を含むことができる。この点に関し、通信プロトコルはプロセス間のデータの転送メカニズムだけでなく、数例をあげれば、暗号化プロトコル、圧縮プロトコル、トンネリングプロトコルなどの補助的な設定や動作も含むことができる。
【０００４】
ある状況においては、クライアントのコンピューティングプロセスとサーバーのコンピューティングプロセスの間の通信プロトコルを予め決めない。特に、サーバーのコンピューティングプロセスが異なる数多くの通信プロトコルを使って多数のクライアントのコンピューティングプロセスと通信するように構成されている場合、クライアントとサーバーのどれか１回のセッションにどのコンピューティングプロセスを利用するかは予め決められない。その結果、クライアントのコンピューティングプロセスは、試行錯誤の過程から適切な通信プロトコルの選択をさせることが多い。具体的には、代表的な状況では、クライアントプロセスは複数の通信プロトコル候補のテーブルを含むことができる。このように、クライアントプロセスは、通信リンクがうまく確立される時まで各通信プロトコル候補を使ってサーバープロセスとの通信リンクの確立を試みなければならない。
【０００５】
サーバープロセス間の通信リンクを確立する前述のプロセスは、究極的には、通信リンクを周知の通信プロトコルの１つを使って確立できるという限りにおいては効果的であろう。また、当業者は、通信リンクを確立するときに最終的に選択される通信プロトコル以外

のある通信プロトコルを使って通信リンクの確立が失敗する度に生じる多大なリソースの無駄を認めるであろう。可能な限り、通信プロトコルを選択するときに試行錯誤の過程を避けて、パフォーマンスレベルを高めるのが好ましいであろう。

【発明の開示】

【発明が解決しようとする課題】

【０００６】

本発明は、プロセス間通信に関する技術の欠点を解決することを目指し、クライアント側のコンピューティングプロセスで通信プロトコルスタックを構成する方法およびシステムを提供する。発明のある好適な側面で、プロセス間通信プロトコル構成システムは、サーバー側プロトコルスタックにおいてサーバー側プロトコルスタックコンポーネントをクライアント側プロトコルスタックコンポーネントにマッピングすることを含むことができる。システムはまた、マッピングに基づいてサーバー側プロトコルスタックのサーバー側プロトコルスタックコンポーネントの従属するもののリスティングを含むことができ、最終的にシステムはリスティングへの発行可能インターフェースを含むことができる。

【０００７】

マッピングは、サーバー側プロトコルスタックのプロトコルスタックコンポーネントと、関連するクライアント側プロトコルスタックに存在しなければならない対応するプロトコルスタックコンポーネントとのリンケージのセットを含むことができる。マッピングはまた、対応するプロトコルスタックコンポーネントのうち選択されたものの属性のセットを含むことができる。発明の好適な側面では、リスティングは、マッピングに基づいたサーバー側プロトコルスタックにおいてサーバー側プロトコルスタックコンポーネントのうちの従属するものの逆順リストを有する逆リスティングとなる。また、発行可能インターフェースはリスティングを内蔵する少なくとも１つのオブジェクトへのネイティブディレクトリインターフェースであることが好ましい。

【課題を解決するための手段】

【０００８】

クライアント側通信プロトコルスタックを構成する方法において、マッピングを調べて、サーバー側プロトコルスタックインスタンスにおいてプロトコルスタックコンポーネントのセットに対応するクライアント側プロトコルスタックコンポーネントのセットを判別することができる。その後、判別したクライアント側プロトコルスタックコンポーネントのセットのリスティングを作成できる。最後に、外部に設けられるクライアントコンピューティングプロセスによるアクセスのためにリスティングを発行できる。好ましくはリスティングを逆順にする。また、調べるステップはさらに、クライアント側プロトコルスタックをサーバー側プロトコルスタックインスタンスと相互運用できるときに、クライアント側プロトコルスタックコンポーネントの少なくとも１つに適用される少なくとも１つの属性を判別するステップを含むことができる。

【０００９】

反対に、クライアント側通信プロトコルスタックを構成する方法は、サーバー側プロトコルスタックインスタンスにまとめられるプロトコルスタックコンポーネントに対応する必要なクライアント側プロトコルスタックコンポーネントのリスティングにアクセスすることを含むことができる。必要なクライアント側プロトコルスタックコンポーネントを含むクライアント側プロトコルスタックインスタンスがすでに存在しているかどうかは、リスティングから判別できる。クライアント側プロトコルスタックインスタンスがすでに存在している場合、クライアント側プロトコルスタックインスタンスを再利用して、サーバー側プロトコルスタックと通信できる。そうでなければ、必要なクライアント側プロトコルスタックコンポーネントを使って、クライアント側プロトコルスタックの新しいインスタンスを作成できる。

【発明を実施するための最良の形態】

【００１０】

本発明は、ダイナミックに選択、構成した通信プロトコルスタックに基づきサーバーコン

ピューティングプロセスと通信するために、クライアントコンピューティングプロセスをインテリジェントに構成する方法、システム、および装置である。本発明によると、プロトコルスタックの様々な対応するプロトコルエレメント又はコンポーネントを、サーバーコンピューティングプロセスと、サーバーコンピューティングプロセスにアクセスするよう構成された一又は複数のクライアントコンピューティングプロセスとの間でマッピングできる。サーバー側プロトコルコンポーネントが見込みクライアント側プロトコルコンポーネントの属性にマッピングする特別定義の属性を有する場合、その属性をマッピングで同様に指定できる。マッピングは指定される属性とともに、サーバーにプロトコルスタックを配備するときに後で使用するために格納できる。プロトコルスタックがサーバーコンピューティングプロセスで使用するために配備される場合、プロトコルスタックの従属プロトコルコンポーネントのリストが生成でき、リストを逆順にできる。従属プロトコルコンポーネントのリストにあるプロトコルコンポーネントの各々は、プロパティの値を検索できる。その結果、プロトコルスタックのクライアント構成は、従属プロトコルコンポーネントの逆順リストと、プロトコルスタックのプロパティの従属性について検索したプロパティの値の両方を使って作成できる。その後、サーバーと通信しようと試みるクライアントプロセスが格納された構成を発見できるように構成を発行する。

【００１１】

クライアントコンピューティングプロセスがサーバーコンピューティングプロセスと通信しようと試みるとき、クライアントコンピューティングプロセスはまず、それ以前の発行からサーバー指定のプロトコル構成を問合せできる。発見したら、クライアントプロセスは、指定のプロトコル構成がすでにクライアントプロセスに存在しているかどうかを判別できる。もし存在していれば、プロトコル構成を再利用して、サーバープロセスと通信できる。存在していなければ、サーバーで指定の構成を作成して、構成の従属性について関連するプロパティ値をもつように構成する。プロトコルスタックが使用可能になったら、クライアントプロセスはそのプロトコルスタックを使用してサーバープロセスと通信できる。

【００１２】

前述のより具体的な図で、図１は発明の構成に従い、アクセスするクライアントプロセスのサーバー側構成用に構成された例示的ではあるが非排他的コンピューティングシステムの概略図である。システムは、コンピュータ通信ネットワーク１２０を介してサーバープロセス１３０に通信連結されるようにプログラムされた一又は複数のクライアントコンピューティングプロセス１１０Ａ、１１０Ｂ、１１０ｎを含むことができる。注目すべきことに、各クライアントコンピューティングプロセス１１０Ａ、１１０Ｂ、１１０ｎは、確立された通信プロトコルを使ってサーバーコンピューティングプロセス１３０と通信できる。確立された通信プロトコルは、例えばＴＣＰ／ＩＰを使うＨＴＴＰなどの物理的およびアプリケーションプロトコルだけでなく、暗号化や圧縮プロセスなどの一又は複数の補助的な手順の動作も含むことができる。サーバー通信プロセス１３０は構成プロセッサ１４０に連結できる。構成プロセッサ１４０は１セットのプロトコル構成１５０を確立できる。具体的には、クライアントのコンピューティングプロセス１１０Ａ、１１０Ｂ、１１０ｎのいずれか一つに対応する必要なコンポーネントがあれば、構成プロセッサ１４０はサーバーコンピューティングプロセス１３０のプロトコルの動作に必要な個々のコンポーネントをマッピングできる。プロトコルスタックをサーバーコンピューティングプロセス１３０で作成しようとするとき、スタックにあるコンポーネントセットのコンポーネントの従属性をリスティングして、その後逆にして逆構成１８０を制作する。さらに、逆構成１８０のコンポーネントの個々のプロパティ値を判別でき、その組合せをプロトコルスタック構成１９０として発行できる。特に、業界でよく知られるように、プロトコルスタック１９０の構成は、コンテンツがネイティブディレクトリインターフェースで認識できるクラスオブジェクトの１つ又はセットへのインターフェースとして、グローバルネーム空間で発行できる。

【００１３】

動作時、クライアントコンピューティングプロセス１１０Ａ、１１０Ｂ、１１０ｎの１つがサーバーコンピューティングプロセス１３０と通信リンクを確立したい場合、クライアントコンピューティングプロセス１１０Ａ、１１０Ｂ、１１０ｎはまず問合せ１００をサーバーコンピューティングプロセス１３０に送って、サーバーコンピューティングプロセス１３０が優先通信プロトコル構成を識別するよう要求する。応答時、サーバーコンピューティングプロセス１３０はプロトコルスタック構成１９０へのアクセスを返すとともに、クライアントコンピューティングプロセス１１０Ａ、１１０Ｂ、１１０ｎのうち要求するものが、プロトコルスタック構成１９０で定義した優先通信プロトコルスタックを選択して構成できる。優先的な通信プロトコル構成を使ってサーバーコンピューティングプロセスと通信するためにクライアントコンピューティングプロセスを構成するプロセスは、マッピングに、サーバー側とクライアント側のプロトコルを互いにリンクすることを含むことができる。また、プロセスは、サーバーコンピューティングプロセスで通信プロトコルスタックを開始することと、マッピングに基づいて対応する発行可能なクライアント側の通信プロトコルスタック構成を展開することを含むことができる。最後に、プロセスは、サーバー指定のプロトコルスタックに基づいて、様々なクライアント側のプロトコルコンポーネントの固有プロトコルスタックを選択、構成することを含むことができる。

【００１４】

前述のプロセスを詳しく示すために、図２は、図１のシステムにおいてサーバー側構成の対応するクライアントプロトコルエレメントとサーバープロトコルエレメントをリンクするプロセスを図示するフローチャートである。

【００１５】

ブロック２１０から始まり、構成のために第１プロトコルセットにロードできる。ブロック２２０で、サーバーコンピューティングプロセスのセットになったプロトコルコンポーネントの各々を、通信が見込まれるクライアントプロセスの対応する従属プロトコルコンポーネントにリンク又はマッピングできる。例えば、サーバーコンピューティングプロセスにおけるＨＴＴＰアプリケーションレベルのプロトコルの使用は、クライアントコンピューティングプロセスのＨＴＴＰにマッピングできる。同様に、サーバーコンピューティングプロセスの物理的な制御レイヤプロトコルとしてのＴＣＰは、クライアントコンピューティングプロセスのＴＣＰにマッピングできる。さらに別の例として、サーバーコンピューティングプロセスの解凍プロセスは、クライアントコンピューティングプロセスの圧縮プロセスにマッピングできる。

【００１６】

いずれの場合も、判断ブロック２３０で、従属プロトコルコンポーネントのうち選択したものの固有の属性を、サーバーコンピューティングプロセスにおいてプロトコルコンポーネントのうち固有の属性に対応するもののアプリケーションに基づいて、適用するべきかどうかを判別できる。属性をマッピングする場合、ブロック２４０のマッピングで属性がそのように指定できる。いずれの場合も、判断ブロック２５０で、処理すべき追加のプロトコルセットがまだ残っているかどうかを判別できる。もし残っていれば、ブロック２６０で構成のためにプロトコルセットにロードでき、ブロック２２０から２６０のプロセスを繰り返す。マッピングで構成すべきプロトコルがもう残っていない場合、ブロック２７０でプロセスは終了する。

【００１７】

様々なプロトコルスタックコンポーネントと対応する属性をマッピングでリンクしてしまえば、プロトコルスタックのある特定のインスタンスの発行される構成はマッピングを使って制作できる。より具体的に、図３は図１のシステムにおけるサーバーの様々なプロトコルエレメントのプロトコルスタックを開始するプロセスを図示する。ブロック３１０で始まり、プロトコルスタックを開始できる。例えば、オブジェクト指向の環境では、プロトコルスタックにまとめられる様々なプロトコルコンポーネントをプロセスアドレス空間内でインスタンス生成して、その中で動作できるようにできる。ブロック３２０で、例えばサーバー側プロトコルスタックの各コンポーネントをトラバースし、マッピングからコ

ンポーネントの従属性を識別することによって、クライアント側従属コンポーネントのリストを作成できる。プロトコルスタックの従属コンポーネントのリストが作成されてしまえば、ブロック３３０で、リストの順番を逆にできる。このように、リストに基づいてプロトコルスタックを構築するクライアントコンピューティングプロセスは、スタックの各コンポーネントがその従属性に依存できるように保証できる。いずれの場合も、判断ブロック３４０で、スタックの従属性が、従属性が利用できるコンポーネントのインスタンスに依存している点で適切かどうかを判別できる。適切でなければ、プロセスはブロック３５０で終了する。そうでなければ、プロセスはブロック３６０に進む。ブロック３６０で、適切な従属性が存在していれば、各通信プロトコルコンポーネントのプロパティの値を検索できる。また、ブロック３７０で、クライアントに配備するべきプロトコルスタックの構成は、逆順にしたリストに基づいて作成できる。最後に、ブロック３８０で、検索したプロパティ値を作成したクライアントプロトコルスタックの構成に付加できる。ブロック３９０で、要求があれば、クライアントプロトコルスタックの構成を何らかの周知のメカニズムにより要求するクライアントプロセスに渡すことができる。発明の好適な側面では、構成はネイティブディレクトリインターフェースからグローバルネーム空間に発行できる。

【００１８】

一旦発行したら、クライアント側プロトコルスタック構成には、サーバーコンピューティングプロセスにより優先通信プロトコルを使ってサーバーコンピューティングとの通信リンクの確立に関わるクライアントコンピューティングプロセスからアクセスできる。重要なことに、クライアントコンピューティングプロセスは、どのプロトコルを使うかを推定するとき試行錯誤をしなくても、また、どのプロトコルを使うかについて予め合意を形成しなくても、適切な通信プロトコルを選ぶことができる。より具体的に、図４は、図１のシステムにおいてサーバー指定のプロトコルスタックに基づいて、様々なクライアント側のプロトコルエレメントの固有プロトコルスタックを選択、構成するプロセスを図示するフローチャートである。

【００１９】

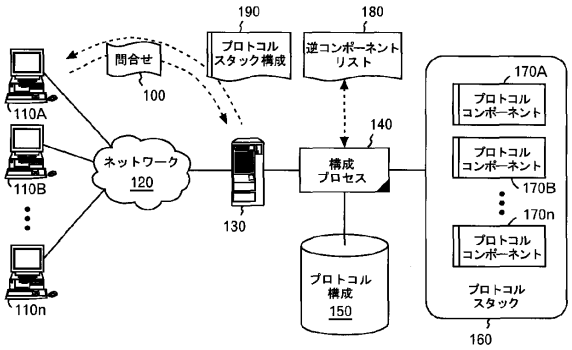
ブロック４０１から始まり、優先プロトコルスタック構成をサーバーコンピューティングプロセスから検索できる。この点について、クライアントコンピューティングプロセスは、優先プロトコルスタック構成に関し、例えば優先プロトコルスタックとのネイティブディレクトリインターフェースから優先プロトコルスタックとのインターフェースにアクセスすることによって、サーバーコンピューティングプロセスに問合せできる。判断ブロック４２０で、優先プロトコルスタックがクライアントコンピューティングプロセスのプロトコルスタックのすでに存在するインスタンスに似ているかどうかを判別できる。似ていなければ、各コンポーネントを適切な従属順で構成内に指定させる新たなプロトコルスタックインスタンスを作成できる。加えて、優先プロトコルスタックで説明したようにコンポーネントの属性を設定でき、さらにコンポーネントのプロパティのプロパティ値を優先プロトコルスタックに従い設定できる。しかし、判断ブロック４２０で、優先プロトコルスタックがクライアントコンピューティングプロセスのプロトコルスタックのすでに存在するインスタンスに似ていれば、さらにブロック４３０で、プロトコルスタックの既存のインスタンスが優先プロトコルスタックが指定する同じ属性を含むかどうかを判別できる。含まなければ、前述のとおりブロック４５０で、クライアントで、優先通信プロトコルスタックの新たなインスタンスを作成できる。そうでなければ、ブロック４４０で、通信プロトコルスタックの既存のインスタンスを、サーバーコンピューティングプロセスとの通信に再利用できる。

【００２０】

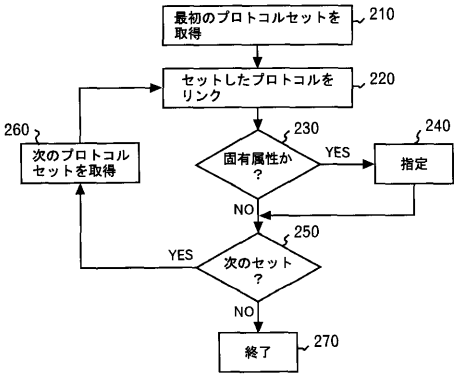
本発明は、ハードウェア、ソフトウェア、又はハードウェアとソフトウェアの組合せで実現できる。本発明の方法およびシステムの実施は、１台のコンピュータシステムで集中的に、又は相互接続された複数のコンピュータシステムに様々なエレメントが広がる分散型で実現できる。本明細書に説明する方法を実施するようになされたあらゆる種類のコンピ

ュータシステム、又はその他の装置が、本明細書に説明する機能を遂行するのに適する。
【００２１】
典型的なハードウェアとソフトウェアの組合せとは、ロードおよび実行されるときに、本明細書に説明する方法を実施するようにコンピュータシステムを制御するコンピュータプログラムを内蔵する汎用コンピュータシステムであろう。本発明は、本明細書で説明する方法の実施を可能にするすべての特徴を持ち、コンピュータシステムでロードされると、これらの方法を実施できるコンピュータプログラム製品に埋め込むこともできる。
【００２２】
本文におけるコンピュータプログラム又はアプリケーションとは、あらゆる言語、コード、又は表記法で、直接、あるいはａ）別の言語、コード、又は表記法への変換とｂ）異なる有形形態での複製の一方又は両方の後のいずれかで、情報処理能力をもつシステムにある特定の機能を遂行させるように意図される命令のセットのあらゆる表現を意味する。重要なことに、本発明は他の特別な形態でも具現でき、従って、発明の範囲を示すとき、前述の明細書ではなく以下の請求項を参照するべきである。
【図面の簡単な説明】
【００２３】
ここで、本発明の実施例を、以下の図面を参照して、単なる例として説明していく。
【００２４】
【図１】本発明の実施例に従いアクセスするクライアントのサーバー側構成用に構成したコンピューティングシステムの概略図である。
【図２】図１のシステムのサーバー側構成において、対応するクライアントのプロトコルエレメントとサーバーのプロトコルエレメントをリンクするプロセスを図示するフローチャートである。
【図３】図１のシステムのサーバーにおいて、様々なプロトコルエレメントのプロトコルスタックを開始するプロセスを図示するフローチャートである。
【図４】図１のシステムのサーバー指定のプロトコルスタックに基づき、様々なクライアント側プロトコルエレメントの固有プロトコルスタックを選択、構成するプロセスを図示するフローチャートである。

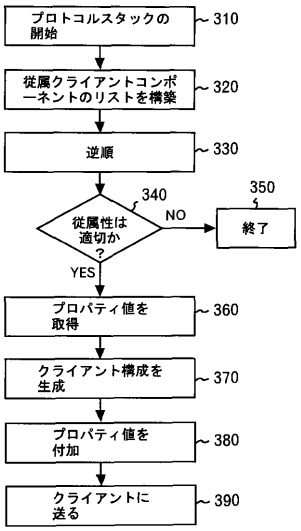
【図１】


190
180
プロトコルスタック構成
逆コンポーネントリスト
問合せ
100
170A
プロトコルコンポーネント
110A
140
ネットワーク
120
構成プロセス
プロトコルコンポーネント
170B
110B
130
170n
プロトコルコンポーネント
プロトコル構成
150
プロトコルスタック
110n
160

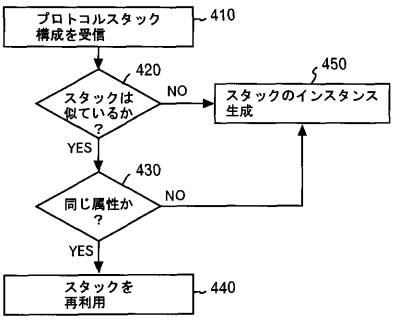
【図２】


最初のプロトコルセットを取得
210
セットしたプロトコルをリンク
220
260
230
240
次のプロトコルセットを取得
固有属性か？
YES
指定
NO
250
YES
次のセット？
NO
終了
270

【図３】


プロトコルスタックの開始
310
従属クライアントコンポーネントのリストを構築
320
逆順
330
340
350
従属性は適切か？
NO
終了
YES
プロパティ値を取得
360
クライアント構成を生成
370
プロパティ値を付加
380
クライアントに送る
390

【図４】


プロトコルスタック構成を受信
410
420
450
スタックは似ているか？
NO
スタックのインスタンス生成
YES
430
同じ属性か？
NO
YES
スタックを再利用
440

フロントページの続き

(72)発明者　ブルカート、エリック
アメリカ合衆国　２７６０４　ノースカロライナ州　ローリー　リッジ・ドライブ　パーシモン　２３４８
(72)発明者　フォンテス、ステファン
アメリカ合衆国　２７５６０　ノースカロライナ州　モーリスビル　ウィンター・ウォーク・サークル　１３３６
(72)発明者　ランゼン、クレーグ
アメリカ合衆国　４８１４４　ミシガン州　ランバートビル　レノックス・コート　３１５７

審査官　阿部　弘

(56)参考文献　米国特許第０６２１２１６０（ＵＳ，Ｂ１）
欧州特許出願公開第０１２９２１６０（ＥＰ，Ａ１）
米国特許第０５９５１６３９（ＵＳ，Ａ）
特開２０００－０９００３０（ＪＰ，Ａ）
特開平０５－０４６４１４（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
H04L 29/06