

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4726539号  
(P4726539)

(45) 発行日 平成23年7月20日(2011.7.20)

(24) 登録日 平成23年4月22日(2011.4.22)

(51) Int.Cl. F I  
G O 6 F 13/00 (2006.01) G O 6 F 13/00 3 5 3 C

請求項の数 12 (全 25 頁)

<p>(21) 出願番号 特願2005-145601 (P2005-145601)                  (22) 出願日 平成17年5月18日 (2005.5.18)                  (65) 公開番号 特開2006-4415 (P2006-4415A)                  (43) 公開日 平成18年1月5日 (2006.1.5)                  審査請求日 平成20年5月19日 (2008.5.19)                  (31) 優先権主張番号 10/872, 119                  (32) 優先日 平成16年6月18日 (2004.6.18)                  (33) 優先権主張国 米国 (US)</p>	<p>(73) 特許権者 500046438                  マイクロソフト コーポレーション                  アメリカ合衆国 ワシントン州 9805                  2-6399 レッドモンド ワン マイ                  クロソフト ウェイ                  (74) 代理人 100077481                  弁理士 谷 義一                  (74) 代理人 100088915                  弁理士 阿部 和夫                  (72) 発明者 アラン エス. ゲラー                  アメリカ合衆国 98052 ワシントン                  州 レッドモンド ワン マイクロソフト                  ウェイ マイクロソフト コーポレーシ                  ョン内</p>
--	---

最終頁に続く

(54) 【発明の名称】 コンテキスト交換を用いたエニユメレーションのセッションのためのフレキシブルなコンテキスト管理

(57) 【特許請求の範囲】

【請求項1】

コンピュータ処理システムであるデータレシーバにネットワーク接続可能な、コンピュータ処理システムであるデータプロバイダを含む環境において、前記データプロバイダが、前記データプロバイダが管理する複数のデータアイテムを複数の別個の応答メッセージによって前記データレシーバに供給する方法であって、

前記データレシーバから、前記データレシーバに供給すべき前記複数のデータアイテムに対する第1の要求を受け取る動作と、

前記第1の要求を読み取って、前記データプロバイダから前記データレシーバへの第1のネットワーク応答のデータ構造内に供給すべき前記複数のデータアイテムのうちの第1の任意の数のデータアイテムを識別する動作と、

前記第1のネットワーク応答のデータ構造内に供給すべき前記複数のデータアイテムのうちの前記第1の任意の数のデータアイテムを識別する第1のコンテキスト情報を生成する動作と、

前記複数のデータアイテムのうちの前記第1の任意の数のデータアイテム及び前記第1のコンテキスト情報を含む前記第1のネットワーク応答を構築する動作と、

前記第1のネットワーク応答を前記データレシーバに送出する動作と

前記データレシーバから、前記データレシーバに供給すべき前記複数のデータアイテムに対する第2の要求を受け取る動作であって、前記第2の要求は、前記第1のコンテキスト情報を含む動作と、

10

20

前記第2の要求に含まれた前記第1のコンテキスト情報を読み取って、前記データプロバイダから前記データレシーバへの第2のネットワーク応答のデータ構造内に供給すべき前記複数のデータアイテムのうちの第2の任意の数のデータアイテムを識別する動作と、  
前記第2のネットワーク応答のデータ構造内に供給すべき前記複数のデータアイテムのうちの前記第2の任意の数のデータアイテムを識別する第2のコンテキスト情報を生成する動作と、

前記複数のデータアイテムの前記第2の任意の数のデータアイテム及び前記第2のコンテキスト情報を含む前記第2のネットワーク応答を構築する動作と、  
前記第2のネットワーク応答を前記データレシーバに送出する動作と  
 を含むことを特徴とする方法。

10

【請求項2】

前記データレシーバに供給すべき前記複数のデータアイテムに対する第1の要求を受け取る前記動作は、前記データレシーバに供給すべき前記複数のデータアイテムに対する内部要求にアクセスする動作を含むことを特徴とする請求項1に記載の方法。

【請求項3】

前記データレシーバに供給すべき前記複数のデータアイテムに対する第1の要求を受け取る前記動作は、前記データレシーバに供給すべき前記複数のデータアイテムに対するネットワーク要求を前記データレシーバから受け取る動作を含むことを特徴とする請求項1に記載の方法。

【請求項4】

前記データレシーバへの第1のネットワーク応答のデータ構造内に供給すべき前記複数のデータアイテムのうちの第1の任意の数のデータアイテムを識別する前記動作は、  
 前記第1のネットワーク応答のデータ構造内に供給すべきデータアイテムの最大数を前記第1の要求から読み取る動作と、

前記複数のデータアイテムのうちの第1の任意の数のデータアイテムは前記データアイテムの最大数に等しいことを決定する動作と  
 を含むことを特徴とする請求項1に記載の方法。

20

【請求項5】

前記データレシーバへの第1のネットワーク応答のデータ構造内に供給すべき前記複数のデータアイテムのうちの第1の任意の数のデータアイテムを識別する前記動作は、  
 前記第1のネットワーク応答のデータ構造内に供給される文字数の最大数を前記第1の要求から読み取る動作と、

前記複数のデータアイテムのうちの第1の任意の数のデータアイテムは前記文字数の最大数以下であることを決定する動作と  
 を含むことを特徴とする請求項1に記載の方法。

30

【請求項6】

前記データレシーバへの第1のネットワーク応答のデータ構造内に供給すべき前記複数のデータアイテムのうちの第1の任意の数のデータアイテムを識別する前記動作は、  
 前記第1のネットワーク応答が必要とする最大応答時間を前記第1の要求から読み取る動作と、

前記第1のネットワーク応答は、前記複数のデータアイテムのうちの第1の任意の数のデータアイテムを与える、前記最大応答時間内の戻りであることを決定する動作と  
 を含むことを特徴とする請求項1に記載の方法。

40

【請求項7】

前記第1及び第2の要求は、Simple Object Access Protocol (SOAP) エンベロープであることを特徴とする請求項1に記載の方法。

【請求項8】

前記第1及び第2の要求は、Remote Method Invocation (RMI) であることを特徴とする請求項1に記載の方法。

【請求項9】

50

前記複数のデータアイテムは、eXtensible Markup Language (XML)要素であることを特徴とする請求項1に記載の方法。

【請求項10】

前記第2のコンテキスト情報は、前記第1のコンテキスト情報と同一の範囲を有することを特徴とする請求項1に記載の方法。

【請求項11】

前記第2のコンテキスト情報は、前記第1のコンテキスト情報と異なる範囲を有することを特徴とする請求項1に記載の方法。

【請求項12】

請求項1乃至11のいずれかに記載の方法をコンピュータに実行させるコンピュータ実行可能命令を含むことを特徴とするコンピュータ・プログラム。

10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、一般には、コンピュータ処理技術に関し、より詳細には、どのコンピュータ処理システムが通信のためのコンテキスト情報を管理するかを制御しながら、多重応答で多重されたデータアイテムをネットワーク配布することに関する。

【背景技術】

【0002】

コンピュータ処理技術は、我々の仕事や遊びのやり方を変容させた。今やコンピュータ処理システムは多種多様な形態をとっており、デスクトップコンピュータ、ラップトップコンピュータ、タブレットPC、パーソナル・デジタル・アシスタント(PDA)、家庭用機器、等がある。コンピュータ処理システムの最も基本的な形態において、コンピュータ処理システムは、システムメモリおよび1つもしくは複数のプロセッサを備える。システムメモリのソフトウェアを、プロセッサによって実行することができ、結果としてコンピュータ処理システムのその他のハードウェアに所望の機能を実行させることができる。

20

【0003】

ネットワーク技術によって、コンピュータ処理システムは、たとえ広大な距離にわたっても通信を行うことができ、それによってコンピュータの機能を拡張することが可能となる。例えば、ネットワーク技術は、電子メール、ウェブブラウジング、ファイル転送、インスタントメッセージ、電子ホワイトボード、ネットワークコラボレーション、等ようなアプリケーションを可能にする。

30

【0004】

ネットワークアプリケーションの中には、大量の情報の転送を伴うものがある。例えば、サイズが数テラバイトのデータベースをダウンロードしたいと思う場合がある。たとえ高帯域幅ネットワーク上においても、このダウンロードには数日を要する可能性がある。かなり信頼性のあるネットワーク環境においてさえ、全期間にわたって、ネットワーク通信が維持できない可能性が高い。接続が失われると、データ転送全体を最初から再び始めなくてはならないことが多い。

【0005】

40

大規模なデータ転送の信頼性および効率性を向上させる一つの従来式の方法は、データを一度に一部分ずつ別個の電子メッセージにして転送する方法である。時には、この方法は単一の要求に回答して自動的に達成することができる。しかしながら、データレシーバがデータ転送プロセスをいくぶん制御できるようにするために、データレシーバは、転送データ全体の一部分を含む各々の電子メッセージに対して要求を送出することができる。例えば、ウェブサイトは、ウェブサイトに関連する大量のデータを有しており、ウェブページ、画像、音声ファイル、他のマルチメディア、スクリプト等がある。ウェブブラウザは、データアイテムの各々に対して別個の要求を送出することが多い。この技術は、「要求駆動のエニユメレーション技術(request-driven enumeration technology)」と呼ぶこととし、この技術においては、データアイテム

50



ロバイダは、第1の要求に対応する第1の応答において供給すべきデータアイテムの第1の部分を識別する。さらに、第1の部分の情報供給されたことを反映する、コンテキスト情報が識別される。次いで、データプロバイダは、どの程度の量のコンテキストをデータレシーバのコンピュータ処理システムに供給すべきかについて決定する。次いで、データプロバイダは第1の応答を構築し、この応答にはデータアイテムの第1の部分が含まれる。また、データプロバイダは、データプロバイダが適当であると考えただけの量のコンテキスト情報を、レシーバのコンピュータ処理システムに供給する。

【0011】

データレシーバは、この供給された第1のコンテキスト情報を、データアイテムの次の部分に対する第2の要求の中に含める。データプロバイダは、この供給された第1のコンテキスト情報を要求と要求との間で、保持する必要がない。代わりに、データプロバイダは、第2の要求から供給された第1のコンテキストを読み取る。このことは、データプロバイダが、データレシーバに送信するデータアイテムの第2の部分を識別するのに役立つ。次いで、データプロバイダは、第2の応答に含めるべき第2のコンテキスト情報を生成する。

10

【0012】

この第2に供給されるコンテキスト情報は、そのときの環境に応じて、先の第1に供給されるコンテキスト情報と同一もしくは異なる範囲を有することがある。例えば、第1に供給されるコンテキスト情報に、コンテキスト情報の全部が含まれることがある。データレシーバが第2の要求を迅速に送信する場合には、データレシーバが能動的に関係して、要求が迅速に連続してくるようと思われるので、次の応答には、比較的少ないコンテキスト情報を含めるか、またはまったく含める必要がなく、このことは、データプロバイダがコンテキスト情報を長くは維持する必要がないことを意味する。

20

【0013】

したがって、本発明の原理によって、データプロバイダは、そのときの環境に応じて、コンテキスト管理におけるデータプロバイダの役割を動的に調整することについて、大幅な柔軟性を得ることができる。本発明のさらなる特徴および利点は、以下の説明に記述しており、部分的には記述から明白であるか、もしくは本発明の実施から知ることができる。本発明の特徴および利点は、添付の本特許請求の範囲に具体的に指摘した機器および組合せによって、実現して獲得することができる。本発明のこれら、および他の特徴は、以下の本明細書の説明と添付の本特許請求の範囲からより完全に明白になるか、もしくは以下の記載に従って本発明を実施することによって知ることができる。

30

【発明を実施するための最良の形態】

【0014】

本発明の上記および他の利点および特徴を獲得する方法について説明するために、上記で簡単に記述した本発明のより具体的な記述を、添付の図面に示した本発明の具体的な実施形態を参照して行う。添付の図面は、本発明の代表的な実施形態のみを示しており、したがって、本発明の範囲を限定するものとは考えないことを理解した上で、本発明を、添付の図面を使用して、さらに具体的かつ詳細に記述し、説明する。

【0015】

本発明の原理は、要求されたデータアイテムの部分がデータレシーバのコンピュータ処理システムからの別個の要求に応答して供給される、要求駆動のエニューメレーションのセッションにおいて要求されたデータアイテムを供給する、データプロバイダのコンピュータ処理システムのためのメカニズムに関する。メカニズムによって、データプロバイダは、どの程度の量のメッセージ間のコンテキスト情報をデータプロバイダによって保持し、およびどの程度の量のコンテキスト情報をデータレシーバに供給するかを、制御することが可能となる。

40

【0016】

図面を参照すると、同一の参照番号は同一の要素を意味しており、適当なコンピュータ処理環境に実装したとして、本発明を示してある。以下の記述は、本発明の図示した実施

50

形態に基づくものであり、本明細書に明示的に記述していない代替実施形態に関して、本発明を限定するものと解釈すべきではない。

【0017】

以下の記述において、本発明は、特に断らない限り、1つまたは複数のコンピュータによって実行される、動作および演算の記号表現を参照して記述してある。したがって、動作および演算は、時にはコンピュータ実行されると記すこともあり、コンピュータの処理ユニットによる構造形式のデータを表す電気信号の操作が含まれることが理解されるであろう。コンピュータの処理ユニットによる操作によって、データが変換されるか、もしくはデータがコンピュータのメモリシステム内の場所に維持され、この操作によって、当業者によく理解されている方法により、コンピュータの演算が再構成されるか、もしくは他の方法で変更される。データが維持されるデータ構造とは、データのフォーマットによって定義される固有の特性を有するメモリの物理的な場所である。しかしながら、本発明は前述の文脈で記述してあるが、限定を意味するものではなく、当業者であれば、以下に記述する動作および演算をハードウェアに実装することもできることを認識するであろう。図1は、装置に有用な例示的コンピュータアーキテクチャの概略図を示す。

10

【0018】

説明の目的で、描いたアーキテクチャは適当な環境の一例を示すものにすぎず、本発明の使用もしくは機能の範囲についてのいかなる限定をも意味するものではない。また、コンピュータ処理システムは、図1に示す構成要素のいずれか、もしくは構成要素の組合せに関して、何らかの依存性もしくは必要条件を有するものとして解釈すべきではない。

20

【0019】

本発明は、他の多数の汎用もしくは専用のコンピュータ処理のまたは通信の、環境もしくは構成において使用可能である。本発明での使用に好適な、周知のコンピュータ処理システム、環境、および構成の例としては、限定はされないが、モバイル電話、ポケットコンピュータ、パーソナルコンピュータ、サーバ、マルチプロセッサシステム、マイクロプロセッサを基部とするシステム、ミニコンピュータ、メインフレームコンピュータ、および上記のシステムもしくは装置のいずれかを含む、分散コンピュータ処理環境がある。

【0020】

コンピュータ処理システムの最も基本的な構成において、コンピュータ処理システム100は、通常、少なくとも1つの処理ユニット102およびメモリ104を備える。メモリ104は、(RAMのような)揮発性、(ROM、フラッシュメモリ、等の)不揮発性、もしくは両者の組合せとすることができる。この最も基本的な構成は、図1に破線106で示してある。

30

【0021】

記憶媒体装置は、付加的な特徴および機能を有することができる。例えば、記憶媒体装置には、限定はされないが、PCMCIAカード、磁気および光ディスク、ならびに磁気テープを含む、追加の(取り外し可能および固定)記憶装置を含めることができる。追加の記憶装置を、図1に取り外し可能記憶装置108および固定記憶装置110によって示してある。コンピュータ記憶媒体には、コンピュータ読み取り可能な命令、データ構造、プログラムモジュール、もしくは他のデータなどの情報を記憶するために任意の方法もしくは技術で実装された、揮発性および不揮発性の、取り外し可能および固定の媒体を含めることができる。メモリ104、取り外し可能記憶装置108、および固定記憶装置110は、すべてコンピュータ記憶媒体の例である。コンピュータ記憶媒体としては、限定はされないが、RAM、ROM、EEPROM、フラッシュメモリ、他のメモリテクノロジー、CD-ROM、DVD(digital versatile disks)、他の光記憶装置、磁気カセット、磁気テープ、磁気ディスク記憶装置、他の磁気記憶装置、および所望の情報を記憶するのに使用可能であるとともに、コンピュータ処理システムによってアクセスが可能である他の任意の媒体がある。

40

【0022】

本明細書において使用する場合には、「モジュール」もしくは「構成要素」という用語

50

は、コンピュータ処理システム上で実行されるソフトウェアのオブジェクトまたはルーチンを意味する場合がある。本明細書に記述する、異なる構成要素、モジュール、エンジン、およびサービスは、コンピュータ処理システムで実行されるオブジェクトもしくはプロセス（例えば、別個のスレッドとして）として実装することができる。本明細書に記述するシステムおよび方法は、ソフトウェアに実装することが望ましいが、ソフトウェアおよびハードウェアへのもしくはハードウェアへの実装も可能であり企図され得る。

#### 【0023】

コンピュータ処理システム100には、例えば、ネットワーク120を介して、ホストが他のシステムおよび装置と通信することを可能にする、通信チャネル112を含めることもできる。通信チャネル112は、通信媒体の例である。通信媒体には、通常、コンピュータ読み取り可能な命令、データ構造、プログラムモジュール、または例えば搬送波もしくは他の搬送メカニズムなどの変調されたデータ信号中の他のデータが含まれるとともに、任意の情報伝達媒体が含まれる。例としてあって限定ではなく、通信媒体には、有線ネットワークおよび有線で直結接続などの有線媒体、ならびに音響、無線、赤外線、および他の無線媒体などの無線媒体が含まれる。本明細書で使用するコンピュータ読み取り可能な媒体という用語には、記憶媒体および通信媒体の両方が含まれる。

10

#### 【0024】

また、コンピュータ処理システム100は、キーボード、マウス、ペン、音声入力構成要素、タッチ入力装置などの、入力構成要素114を有することもできる。出力構成要素116としては、スクリーンディスプレイ、スピーカ、プリンタ、等、およびスクリーンディスプレイ、スピーカ、プリンタ、等を駆動するためのレンダリングモジュール（「アダプタ」と呼ばれることが多い）がある。コンピュータ処理システム100は、電源118を備える。構成要素はすべて、本技術分野において周知であり、ここでは詳しく述べる必要はない。

20

#### 【0025】

図2は、データレシーバのコンピュータ処理システム210にネットワーク接続する機能を有するデータプロバイダのコンピュータ処理システム220を含む、ネットワーク環境200を示す。本明細書の説明および本特許請求の範囲においては、「コンピュータ処理システム」とは、システムメモリおよび、システムメモリからの命令を実行する機能のある少なくとも1つのプロセッサを有する任意の装置またはシステムとして定義される。代替および追加として、コンピュータ処理システムは、全体がハードウェアに実装された場合でも、任意の論理処理の機能を有することができる。したがって、データレシーバのコンピュータ処理システム210およびデータプロバイダのコンピュータ処理システム220は、必須ではないが、コンピュータ処理システム100について上述したように、構築することができる。

30

#### 【0026】

また、コンピュータ処理システム210および220に対する、「データプロバイダ」および「データレシーバ」という用語の使用は、データの受信がコンピュータ処理システム210によってサポートされる唯一の機能であることも、データの供給がコンピュータ処理システム220によってサポートされる唯一の機能であることもまた示唆するものではない。実際に、コンピュータ処理システムは、非常に複雑な汎用のコンピュータ処理システムである場合もあるが、これは必要条件ではない。データレシーバのコンピュータ処理システム210は、本明細書では「データレシーバ」と呼ぶこともある。同様に、データプロバイダのコンピュータ処理システム220は、本明細書では「データプロバイダ」と呼ぶこともある。

40

#### 【0027】

データプロバイダ220は、データレシーバ210に供給すべきある数のデータアイテム221を管理する。また、データプロバイダ220は、データレシーバ210に供給しない他のデータアイテムも管理することができる。図に示した場合には、データプロバイダ220は、場合によっては縦方向の省略符号221Gで表すような他のデータアイテム

50

とともに、データアイテム 221A から 221F をデータレシーバのコンピュータ処理システムに供給することになっている。

【0028】

データプロバイダ 220 は、要求駆動のエニューメレーション技術によるセッションのとおりに、メッセージ交換 230 を使用することによって、データアイテム 221 をデータレシーバ 210 に供給する。具体的には、データアイテム 221 は、単一要求への単一の応答においては、データプロバイダに供給されない。代わりに、データアイテム 221 は、一度に一部分ずつ、データレシーバ 210 に供給され、各部分は一つの要求に応答して配布される。一実施形態においては、要求駆動のエニューメレーション技術によるセッションは、データレシーバ 210 がデータプロバイダ 220 に予備のエニューメレートの要求を 10 伝送することによって開始される。データプロバイダ 220 は、エニューメレートの応答をデータレシーバ 210 に返送することによって、セッションを確認することができる。以下にさらに詳細に記述する一つの具体例において、エニューメレートの要求および応答は、`eXtensible Markup Language (XML)` 文書を含む、`Simple Object Access Protocol (SOAP)` エンベロープの形式とすることができる。しかしながら、エニューメレートの要求および応答は、`Remote Method Invocation (RMI)` メッセージとすることもできる。

【0029】

図 2 に示すように、データレシーバ 210 は、第 1 の要求 231A を伝送する。応答として、データプロバイダ 220 は、第 1 の要求 231A に対する第 1 の応答 231B にお 20 いて、データアイテム 221A および 221B を供給する。このプロセスを、データレシーバ 210 がデータアイテム 221 のすべてを受け取るまで反復することができる。具体的には、データレシーバ 210 は、第 2 の要求 232A を伝送する。応答として、データプロバイダ 220 は、第 2 の要求 232A に対する第 2 の応答 232B において、データアイテム 221C および 221D を供給する。次いで、データレシーバ 210 は、第 3 の要求 233A を伝送する。応答として、データプロバイダ 220 は、第 3 の要求 233A に対する第 3 の応答 233B において、データアイテム 221E および 221F を供給する。このプロセスは、縦方向の省略符号 221G によって表すさらなるデータアイテムに対して、縦方向の省略符号 234 で表すように、継続することができる。

【0030】

図に示した場合には、少なくとも 6 つのデータアイテムが、各要求に対して一度に 2 つのデータアイテムずつ、データレシーバ 210 に供給されるものとして示してある。しかしながら、本発明の原理は、この実施形態に限定はされない。要求駆動のエニューメレーション技術による任意の単一のセッションを使用して、任意の数のデータアイテムを伝送することができる。さらに、データプロバイダ 220 は、任意の所与の要求に対する任意の所与の応答において、任意の数のデータアイテムを（もしくはデータアイテムの一部でも）伝送することができる。また、データレシーバ 210 へのデータアイテムのエニューメレーションは要求駆動であるので、データレシーバ 210 は、さらなる要求の伝送を停止するだけで、データ転送を停止することができる。

【0031】

図 3 は、データプロバイダが 2 つのコンピュータ処理システム間のコンテキスト管理の均衡に対する制御を保持することを可能にしながら、データプロバイダが、データアイテムをデータレシーバに供給する方法 300 のフローチャートを示す。同様に、図 4 は、この役割をするデータプロバイダをデータレシーバがサポートする方法 400 のフローチャートを示す。方法 300 および方法 400 は、ネットワーク環境 200 において実行できるので、図 3 および図 4 を、図 2 を頻繁に参照して説明する。

【0032】

図 3 を参照すると、データプロバイダ 220 は、データレシーバのコンピュータ処理システムに供給すべきデータアイテムに対する要求にアクセスする（動作 301）。この要求は、例えばファンクションコールによって行われる、データプロバイダ 220 における 50



内部要求とすることができる。あるいはまた、この要求は、他のコンピュータ処理システムからの要求を受け取ることによってアクセスすることができる。図2に図示した本実施形態においては、データプロバイダ220は、データレシーバ210からの第1の要求231Aを受け取ることによって、この要求にアクセスすることができる。一実施形態においては、データレシーバからの後続の要求のみならずこの要求は、各々、データレシーバ210が出す、「プル要求」である。ここに記述する本実施形態においては、「プル要求」は、SOAPエンベロープに含めることができ、結果として、XML文書の形式にすることができる。「プル要求」の一例を、以下により詳細に示す。「プル要求」は、RMIとすることもできる。

#### 【0033】

次いで、データプロバイダ220は、データプロバイダとデータレシーバとの間のコンテキスト管理における均衡を制御しながら、データレシーバ210にデータアイテムを供給する機能の、結果指向の動作を実行する(ステップ310)。図示した本実施形態においては、このことは、少なくとも動作311から314、および321から323を含むが、この結果を達成する任意の動作で十分である。

#### 【0034】

具体的には、データプロバイダ220は、少なくとも部分的には要求内の情報に基づいて、データレシーバのコンピュータ処理システムに供給すべきデータアイテムの収集を識別する(動作311)。データアイテムは、要求中に直接的に指定することができる。さらに、データアイテムは、データアイテムの収集(本明細書においてはデータアイテムの初期収集もしくは複数のデータアイテムとも呼ぶ)を指定することによって間接的に指定することもできる。この要求には、データアイテムの最終収集を生成するためにデータアイテムの初期収集に適用するフィルタを含めることができる。一実施形態においては、識別しようとするアイテムの収集は、要求中のセッション識別子を読み取ることによって実行される。セッション識別子は、先のエニューメレートの実要求と関係づけることができ、この要求においては、データアイテムが明白に識別されている。この先のエニューメレートの実要求には、データアイテムを識別するためのフィルタを含めておくこともできる。

#### 【0035】

次いで、データプロバイダ220は、データアイテムの収集の第1の部分がデータレシーバのコンピュータ処理システムに供給されることを反映する第1のコンテキストを識別する(動作312)。データアイテムの収集が、例えば、XML要素である場合には、コンテキストには、先に供給された最後のXML要素のもしくは供給しようとする次のXML要素の識別子を、またはデータアイテムの収集のどの部分がすでにデータレシーバ210に供給されたかおよびデータアイテムのどの部分をさらに供給すべきかを示すのに有用な他の任意の情報を、含めることができる。

#### 【0036】

次いで、データプロバイダ220は、どの程度の量の第1のコンテキストをデータレシーバ210に供給すべきかを決定する(動作313)。このことには、動作312において識別された全部の第1のコンテキストをひとつも含めないか、一部もしくは全部を含めることができる。ここで留意すべきことは、コンテキスト情報をデータレシーバ210に供給できるとしても、データプロバイダは、コンテキストを後に検証するために、供給されたコンテキストの一部もしくは全部を重複して保持することができることである。どの程度の量のコンテキストをデータレシーバ210に供給するかの決定は、任意の決定基準のとおり実施することができる。関係する決定基準としては、例えば、データプロバイダ220もしくはデータプロバイダ220の利用可能なメモリの、現在の処理量および利用可能な処理能力、データレシーバ210もしくはデータレシーバ210の利用可能なメモリの、現在の処理量および利用可能な処理能力、データレシーバ210とデータプロバイダ220との間の利用可能なネットワーク帯域幅、ネットワークレシーバもしくはネットワークレシーバに関連するユーザの重要性、等がある。次いで、データプロバイダ220は、データレシーバ210に適宜、第1のコンテキストを供給する(動作314)。

10

20

30

40

50

## 【 0 0 3 7 】

さらに、データプロバイダ 2 2 0 は、データレシーバへの第 1 のネットワーク応答に供給すべきデータアイテムの収集の第 1 の部分（全体よりも少ない）を識別する（動作 3 2 1）。「第 1 の」という用語は、必ずしも、2 つのコンピュータ処理システム間に先のエニユメレート要求もしくは応答がなかったことを意味するものではない。実際に、以下に示す一実施形態においては、プル演算を使用してデータアイテムを一度に一部分ずつ取り出すことのできるセッション中のポイントに達するために、エニユメレートの要求およびエニユメレートの応答の形式の全体的な要求 / 応答の交換がある。

## 【 0 0 3 8 】

次いで、データプロバイダは、第 1 の部分を含む第 1 のネットワーク応答を構築する（動作 3 2 2）。応答の一例を、‘プル応答’の形式で以下に記述する。‘プル応答’の形式は、特有の XML 構造を有する SOAP エンベロープの形式をとる。次いで、第 1 のネットワーク応答は、データレシーバ 2 1 0 に送出される（動作 3 2 3）。一実施形態においては、データプロバイダ 2 2 0 は、第 1 のネットワーク応答自体に、データレシーバ 2 1 0 に供給される第 1 のコンテキストを含める。図 2 においては、第 1 のネットワーク応答は、ネットワーク応答 2 3 1 B として示してある。

## 【 0 0 3 9 】

エニユメレーションのセッションは要求駆動であるので、データプロバイダ 2 2 0 は、セッションをさらに進行させる前に、次の要求を待ち受ける（動作 3 3 1）。図 4 を参照すると、データレシーバ 2 1 0 は、この段階において、セッションが続く前に、さらなる動作を実行することになっている。図 4 の方法 4 0 0 は、第 1 のネットワーク応答であっても、そうでなくても、エニユメレーションのセッション内でネットワーク応答を受け取るときに、データレシーバ 2 1 0 がとる動作を記述している。具体的には、データレシーバ 2 1 0 はネットワーク応答を受け取り（動作 4 0 1）、ネットワーク応答中に供給されるコンテキストを含む次の要求を生成し（動作 4 0 2）、次いで、次の要求をデータプロバイダ 2 2 0 に伝送する（動作 4 0 3）。コンテキストは、先のネットワーク応答から次のネットワーク応答へ直接移すことができるが、これは必ずしもそうでなくてもよい。おそらく、先のネットワーク応答に供給されたコンテキストの一部分だけが、次の要求に供給される。さらに、次の要求に供給されるコンテキスト情報は、先のネットワーク応答においてコンテキスト情報が符号化された方法と比較して、異なる符号化を施すこともできる。さらに、次の要求に供給されるコンテキストは、データレシーバによってさらに処理しておく（例えば、暗号化および / もしくは圧縮しておく）ことができる。

## 【 0 0 4 0 】

データアイテムの収集の残部の任意の部分に対する第 2 の要求もしくは任意の後続の要求を受け取ると（動作 3 4 1）、データプロバイダ 2 2 0 は、第 2 の要求から先に供給されたコンテキストを読み取り（動作 3 4 2）、複数のデータアイテムの第 2 の部分がデータレシーバのコンピュータ処理システムに供給されたことを反映する第 2 のコンテキストを識別して（動作 3 5 1）、どの程度の量の次のコンテキストをデータレシーバのコンピュータ処理システムに供給すべきかを決定する（動作 3 5 2）。

## 【 0 0 4 1 】

データレシーバ 2 1 0 に供給されるコンテキストの部分は、先にデータレシーバ 2 1 0 に供給されたコンテキストと同じ範囲とすることができるが、これは必ずしもそうでなくてもよい。データプロバイダ 2 2 0 は、状況によって保証される場合には、先に供給したものと大きく異なるコンテキストを供給することができる。例えば、データプロバイダ 2 2 0 が、データレシーバ 2 1 0 が要求駆動のエニユメレーションに能動的に関係して、先の応答を受け取るといつでも迅速に要求を発行していることを発見すると仮定する。この場合には、セッションは迅速に完了して、かなり長い時間にわたってメモリ資源を拘束することがないことが予測されるので、データプロバイダ 2 2 0 が、データレシーバ 2 1 0 にまったく供給することなく、全体コンテキストを維持するコストは小さくすることができる。次いで、データプロバイダ 2 2 0 は、データプロバイダ 2 2 0 がデータレシーバ 2

10

20

30

40

50

10 に送ろうと考える次のコンテキストの部分を供給する（動作353）。

【0042】

データプロバイダ220は、また、要求から読み取った先のコンテキストを使用して、データレシーバに供給すべきデータアイテムの収集の次の部分を識別する（動作361）。次いで、データプロバイダ220は、データアイテムの収集の次の部分を含む、次のネットワーク応答を構築する（動作362）。次いで、次のネットワーク応答が、データレシーバ210に送出される（動作363）。この段階において、供給すべきデータアイテムの収集のデータアイテムがもうない（決定ブロック370においてNoである）場合には、セッションはこの時点で終了する。しかしながら、さらにデータアイテムがある（決定ブロック370においてYesである）場合には、セッションは動作331において次の要求を待ち受けることによって継続する。

10

【0043】

本発明の一般的な原理について記述したが、次に本発明の具体的な一実施形態をより詳細に記述する。この具体的な実施形態においては、セッションにおける要求および応答の各々は、XML文書を含むSOAPエンベロープとして構築される。

【0044】

上記に簡潔に述べたエニューメレータの要求の形式を以下に示す。前述のように、データレシーバ210は、要求駆動のエニューメレーションのセッションを開始するために、データプロバイダ220にエニューメレータの要求を伝送する。分かりやすくするために行番号を付与してある。

20

【0045】

【表 1】

1) <s:Envelope ...>	
2) <s:Header ...>	
3) <wsa:Action>	
4) http://schemas.xmlsoap.org/ws/2004/04/enumeration/Enumerate	
5) </wsa:Action>	10
6) <wsa:MessageID>xs:anyURI</wsa:MessageID>	
7) <wsa:To>xs:anyURI</wsa:To>	
8) ...	
9) </s:Header>	
10) <s:Body ...>	20
11) <wsen:Enumerate ...>	
12) <wsen:Filter Dialect="xs:anyURI"?> xs:any </wsen:Filter> ?	
13) ...	
14) </wsen:Enumerate>	
15) </s:Body>	
16) </s:Envelope>	30

## 【 0 0 4 6 】

行 1 から行 1 6 は、SOAP エンベロープ全体を定義する XML 要素を表す。行 2 から行 9 は、SOAP エンベロープのためのヘッダー要素を表す。行 3 から行 5 は、アクションがエニユメレート（Enumerate）の要求であることを定義する（行 4 の「Enumerate」を参照）アクションの XML 要素を表す。行 6 は、セッション識別子を識別する。行 7 は、データプロバイダのアドレスを表す。行 8 は、ヘッダーの XML 要素内に追加の XML 要素がある場合があることを表す。行 10 から行 15 は、SOAP ボディの XML 要素を表す。行 11 から行 14 は、エニユメレーションのセッションに必要な情報を定義するエニユメレートの XML 要素を表す。例えば、行 12 は、フィルタ要素を識別する。このフィルタ要素は、フィルタダイアレクトの属性を含み、これはフィルタを表現するためのクエリ言語（例えば、SQL、XPath、もしくは他の URI）を定義する。さらに、フィルタ要素は「任意の」属性を含み、これは、指定のクエリ言語を使用してフィルタを表現する。本明細書の説明および本特許請求の範囲において、「クエリ言語」とは、最初の複数データアイテムをひとつも含まないか、一部もしくは全部を含むフィルタリングされたデータアイテムを生成するために、最初の複数データアイテムに対するフィルタを表現することに使用することができる、任意の組のセマンティックな規則である。

## 【 0 0 4 7 】

以下は、上記の形式に続く、エニユメレートの要求の一例であり、分かりやすくするために行番号を付与してある。

【 0 0 4 8 】

【 表 2 】

(01) <s:Envelope xmlns:S='http://www.w3.org/2003/05/soap-envelope'	
(02) xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/03/addressing'	
(03) xmlns:wxf='http://schemas.xmlsoap.org/ws/2004/04/enumeration'	
(04) <s:Header>	
(05) <wsa:Action>	10
(06) http://schemas.xmlsoap.org/ws/2004/04/enumeration/Enumerate	
(07) </wsa:Action>	
(08) <wsa:MessageID>	
(09) uuid:e7c5726b-de29-4313-b4d4-b3425b200839	
(10) </wsa:MessageID>	20
(11) <wsa:To>http://www.example.com/relayAgent/enum19</wsa:To>	
(12) </s:Header>	
(13) <s:Body>	
(14) <wsen:Enumerate/>	
(15) </s:Body>	30
(16) </s:Envelope>	

【 0 0 4 9 】

行(05~07)は、このメッセージがエニユメレート(Enumerate)の要求であり、データ源がエニユメレートの応答メッセージで応答することが期待されることを示す。w s e : F i l t e r 要素は存在せず、結果として得られるエニユマレーションのコンテキストは、すべての利用可能な要素を返すことが期待される。

【 0 0 5 0 】

データプロバイダ 2 2 0 は、要求駆動のエニユメレーションのセッションを完全に初期化するために、エニユメレートの要求に対して、エニユメレートの応答で応答するとともに、以下の形式に準拠することが可能であり、ここで分かりやすくするために行番号を付与してある。

【 0 0 5 1 】

40

## 【表 3】

- |  |    |
|--|----|
| 1) <s:Envelope ...>  |    |
| 2) <s:Header ...>  |    |
| 3) <wsa:Action>  |    |
| 4) http://schemas.xmlsoap.org/ws/2004/04/enumeration/EnumerateResponse |    |
| 5) </wsa:Action>   | 10 |
| 6) ...   |    |
| 7) </s:Header>   |    |
| 8) <s:Body ...>  |    |
| 9) <wsen:EnumerateResponse ...>  |    |
| 10) <wsen:EnumerationContext>...</wsen:EnumerationContext>             | 20 |
| 11) ...  |    |
| 12) </wsen:EnumerateResponse>  |    |
| 13) </s:Body>  |    |
| 14) </s:Envelope>  |    |

## 【0052】

行1から行14は、SOAPエンベロープ全体を定義するXML要素を表す。行2から行7は、SOAPエンベロープのヘッダー要素を表す。行3から行5は、アクションがエニユメレーションの応答であることを定義するアクションのXML要素を表す（行4の「EnumerateResponse」を参照）。行6は、行2から行7にわたるヘッダーのXML要素内に追加のXML要素がある可能性があることを表している。行8から行13は、SOAPボディのXML要素を表す。行9から行12は、エニユメレーションのセッションの初期化を完了するために使用する情報を定義するエニユメレーションの応答のXML要素を表す。例えば、行10はエニユメレーションのコンテキストのXML要素であり、これには、セッションにおける将来の任意の「プル要求」のためのコンテキストを識別するのに十分な情報を含む最初のコンテキストを含めることができる。EnumerateContext要素は、新規のエニユメレーションのコンテキストのXML表現を含む。データレシーバ210は、「プル応答」メッセージ（以下にさらに記述する）がこのエニユメレーションのコンテキストを更新するまで、かつ更新しない限り、このXMLデータをこのエニユメレーションのコンテキストのための「プル要求」中に渡す。行11は、このエニユメレーションのコンテキストのXML要素内に、さらにXML要素が含まれる可能性があることを表す。

## 【0053】

以下に示すのは、上記の形式に続く仮定のエニユメレーションの応答である。

## 【0054】

## 【表 4】

(01)	<s:Envelope xmlns:S='http://www.w3.org/2003/05/soap-envelope'	
(02)	xmlns:wxf='http://schemas.xmlsoap.org/ws/2004/04/enumeration'	
(03)	xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/03/addressing'	
(04)	>	
(05)	<s:Header>	10
(06)	<wsa:Action>	
(07)	http://schemas.xmlsoap.org/ws/2004/04/enumeration/EnumerateResponse	
(08)	</wsa:Action>	
(09)	<wsa:RelatesTo>	
(10)	uuid:e7c5726b-de29-4313-b4d4-b3425b200839	
(11)	</wsa:RelatesTo>	20
(12)	</s:Header>	
(13)	<s:Body>	
(14)	<wsen:EnumerateResponse>	
(15)	<wsen:EnumerationContext>	
(16)	123	
(17)	</wsen:EnumerationContext>	30
(18)	</wsen:EnumerateResponse>	
(19)	</s:Body>	
(20)	</s:Envelope>	

## 【 0 0 5 5 】

行 0 6 から行 0 8 は、メッセージがエニユメレートの応答メッセージであることを示す。行 1 5 から行 1 7 は、下記の「プル演算」をサポートするエニユメレーションコンテキストの XML 表現である。

40

## 【 0 0 5 6 】

この「プル演算」は、データレシーバ 2 1 0 が「プル演算」のメッセージをデータプロバイダ 2 2 0 に送ることによって開始することができる。以下は、「プル要求」のための例示的形式であり、分かりやすくするために行番号を付与してある。

## 【 0 0 5 7 】

## 【表5】

- 1) <s:Envelope ...>
- 2) <s:Header ...>
- 3) <wsa:Action>
- 4) http://schemas.xmlsoap.org/ws/2004/04/enumeration/Pull
- 5) </wsa:Action> 10
- 6) <wsa:MessageID>xs:anyURI</wsa:MessageID>
- 7) <wsa:ReplyTo>wsa:EndpointReference</wsa:ReplyTo>
- 8) <wsa:To>xs:anyURI</wsa:To>
- 9) ...
- 10) </s:Header> 20
- 11) <s:Body ...>
- 12) <wsen:Pull ...>
- 13) <wsen:EnumerationContext>...</wsen:EnumerationContext>
- 14) <wsen:MaxTime>xsd:duration</wsen:MaxTime> ?
- 15) <wsen:MaxElements>xsd:long</wsen:MaxElements> ?
- 16) <wsen:MaxCharacters>xsd:long</wsen:MaxCharacters> ? 30
- 17) ...
- 18) </wsen:Pull>
- 19) </s:Body>
- 20) </s:Envelope>

## 【0058】

40

行1から行20は、SOAP構造の「プル要求」である、XML要素全体を定義している。行2から行10は、ヘッダーのXML要素を定義している。行3から行5は、アクション要素であり、この要素においては、アクションはプル要求として定義される（行4の「Pull」を参照のこと）。行6はメッセージ識別子のXML要素である。この要素は、「プル要求」をセッションと関係づける識別子を含む。行7は、「返信先」のXML要素であり、これは「プル要求」への応答のためのアドレスを供給する。行8は、データプロバイダ220のアドレスを指定する、アドレスのXML要素である。行9は、ヘッダーのXML要素に含まれるさらなるXML要素がある可能性があることを示す。

## 【0059】

行11から行19は、ボディのXML要素を表す。具体的には、行12から行18は、

50



‘プル’のXML要素を表し、これは‘プル要求’に使用される情報を含む。例えば、行13はエnumレーションのコンテキスト要素であり、この要素においては、先に供給されたエnumレーションの要素が挿入される。このエnumレーションのコンテキストは、セッションのコンテキストが何かを識別するために、データプロバイダ220によって使用される。

【0060】

行14は、最大時間のXML要素であり、これはデータプロバイダ220が‘プル要求’を受け取った後にデータプロバイダ220が関連する‘プル応答’を供給する前に許容すべき最大時間を反映している。関連する‘プル応答’を送る前に、データプロバイダ220は、この最大時間を超過しないことを保証にすることになる。

10

【0061】

行15は、最大要素数のXML要素であり、これはプロバイダ220が関連する‘プル応答’において許容すべきXML要素の最大数を反映する。関連する‘プル応答’を送る前に、データプロバイダ220は、プル応答がこの最大数より多いXML要素を含まないことを保証にすることになる。

【0062】

行16は、最大文字数のXML要素であり、これはプロバイダ220が関連するプル応答内に許容すべき文字の最大数である。関連する‘プル応答’を送る前に、データプロバイダ220は、‘プル応答’がこの最大文字数より多くを含まないことを保証にすることになる。

20

【0063】

以下は、‘プル要求’の例であり、分かりやすくするために行番号を付与してある。

【0064】

【表 6】

```

(01) <s:Envelope xmlns:S='http://www.w3.org/2003/05/soap-envelope'
(02)   xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/03/addressing'
(03)   xmlns:wsen="http://schemas.xmlsoap.org/ws/2004/04/enumeration">
(04) <s:Header>
(05)   <wsa:Action>
(06)     http://schemas.xmlsoap.org/ws/2004/04/enumeration/Pull
(07)   </wsa:Action>
(08)   <wsa:MessageID>
(09)     uuid:e7c5726b-de29-4313-b4d4-b3425b200839
(10)   </wsa:MessageID>
(11)   <wsa:To>http://www.example.com/relayAgent</wsa:To>
(12) </s:Header>
(13) <s:Body>
(14)   <wsen:Pull>
(15)     <wsen:EnumerationContext>123</wsen:EnumerationContext>
(16)     <wsen:MaxTime>P30S</wsen:MaxTime>
(17)     <wsen:MaxElements>10</wsen:MaxElements>
(18)   </wsen:Pull>
(19) </s:Body>
(20) </s:Envelope>

```

## 【0065】

表 2 の行 ( 0 5 ) から行 ( 0 7 ) は、このメッセージは 'プル要求' であり、データ源は、'プル応答' メッセージで応答すると期待される。行 ( 1 6 ) は、応答メッセージは、'プル要求' メッセージを受け取って後、30 秒以下で生成しなくてはならないことを示している。行 ( 1 7 ) は、10 個以下の要素が、'プル応答' メッセージのボディ内に戻されるべきことを示す。

## 【0066】

'プル要求' を受け取り、処理するとき、データレシーバ 210 は、'プル応答' を生成することができる。以下は、'プル応答' の例示的な形式であり、分かりやすくするために行番号を付与してある。

## 【0067】

10

20

30

40

50

【表 7】

1) <s:Envelope ...>	
2) <s:Header ...>	
3) <wsa:Action>	
4) http://schemas.xmlsoap.org/ws/2004/04/enumeration/PullResponse	
5) </wsa:Action>	10
6) ...	
7) </s:Header>	
8) <s:Body ...>	
9) <wsen:PullResponse ...>	
10) <wsen:EnumerationContext>...</wsen:EnumerationContext> ?	20
11) <wsen:Items> ?	
12) <xs:any> enumeration-specific element </xs:any> +	
13) </wsen:Items>	
14) <wsen:EndOfSequence/> ?	
15) ...	
16) </wsen:PullResponse>	30
17) </s:Body>	
18) </s:Envelope>	

## 【 0 0 6 8 】

行 1 から行 1 8 は、SOAP エンベロープの XML 要素全体を表す。行 2 から行 7 は、SOAP エンベロープのためのヘッダー要素を表す。行 3 から行 5 は、アクションがプル応答であることを定義するアクションの XML 要素を表す（行 4 の「PullResponse」を参照のこと）。行 6 は、行 2 から行 7 をまたぐヘッダーの XML 要素内に追加の XML 要素がある可能性があることを表す。 40

## 【 0 0 6 9 】

行 8 から行 1 7 は、ボディ XML の要素を表す。行 9 から行 1 6 は、応答の部分である要素を含む、プル応答の XML 要素を表す。行 1 0 は、次の要求に対するエnumレーションのコンテキストを含むエnumレーションのコンテキストの XML 要素である。行 1 1 から行 1 3 は、応答内に供給すべきデータアイテムの全部分を含む、アイテムの XML 要素を表す。行 1 2 は、実際にデータアイテムを包含する下層の XML 要素を表す。行 1 4 は、シーケンスの終了の XML 要素であり、これは、すべてのデータアイテムがデータレシーバ 2 1 0 に供給されたときに、設定することができる。このようにして、データレシーバ 2 1 0 に、エnumレーションのセッションの終わりを知らせることができる。ポ 50

ディのXML要素は、1つもしくは複数のデータアイテムがスキップされたために、供給されるデータアイテムが連続ではなくなるときを指定する、別の任意選択のXML要素を有することができる。このXML要素は、なぜデータアイテムがスキップされたかも指定する。例えば、データアイテムはそのときに使用中であり、したがってアクセスが制限されたためである。さらに、データアイテムは、'プル要求'によって課せられる最大時間、最大要素数、もしくは最大文字数の制限のために、応答の中にすべて含めることができなかったためである。

【0070】

以下は、'プル応答'の例であり、分かりやすくするために行番号を付与してある。

【0071】

## 【表 8】

(01)	<s:Envelope xmlns:S='http://www.w3.org/2003/05/soap-envelope'	
(02)	xmlns:wsen='http://schemas.xmlsoap.org/ws/2004/04/enumeration'	
(03)	xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/03/addressing'	
(04)	>	
(05)	<s:Header>	10
(06)	<wsa:Action>	
(07)	http://schemas.xmlsoap.org/ws/2004/04/enumeration/PullResponse	
(08)	</wsa:Action>	
(09)	<wsa:RelatesTo>	
(10)	uuid:e7c5726b-de29-4313-b4d4-b3425b200839	
(11)	</wsa:RelatesTo>	20
(12)	</s:Header>	
(13)	<s:Body>	
(14)	<wsen:PullResponse>	
(15)	<wsen:Items xmlns:xx='http://fabrikam123.com/schema/log'>	
(16)	<xx:LogEntry id="1">System booted</xx:LogEntry>	30
(17)	<xx:LogEntry id="2">AppX started</xx:LogEntry>	
(18)	<xx:LogEntry id="3">John Smith logged on</xx:LogEntry>	
(19)	<xx:LogEntry id="4">AppY started</xx:LogEntry>	
(20)	<xx:LogEntry id="5">AppX crashed</xx:LogEntry>	
(21)	</wsen:Items>	
(22)	<wsen:EndOfSequence/>	40
(23)	</wsen:PullResponse>	
(24)	</s:Body>	
(25)	</s:Envelope>	

## 【0072】

表3の行(06)から(08)は、このメッセージが‘プル応答’メッセージであることを示している。行(15)から(19)は、この‘プル要求’によって返される5つの要素である。wsen:EndOfSequence(行20)の要素の存在は、それ以

上、利用可能な要素はないこと、およびエニュメレーションのコンテキストはすでに無効であることを示す。

【 0 0 7 3 】

一つの最適化策として、開放演算を実施して、データアイテムが供給される前に、データレシーバ 2 1 0 がセッションを終了することを可能にすることができる。‘開放演算’は、データレシーバ 2 1 0 が‘開放要求’をデータプロバイダ 2 2 0 に送ることによって開始される。例示の開放メッセージは、以下の形式を有し、分かりやすくするために行番号を付与してある。

【 0 0 7 4 】

【表 9】

- |  |    |
|--|----|
| 1) <s:Envelope ...>  |    |
| 2) <s:Header ...>  |    |
| 3) <wsa:Action>  |    |
| 4) http://schemas.xmlsoap.org/ws/2004/04/enumeration/Release |    |
| 5) </wsa:Action>   | 20 |
| 6) <wsa:MessageID>xs:anyURI</wsa:MessageID>                  |    |
| 7) <wsa:ReplyTo>wsa:EndpointReference</wsa:ReplyTo>          |    |
| 8) <wsa:To>xs:anyURI</wsa:To>                                |    |
| 9) ...   |    |
| 10) </s:Header>  |    |
| 11) <s:Body ...>   | 30 |
| 12) <wsen:Release ...>                                       |    |
| 13) <wsen:EnumerationContext>...</wsen:EnumerationContext>   |    |
| 14) ...  |    |
| 15) </wsen:Release>  |    |
| 16) </s:Body>  | 40 |
| 17) </s:Envelope>  |    |

【 0 0 7 5 】

行 4 は、メッセージを‘開放要求’であると識別し、行 1 2 から行 1 5 は、開放されているエニュメレーションのセッションに対するエニュメレーションのコンテキストを識別する。

【 0 0 7 6 】

‘開放要求’の処理が成功すると、データプロバイダ 2 2 0 は、‘開放応答’を供給することが可能となり、‘開放応答’は以下の例示的な形式を有し、分かりやすくするために行番号を付与してある。

10

20

30

40

50

【 0 0 7 7 】

【 表 1 0 】

- 1) <s:Envelope ...>
- 2) <s:Header ...>
- 3) <wsa:Action>
- 4) <http://schemas.xmlsoap.org/ws/2004/04/enumeration/ReleaseResponse>
- 5) </wsa:Action>
- 6) ...
- 7) </s:Header>
- 8) <s:Body />
9. </s:Envelope>

10

20

【 0 0 7 8 】

行 4 は、メッセージを ' 開放応答 ' として識別する。

【 0 0 7 9 】

したがって、本発明の原理によって、データプロバイダは、そのときの環境に応じて、コンテキスト管理におけるデータプロバイダの役割を動的に調整することにおいて、広く柔軟性を有することが可能となる。

【 0 0 8 0 】

本発明は、本発明の趣旨もしくは本質的な特徴から逸脱することなく、他の特有の形態に具現することができる。記述した実施形態は、すべての点で、ただ説明的であって限定的ではないとしてだけと考えるべきである。したがって、本発明の範囲は、前述の説明ではなく、添付の本特許請求の範囲によって示されるものである。本特許請求の範囲の意味および同等の範囲に入るすべての変更は、本特許請求の範囲に含めるものである。

30

【 図面の簡単な説明 】

【 0 0 8 1 】

【 図 1 】本発明の特徴を実現することのできる、適当なコンピュータ処理システムを示す図である。

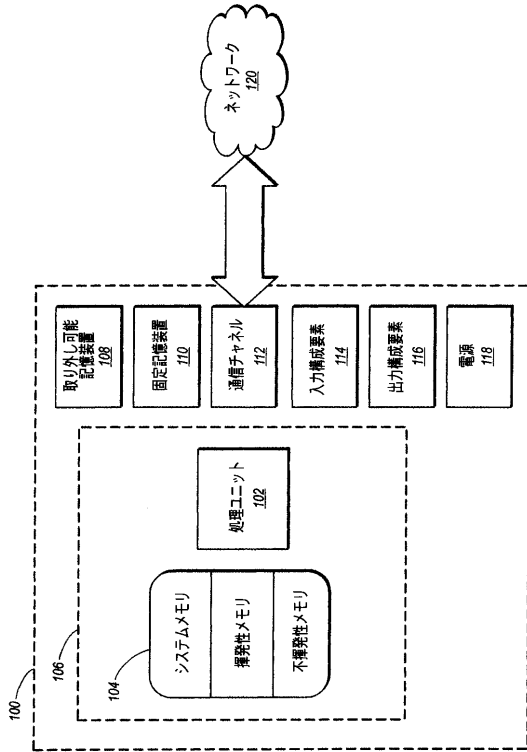
【 図 2 】データプロバイダのコンピュータ処理システムが、本発明の原理のとおり要求駆動のエニューメレーションのメッセージ交換を使用することによって、データレシーバのコンピュータ処理システムに、いくつかのデータアイテムを供給する、ネットワーク環境を示す図である。

40

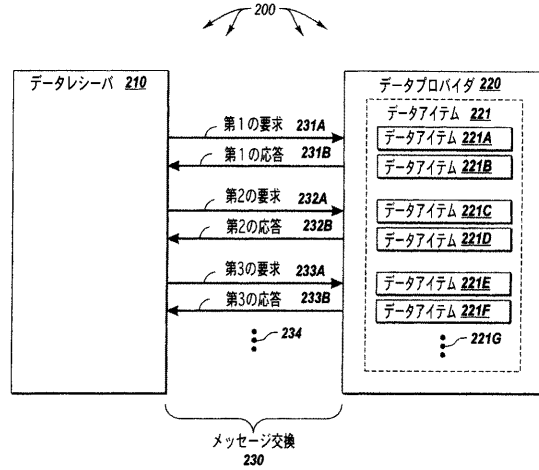
【 図 3 】本発明の原理のとおり、データプロバイダがコンテキスト管理の均衡を制御することを可能にしながら、データプロバイダのコンピュータ処理システムが、いくつかのデータアイテムをデータレシーバのコンピュータ処理システムに供給する方法のフロー図である。

【 図 4 】データレシーバが、データレシーバのコンピュータ処理システムと協働して、2つのコンピュータ処理システム間のコンテキスト管理の責任の均衡を制御する、データプロバイダのコンピュータ処理システムを支援する方法のフロー図である。

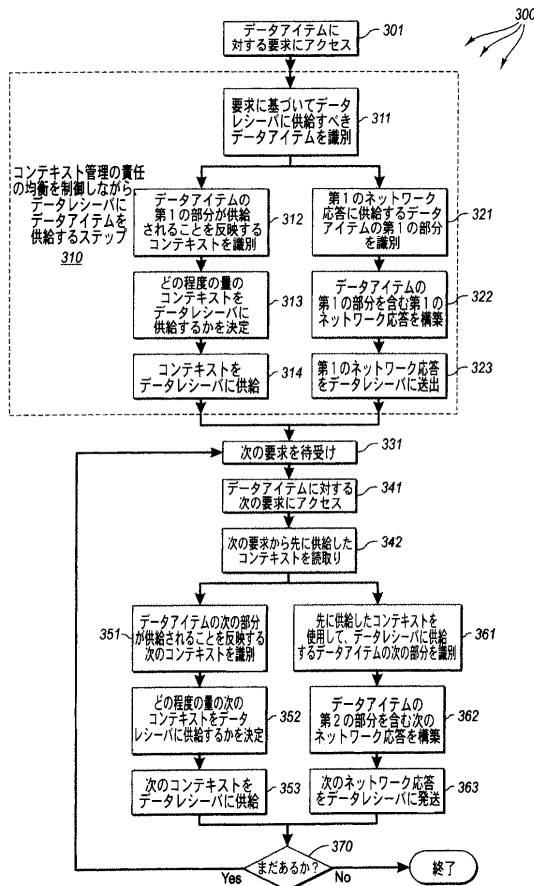
【図1】



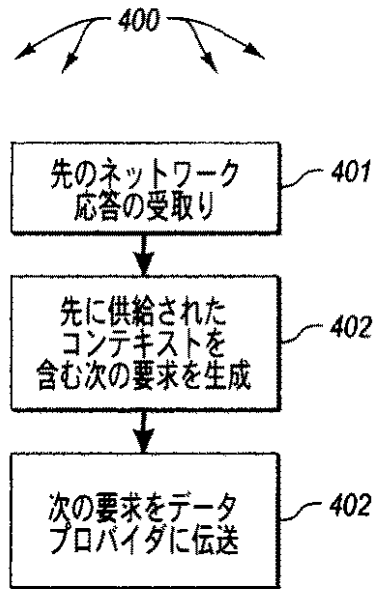
【図2】



【図3】



【図4】





---

フロントページの続き

- (72)発明者 ドナルド エフ. ボックス  
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ  
イクロソフト コーポレーション内
- (72)発明者 ルイス フェリペ カブレラ  
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ  
イクロソフト コーポレーション内
- (72)発明者 レイモンド ダブリュー. マックコラム  
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ  
イクロソフト コーポレーション内

審査官 田上 隆一

(56)参考文献 国際公開第2004/023341(WO, A1)

(58)調査した分野(Int.Cl., DB名)  
G06F 13/00