

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4441249号

(P4441249)

(45) 発行日 平成22年3月31日(2010.3.31)

(24) 登録日 平成22年1月15日(2010.1.15)

(51) Int.Cl.

F I

G 0 6 F 13/00 (2006.01)

G 0 6 F 13/00 3 5 3 B

請求項の数 3 (全 24 頁)

(21) 出願番号	特願2003-413805 (P2003-413805)	(73) 特許権者	596170170
(22) 出願日	平成15年12月11日(2003.12.11)		ゼロックス コーポレーション
(65) 公開番号	特開2004-192652 (P2004-192652A)		XEROX CORPORATION
(43) 公開日	平成16年7月8日(2004.7.8)		アメリカ合衆国、コネチカット州 068
審査請求日	平成18年12月8日(2006.12.8)		56、ノーウォーク、ビーオーボックス
(31) 優先権主張番号	10/317,580		4505、グローバー・アヴェニュー 4
(32) 優先日	平成14年12月12日(2002.12.12)		5
(33) 優先権主張国	米国 (US)	(74) 代理人	100075258
			弁理士 吉田 研二
		(74) 代理人	100096976
			弁理士 石田 純
		(72) 発明者	マーク ダブリュ ニューマン
			アメリカ合衆国 カリフォルニア サンフ
			ランシスコ ボカナ ストリート 164

最終頁に続く

(54) 【発明の名称】 ネットワークコンピューティング環境においてコンテキストプロパティメタデータを利用するための装置

(57) 【特許請求の範囲】

【請求項 1】

コンテキスト履歴を累積するための装置であって、
前記装置のコンテキストと、他のコンポーネントに関連づけられた相互作用の情報と、
を有するコンテキスト履歴と、

少なくとも1つの汎用インタフェースと、

リクエストコンポーネントの相互作用の際に、前記リクエストコンポーネントから、前記リクエストコンポーネントのコンテキストと、動作の実行を要求するリクエストと、を受信する受信メカニズムと、

前記動作による前記装置のコンテキストの変更を検出するよう構成された検出メカニズムと、

前記装置のコンテキストの変更を、前記コンテキスト履歴中の前記装置のコンテキストと結合するよう構成されている受信結合メカニズムと、

前記リクエストコンポーネントの相互作用と、前記リクエストコンポーネントのコンテキストとを、前記コンテキスト履歴中の前記相互作用の情報と結合するように構成されたリクエスト結合メカニズムと、

前記結合の結果のコンテキスト履歴を保存するよう構成された保存メカニズムと、
を備えることを特徴とする装置。

【請求項 2】

請求項 1 に記載の装置であって、

10

20

さらに、前記コンテキスト履歴を提示するよう構成された提示メカニズムを備えることを特徴とする装置。

【請求項 3】

コンテキスト履歴を累積するための装置であって、

相互作用のコンテキスト履歴と、登録されたリスナーコンポーネントのリストと、を保持する保持メカニズムと、

リクエストコンポーネントから、前記リクエストコンポーネントのコンテキストと動作を指定する情報とを含んだ要求を受け取るよう構成された受信メカニズムと、

前記受信メカニズムによって受け取られた前記要求に対して、コンテキストアクセス許可を含むコンテキストアクセス方針を適用するよう構成されたコンテキストセキュリティメカニズムと、

コンテキスト確認処理として、前記要求が前記コンテキストアクセス許可のうちの少なくともひとつを満足していることを確認するよう構成されたコンテキスト許可メカニズムと、

前記要求について前記コンテキスト確認処理が成功した場合に前記動作を実行する動作メカニズムと、

前記リクエストコンポーネントのコンテキストと、前記要求と、前記動作の状態と、を前記コンテキスト履歴とともに保存する保存メカニズムと、

前記登録されたリスナーコンポーネントのうちの少なくとも 1 つに対して、実行された前記動作の通知を行う通知メカニズムと、

を備えることを特徴とする装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は一般に、メタデータを使用し、共有するシステムと方法に関する。

【背景技術】

【0002】

データ通信環境において、さまざまなベンダが特定のサービスのための製品を提供している。これまで、環境内の任意のコンポーネントが相互に通信できるよう、ドメインごとに異なる所定のプロトコル一式が必要とされてきた。ベンダによって必要な所定のプロトコルが異なるという問題は、既存のプロトコル基準を採用することで一部対処されている。しかしながら、プロトコルの基準も異なる。

【0003】

【特許文献 1】米国特許出願公開第 2002/0156795 号

【発明の開示】

【発明が解決しようとする課題】

【0004】

任意のコンポーネントが互いの優先されるべき知識（アプリオリ・ナレッジ:a priori knowledge）を持たずに通信しようとする場合、両当事者は特にドメイン別のプロトコルを優先的に知っていなければならない。ひとつの任意のコンポーネントは、標準化されたプロトコルのひとつまたはいくつかを理解するよう特にプログラムされている必要がある。このように、新しい、未知のコンポーネントが方程式に入った場合、アプリケーションは、その新規コンポーネントと通信するのに使用される新たな標準化されたプロトコルを理解するようプログラムし直さなければならない。このように、各アプリケーションは、先ず特定の標準化されたプロトコル一式を使用するよう明確に書かなければ、そのプロトコルと関連するコンポーネントと通信することができない。

【0005】

ドメイン別のインタフェースを用いる J i n i TM等のシステムにおいて、P D A システ

ム等のコンポーネントがプリンタ等の他のコンポーネントと通信するために、PDAシステムはプリンタのプログラムインタフェースの意味規則に関する優先知識を持っていないなければならない。つまり、印刷の方法を知っているコンポーネントでも、特定のコンポーネントのためのインタフェースと通信する方法がわかるように明確にプログラムされないかぎり、ファイルシステム、スキャナデバイスまたはネットワーク翻訳サービス間のデータ転送方法がわからないかもしれない。

【0006】

現在、ユビキタスコンピューティング環境は、ネットワークの効果を見込んでいない。つまり、環境に新たなコンポーネントが加えられた場合、追加されたコンポーネントの機能を環境に加えること以上の能力はない。

10

【0007】

さらに、システムの中には、システム内のコンポーネントに関して、コンテキスト情報を臨時に保持するものもある。このようなコンテキスト情報はしばしば、そのコンポーネントがユーザか、ドキュメントか、アプリケーションか等、それが代表するコンポーネントの種類に応じて、各種のフォーマットで表現される。その結果、システム内の任意のコンポーネントは、特にその任意のコンポーネントが相互に関する優先知識を持っていない場合、現在のコンテキスト情報を相互に提供することが困難または不可能となることが多い。

【0008】

コンテキストシステムのひとつの問題は、信頼性を確保することである（たとえば、あるプロパティ値が不適当に変更されていないか検証する）。別の問題は、ユビキタスコンピューティング環境は非常に流動的であるため、その環境内のシステムの挙動を理解することが非常に困難である点である。どのようにプロパティ値が変化したか、およびなぜそのシステムが特定の方法で行動したかを検証することができれば有利である。

20

【0009】

ユビキタスコンピューティング環境において検索メカニズムを使用するひとつの問題は、これらのメカニズムが検索されたコンポーネントの非常に長いリストを戻してきながら、ユーザが現在の状況にとって有益なコンポーネントを理解し、選択するのに役立つ情報を提供しない場合が多いことである。ユーザに対し、現状に関係のある情報を優先的に提供できれば有利である。

30

【課題を解決するための手段】

【0010】

本明細書では、各コンポーネントが、それが何かを明らかにすることのできるコンポーネントコンテキストを持つ環境におけるコンポーネント間の相互作用のコンテキスト履歴を累積するための方法とシステムの実施形態を開示する。このコンポーネントは、リクエストコンポーネントとサービスコンポーネントを備える。リクエストコンポーネントは、そのコンテキストを動作要求と一緒にサービスコンポーネントに渡す。リクエストコンポーネントのコンテキストは、サービスコンポーネントのコンテキストメタデータとして記録され、コンテキスト履歴の一部となる。

【0011】

また、それが何かを明らかにすることのできるコンポーネントコンテキストにおける変更のコンテキスト履歴を累積するために使用される方法、装置、プログラム製品の実施形態も開示される。この装置とプログラム製品は、コンポーネントコンテキストの一部を変更する動作を検出し、その部分をコンテキスト履歴と結合する。次に、この装置とプログラム製品は、その動作に応答してその部分を変更する。

40

【0012】

さらに、コンポーネントが要求を受け取り、その要求にコンテキストアクセス方針を適用する方法、装置、プログラム製品の別の実施形態が開示される。コンテキストアクセス方針を要求に適用した後、コンポーネントは、その要求が一連のコンテキストアクセス許可のひとつを満足するかを確認し、満足する場合は、その要求を満たす。本発明に係る方

50

法の1つの態様では、コンピュータ制御による方法であって、要求を受け取るステップと、前記要求にコンテキストアクセス方針を適用するステップと、前記要求が、一連のコンテキストアクセス許可のひとつを満足することを確認するステップと、前記確認ステップに応答して前記要求を満たすステップと、を含むことを特徴とする。

【発明を実施するための最良の形態】

【0013】

本発明の実施の形態（以下、実施形態）によるコンテキスト情報を提供するためのシステム10が図1に示されている。システム10は、コンピュータ12、プリンタ14（多機能デバイスを含む）、携帯情報端末機器（PDA）16およびサーバ18（以下、「コンポーネント12-18」とする）を含み、これらがネットワーク20によって相互に連結されている。本実施形態により、同じ、または異なる通信プロトコルおよび／またはデータタイプを使うコンポーネントは、そのコンポーネントがドメイン別のインタフェース、プロトコルまたはデータフォーマットを使用することなく、相互にコンテキスト情報を転送することが可能となる。さらに、本実施形態は、ユーザ、デバイスまたはアプリケーションが、相互の優先知識を持つコンポーネントを必要とすることなく、直接相互を呼び出し、現在のコンテキスト情報およびその他のデータを提供することを可能とする手立てとなる。

【0014】

図2に示すように、本発明の実施形態では、コンピュータ12は中央処理ユニット（CPU）22、メモリ24、I/Oユニット26で構成され、これらが1つまたは複数のバスによって相互に連結されている。

【0015】

図3において、図1に関連して上述したように、コンピュータ12がPDA16に連結されている。PDA16は、コンテキストインタフェース、通知（notification）インタフェース、ユーザインタフェースおよびデータソースインタフェースからなる汎用インタフェースセット16aをメモリに保存しているか、あるいはこれにアクセスできる。これを以下、「関連している」と表現する。具体的なインタフェースの数および／または組み合わせは異なってもよく、デバイスPDA16の種類や、それが望む、または提供する能力および／またはサービスに依存する。また、PDA16、したがって汎用インタフェースセット16aはいつでも、インタフェースの追加、削除または変更を行うよう更新できる。

【0016】

汎用インタフェースセット16a内の各インタフェースは、PDA16に特有でありながら、コンピュータ12が理解、実行し、それがPDA16とコンテキストデータを通信、転送し、PDA16に関するコンテキストデータの変化について、コンピュータ12にイベント通知を送り、コンピュータ12がユーザインタフェースを受け取って、コンピュータ12のユーザが変更後のコンテキストデータを見ることができるようにし、あるいはコンピュータ12がPDA16からのデータを受け取ることができるようにする命令、動作セットおよび／またはその他のデータによって構成される。

【0017】

特に、コンテキストインタフェースは、コンテキストオブジェクト16cを要求し、これにアクセスするgetContext（）動作によって構成されている。コンテキストデータには、PDA16に関する情報、たとえばその種類、所有者、使用履歴、PDA16が現在使用中か、あるいはその他の動作状態情報、アイデンティティ、ネットワーク20上のロケーション、物理的ロケーション、管理ドメイン、PDA16またはPDA16に記憶されたファイルの一人または複数のユーザに関する情報、あるいはその他、PDA16が提供するその他の種類の環境情報等が含まれる。さらに、コンテキストデータには、コンピュータ12が理解し、実行することのできる、PDA16に特有のコンピュータ言語の命令も含まれる。

【0018】

本発明の実施形態において、PDA 16に関連するコンテキストオブジェクト16cは、get Context () 動作を通じてコンピュータ12によって取得されるようにすることができる。コンテキストオブジェクト16cは、get Context () 動作によって戻されるひとつまたは複数の動作、たとえばget Property () やset Property () 動作をサポートするオブジェクトのライブでのリモートリファレンスおよびその他、コンピュータ12がPDA 16に関連する現在および／またはこれまでのコンテキストデータにアクセスすることを可能にする命令等である。ただし、オブジェクト16cは、コンテキストデータを直接含んでもよい。get Property () 動作は、コンテキストデータをPDA 16からコンピュータ12に戻すよう要求する命令を含むことができるため、コンピュータ12はPDA 16に関連するコンテキストデータを
10 読み取れる。set Property () 動作は、PDA 16にコンテキストデータを提供するためにコンピュータ12が実行する命令とデータを含むため、コンピュータ12はPDA 16のコンテキストデータを更新または変更できる。

【0019】

通知インタフェース（たとえば、プロキシオブジェクトによって提供される）はregister () 動作によって構成され、これには、コンピュータ12が、PDA 16のコンテキストデータへの変更に関する通知を受け取るための、PDA 16に関する聞き手（リスナー：listener）としてコンピュータ自身を登録することができるようにするために
20 実行する命令、動作およびデータを含めることができる。通知インタフェースには、コンポーネントパラメータとコンテキストパラメータを含むひとつまたは複数のパラメータを渡すことができる。コンポーネントパラメータは、通知の受信者としてコンピュータ12を識別する。コンテキストパラメータは、コンピュータ12に関する現在および／または現在までのコンテキストデータであり、これは、コンピュータ12に通知を送信すべきか否かを決定するための、PDA 16に関するひとつまたは複数のプロパティを表す。

【0020】

ユーザインタフェースは、ユーザウィンドウを生成するためのget UI () 動作によって構成される。特に、get UI () 動作は、コンピュータ12のユーザがPDA 16の機能（PDA 16に関連するコンテキストデータへのアクセスを含む）にアクセスすることを可能にするユーザインタフェースウィンドウを生成し、表示するためにコンピュータ12が実行できる命令を有するオブジェクトを、コンピュータ12に戻す。
30

【0021】

データソースインタフェースは、コンピュータ12がPDA 16からデータを受信できるようにするデータ転送セッションを確立するためのbegin Transfer Session () 動作によって構成される。さらに、begin Transfer Session () 動作は、コンテキストパラメータ等のパラメータを受け取ることができる。本発明の実施形態において、コンピュータ12は、そのアイデンティティをPDA 16に知らせるためにそのコンテキストオブジェクトを呼ぶと、これをパラメータとしてbegin Transfer Session () 動作に渡す。PDA 16は、コンテキストパラメータの中で提供されるコンテキストデータに基づき、コンピュータ12にデータを転送するか、データ転送中にその挙動を変更するかを決定する。
40

【0022】

上記のインタフェースと関連する動作は各々、モバイルコードで構成することができる。本発明の実施形態において、汎用インタフェースセット16aがコンピュータ12に提供されると、本明細書で説明したように、コンピュータ12が特定のインタフェースに関連する一連の命令およびその他のデータにアクセスし、これを実行することが可能となる。またさらに、インタフェースは他のインタフェースへの一連の命令またはリファレンスで構成することもでき、この場合、コンピュータ12はそれに応じてデータを利用し、あるいは命令を実行する。

【0023】

本発明の実施形態において、上記のモバイルコードを使用するか否かは任意である。

10

20

30

40

50

【0024】

データオブジェクト16bは、PDA16のプロキシオブジェクトであり、PDA16から受信され、コンピュータ12に保存される。コンピュータ12はデータオブジェクト16bを通じて汎用インタフェースセット16aにアクセスできる。より具体的には、データオブジェクト16bは、コンピュータ12が知り、理解できると想定される、PDA16に関連する汎用インタフェースセット16aの中のインタフェースによって定義される各種の動作をサポートする。たとえば、データオブジェクト16bは、PDA16の開発者が決定したプロトコルおよび／またはデータフォーマットを使ったPDA16との通信を専門とするコンテキストインタフェースのカスタム実装を提供する。

【0025】

図4において、はじめにステップ30で、コンピュータ12は探索プロセスを実行し、PDA16がコンピュータにコンテキストデータを供給できるか判断する。

【0026】

次に、ステップ32で、探索されたPDA16は、データオブジェクト16bをコンピュータ12に戻す。コンピュータ12は、受け取ったデータオブジェクト16bを点検し、どの汎用インタフェース（ひとつまたは複数）がPDA16に関連しているか判断する。コンピュータ12は、PDA16が少なくともコンテキストインタフェースと関連していると判断し、したがって、PDA16はコンピュータにコンテキストデータを提供できる。

【0027】

次に、ステップ34で、コンピュータ12はデータオブジェクト16bにおいて定義された手順、命令および／またはデータを使い、PDA16と関連するget Context（）インタフェースを呼び出し、コンテキストオブジェクト16cをPDA16に要求する。コンピュータ12がコンテキストオブジェクト16cを通じてコンテキストデータへのアクセスを要求するため、オブジェクト16cに含まれる命令は、その要求をPDA16によってサポートされる第一のプロトコルに翻訳して、コンテキストデータへのアクセスを実現することができる。

【0028】

次に、ステップ36で、コンピュータ12はコンテキストオブジェクト16cを受け取り、関連するget Property（）動作を呼び出して、PDA16からコンテキストデータを取り出す。さらに、コンピュータ12がget Property（）動作を実行する時、PDA16は命令、動作またはデータを直接コンピュータ12に戻し、コンピュータが、PDA16から転送されているコンテキストデータを理解することができるようにすることもできる。また、コンテキストオブジェクト16cはコンテキストデータの要求をPDA16に送信するものの、オブジェクト16cがコンテキストデータを含むこともあり、この場合、コンピュータ12はその中のデータにアクセスする。いずれの場合も、コンピュータ12はPDA16からコンテキストデータを受け取る。

【0029】

別の実施形態において、ステップ30-36は、ステップ30-32でデータオブジェクト16bを点検し、PDA16が通知インタフェースおよびユーザインタフェースとも関連していることを判断する点を除き、本発明の実施形態について先に説明したものと同一方法で実行される。コンピュータ12は、したがって、PDA16と関連するコンテキストデータの変化に関するイベント通知を受信するリスナーとしてPDA16に登録できる。

【0030】

別の実施形態において、ステップ30-36は、ステップ30-32でコンピュータ12が受信したデータオブジェクト16bを点検し、PDA16がデータソースインタフェースとも関連していることを判断する点を除き、本発明の実施形態について先に説明したものと同一方法で実行される。このように、PDA16は連続するデータストリーム（たとえば、ストリーミングビデオ）等、それが関連するコンテキストデータのほかに、コン

10

20

30

40

50

ピュータ12の他のタイプのデータを送信することができる。

【0031】

図5に示すとおり、プリンタ14はコンテキストインタフェースを含む汎用インタフェースセット14aと関連している。この実施形態におけるコンテキストインタフェースは、図3-5に関して上述したコンテキストインタフェースと同じであるが、この実施形態の場合、コンピュータ12によって実行されるプリンタ14専用の命令がある点だけが異なる。

【0032】

図6において、はじめにステップ50で、コンピュータ12は図3のステップ30-32について上述した探索プロセスを実行するが、ステップ50ではデータオブジェクト14bを探索する。さらに、コンピュータ12は、プリンタ14が少なくともコンテキストインタフェースと関連していると判断し、プリンタ14はコンピュータ12にコンテキストデータを供給することができる。

10

【0033】

次に、ステップ52で、コンピュータ12は、図3のステップ34についてすでに説明したように、データオブジェクト14bで定義された手順、命令および／またはデータを使い、プリンタ14と関連する `getContext()` インタフェースを呼び出して、コンテキストオブジェクト14cを要求する。このようにして、コンピュータ12はデータオブジェクト14bを通じてプリンタ14からコンテキストオブジェクト14cを受け取る。

20

【0034】

次に、ステップ54で、コンピュータ12はデータオブジェクト14bに含まれる `getProperty()` 動作を呼び出して、プリンタ14と関連するコンテキストデータを取り出す。この実施形態において、プリンタ14と関連するコンテキストデータは、サーバ18等の中央ロケーションに保存することができるが、コンピュータ12またはPDA16等、別の場所にコンテキストデータを保存することも可能である。このようにして、`getProperty()` 動作に含まれ、コンピュータ12によって実行される命令、動作またはデータは、プリンタ14と関連するコンテキストデータをサーバ18から取り出すようプリンタ12に命令する。さらに、命令、動作またはデータは、ネットワーク20上でのコンテキストデータのロケーション（つまり、サーバ18）やその他、コンピュータ12が実行し、それがサーバ18から転送されているコンテキストデータを取り出して理解することができるようにする色々な命令を含むことがある。

30

【0035】

コンポーネントがコンテキストデータを記録する能力により、そのコンポーネントは、そのコンポーネントによって、またはそのコンポーネントのために実行される動作のコンテキスト履歴を得ることができる。このコンテキスト履歴は、ユビキタスコンピューティング環境におけるコンポーネントの使用を簡素化するために使用できる。

【0036】

コンポーネントは、そのコンポーネントを操作することのできる人またはその他の生物学的創造物と強力に関連させることが可能であり、これらのコンポーネントをエンティティと呼ぶ。その他のコンポーネントは人との強い関連を持たず、単純にコンポーネントと呼ばれる。

40

【0037】

コンポーネントには、コンテキストを持ち、そのコンテキストを明らかにすることができるデバイスが含まれる。リクエスタコンポーネント (requester component) は、そのリクエスタコンポーネントのコンテキストを動作の要求と一緒にサービスコンポーネントに送信することにより、サービスコンポーネントと相互作用する。サービスコンポーネントはリクエスタコンポーネントのコンテキストを受信するため、サービスコンポーネントはリクエスタコンポーネントのコンテキストと要求された動作の識別とを保存することができる。サービスコンポーネントが任意データを任意で含めることを可能にする任意拡張

50

可能なデータ構造を使用する能力を有する場合、サーバコンポーネントは他のコンポーネントとの相互作用のコンテキスト履歴を、メタデータとして維持することができる。リクエストコンポーネントのコンテキストは、サービスコンポーネントのコンテキストからの情報と組み合わせ、サービスコンポーネントのコンテキスト履歴に、何が行われたかだけでなく、どのエンティティ／コンポーネントがその動作を要求したか、いつ、どこでその動作が要求されたか、および要求された動作の結果も含めることができる。

【0038】

さらに、コンポーネントは、コンポーネント自身が行ったコンテキストの変更（たとえば、プロパティ変更）の履歴も維持できる。

【0039】

コンテキストメタデータにはプロパティメタデータが含まれ、アクセス管理（許可）、アクセス方針、履歴も含めることができる。アクセス管理は、どのコンポーネント／エンティティがそのコンテキスト（またはコンテキストの中の特定のコンテキストプロパティ）にアクセスできるか、あるいはコンポーネント／エンティティに動作を実行するよう要求できるかを決定する。アクセス方針は、サービスコンポーネントにアクセスして要求を出すコンポーネント／アイデンティティを特定するのに必要な認証のレベルを決定し、コンテキスト履歴は、コンテキストレベルで実行された動作の履歴とコンテキストプロパティレベルで実行された動作の履歴の両方を含む。コンテキストプロパティレベルの履歴は、プロパティについて行われたアップデートの全部またはその中から選択されたもののログを維持する。このように、プロパティを前の状態まで辿ることができる。プロパティ履歴記録のひとつの実施形態を、以下に、図11に関して説明する。

【0040】

ユビキタスコンピューティング環境のひとつの問題は、この環境においてユーザがコンポーネントを選択することが面倒で、エラーを招きやすい点である。過去のコンポーネントの使用の履歴（使用履歴）を維持することにより、ユーザは簡単に以前の構成を呼び出して再利用または変更できる。さらに、履歴メタデータを使い、過去の使用パターンを検出し、意味形成（センスメイキング: sensemaking）のための情報を収集し、これらのパターンに基づく推論を確立することも可能である。

【0041】

実施形態のひとつの面は、コンテキスト情報に関するメタデータを保持することである。このコンテキスト情報は、信頼性とシステム挙動の説明を提供することができる。つまり、信頼性は、誰が所与のプロパティを更新したか、それが更新されたのはどれほど最近か、および現在値は何に基づいて決定されたかを知ることによって改善される。システム挙動の説明は、コンテキストプロパティの現在値の根拠およびコンテキストプロパティを変更するためにとられた動作の説明を知ることによって提供される。

【0042】

デバイスおよび／または動作からのコンテキストは、積み重ねることができる。つまり、特定のコンポーネント／エンティティは、それと相互作用したことがあるその他のコンポーネント／エンティティのコンテキストを含むことができる。

【0043】

コンポーネント／エンティティからのコンテキストは、連鎖させることもできる。つまり、各コンポーネントのコンテキストをコンポーネント間のデータフローに追加できる。たとえば、リクエストエンティティがプレゼンテーションコンポーネントに情報を提供する場合を考える。プレゼンテーションコンポーネントによって提供されたプレゼンテーション情報は、プレゼンテーションコンポーネントのコンテキストを含むであろう。そこで、エンティティがストレージコンポーネントにその情報を保存したいとすると、ストレージコンポーネントのコンテキストもプレゼンテーション情報と一緒に含めることができる。さらに、エンティティのコンテキストも含めることができる。このように、プレゼンテーション情報の中のメタデータは、そのデータの由来を提供する。

【0044】

メタデータの連鎖により、コンポーネントはそれがアクセスするデータの履歴全体を判断できる。この情報には、どのコンポーネントがそのデータの供給源か、どのコンポーネントがそのデータにアクセスし、および／またはそのデータについて動作したか、いつ、どこでそのデータが保存されたか等がある。各コンポーネントがデータを提供するため、提供するコンポーネントは既存のメタデータとともにそのメタデータを提供し、したがって、これを受け取るコンポーネントは、そのコンポーネント自身のメタデータと、既存のメタデータおよび提供するコンポーネントのメタデータを統合することができる。

【0045】

履歴メタデータの結合は、`getProperty()` および `setProperty()` 方式にフィールドを追加し、`getPropertyHistory()` 方式を追加することによってサポートされる。

【数1】

```
Object getProperty(String key, RNC requester);
Void setProperty(String key, Object value,
    RNC requester, Cause cause, Date expiration);
History getPropertyHistory(String key, RNC requester,
    Date sinceWhen);
```

`requester` の引数は、認証可能または不可能なコンポーネント、エンティティまたはアプリケーションとすることができる。`cause` の引数は、リクエストによって定義され、アップデートの理由に関するより詳しい情報を得るためにリクエストに戻されるトークンとすることができる。さらに、`getProperty` と `setProperty` はどちらも、リクエストのアイデンティティを提供する証明を含めるよう拡張できる。

【0046】

リクエストコンポーネントは、他のコンポーネントの履歴メタデータを取得して、他のコンポーネントのコンテキスト履歴を取得するか、あるいは特定の信頼性基準を満たす数値を得ることができる。したがって、リクエストコンポーネントは、信頼できない、あるいは認証されていないエンティティ／コンポーネントが行った他のコンポーネントのプロパティへのアップデートや、認証メカニズムを迂回しながら行われたアップデート、あるいは古くなった数値を拒否できる。

【0047】

コンポーネントはまた、そのコンテキスト履歴に、自分から行ったコンテキスト変更も追加できる。

【0048】

図7から図11は、メタデータを保存するために考えられるひとつの方法を示す。当業界で周知の別のメカニズムも使用できる。

たとえば、以下の形態による記憶メタデータ：

【数2】

```
SE.histcontext.operation02843.timestamp = May 27, 2001
SE.histcontext.operation02843.operation = create prop "location"
```

または

```
SE.histcontext.operation02843{
    Timestamp = May 27, 2001
    Operation = create prop "location"
    ...}
```

あるいは上記以外の表現でもよい。

【0049】

図7は、プロパティ集合703を含むコンテキスト集合701を表すコンテキスト図700を示し、これには、オプションのコンテキストアクセス方針705、オプションのアクセス管理集合77、オプションの履歴集合709を含めることができる。プロパティ集合703は、コンテキスト集合701に関連するコンポーネント／エンティティのプロパティに関するメタデータを含み、これについて、図10を参照しながら説明する。

【0050】

コンテキスト集合701には、オプションのコンテキストアクセス方針705を含めることができる。オプションのコンテキストアクセス方針705は、そのコンポーネントへのアクセスを許可するためのメカニズムを特定するために使用される。この方針の一部は、コンポーネントに対し、アクセスを要求しているエンティティ／コンポーネントの認証を行うよう要求することができ、また、特定されたエンティティ／コンポーネントがコンテキスト集合701を含むコンポーネントにアクセスできるか否かを実行またはその他の方法で指示することができる。方針の例として、サービスコンポーネントと異なるロケーションにある、他の方法で認証されたエンティティへのアクセスを否認する方針や、瑕疵のある方針にかかわらず、特定のエンティティ／コンポーネントのアクセスを許可するバイパス方針、および／またはコンテキストモニタ等へのアクセスを許可する方針がある。

【0051】

さらに、コンテキスト集合701は、オプションのアクセス管理集合707を含むことができる。オプションのアクセス管理集合707は、アクセス許可を特定のエンティティまたは特定の種類のエンティティに関連付けるメタデータを含むことができる。オプションのアクセス管理集合707については、図9を参照しながらさらに詳しく説明する。

【0052】

さらに、コンテキスト集合701は、コンテキスト集合701と関連するコンポーネントへの過去の動作を表す履歴メタデータを含むオプションの履歴集合709を含んでもよい。オプションの履歴集合709については、図8を参照しながらさらに詳しく説明する。他のプロパティ関係のデータ履歴は、プロパティ集合703の中で維持される。

【0053】

図8は、コンテキスト履歴集合709を表すコンテキスト履歴図720を示す。オプションの履歴集合709は、少なくともひとつのコンテキスト履歴エントリ723を含む。コンテキスト履歴エントリ723は、動作フィールド725、リクエストフィールド727、時間フィールド729、理由フィールド731および状態フィールド733を含むことができる。

【0054】

動作フィールド725には、コンテキスト集合701にどの動作が加えられたか（あるいは加えられようとしたか）を示す情報が含まれる。これらの動作には、たとえば、コンテキストのプロパティ、アクセス管理、履歴の追加、削除、維持、変更等がある。

【0055】

リクエストフィールド727には、動作フィールド725に示される動作を呼び出したエンティティおよび／またはコンポーネントを特定する情報が含まれる。時間フィールド729には、その動作が要求され、および／または実行された時のタイムスタンプが含まれる。理由フィールド731には、要求するコンポーネント／エンティティによって明示される動作の理由を示す情報が含まれる。状態フィールド733には、コンテキスト履歴エントリ723の管理を助ける各種の状態が含まれる。

【0056】

コンテキスト履歴集合709は、エンティティ／コンポーネントのコンテキストの使用履歴をまとめるのに使用できる。オプションの履歴集合709は、プロパティの削除と追加に関する履歴だけを含めばよい。特定のプロパティに関する履歴（またはコンテキスト集合701の中の他のエントリ）は、プロパティコンテキストの一部として保持できる（たとえば、図10、図11参照）。

【0057】

10

20

30

40

50

図9は、アクセスエントリ743を含むアクセス管理集合741を表すアクセス管理図740を示す。アクセスエントリ743は、アクセス管理集合741に関連するコンテキストまたはプロパティ（図10参照）にアクセスするための許可を有する（許可記録747の中に保存されている）エンティティまたはコンポーネントを特定するエンティティ／コンポーネント記録745を含む。ある実施形態において、コンテキストおよびコンテキストプロパティに関するアクセス管理集合は同様の構造を有する。

【0058】

エンティティ／コンポーネント記録745によって特定されるエンティティ／コンポーネントは、単独のユーザ、ユーザグループ、コンポーネント、コンポーネントの種類あるいはその他特定可能なエンティティおよび／またはコンポーネントのいずれであってもよい。

10

【0059】

図10は、プロパティエントリ753を含むプロパティ集合703を表すコンテキストプロパティ図750を示す。プロパティエントリ753は、そのプロパティの識別を含むキーフィールド755と、そのプロパティの数値を含む現在値フィールド757を含むことができる。プロパティエントリ753にはまた、オプションの有効期限フィールド759とオプションのプロパティアクセス方針フィールド761を含めてもよい。

【0060】

キーフィールド755は、プロパティ識別子（一般に、人間が判読可能なテキストで、おそらくは国際化の対象となっている）を含む。

20

【0061】

オプションの有効期限フィールド759により、プロパティ値を設定しているコンポーネントは、その数値が古くなる（信頼できないものとなる）と予想される時期を示すことができる。

【0062】

プロパティエントリ753は、エンティティ／コンポーネントがオプションのプロパティアクセス方針フィールド761を含むプロパティにアクセスし、および／またはこれについて動作することができるか否かを指示するのに使用できるオプションのプロパティアクセス方針フィールド761を含むことができる。方針の例としては、サービスコンポーネントと異なるロケーションにある、他の方法で認証されたエンティティへのアクセスを否認する方針や、既存の方針にかかわらず、特定のエンティティ／コンポーネントのアクセスを許可するバイパス方針、および／またはコンテキストモニタ等へのアクセスを許可する方針がある。

30

【0063】

主要キー／数値プロパティに加え、プロパティエントリ753には、オプションのプロパティ履歴集合763および／またはオプションのアクセス管理集合765を含めることができる。

【0064】

オプションのアクセス管理集合765の構造は、アクセス管理集合741と同じにすることができ（ただし、同じにする必要はない）、これをコンテキストではなくプロパティに適用する。

40

【0065】

オプションのプロパティ履歴集合763は一般に、コンテキスト履歴集合709とは異なり、使用および／または変更履歴を特定のプロパティとまとめるのに使用される。オプションのプロパティ履歴集合763については、図11を参照しながら説明する。

【0066】

図11は、プロパティ履歴エントリ773を含むプロパティ履歴集合763を表すプロパティ履歴図770を示す。プロパティ履歴エントリ773は、特定のプロパティに加えられる変更履歴を維持するのに使用できる。プロパティ履歴エントリ773には、新数値フィールド775、旧数値フィールド777、リクエストフィールド779、時間フィー

50

ルド781および理由フィールド783を含めることができる。

【0067】

新数値フィールド775は、時間フィールド781内に示される時刻の時点で現在値フィールド757に設定される数値を記憶する。旧数値フィールド777は、時間フィールド781に示された時刻の直前の、現在値フィールド757の数値を記憶する。リクエストフィールド779は、プロパティの変化の原因となったコンポーネント／エンティティを特定する数値を記憶する。理由フィールド783は、なぜそのプロパティが変更されたかを示す数値を含むことができる。

【0068】

コンテキスト履歴集合709とプロパティ履歴集合763を維持することにより、コンポーネントはそのコンポーネントと他のコンポーネントとの間の相互作用のコンテキスト履歴を累積する。さらに、コンテキストモニタも、そのコンテキストモニタが使用されている環境内でのコンポーネント間の相互作用のコンテキスト履歴を累積できる。各コンポーネントが有するコンポーネントコンテキストは、他のコンポーネントまたはコンテキストモニタによって明らかにできる。コンポーネント間の動作は、リクエストコンポーネントが動作要求とリクエストコンポーネントのコンテキストをサービスコンポーネントに送信することによって誘発される。サービスコンポーネントは、リクエストコンポーネントのコンテキストをサービスコンポーネントのコンテキストメタデータとして記録することができる。動作とリクエストコンポーネントのコンテキストは、他のコンポーネントまたはコンテキストモニタによっても記録されることが可能である。記録されたメタデータを結合することにより履歴メタデータが得られ、これを使ってコンポーネント間の将来の相互作用を簡略化できる。

【0069】

このほか、コンポーネントは自分のプロパティを変更でき、これらの変更もまたそのコンポーネントのコンテキスト履歴の中に累積される。

【0070】

図12は、コンテキストアクセスプロセス1200を示しており、「開始」1201から始まり、「要求受取」手順1203へと続く。「要求受取」手順1203は、利用可能な通信メカニズムを使い、リクエストコンポーネントからの動作およびコンテキストを受け取る。この動作には、プロパティをコンテキスト集合701の抽出レベルにおいて定義、削除されるようにする動作、プロパティをプロパティ集合703の抽出レベルにおいて変更、アクセスされるようにする動作、コンテキスト履歴をコンテキスト集合701の抽出レベルでアクセスされるようにする動作、およびコンポーネントのコンテキストにアクセスし、これを保持する動作が含まれる。

【0071】

要求が受け取られると、コンテキストアクセスプロセス1200は、不完全な状態（状態フィールド733に記憶できる）を有するコンテキスト履歴エントリ723を保存する「試行履歴保存」手順1205に続く。次に、「コンテキストアクセスチェック」決定手順1207がオプションのコンテキストアクセス方針705とオプションのアクセス管理集合707を使い、リクエストコンポーネントのアイデンティティを検証し、そのリクエストコンポーネントがコンテキスト集合701の抽象レベルで要求された動作を実行することを許可されているかを検証する。

【0072】

「コンテキストアクセスチェック」決定手順1207は、その動作を要求したエンティティ／コンポーネント上でさまざまなチェックを実行する。これらのチェックには、1）あらゆるエンティティ／コンポーネントによるすべての動作を許可する、2）エンティティ／コンポーネントの識別を認証する、3）エンティティ／コンポーネントが要求された動作を実行する許可を持っていることを検証する、4）特定のエンティティ／コンポーネントについてユーザが指定した無効化（オーバーライド：override）があるか否かをチェックする、およびその他、オプションのコンテキストアクセス方針705によって定義さ

れる、あるいはオプションによるアクセス管理集合 707 によって要求されるその他のチェックのうちの少なくともひとつまたは複数が含まれる。

【0073】

「コンテキストアクセスチェック」決定手順 1207 がリクエスタコンポーネント／エンティティへのアクセスを拒否すると、コンテキストアクセスプロセス 1200 は、試行された動作をログ（記録）し、状態フィールド 733 を更新する「コンテキスト拒絶履歴保存」手順 1211 に続く。さらに、リクエスタコンポーネント／エンティティにも、アクセスの拒否を知らせることができる。

【0074】

これに対し、「コンテキストアクセスチェック」決定手順 1207 がリクエスタコンポーネント／エンティティによるコンテキスト集合 701 へのアクセスを許可した場合、「コンテキスト動作」決定手順 1215 が、要求された動作はコンテキスト集合 701 の抽象レベルで実行されるべきかを決定する。そうであれば、コンテキストアクセスプロセス 1200 は、「コンテキスト動作実行」手順 1217 に進み、ここで要求された動作が実行される。

10

【0075】

要求された動作が完了すると、コンテキストアクセスプロセス 1200 は、「コンテキスト履歴保存」手順 1218 に進み、この手順は「試行履歴保存」手順 1205 によって保存される履歴記録の中のオプションの履歴集合 709 に動作の結果を保存し、これに従って状態フィールド 733 を更新する。

20

【0076】

しかしながら、「コンテキスト動作」決定手順 1215 において、要求された動作がプロパティ集合 703 の抽象レベルで実行されることになった場合、コンテキストアクセスプロセス 1200 は「プロパティアクセスチェック」決定手順 1219 に進み、これがキーフィールド 7155 を使って特定されたプロパティをプロパティエントリ 753 の中に位置づけ、「コンテキストアクセスチェック」決定手順 1207 によって実行されたものと同様のチェックを行うことができる（ただし、オプションのプロパティアクセス方針フィールド 761 および／またはオプションのアクセス管理集合 765 を使い、プロパティ集合 703 の抽象レベルとプロパティエントリ 753 そのもののの中で、チェックを行うことができる）。要求された動作が許可されないと、コンテキストアクセスプロセス 1200 は「プロパティ拒絶履歴保存」手順 1223 に進み、ここで、新たなプロパティ履歴エントリ 773 がオプションのプロパティ履歴集合 763 に追加され、そのプロパティについて要求された動作が拒否されたことが示される。次に、コンテキストアクセスプロセス 1200 は、「コンテキスト履歴保存」手順 1218 に進み、状態フィールド 733 が更新される。

30

【0077】

「プロパティアクセスチェック」決定手順 1219 において、要求された動作が許可されると、コンテキストアクセスプロセス 1200 は「プロパティについて動作」手順 1225 に進み、特定された製品について要求された動作が試みられる。要求された動作の結果の状態は「プロパティ履歴保存」手順 1227 に送られ、これが新規のプロパティ履歴エントリ 773 をオプションのプロパティ履歴集合 763 に追加し、このようにして要求されたプロパティ動作の状態と結果が得られる。次に、コンテキストアクセスプロセス 1200 は「コンテキスト履歴保存」手順 1218 に進み、状態フィールド 733 が更新される。

40

【0078】

あるコンポーネントのコンテキストについて許可される動作には、プロパティ集合 703 にプロパティを追加する、あるいはここからプロパティを削除すること、オプションの履歴集合 709 へのアクセスを読み取ること、およびオプションのアクセス管理集合 707 への変更を許可することが含まれる。特定のプロパティの変更履歴は、そのプロパティそのものとともに、オプションのプロパティ履歴集合 763 の中に保存される。

50

【0079】

あるプロパティについて許可される動作には、現在値フィールド757の数値および／またはオプションの有効期限フィールド759を変更すること、これらの数値にアクセスすること、あるいはオプションのプロパティ履歴集合763にアクセスすることが含まれる。

【0080】

前述の説明に基づき、当業者はまた、オプションによる履歴集合709にアクセスする方法や、あるエンティティに個別で自己の履歴を更新させる方法を理解できるであろう。

【0081】

図13は、サービスコンポーネントがその履歴メタデータを提供するのに使用できる履歴供給プロセス1300を示す。履歴供給プロセス1300は、リクエストコンポーネント（たとえば、`getPropertyHistory()`方式の呼び出しによる履歴検索メカニズム等）から呼び出された履歴要求を受け取ったところで、「開始」1301から始まる。履歴供給プロセス1300は、要求が十分に形成されていることを検証できる「履歴要求点検」手順1303に続く。「履歴供給」手順1307は、要求された履歴メタデータを組み立て、組み立てられたメタデータをリクエストコンポーネントに送信する。

10

【0082】

エンティティ／コンポーネントのコンテキストに履歴およびアクセス情報が含まれると、リクエストコンポーネント／エンティティは他のエンティティ／コンポーネントに履歴および許可情報を要求し、これらのコンポーネントのうちどれにリクエストコンポーネント／エンティティがアクセスできるかを判断して、利用可能な履歴コンテキストに基づき、ユーザにとってのオプションの優先順位を付けることができる。この優先順位付けは、コンポーネントのエンティティのアイデンティティおよび／またはコンテキストモニタから利用できる情報に基づいて行われる。この情報は、環境内のコンポーネントに要求し、ふるいにかけて使用できないコンポーネントを排除し、エンティティの選好に基づいて、また環境の状況に応じて変更する。この情報は、構成再構築メカニズムにより、構成確立メカニズムによってどのようなコンポーネント構成が確立できるかを判断するために使用される。

20

【0083】

コンポーネント探索サービスは通常、探索コンポーネントがアクセスしたすべてのエンティティ／コンポーネントに関する情報を戻す。これはしばしば、大量の情報を探索コンポーネントのユーザに提供することになる。ユーザは、自分にとって関心のないコンポーネントをふるい落とすためのテンプレートを提供することができる。

30

【0084】

本発明のひとつの実施形態は、リクエストコンポーネント／エンティティおよび検索されているコンポーネントの両方の背景コンテキスト情報に基づいて、コンポーネント検索結果の優先順位付けとフィルタリングを行う方法を提供する。あるエンティティに関するコンテキスト情報には、アイデンティティ、ロケーション、優先性、利用履歴等の情報が含まれる。コンポーネントに関するコンテキスト情報には、サポートされているデータのタイプ、ロケーション（これを持っている場合）、ロケーション要件、利用履歴、セキュリティ方針等の情報が含まれる。エンティティとコンポーネントの情報を組み合わせ、アクセス管理上の制限によって、所与のエンティティ／コンポーネントが使用できないコンポーネントを排除することができる。この情報を組み合わせることにより、現状においてそのユーザにとって最も有益なものは何かに基づき、利用可能なコンポーネント／エンティティを提示する優先順位も付けることができる。

40

【0085】

コンテキスト情報は、2つの方法で分類される。つまり、動的対静的および状態対要求である。コンテキスト状態は、あるコンポーネント／エンティティのプロパティの数値を表す。要求コンテキストは、あるコンポーネント／エンティティのユーザにとっての前提

50

条件を表す。動的コンテキストは、コンテキスト情報が実行中に更新されているときのものである。静的コンテキストは、定義されたときに特定され、定義し直されるまで変更されない。このようなコンテキストの表現は、そのコンテキストが実際にどのように取得され、検証されるかに無関係である。

【0086】

本発明のひとつの実施形態は、これらの属性を、データタイプ、アイデンティティアクセス制御、ロケーションアクセス制御等、他の静的コンテキスト要件で補うことができる。特定された要件のすべてを満たすことができないコンポーネントは、ユーザに提示されるコンポーネントのリストから直ちに除外される。つまり、コンポーネントはコンテキスト要件を持ち、エンティティ（人）は静的および動的コンテキスト状態を持ち、フィルタリング時にこれらの状態と要件を組み合わせることができる。

10

【0087】

本発明のさらに別の面は、エンティティの使用履歴をコンポーネントの使用履歴と組み合わせ、戻されたコンポーネントを人気（またはその他のメトリクス）によって優先順位付けする。人気のメトリクスは、過去において使用されたコンポーネントが将来においても再び使用される可能性が高いため、興味深い。

【0088】

利用可能なコンポーネント／エンティティおよび／またはコンポーネントの相互作用をフィルタにかけ、これに優先順位を付ける方法のひとつは、その履歴情報をフィルタにかけ、これに優先順位を付けるプロセスを適用することである。

20

【0089】

図14は、環境内のコンポーネントから履歴コンテキストを収集するために使用できるフィルタ／優先順位付けプロセス1400を示す。フィルタ／優先順位付けプロセス1400は、ユーザ、エンティティの要求、環境の変化の検出あるいはその他のトリガにより、「開始」1401から始まる。フィルタ／優先順位付けプロセス1400は、「初期化」手順1403に進み、ここで、要求された初期化が行われ、次に、「履歴要求」手順1405に進む。「履歴要求」手順1405はまず、どのコンポーネントが環境内にあるかを検出し、履歴要求コマンドならびに創出（オリジネーティング: originating）コンポーネントのコンテキストとともに、環境内の関係するコンポーネント／エンティティの各々に送信する。「履歴受取」手順1407は、コンポーネント／エンティティが「履歴要求」手順1405からの要求に応える際（図13参照）に、そこから履歴を受け取る。リクエストコンポーネントは次に、「履歴に基づくコンポーネントフィルタ」手順1409を使い、指定されたフィルタプロパティによる特定に従って、コンポーネントの履歴情報および／または受け取った履歴情報に基づき、コンポーネントをフィルタにかけ、これに優先順位を付ける。「履歴に基づくコンポーネントフィルタ」手順1409の結果は「履歴に基づくコンポーネント優先順位付け」手順1411に送られ、これが優先順位決定パラメータに応じてその結果を順番に並べる。コンポーネントを特定する優先順位付けされた情報は次に、「結果提示」手順1413によってコンポーネントの順位リストとして提示され、これにより、エンティティ／コンポーネントは、たとえばその過去のコンテキスト履歴およびその他利用可能な情報に基づいて、どのコンポーネントを使用するかを選択できる。

30

40

【0090】

フィルタリングと優先順位付けがコンポーネント選択の援助を提供するのに不十分な場合がある。

【0091】

多数のエンティティ／コンポーネントが各々のコンテキスト履歴を保持しているユビキタスコンピューティング環境において、エンティティ／コンポーネントが位置する環境またはその環境の使用に関する推論を建てることは可能である。したがって、その環境の中の状況が変化した場合、エンティティ／コンポーネントの相対的重要性もまた変化する可能性がある。たとえば、プレゼンテーション中、プレゼンターにとってはおそらくメディ

50

アデバイスがその部屋の中で最も高い関心の的であり、一方、画像変換サービスやキャラクター認識コンポーネント等のコンポーネントはそれほどの重要性を持たないであろう。反対に、室内で、その他のデバイスより後者のコンポーネントのほうが容易により大きな重要性を持つことになる状況もある。

【0092】

本発明のひとつの実施形態は、環境、コンポーネント／エンティティのグループおよび／または情報を観察することのできるコンテキストモニタまたはコンテキストモニタシステムである。これらのモニタは、コンポーネントと組み合わせる必要がないという点でコンポーネントを必要とせず、コンテキストモニタの機能は、どのコンポーネント／エンティティが、そのコンテキストモニタが担当するモニタ対象の環境に入ってきたかを観察し、環境の状況的評価を提供することである。

10

【0093】

たとえば、会議室用のコンテキストモニタは、そのドメイン内に、その会議室内のすべてのプレゼンテーション媒体を有することができる。このコンテキストモニタはまた、室内のエンティティならびに照明、室温といった環境管理を監視することもできる。このコンテキストモニタはさらに、以前の事象の履歴も保持できる。このコンテキストモニタは、特定の数の人が室内にいることを検出し、情報を分析して環境の状況的評価を推定することが可能である。

【0094】

このコンテキストモニタは、室内のコンポーネントについて行われた動作を観察、記録し、この情報は推論エンジンに供給される。コンテキストモニタによって収集された、コンポーネントに対する動作に関するデータは、フィルタにかけ、コンポーネント／エンティティおよび推論エンジンに供給することができる。コンポーネントは、複数のコンテキストモニタで監視してもよい（エンティティは、そのエンティティのオフィスのためのコンテキストモニタおよびそのエンティティのプロジェクトグループのためのコンテキストモニタと関連することができる）。

20

【0095】

図15は、コンテキストモニタを開始するのに使用されるコンテキストモニタ初期化プロセス1500を示す。コンテキストモニタ初期化プロセス1500は、コンテキストモニタサービスが開始されると「開始」1501から始まり、必要な初期化を実施する「初期化」手順1503に進む。次に、コンテキストモニタ初期化プロセス1500は、「探索スレッド開始」手順1505と「推論エンジンスレッド開始」手順1507を使って、さまざまなスレッドを開始する。当業者は、図15にはこれらのスレッドによって特定の順序で開始されることが示されているものの、他の同等の実施形態でも別の順序でそのスレッドを開始することが可能であることを理解できる。

30

【0096】

図16は、「探索スレッド開始」手順1505によって呼び出すことができ、「開始」1513から始まる探索スレッド1512を示す。一旦開始されると、「初期化」手順1515はそのスレッドについての初期化を実施し、探索スレッド1512は「コンポーネント検出」手順1517に続く。「コンポーネント検出」手順1517は、環境内の他のエンティティ／コンポーネントを探索する。各エンティティ／コンポーネントが探索されると、探索スレッド1512は「コンポーネントに関するモニタスレッド開始」手順1519を開始し、探索されたコンポーネントと、監視対象となる環境内の他のコンポーネントとの間の通信と動作を監視するスレッドを始める。探索スレッド1512は次に、「コンポーネント検出」手順1517に戻り、環境内の別のエンティティ／コンポーネントを探索する。

40

【0097】

「コンポーネント検出」手順1517は、各種の方法でエンティティ／コンポーネントを検出できる。

【0098】

50

図17は、「推論エンジンスレッド開始」手順1507によって呼び出すことのできる推論エンジンスレッド1520を示す。推論エンジンスレッド1520は「開始」1521から始まり、「初期化」手順1523に進んで必要な初期化機能を実行する。次に、推論エンジンスレッド1520は「環境安定性判断」手順1525に進み、探索スレッド1512の活動を監視して、そのコンテキストモニタによって監視される環境がいつ、合理的推論をサポートするのに十分安定するかを判断する。推論エンジンそのものは、この判断を下すのに役立てるために使用でき、禁止対象の環境の過去における使用、ネットワーク接続されたカレンダーリングシステムの日付、時刻、状態およびその他、推論エンジンが利用できる情報を考慮することができる。

【0099】

「環境安定性判断」手順1525が満たされると、推論エンジンスレッド1520は「予備推論分析」手順1527に進み、環境内にあると検出されたエンティティ／コンポーネントを分析し、環境状況の予備評価を行う。

【0100】

次に、「エンティティ／コンポーネント要求受取」手順1529は、リクエストコンポーネントからの要求を待つ。要求を受け取ると、推論エンジンスレッド1520は「各要求分析」手順1531に進み、その要求に照らして、おそらく「予備推論分析」手順1527の結果を使って、その状況に推論エンジンを適用する。この分析結果は、「要求への応答」手順1533においてコンポーネント／エンティティ／コンピュータに戻され、推論エンジンスレッド1520は「エンティティ／コンポーネント要求受取」手順1529

【0101】

図18は、「コンポーネント検出」手順1517によって呼び出すことができるコンポーネント監視スレッド1550を示し、これは「開始」1551から始まり、「初期化」手順1553に進んで必要な初期化を行う。次に、コンポーネント監視スレッド1550は「履歴収集」手順1555に進み、ここでコンポーネント／エンティティからのコンテキスト履歴を取得し、コンテキスト履歴のどの部分が現在の環境に関係があるかを判断する。コンポーネント／エンティティからの関係するコンテキスト履歴は、「関係履歴の推論データベースとの併合」手順1557によって推論データベースと併合される。次に、コンポーネント監視スレッド1550は、「コンポーネント相互作用監視および推論データベース更新」手順1559に進み、これがエンティティ／コンポーネントの他のエンティティ／コンポーネントとの相互作用を監視し、推論データベースを更新する。「コンポーネント相互作用監視および推論データベース更新」手順1559は、継続的にコンポーネント／エンティティの相互作用を監視し、更新を実行する。

【0102】

「コンポーネント相互作用監視および推論データベース更新」手順1559が、監視対象のエンティティ／コンポーネントがある期間にわたり、他のコンポーネントと相互作用していないと判断した場合、コンポーネント監視スレッド1550はその監視対象のエンティティ／コンポーネントに休眠とマークする。定期的に、コンポーネント監視スレッド1550は「コンポーネント休眠」決定手順1561に進み、エンティティ／コンポーネントについて、それが休眠とマークされているか否かチェックする。そのエンティティ／コンポーネントが休眠とマークされていない場合、コンポーネント監視スレッド1550は「コンポーネント相互作用監視および推論データベース更新」手順1559に戻る。

【0103】

しかしながら、「コンポーネント休眠」決定手順1561において、禁止対称のエンティティ／コンポーネントが休眠とマークされていると、監視対象のエンティティ／コンポーネントは、そのコンテキストモニタによって監視される環境内にはなくなっている可能性がある。すると、コンポーネント監視スレッド1550は「タイマスタート」手順1563に進み、タイムアウトタイマをスタートさせ、次に「休眠コンポーネントポーリング」手順1565に進んで監視対象のエンティティ／コンポーネントにメッセージを送り、

10

20

30

40

50

応答を呼び出す。「タイマスタート」手順1563によってセットされたタイマが切れると、「コンポーネント応答」決定手順1567は、監視対象のモニタ／コンポーネントが応答したか否かを判断する。監視対象のエンティティ／コンポーネントが応答していれば、コンポーネント監視スレッド1550は「コンポーネントに非休眠とマーク」手順1569に進み、これがその監視対象のエンティティ／コンポーネントに非休眠とマークする。次に、コンポーネント監視スレッド1550は「コンポーネント相互作用監視および推論データベース更新」手順1559に進み、監視対象のエンティティ／コンポーネントの監視を続ける。

【0104】

これに対し、「コンポーネント応答」決定手順1567において、監視対象のエンティティ／コンポーネントが応答していないと、コンポーネント監視スレッド1550は「スレッド終了」手順1571に進み、リソースを解除してスレッドを終了する。

10

【0105】

実施形態の別の面は、コンテキスト情報を使ってセキュリティ情報にアクセスし、これに使用することである。つまり、セキュリティデータは、（先行技術において行われているように）プログラムインタフェースを通じてセキュリティ情報を明確に表現することを要求するのではなく、コンテキストプロパティとして維持することができる。コンポーネントが相互からデータを得る、あるいは相互に対する動作を要求すると、そのコンテキストによって提供されるセキュリティ情報をどのように使用するか（またはこのセキュリティ情報を使用するか否か）は関係者に任せられる。したがって、異なるセキュリティモデルおよびテクニックについて、均一なプログラムインタフェースを使用することができる。

20

【0106】

リクエストコンポーネント／エンティティが、サービスコンポーネントに対する動作を実行したい場合、リクエストコンポーネントはセキュリティ情報（たとえば証明書、パスワードまたはそのアイデンティティのその他の証明）をサービスコンポーネントに伝えなければならない。このセキュリティ情報は、コンテキストプロパティとすることができる。コンポーネント間のすべての通信において、コンテキストをパラメータとして渡すことが必要となるため、リクエストコンポーネントからのセキュリティ情報をサービスコンポーネントに提供することができる。サービスコンポーネントがセキュリティ情報を使用すると決定した場合、これは、リクエストコンポーネント／エンティティの証明を認証機関に送り、検証を受けることができる。また、エンティティは、リクエストコンポーネント／エンティティによるその後の改良のために、ゲスト証明を提供することも可能である。

30

【0107】

他のコンポーネントが「知る」ことのできるプロパティキーを有する、きわめて一般的に使用されると特定のコンテキストパラメータがある。同様に、確実な証明で認証されるコンポーネントは、よく知られたキーを持つコンテキストプロパティを有する。セキュリティプロパティは単なるコンテキスト情報であるため、セキュリティプロパティは、他のコンテキストプロパティと同様に、コンポーネントのニーズに応じて要求してもしなくてもよい。さらに、証明プロパティをコンポーネントの残りのプロパティとともに提供することができる。

40

【0108】

サービスコンポーネント／エンティティに、セキュリティ情報を使用するか否かを決定させることにより、サービスコンポーネントは、特定の状況におけるアクセス方針の例外を容易に作ることができる。

【0109】

オプションのコンテキストアクセス方針705とオプションのプロパティアクセス方針フィールド761はそれぞれ、オプションのアクセス管理集合707およびオプションのアクセス管理集合765（プロパティ用）とともに使用することができる。これらの方針には、どのような種類の認証が必要で、どのような状況において認証を無効化できるかを

50

決定するルールが含まれる。

【0110】

さらに、本発明の実施形態のこの面により、そのコンポーネントによって使用されるプログラムインタフェースを変更することを必要とせずに、認証戦略を変更する柔軟性が得られる。相互に信用される証明付与／認証機関が適当な場合もあれば、エンティティが相互に関する公開鍵を保有する、「公開鍵暗号方式」を使用するコンポーネントもある。さらに別のケースでは、ひとつのコンポーネントを別のコンポーネントに「ログオン」できるようにするパスワードを必要とする。これらのケースはいずれも、そのコンポーネントのコンテキストを使用する、同じプログラムインタフェースによって扱われる。

【図面の簡単な説明】

10

【0111】

【図1】本発明の実施形態によるコンテキスト情報を提供するためのシステムの斜視図である。

【図2】コンテキスト情報を提供するためのシステムにおいて利用される任意コンポーネントの一例のブロック図である。

【図3】本発明の実施形態によるコンテキスト情報を提供するためのシステムの部分斜視図である。

【図4】コンテキスト情報を提供するプロセスのフローチャートである。

【図5】本発明の実施形態によるコンテキスト情報を提供するためのシステムの部分斜視図である。

20

【図6】コンテキスト情報を提供するプロセスのフローチャートである。

【図7】本発明の実施形態によるコンポーネントのコンテキストの組成を示す図である。

【図8】本発明の実施形態によるコンポーネントのコンテキスト履歴の組成を示す図である。

【図9】本発明の実施形態によるコンポーネントのコンテキストアクセス制御の組成を示す図である。

【図10】本発明の実施形態によるコンポーネントのプロパティの組成を示す図である。

【図11】本発明の実施形態によるコンポーネントのプロパティ履歴の組成を示す図である。

【図12】ある実施形態によるコンテキストアクセスプロセスを示す図である。

30

【図13】ある実施形態によるコンテキスト履歴を提供するプロセスを示す図である。

【図14】ある実施形態によるフィルタ／優先順位付けプロセスを示す図である。

【図15】ある実施形態によるコンテキストモニタを初期化するプロセスを示す図である。

。

【図16】ある実施形態によるコンテキストモニタの探索スレッドを示す図である。

【図17】ある実施形態による推論エンジンスレッドを示す図である。

【図18】ある実施形態によるコンポーネントモニタスレッドを示す図である。

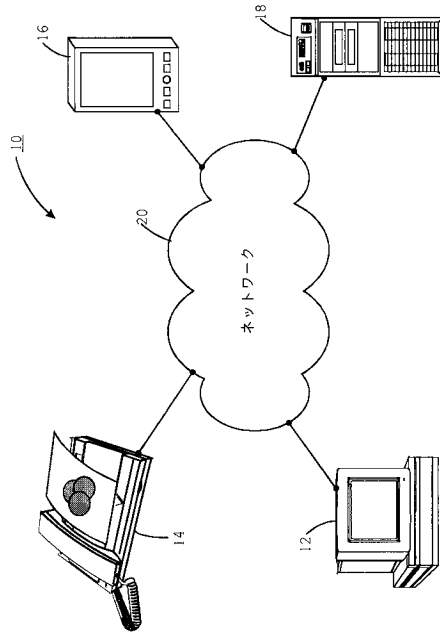
【符号の説明】

【0112】

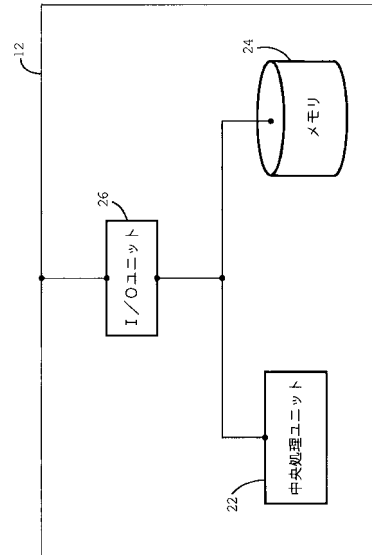
10 システム、12 コンピュータ、14a 汎用インタフェースセット、14b データオブジェクト、14c コンテキストオブジェクト、14 プリンタ、16a 汎用インタフェースセット、16b データオブジェクト、16c コンテキストオブジェクト、18 サーバ、20 ネットワーク、22 中央処理ユニット、24 メモリ、26 I/Oユニット。

40

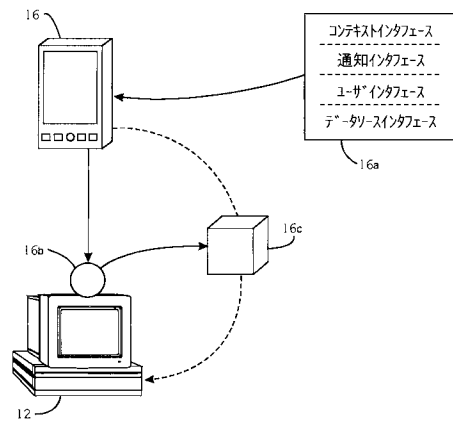
【図 1】



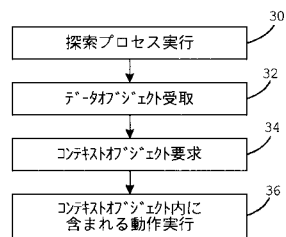
【図 2】



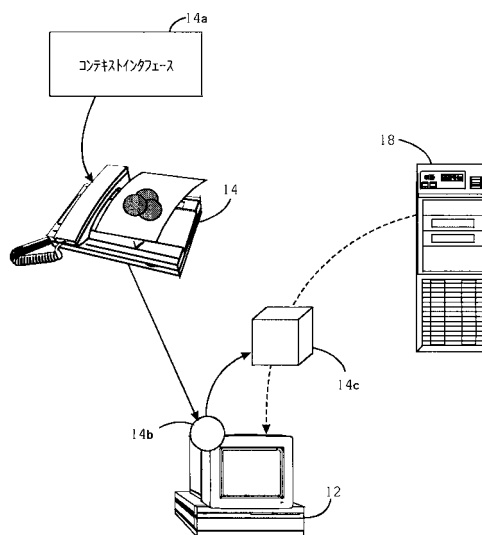
【図 3】



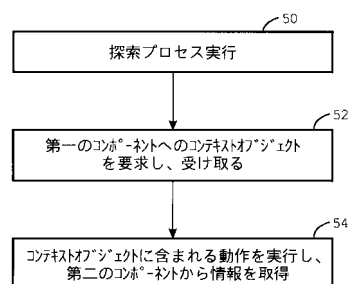
【図 4】



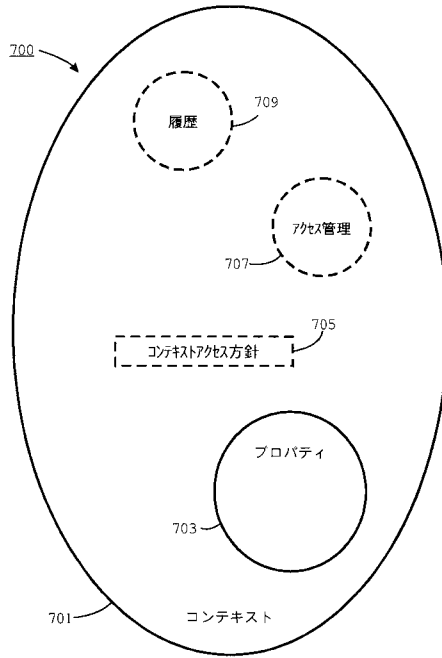
【図 5】



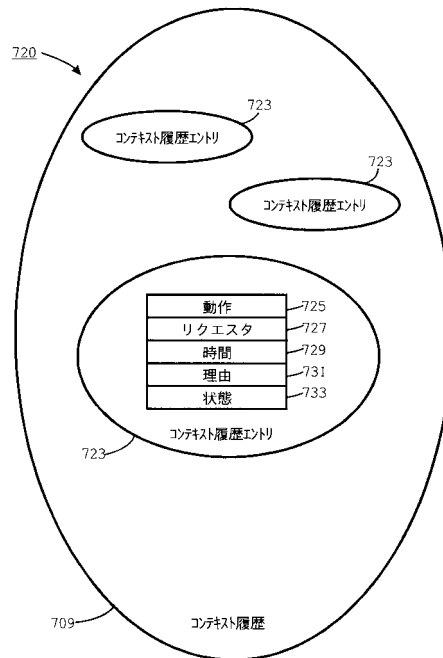
【図 6】



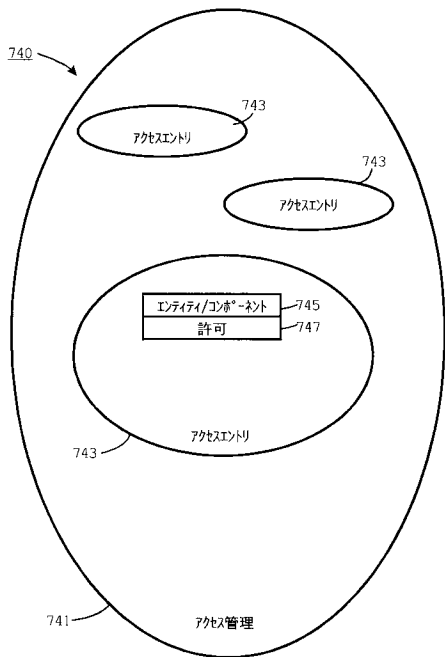
【図 7】



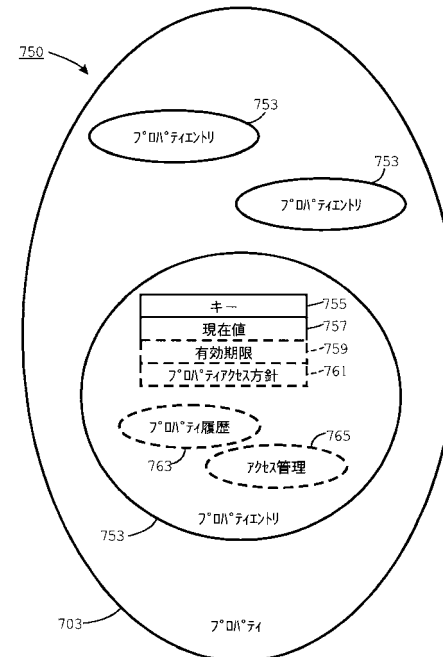
【図 8】



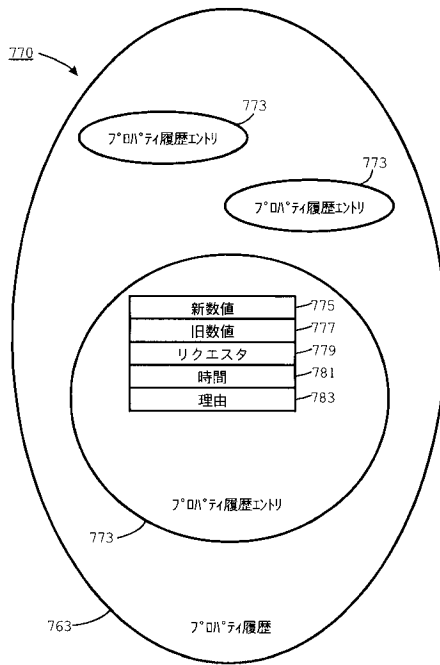
【図 9】



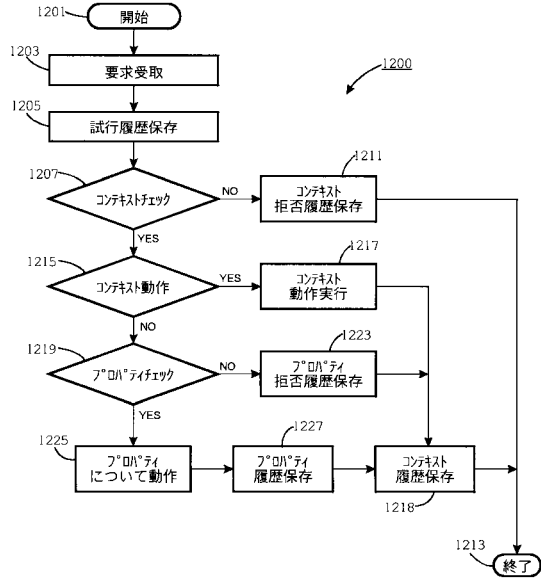
【図 10】



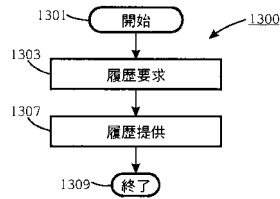
【図 11】



【図 12】



【図 13】



【図 14】

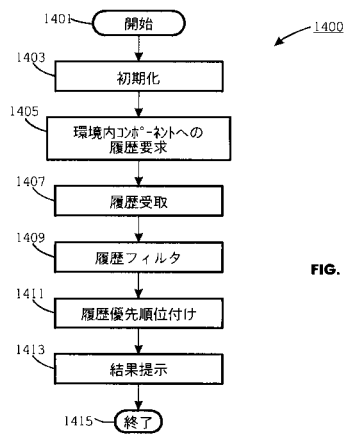
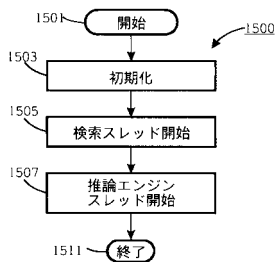
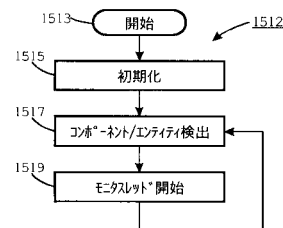


FIG. 14

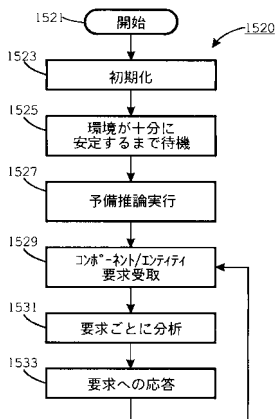
【図 15】



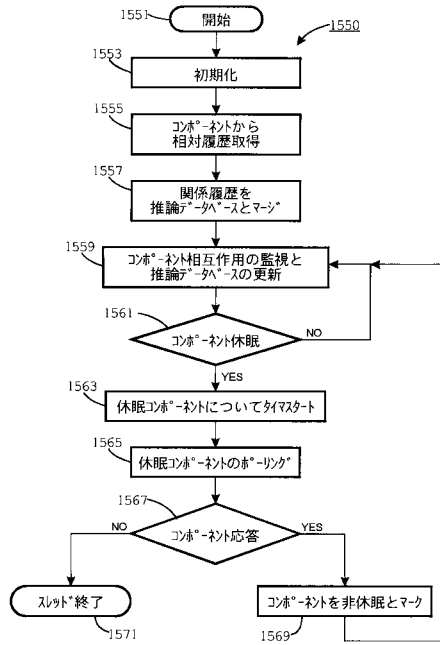
【図 16】



【図 17】



【図 18】



フロントページの続き

- (72)発明者 ダブリュ ケイス エドワード
アメリカ合衆国 カリフォルニア サンフランシスコ ウィンフィールド ストリート 83
(72)発明者 ジャナ ゼット セディヴィ
アメリカ合衆国 カリフォルニア パロ アルト カートナー アベニュー #22 250

審査官 須藤 竜也

- (56)参考文献 特開2002-191039 (JP, A)
特開2002-334162 (JP, A)
特開2002-269040 (JP, A)

- (58)調査した分野(Int.Cl., DB名)
G06F 13/00