(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第4716729号
(P4716729)

(45)発行日　平成23年7月6日(2011.7.6)　(24)登録日　平成23年4月8日(2011.4.8)

(51)Int.Cl.　　F I
G06F 21/24 (2006.01)　　G06F 12/14 520A
G06F 12/00 (2006.01)　　G06F 12/14 560B
　　G06F 12/00 513D
　　G06F 12/00 537A

請求項の数 7　(全 25 頁)

(21)出願番号　特願2004-543023 (P2004-543023)
(86)(22)出願日　平成15年9月25日 (2003.9.25)
(65)公表番号　特表2006-502494 (P2006-502494A)
(43)公表日　平成18年1月19日 (2006.1.19)
(86)国際出願番号　PCT/US2003/030337
(87)国際公開番号　WO2004/034186
(87)国際公開日　平成16年4月22日 (2004.4.22)
審査請求日　平成18年9月22日 (2006.9.22)
(31)優先権主張番号　10/264,243
(32)優先日　平成14年10月3日 (2002.10.3)
(33)優先権主張国　米国(US)

(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】データに関するセキュリティを提供する方法、及びそのコンピュータ・プログラム

(57)【特許請求の範囲】
【請求項１】
ユーザ・データに関するセキュリティを提供する方法であって、
前記ユーザ・データを含むデータベースにアクセス可能なサーバ・コンピュータが、
前記サーバ・コンピュータにネットワークを介して接続されたクライアント・コンピュータの所与のユーザについてのユーザ・データが存在する複数のフィールドのそれぞれを反映する少なくとも１つのセキュリティ・リストを生成するステップであって、前記複数のフィールドのそれぞれは前記所与のユーザについての定義済みセキュリティ規則を有している、前記生成するステップと、
前記所与のユーザについて、前記少なくとも１つのセキュリティ・リストのユーザ固有インスタンスを生成するステップであって、
ａ）前記セキュリティ・リストにおける各エントリの各フィールドについて、ユーザ・データが前記所定のユーザに固有であるフィールドについて存在するかどうかを決定するステップと、
ｂ）存在しない場合に、前記セキュリティ・リストから前記エントリを取り除くステップと
を含む、前記生成するステップと、
前記ユーザ固有インスタンスの生成後に、前記データベースに対して発行された前記所与のユーザからのクエリを受信するステップであって、前記クエリが少なくとも１つのフィールドと当該少なくとも１つのフィールドに関連付けられた値とを有する、前記受信す

るステップと、
前記セキュリティ・リストが前記クエリの前記少なくとも１つのフィールドに対応するエントリを含むかどうかを決定するために前記セキュリティ・リストにアクセスするステップと、
前記セキュリティ・リストが前記クエリの前記少なくとも１つのフィールドに対応するエントリを含むことに応じて、前記クエリの前記少なくとも１つのフィールドに関連付けられたセキュリティ規則を実施するステップであって、前記クエリ中のフィールドについて特定されたセキュリティ規則が前記ユーザ・データの存在に基づいて実施されて、前記クエリの実行をさせないこと又は前記クエリの発行のログを記録することが行われる、前記実施するステップと
を実行することを含む、前記方法。
【請求項２】
前記クエリがＳＱＬクエリである、請求項１に記載の方法。
【請求項３】
前記サーバ・コンピュータが、前記セキュリティ規則を呼び出すステップをさらに実行することを含み、当該呼び出しが、前記クエリを実行すべきかどうかを判断することを含む、請求項１に記載の方法。
【請求項４】
前記サーバ・コンピュータが、前記クエリの前記少なくとも１つのフィールドについて前記ユーザ・データの値にかかわらず、前記クエリの前記少なくとも１つのフィールドについて前記ユーザ・データの存在を判断した後、前記セキュリティ規則を呼び出さなければならないかどうかを判断するステップをさらに実行すること含む、請求項１に記載の方法。
【請求項５】
前記ユーザ固有インスタンスを生成するステップが、前記所与のユーザがログオンする時に実行される、請求項１に記載の方法。
【請求項６】
前記サーバ・コンピュータが、
前記サーバに接続されたクライアントのユーザからのサインオンに応じて、当該ユーザに関するアクセス型のセキュリティ・リストを取り出すステップであって、当該アクセス型はセキュリティ型の一つであり、当該アクセス型のセキュリティ・リストは、フィールド型の値であるフィールド値によらず、当該セキュリティ・リストにおいて定義されたユーザ及びフィールド型に基づいてアクセスを制限するために使用され、前記フィールド型はファースト・ネーム、ラスト・ネーム又は住所を包含する、前記取り出すステップと、
前記取り出されたアクセス型のセキュリティ・リストのフィールドごとにループを開始するステップであって、
ユーザ・データの対応するフィールド内にエントリが存在するかどうかを検査するステップと、
前記エントリが存在しない場合に、前記アクセス型のセキュリティ・リストから処理中のフィールドを取り除くステップと、
前記エントリが存在する場合に、前記検査するステップに戻り、別のフィールド内のエントリについて検査するステップと
を含む、前記開始するステップと、
前記フィールドごとの前記検査が終了したことに応じて、値型のセキュリティ・リストを検索するステップであって、当該値型はセキュリティ型の一つであり、当該値型のセキュリティ・リストは、当該値型のセキュリティ・リストにおいて定義されたユーザに関する情報及びフィールド型の値である特定のフィールド値に基づいてアクセスを制限するために使用される、前記検索するステップと、
前記検索された値型のセキュリティ・リストのフィールドごとにループを開始するステップであって、

ユーザ・データの対応するフィールド内にエントリが存在するかどうかを決定するステップと、
前記エントリが存在しない場合に、前記値型のセキュリティ・リストから処理中のフィールドを取り除くステップと、
前記エントリが存在する場合に、対応するユーザ固有エントリの値を前記値型のセキュリティ・リストに格納するステップと、
を含む、前記開始するステップと
をさらに実行することを含む、請求項１に記載の方法。
【請求項７】
ユーザ・データに関するセキュリティを提供するコンピュータ・プログラムであって、サーバ・コンピュータに、請求項１～６のいずれか一項に記載の方法の各ステップを実行させるコンピュータ・プログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、一般には、データ処理に関し、より詳細には、データを物理的に表現する特定の様式とは独立にデータにアクセスすることに関する。
【背景技術】
【０００２】
データベースは、コンピュータ化された情報格納／取り出しシステム（informationstorage and retrieval system）である。リレーショナル・データベース管理システムは、データを格納し取り出すのにリレーショナル（relational）技法を使用するコンピュータ・データベース管理システム（ＤＢＭＳ、databasemanagement system）である。最も優勢なタイプのデータベースは、リレーショナル・データベース、すなわちデータを多くの異なるやり方で再編成しアクセスできるように定義した表形式の（tabular）データベースである。
【０００３】
具体的なアーキテクチャによらず、ＤＢＭＳでは、要求側エンティティ（たとえば、アプリケーションまたはオペレーティング・システム）が、指定したデータベースへのアクセスを要求を、データベース・アクセス要求を発行することによって行う。そのような要求は、たとえば、単純なカタログ・ルックアップまたはデータベース内の指定したレコードを読み出し、変更し、追加するように動作するトランザクション、およびトランザクションの組合せを含むことができる。こうした要求は、ＳＱＬ（Structured Query Language、構造化問い合わせ言語）などの高水準問い合わせ言語を用いて行われる。例示すると、ＳＱＬは、インターナショナル・ビジネス・マシーンズ（ＩＢＭ、InternationalBusiness Machines）社のＤＢ２、マイクロソフト（Microsoft）社のＳＱＬ　Ｓｅｒｖｅｒや、オラクル（Oracle）社、サイベース（Sybase）社、およびコンピュータ・アソシエーツ（ComputerAssociates）社製のデータベース製品などのデータベースから情報を得て更新するために対話的な問い合わせを行うのに使用される。「問い合わせ」という用語は、保管されているデータベースからデータを取り出すための１組のコマンドの名称である。問い合わせは、プログラマおよびプログラムに、選択、挿入、更新、データのロケーションの発見などをさせるコマンド言語の形をとっている。
【０００４】
データベースの文脈における１つの顕著な問題はセキュリティである。データベースは、しばしば、アクセスから保護すべきほどにセキュリティの必要な、秘密の（confidential）、そうでなければ慎重さを要する（sensitive）資料を含んでいる。たとえば、医療記録は個人性および秘密性が高いと考えられている。したがって、医療記録へのアクセスは、通常、選ばれたユーザに制限される。このために、従来のデータベース管理システムでは、認可のレベルを指定するユーザ・プロファイルを実装する場合がある。ユーザがある特定のデータにアクセスしてよいかどうかは、それぞれのプロファイルに指定されたそ

のユーザの認可のレベルに依存する。
【発明の開示】
【発明が解決しようとする課題】
【０００５】
しかし、以上のアプローチは、きわめて柔軟性に乏しく静的である。他方、データは、きわめて動的である（すなわち、時間とともに変化する）。その結果、ユーザの従来のデータベースに課されたセキュリティは、そのユーザがアクセスしようと試みるデータに関して適切であることも、そうでないこともある。たとえば、ＨＩＶテストの結果にアクセスしようと試みるあるユーザを考えよう。このユーザがＨＩＶテストの結果にアクセスさせないようにするのを、その値（すなわち、ポジティブかネガティブか）にかかわらず、このユーザ自身の医療記録に、このユーザがＨＩＶテストを受けたことが反映されている場合だけにすることが望ましいことがある。第１のアクセスの試行中に、このユーザの医療記録がＨＩＶテストを反映していない場合には、このユーザは、ＨＩＶテストの結果について医療記録の問い合わせを行うことが許される。同じユーザが、引き続いてＨＩＶテストを受け、それが次いでこのユーザの医療記録に反映された場合には、これに引き続くＨＩＶテストの結果にアクセスする試行は許されないはずである。あるいは、このユーザがＨＩＶテストの結果にアクセスさせないようにするのを、このユーザ自身の医療記録に、ポジティブというＨＩＶテストの結果が反映されている場合だけにすることが望ましいこともある。いずれの場合でも、従来のデータベース管理システムでは、このレベルの柔軟性が可能ではない。
【０００６】
したがって、データベースのための改良されたセキュリティ機構が必要とされている。
【課題を解決するための手段】
【０００７】
本発明は、一般に、データベース・セキュリティのための方法、システム、および製造物を対象とする。
【０００８】
一実施形態では、データに関するセキュリティを提供する方法が提供される。この方法は、ユーザがデータベースに対して発行した、少なくとも１つのフィールドおよび前記少なくとも１つのフィールドに関連する値で構成される問い合わせを受け取ること、および、前記少なくとも１つのフィールドについての前記ユーザに固有のデータの存在に基づいて、前記少なくとも１つのフィールドについて前記ユーザ向けに指定されたセキュリティ規則を呼び出さなければならないかどうかを判断することを含む。
【０００９】
データに関してセキュリティを提供する別の方法は、所与のユーザについてデータが存在する複数のフィールドのそれぞれを反映する少なくとも１つのセキュリティ・リストを生成し、前記複数のフィールドのそれぞれは前記所与のユーザについての定義済みセキュリティ規則を有しており、前記所与のユーザがデータベースに対して発行した問い合わせを受け取り、前記問い合わせは、少なくとも１つのフィールドおよび前記少なくとも１つのフィールドに関連する値で構成されており、前記問い合わせの前記少なくとも１つのフィールドが前記少なくとも１つのセキュリティ・リストに反映されているかどうかを判断するために、前記少なくとも１つのセキュリティ・リストにアクセスし、前記問い合わせの前記少なくとも１つのフィールドが前記少なくとも１つのセキュリティ・リストに反映されている場合に、前記セキュリティ規則を実施する（enforce）ことを含む。
【００１０】
別の実施形態では、特定の物理データ表現を有するデータにセキュリティを提供する方法は、抽象的問い合わせを定義するために複数の論理フィールドを含む問い合わせ仕様を提供し、前記複数の論理フィールドを前記データの物理エンティティに対応付ける対応付け規則を提供し、前記複数の論理フィールドについてユーザ固有のセキュリティ規則を提供し、ユーザが前記データに対して発行した抽象的問い合わせを受け取り、前記抽象的問

い合わせは、前記問い合わせ仕様に従って定義され、少なくとも１つの論理フィールドおよび前記少なくとも１つの論理フィールドに関連する値で構成されており、前記少なくとも１つの論理フィールドについての前記ユーザに固有のデータの存在に基づいて、前記少なくとも１つの論理フィールドについて前記ユーザ向けに指定されたセキュリティ規則を呼び出さなければならないかどうかを判断することを含む。

【００１１】

別の実施形態では、ユーザがデータベースに対して発行した、少なくとも１つのフィールドおよび前記少なくとも１つのフィールドに関連する値で構成される問い合わせを受け取り、前記少なくとも１つのフィールドについての前記ユーザに固有のデータの存在に基づいて、前記少なくとも１つのフィールドについて前記ユーザ向けに指定されたセキュリティ規則を呼び出さなければならないかどうかを判断することを含むセキュリティ検証動作を、実行時にコンピュータに行わせるセキュリティ検証命令を含む、コンピュータ可読媒体が提供される。

【００１２】

別の実施形態では、セキュリティ検証動作を、実行時にコンピュータに行わせるセキュリティ検証命令を含む、コンピュータ可読媒体が提供される。前記セキュリティ検証動作は、所与のユーザについてデータが存在する複数のフィールドのそれぞれを反映する少なくとも１つのセキュリティ・リストを生成し、前記複数のフィールドのそれぞれは、前記所与のユーザについての定義済みルールを有し、前記所与のユーザがデータベースに対して発行した問い合わせを受け取り、前記問い合わせは、少なくとも１つのフィールドおよび前記少なくとも１つのフィールドに関連する値で構成されており、前記問い合わせの前記少なくとも１つのフィールドが前記少なくとも１つのセキュリティ・リストに反映されているかどうかを判断するために、前記少なくとも１つのセキュリティ・リストにアクセスし、前記問い合わせの前記少なくとも１つのフィールドが、前記少なくとも１つのセキュリティ・リストに反映されている場合に、前記セキュリティ規則を実施することを含む。

【００１３】

別の実施形態では、格納された情報をその媒体上に含む、コンピュータ可読媒体であって、前記情報が、抽象的な問い合わせを定義するために複数の論理フィールドを含む問い合わせ仕様と、前記複数の論理フィールドを前記データの物理エンティティに対応付ける複数の対応付け規則と、前記複数の論理フィールドについての複数のユーザ固有のセキュリティ規則と、ユーザが前記データに対して発行した抽象的問い合わせを受け取るのに応答してセキュリティ検証動作をコンピュータに行わせるように実行可能なランタイム・コンポーネントとを含み、前記抽象的問い合わせは前記問い合わせ仕様に従って定義され、少なくとも１つの論理フィールドおよび前記少なくとも１つの論理フィールドに関連する値で構成される。前記セキュリティ検証動作は、前記少なくとも１つの論理フィールドについての前記ユーザに固有のデータの存在に基づいて、前記少なくとも１つの論理フィールドについて前記ユーザ向けに指定されたセキュリティ規則を呼び出さなければならないかどうかを判断することを含む。

【００１４】

本発明の上に挙げた諸特徴を達成する態様を詳細に理解できるように、上で簡潔に要約した本発明のより具体的な説明を、添付の図面に示すその実施形態への参照によって得ることができる。

【００１５】

しかし、本発明には他の均等に効果的な実施形態を許すことができるため、添付の図面には、本発明の典型的な実施形態のみを示しており、したがってその範囲を限定するものと見なすべきでないことに留意されたい。

【発明を実施するための最良の形態】

【００１６】

導入

本発明は、一般には、データへのアクセスを制限するためのシステム、方法、および製品（articleof manufacture）を対象とする。一般に、データ・アクセスの制限は、問い合わせ内で要求中のデータとその問い合わせを発行するユーザとの間の関連付けに従って行う。

【００１７】

一実施形態では、セキュリティ規則を、データの論理モデルの一部としてインプリメントする。論理モデルは、下位にある（underlying）データ・リポジトリの論理ビュー（logical view）を提供する、データ・リポジトリ抽象化（datarepository abstraction）レイヤとしてインプリメントする。このようにして、データを、そのデータを物理的に表現する特定の様式とは独立させる。また、問い合わせ抽象化レイヤ（queryabstraction layer）も提供するが、これはデータ・リポジトリ抽象化レイヤに基づく。ランタイム・コンポーネントは、抽象的問い合わせを、特定の物理データ表現と突き合わせて使用できる形への翻訳を行う。

【００１８】

本発明の一実施形態は、たとえば図１に示し以下で説明するコンピュータ・システムなどのコンピュータ・システムとともに使用するためのプログラム製品として実装する。このプログラム製品のプログラムは、（本明細書で説明する諸方法を含む）諸実施形態の機能を定義するものであり、また様々な信号担持媒体上に収容することができる。例示的な信号担持媒体は、（ｉ）非書き込み可能なストレージ・メディア（たとえば、ＣＤ－ＲＯＭドライブが読み取り可能なＣＤ－ＲＯＭディスクなどコンピュータ内部の読み取り専用記憶装置）上に永続的に格納した情報、（ｉｉ）書き込み可能なストレージ・メディア（たとえば、ディスケット・ドライブ内部のフロッピー（Ｒ）・ディスクやハードディスク・ドライブ）上に格納した変更可能な情報、または（ｉｉｉ）無線通信を含めて、コンピュータ・ネットワークや電話網を通してなどの、通信媒体がコンピュータへと運ぶ情報を含むが、これに限定されるものではない。最後の実施形態は、インターネットおよび他のネットワークからダウンロードした情報を特に含む。そのような信号担持媒体は、本発明の機能を支持するコンピュータ可読命令を担うとき、本発明の諸実施形態に相当する。

【００１９】

一般に、本発明の諸実施形態をインプリメントするために実行するルーチンは、オペレーティング・システムの一部、あるいは、特定のアプリケーション、コンポーネント、プログラム、モジュール、オブジェクト、または命令の系列とすることができる。本発明のソフトウェアは、通常、ネイティブ（native）コンピュータが機械可読フォーマット、したがって実行可能命令へと翻訳することになる数多くの命令からなる。また、プログラムは、そのプログラムにローカルに存在する、あるいは、メモリ内またはストレージ装置上に見出される変数およびデータ構造からなる。さらに、本明細書でこの後説明する様々なプログラムは、本発明の特定の実施形態でそれらプログラムを実装する目的となる適用によって特定することができる。しかし、以下のどのような特定の命名法も便宜的に用いるものに過ぎず、したがって本発明は、そのような命名法の特定しまたは示唆する、あるいはその双方に相当する任意の特定の適用のみにおける使用に限定されるべきでないことを理解されたい。

【００２０】

環境の物理ビュー（physical view）

図１には、本発明の諸実施形態を実装できるネットワーク化システム１００の構成図を示している。一般に、ネットワーク化システム１００は、クライアント（たとえば、ユーザの）コンピュータ１０２（そのようなクライアント・コンピュータ１０２　３台を示す）および少なくとも１つのサーバ１０４（そのようなサーバ１０４　１台を示す）を含む。クライアント・コンピュータ１０２およびサーバ・コンピュータ１０４は、ネットワーク１２６を介して接続されている。一般に、ネットワーク１２６は、ＬＡＮ（local area network、構内通信網）またはＷＡＮ（wide area network、広域通信網）あるいはそれらの組合せであってよい。特定の実施形態では、ネットワーク１２６はインターネットで

ある。

【００２１】

クライアント・コンピュータ１０２は、バス１３０を介してメモリ１１２、ストレージ１１４、入力装置１１６、出力装置１１９、およびネットワーク・インターフェース装置１１８に接続される中央処理装置（ＣＰＵ、Central Processing Unit）１１０を含む。入力装置１１６は、クライアント・コンピュータ１０２に入力を与えるためのどのような装置とすることもできる。たとえば、キーボード、キーパッド、ライト・ペン、タッチ・スクリーン、トラック・ボール、または音声認識ユニット、オーディオ／ビデオ・プレイヤーなどを使用することもできる。出力装置１１９は、ユーザに出力を与えるどのような装置、たとえば、従来のどのようなディスプレイ画面とすることもできる。入力装置１１６とは別個に示しているが、出力装置１１９と入力装置１１６を組み合わせることもできる。たとえば、タッチ・スクリーンを組み込んだディスプレイ画面、キーボードを組み込んだディスプレイ、またはテキスト・スピーチ変換器を組み合わせた音声認識ユニットを使用することもできる。

【００２２】

ネットワーク・インターフェース装置１１８は、クライアント・コンピュータ１０２とサーバ・コンピュータ１０４の間のネットワーク１２６を介したネットワーク通信を可能にするように構成されたどのような入口／出口装置（entry/exit device）であってもよい。たとえば、ネットワーク・インターフェース装置１１８は、ネットワーク・アダプタまたは他のネットワーク・インターフェース・カード（ＮＩＣ、networkinterface card）とすることもできる。

【００２３】

ストレージ１１４は、好ましくは、ＤＡＳＤ（Direct Access StorageDevice、直接アクセス記憶装置）である。これは単一のユニットとして示しているが、固定ディスク・ドライブ（fixed disc drives）、フロッピー（Ｒ）・ディスク・ドライブ、テープ・ドライブ、リムーバブル・メモリ・カード、または光ストレージなど、固定またはリムーバブル記憶装置あるいはその両方の組合せとすることもできる。メモリ１１２およびストレージ１１４は、複数の１次および２次ストレージ装置にまたがる１つの仮想アドレス空間の一部とすることもできる。

【００２４】

メモリ１１２は、好ましくは、本発明の必要なプログラミングおよびデータ構造を保持するのに十分な大きさのＲＡＭ（random access memory、ランダム・アクセス・メモリ）である。メモリ１１２を単一のエンティティとして示しているが、メモリ１１２は、実際には複数のモジュールを含むことができ、メモリ１１２は、高速のレジスタおよびキャッシュから低速ながらより大きなＤＲＡＭチップに至る、複数のレベルで存在することができることを理解されたい。

【００２５】

例示的に、メモリ１１２は、オペレーティング・システム１２４を収容している。有利になるよう使用してよい、例示的なオペレーティング・システムとしては、Ｌｉｎｕｘおよびマイクロソフト社（Microsoft）のＷｉｎｄｏｗｓ（登録商標）がある。より一般には、本明細書に開示する機能をサポートするどのようなオペレーティング・システムを使用してもよい。

【００２６】

また、メモリ１１２は、ＣＰＵ１１０上で実行すると、様々なサーバ１０４の間をナビゲートし（navigating）、サーバ１０４の１つまたは複数にあるネットワーク・アドレスを見出すためのサポートを提供する、ブラウザ・プログラム１２２を収容するものとして示している。一実施形態では、ブラウザ・プログラム１２２は、ＷｅｂベースのＧＵＩ（GraphicalUser Interface、グラフィカル・ユーザ・インターフェース）を含み、これにより、ユーザはＨＴＭＬ（HyperText Markup Language、ハイパーテキスト記述言語）情報を表示することができるようになる。しかし、より一般には、ブラウザ・プログラム１

２２は、サーバ・コンピュータ１０４から送信される情報をレンダリングする能力のある（好ましくはＧＵＩベースの）どのようなプログラムであってもよい。

【００２７】

サーバ・コンピュータ１０４は、クライアント・コンピュータ１０２と類似のやり方で物理的に配置することができる。したがって、サーバ・コンピュータ１０４を、バス１３６が互いに結合するＣＰＵ１３０、メモリ１３２、およびストレージ装置１３４を含むものとして一般的に示している。メモリ１３２は、サーバ・コンピュータ１０４上に配置された必要なプログラミングおよびデータ構造を保持するのに十分な大きさのＲＡＭ（random access memory、ランダム・アクセス・メモリ）とすることができる。

【００２８】

サーバ・コンピュータ１０４は、一般に、メモリ１３２内にあるように示したオペレーティング・システム１３８の制御下にある。オペレーティング・システム１３８の例は、ＩＢＭ　ＯＳ／４００（登録商標）、ＵＮＩＸ（登録商標）、Ｍｉｃｒｏｓｏｆｔ　Ｗｉｎｄｏｗｓ（登録商標）などである。より一般に、本明細書に記載の機能をサポートする能力のあるどのようなオペレーティング・システムを使用するのでもよい。

【００２９】

メモリ１３２は、１つまたは複数のアプリケーション１４０および抽象的問い合わせインターフェース（abstract query interface）１４６をさらに含む。アプリケーション１４０および抽象的問い合わせインターフェース１４６は、コンピュータ・システム１００内の様々なメモリおよびストレージ装置に様々なときに常駐する複数の命令を含むソフトウェア製品である。サーバ１０４内の１つまたは複数のプロセッサ１３０により読み込まれ実行されると、アプリケーション１４０および抽象的問い合わせインターフェース１４６は、コンピュータ・システム１００に、本発明の様々な態様を実施する諸ステップおよび諸要素を実行するのに必要な諸ステップを実行させる。アプリケーション１４０（ならびに、より一般には、オペレーティング・システム１３８および最も高水準ではユーザを含めて、どのような要求側エンティティ（requestingentity）も）データベース（たとえば、まとめてデータベース１５６と呼ぶデータベース１５６$_1$．．．１５６$_N$）に対して問い合わせを発行する。例示的に、データベース１５６を、ストレージ１３４内のＤＢＭＳ（databasemanagement system、データベース管理システム）の一部として示している。データベース１５６により、特定の物理表現によらず、どのようなデータの集まりも代表させている。例示として、データベース１５６は、（ＳＱＬ問い合わせによってアクセス可能な）リレーショナル・スキーマに従って、または（ＸＭＬ問い合わせによってアクセス可能な）ＸＭＬスキーマに従って編成することができる。しかし、本発明は、特定のスキーマに限定されるものではなく、現在知られていないスキーマへの拡張を企図するものである。本明細書で使用するように、「スキーマ」という用語は、データのある特定の構成を総称的に指すものとする。

【００３０】

一実施形態では、アプリケーション１４０の発行する問い合わせの定義は、アプリケーション１４０のそれぞれとともに含まれるアプリケーション問い合わせ仕様１４２に従って行う。アプリケーション１４０の発行する問い合わせは、あらかじめ定義する（すなわち、アプリケーション１４０の一部としてハード・コードする）のでもよく、または入力（たとえば、ユーザ入力）に応答して生成するのでもよい。どちらの場合でも、問い合わせ（本明細書では「抽象的問い合わせ」（abstract query）と呼ぶ）の作成／実行は、抽象的問い合わせインターフェース１４６の定義する論理フィールドを用いて行う。具体的には、抽象的問い合わせで使用する論理フィールドを、抽象的問い合わせインターフェース１４６のデータ・リポジトリ抽象化コンポーネント（datarepository abstraction component）１４８によって定義する。抽象的問い合わせの実行は、ランタイム・コンポーネント１５０が行い、これはまず、抽象的問い合わせを、ＤＢＭＳ１５４に含まれるデータの物理表現と一貫性のある形式へと変換する。

【００３１】

一実施形態では、データ・リポジトリ抽象化コンポーネント１４８は、セキュリティ情報１６２で構成される。セキュリティ情報１６２は、一般に、ユーザまたは複数のユーザの選択されたデータに対するアクセス特権を定義するセキュリティ規則を含む。ランタイム・コンポーネント１５０は、問い合わせを発行する特定のユーザ、およびその問い合わせの実行によってアクセスされる／返されることになる特定のデータに応じて、セキュリティ情報１６２をエンフォースする（enforce）ように動作する。一般に、セキュリティ規則は、特定のユーザ、ユーザのグループ、またはすべてのユーザに対して定義することができる。あるユーザが１人または複数の他のユーザと関連をもっているかどうかは、プロファイル１５８に定義する。したがって、プロファイル１５８は、データベース１５６にアクセスを試みるユーザそれぞれに対して存在する可能性がある。

【００３２】

一実施形態では、問い合わせの諸要素の指定は、ユーザがＧＵＩ（graphicaluser interface）によって行う。ＧＵＩの内容は、アプリケーション１４０によって生成される。特定の実施形態では、ＧＵＩの内容は、クライアント・コンピュータ・システム１０２上でブラウザ・プログラム１２２によってレンダリングすることのできるＨＴＭＬ（hypertextmarkup language）コンテンツである。したがって、メモリ１３２は、クライアント・コンピュータ１０２からの要求を処理するように適合されたＨＴＴＰ（HypertextTransfer Protocol、ハイパーテキスト転送プロトコル）サーバ・プロセス１５２（たとえば、Ｗｅｂサーバ）を含む。たとえば、サーバ・プロセス１５２は、例示的にサーバ１０４上に常駐している、データベース１５６にアクセスする要求に応答することができる。データベース１５６から来る、データに対するクライアント要求により、アプリケーション１４０が呼び出される。アプリケーション１４０は、プロセッサ１３０によって実行されると、サーバ・コンピュータ１０４に、データベース１５６にアクセスすることを含めて、本発明の様々な態様を実施する諸ステップまたは諸要素を実行させる。一実施形態では、アプリケーション１４０は、ブラウザ・プログラム１２２によってレンダリングされるＧＵＩ要素を作成するように構成された複数のサーブレットを含む。

【００３３】

図１は、ネットワーク接続したクライアント・コンピュータ（networkedclient computer）１０２およびサーバ・コンピュータ１０４のためのあるハードウェア／ソフトウェア構成に過ぎない。本発明の実施形態は、コンピュータ・システムが、複雑なマルチ・ユーザのコンピューティング装置、シングル・ユーザのワークステーション、またはそれ自体に不揮発性ストレージのないネットワーク・アプライアンスであるのを問わず、類似のどのようなハードウェア構成にも適用可能である。さらに、ＨＴＭＬを含めて、特定の記述言語（markuplanguage）に言及しているが、本発明は、標準であれ変形であれ、特定の言語に限定されるものではないことが理解されている。したがって、本発明が他の記述言語ならびに非記述言語に適合可能であること、および、本発明が特定の記述言語における変更ならびに現在知られていない他の言語にも適合可能であることが当業者には理解されよう。同様に、図１に示すＨＴＴＰサーバ・プロセス１５２は、例示的なものに過ぎず、知られているおよび知られていないどのようなプロトコルをサポートするように適合された他の実施形態をも企図するものである。

【００３４】

環境の論理／ランタイムビュー（logical/runtime view）

図２～３に、本発明のコンポーネントの例示的な関係図２００を示している。要求側エンティティ（たとえば、アプリケーション１４０のうちの１つ）により、その要求側エンティティのそれぞれのアプリケーション問い合わせ仕様１４２の定義する問い合わせ２０２が発行される。その結果である問い合わせ２０２を、本明細書では一般に「抽象的問い合わせ」と呼ぶが、これは、その問い合わせの作成を、ＤＢＭＳ１５４内の下位にある物理データ・エンティティを直接参照することによるのではなく、抽象的な（すなわち、論理的な）フィールドに従って行うためである。その結果、使用される特定の下位にあるデータ表現とは独立な抽象的問い合わせを定義することができる。一実施形態では、アプリ

ケーション問い合わせ仕様１４２は、データ選択に使用する基準（選択基準２０４）と、選択基準２０４に基づく、返すべきフィールドの明示的指定（返却データ指定２０６）のどちらも含むことができる。

【００３５】

図３に示す抽象的問い合わせ２０２に対応する例示的な抽象的問い合わせを、下の表Ｉに示す。例示として、抽象的問い合わせ２０２はＸＭＬを用いて定義している。しかし、有利になるように他のどのような言語を使用してもよい。

【００３６】

表Ｉ－問い合わせの例

```
001 <?xmlversion="1.0"?>
002 <!--Query string representation:(FirstName = "Mary" AND LastName =
003 "McGoon") OR State= "NC"-->
004 <QueryAbstraction>
005 <Selection>
006 <ConditioninternalID="4">
007<Condition field="FirstName" operator="EQ"value="Mary"
008 internalID="1"/>
009<Condition field="LastName" operator="EQ"value="McGoon"
010 internalID="3"relOperator="AND"></Condition>
011</Condition>
012 <Conditionfield="State" operator="EQ" value="NC"internalID="2"
013relOperator="OR"></Condition>
014 </Selection>
015 <Results>
016<Field name="FirstName"/>
017<Field name="LastName"/>
018<Field name="State"/>
019 </Results>
020 </QueryAbstraction>
```

【００３７】

例示的には、表１に示す抽象的問い合わせは、選択基準を含む選択指定（００５～０１４行）および結果仕様（０１５～０１９行）を含む。一実施形態では、１つの選択基準は、（論理フィールドに対する）フィールド名、比較演算子（＝、＞、＜など）および値の式（value expression）（そのフィールドを比較する相手）からなる。一実施形態では、結果仕様は、問い合わせ実行の結果として返されることになっている抽象的フィールドのリストである。抽象的問い合わせ内の結果仕様は、フィールド名および整列基準からなるものとすることができる。

【００３８】

アプリケーション問い合わせ仕様１４２により指定し、抽象的問い合わせ２０２を作成するのに使用する論理フィールドの定義は、データ・リポジトリ抽象化コンポーネント１４８によって行う。一般に、データ・リポジトリ抽象化コンポーネント１４８では、情報を１組の論理フィールドとして公開しており、これは（ユーザ入力問い合わせ条件に応答する状態にあるのでもよい）アプリケーション１４０の発行する問い合わせ（たとえば、抽象問い合わせ２０２）の内部でデータ選択のための基準を指定し、問い合わせ動作から返される結果のデータの形式を指定するのに使用することができる。論理フィールドは、ＤＢＭＳ１５４内で使用中の下位にあるデータ表現とは独立に定義し、それにより、下位のデータ表現と緩やかに結合する問い合わせを形成することができるようになる。

【００３９】

一般に、データ・リポジトリ抽象化コンポーネント１４８は、まとめてフィールド仕様（fieldspecification）２０８として参照する、複数のフィールド仕様２０８$_1$、２０８

$_2$、．．．（例として２つを示す）を含む。特に、フィールド仕様は、抽象的問い合わせの作成のために利用可能な論理フィールドのそれぞれに対して提供される。一実施形態では、フィールド仕様２０８は、論理フィールド名２１０$_1$、２１０$_2$（まとめて、フィールド名２１０）および関連するアクセス・メソッド（accessmethod）２１２$_1$、２１２$_2$（まとめて、アクセス・メソッド２１２）を含む。また、この例示的な実施形態では、フィールド仕様２０８は、１つまたは複数のカテゴリ名２１６$_1$、２１６$_2$（まとめて、カテゴリ名２１６）を含む。カテゴリ名により、あるグループの論理フィールド名が関連付けられる。たとえば、図３では、フィールド仕様２０８$_1$と２０８$_2$のどちらでも、人口統計（demographic）カテゴリ２１６$_1$と個人（person）カテゴリ２１６$_2$が指定されている。しかし、このカテゴリの使用は、特定の実施形態を代表させているものに過ぎず、他の実施形態ではカテゴリは利用されていない。

【００４０】

アクセス・メソッド２１２により、論理フィールド名が、データベース（たとえば、データベース１５６のうちの１つ）内の特定の物理データ表現２１４$_1$、２１４$_2$、．．．２１４$_N$と関連付けられる。例示として、図２には、２つのデータ表現、すなわちＸＭＬデータ表現２１４$_1$およびリレーショナル・データ表現２１４$_2$を示している。しかし、物理データ表現２１４$_N$では、知られているものにせよ知られていないものにせよ、他のどのようなデータ表現も企図されていることを示している。

【００４１】

一実施形態では、単一のデータ・リポジトリ抽象化コンポーネント１４８が、２つ以上の物理データ表現２１４に関する（関連するアクセス・メソッドをともなう）フィールド仕様を含む。一代替実施形態では、異なる単一のデータ・リポジトリ抽象化コンポーネント１４８が別個の物理データ表現２１４ごとに提供されている。また別の実施形態では、複数のデータ・リポジトリ抽象化コンポーネント１４８が提供され、ここでは、データ・リポジトリ抽象化コンポーネント１４８のそれぞれにより、（１つまたは複数の物理データ表現２１４を含む可能性のある）同じ下位の物理データの異なる部分を露出（expose）する。このようにして、単一のアプリケーション１４０を複数のユーザで同時に使用して、同じ下位データにアクセスを行い、ただしそのアプリケーションに露出されている下位データの特定の部分はそれぞれのデータ・リポジトリ抽象化コンポーネント１４８が決定するようにすることができる。

【００４２】

サポートすべき論理フィールドの異なるタイプの数に応じて、任意の数のアクセスが企図されている。一実施形態では、単純（simple）フィールド、フィルタされた（filtered）フィールド、および合成（composed）フィールド用のアクセス・メソッドが提供される。フィールド仕様２０８$_1$および２０８$_2$ではそれぞれ、単純フィールド・アクセス・メソッド２１２$_1$および２１２$_2$を例示している。単純フィールドは、下位の物理データ表現内の特定のエンティティに直接に対応付けられる（たとえば、所与のデータベースの表（table）および列（column）に対応付けられたフィールド）。例示として、図３に示す単純フィールド・アクセス・メソッド２１２$_1$は、論理フィールド名２１０$_1$（「FirstName」（ファースト・ネーム））を「contact」（連絡先）という名前の表の中の「f_name」という名前の列に対応付ける。フィルタされたフィールド（図２～３には例を示していない）は、関連する物理エンティティを識別し、物理データ表現内部の項目の特定のサブセットを定義するのに使用する規則を提供する。フィルタされたフィールドの例は、ニューヨーク郵便番号（ZIPcode）フィールドであり、このフィールドは郵便番号の物理表現に対応し、データをニューヨーク州に対して定義されている郵便番号だけに制限するものである。合成アクセス・メソッド（図２～３には例を示していない）では、アクセス・メソッド定義の一部として与えられる式を用いて１つまたは複数の物理フィールドから論理フィールドを計算する。このようにして、下位のデータ表現内に存在しない情報を計算することができる。１つの例は、売上税（salestax）フィールドであり、このフィールドは、販売価格フィールドに売上税率を乗じることによって合成される。

【００４３】
下位データのどのような所与のデータ型（たとえば、日付、十進数など）に対するフォーマットも、変わる可能性があることを企図している。したがって、一実施形態では、フィールド仕様２０８は、下位データのフォーマットを反映する型属性を含む。しかし、別の実施形態では、フィールド仕様２０８のデータ・フォーマットは、関連する下位の物理データとは異なり、その場合、データを要求側エンティティの仮定する正しいフォーマットで返すのはアクセス・メソッドの役目である。このため、アクセス・メソッドは、下位の物理データの実際のフォーマットだけでなく、データのどのようなフォーマットが仮定されているか（すなわち、論理フィールドに従って）について情報を持っていなければならない。そうして、アクセス・メソッドは下位の物理データを論理フィールドのフォーマットへと変換することができる。

【００４４】
例として、図２～３に示すデータ・リポジトリ抽象化コンポーネント１４８のフィールド仕様２０８により、リレーショナル・データ表現２１４$_2$で表現されたデータに対応付けられた論理フィールドを代表させている。しかし、データ・リポジトリ抽象化コンポーネント１４８の他のインスタンスでは、論理フィールドを、ＸＭＬなどの、他の物理データ表現に対応付けている。

【００４５】
一実施形態では、１つまたは複数のフィールド仕様２０８は、図１を参照して上で手短かに説明したセキュリティ情報１６２付きで構成される。例示しているこの実施形態では、カテゴリ定義とフィールド定義のどちらにも、関連するセキュリティ情報１６２がある。しかし、すべてのカテゴリ定義およびすべてのフィールド定義が、必ずしもセキュリティ情報を含む必要はないことを理解されたい。カテゴリは２つ以上の論理フィールドの間の関連を定義するものなので、同じカテゴリに属する論理フィールドは、同じカテゴリからのセキュリティ情報を継承する。たとえば、フィールド定義２０８$_1$および２０８$_2$はDemographic（人口統計）カテゴリ２１６$_1$およびPerson（個人）カテゴリ２１６$_2$にそれぞれ属しており、したがって、それぞれのセキュリティ情報１６２$_1$および１６２$_2$を継承する。

【００４６】
セキュリティ情報１６２のさらなる諸態様を表ＩＩを参照して説明することができるが、この表には、図３に示すデータ・リポジトリ抽象化コンポーネント１４８に対応する、ある例示的なデータ・リポジトリ抽象化コンポーネントを示している。例示として、データ・リポジトリ抽象化１４８はＸＭＬを用いて定義されている。しかし、有利になるように他のどのような言語を使用してもよい。

【００４７】
表ＩＩ－データ・リポジトリ抽象化の例

```
001 <?xmlversion="1.0"?>
002 <DataAbstractionversion="1.0"
003xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
004xsi:noNamespaceSchemaLocation="DataAbstraction.xsd">
005
006 <Categoryname="Person">
007
008 <Securitytype="value">
009 <SecurityRule>
010<User>All</User>
011 <Action>NoRunAndLog</Action>
012</SecurityRule>
013 </Security>
014
```

```
015<Field name="First Name">
016<AccessMethod>
017<Simple attrName="FNAME" entityName="PERSONINFO"/>
018</AccessMethod>
019<Type baseType="char"></Type>
020</Field>
021
022<Field name="Last Name">
023<AccessMethod>
024<Simple attrName="LNAME" entityName="PERSONINFO"/>
025</AccessMethod>
026<Type baseType="char"></Type>
027<Security type="value">
028 <SecurityRule>
029<User>All</User>
030<Action> RunAndLog</Action>
031</SecurityRule>
032<SecurityRule>
033<User> securityOfficers </User>
034<Action> RunAndLog </Action>
035</SecurityRule>
036<SecurityRule>
037<User> Mary_McGoon </User>
038<Action> NoAction </Action>
039</SecurityRule>
040</Security>
041 </Field>
042 </Category>
043
044 <Category name="Test">
045 <Field name="HIVTest">
046<AccessMethod>
047<Simple attrName="RESULTS" entityName="TESTS"/>
048</AccessMethod>
049<Type baseType="char"></Type>
050 <Securitytype="access">
051 <SecurityRule>
052<User>All</User>
053<Action> RunAndLog</Action>
054</SecurityRule>
055<SecurityRule>
056<User> securityOfficers </User>
057<Action> RunAndLog </Action>
058</SecurityRule>
059<SecurityRule>
060<User> Mary_McGoon </User>
061<Action> LogAndDoNotRun </Action>
062 </SecurityRule>
063</Security>
064 </Field>
```

065 </Category>
066</DataRepository>

【００４８】

表ＩＩからわかるように、それぞれのカテゴリ定義およびフィールド定義と関連付けられたセキュリティ情報は、一般に、セキュリティ型（security type）およびセキュリティ規則を含む。セキュリティ規則では、その規則が適用されるユーザまたは複数のユーザ、および規則が呼び出されたときに取るべき特定のアクションを指定している。例示的なアクションは、問い合わせの実行をさせずその試行をログに記録すること、問い合わせを実行しその実行をログに記録すること、またはアクションを取らないことを含む。もちろん、他のアクションも企図するものである。

【００４９】

一実施形態では、セキュリティ型は「値」型、「アクセス」型、またはその両方とすることができる。本例では、「Last Name」フィールド定義で値型（value type）セキュリティを例示し、「HIV Test」フィールド定義でアクセス型（accesstype）セキュリティを例示している。本明細書に定義するように、値型セキュリティ規則では、アクセスをユーザ固有の情報および特定のフィールド値に基づいて制限する（フィールド値の例としては、Mary、McGoon、1111Easy Streetがある）。対照的に、アクセス型セキュリティ規則では、フィールド値によらず、ユーザおよびフィールド型に基づいてどのようなアクセスも制限する（フィールド型の例としては、ファースト・ネーム（firstname）、ラスト・ネーム（last name）、住所がある）。例示の目的で、「Last Name」（ラスト・ネーム）フィールドを考える。ユーザが自分自身のラスト・ネームにアクセスしないようにすることが望ましいが、他のどのラスト・ネームについてもその限りではないと仮定する。この場合、値型セキュリティ規則を「LastName」フィールドに関してすべてのユーザについてインプリメントするが、これは表ＩＩの例示的なデータ・リポジトリ抽象化で行っている通りである（０２８～０３１行を参照されたい）。セキュリティ情報をこのように付加することにより、ランタイム・コンポーネント１５０は、ユーザが「LastName」フィールドにアクセスしようと試みたときに、あらかじめ定義したセキュリティ・アルゴリズム１５１（図１）を実施しなければならないことを（ランタイム時に）気付かされる。ランタイム・コンポーネントの様々な態様は、以下でより詳細に説明を行う。

【００５０】

別の例示として、「HIV Test」フィールドの場合を考える。このフィールドに含まれる情報が広範囲で慎重さを要するものであることを考えると、特定のユーザがこのフィールドにまったくアクセスしないようにするのが望ましいこともあろう。たとえば、本人がＨＩＶポジティブであるユーザが「HIVTest」フィールドにアクセスしないようにするのが望ましいとしよう。この場合、セキュリティ情報は、アクセス型セキュリティ制限を含む（０４５～０６４行を参照されたい）。ランタイム中、ランタイム・コンポーネント１５０のセキュリティ・アルゴリズム１５１により、「HIVTest」フィールドにアクセスしようと試みるあるユーザをそうすることから制限するかどうか、または、より一般に、（そのユーザがＨＩＶポジティブであるかどうかに従って）なにかセキュリティ・アクションを取らなければならないかどうかを判断する。したがって、アクセス型制限の場合、要求されている「HIVTest」フィールドにある特定の値は、関係がない。

【００５１】

表ＩＩ（０２２～０４１行）の論理の「Last Name」フィールドで示しているように、所与のフィールド／カテゴリ定義のためのセキュリティ情報は、セキュリティ規則を２つ以上含んでもよい。特に、異なるセキュリティ規則を、ユーザの異なるクラスに適用することができる。論理「LastName」フィールドのための定義では、ユーザすべてに対する第１のセキュリティ規則、あるグループのユーザに対する第２のセキュリティ規則、ある１人のユーザに対する第３のセキュリティ規則を例示している。一実施形態では、（アプリケーション１４０（図１）の一部であってもよい）最適適合（best-fit）アルゴリズムを使用して、特定のユーザ・クラスにどの制限をかけるかを判断する。たとえば、ユーザMa

ry_McGoonが「LastName」フィールドにアクセスしようと試みた場合、ユーザ・プロファイルMary_McGoonに関する規則を使用する。より的確な（exact）規則指定が見つかったため（すなわち、ユーザMary_McGoonに固有のセキュリティ規則）、「all」（すべて）ユーザおよび「securityofficer」（セキュリティ責任者）に対する規則は無視される。

【００５２】

しかし、一実施形態では、最適適合アルゴリズムによって判断した規則をオーバーライドすることがある。表ＩＩが実例となっている通り、データ・リポジトリ抽象化コンポーネント１４８は、階層的ツリー構造をもつことができる。表ＩＩの例示的なデータ・リポジトリ抽象化では、提示してある最も上位のノードは人口統計カテゴリである。次に上位のノードは個人カテゴリであり、これには子としてファースト・ネーム・フィールドおよびラスト・ネーム・フィールドがある。したがって、一実施形態では、最厳格規則（the strictest rule）は、ツリーの中の最上位ノードに関連付けられている規則である。この例では、最厳格規則は、人口統計カテゴリのセキュリティ情報に関連付けられている規則である。したがって、ユーザMary_McGoonに固有のセキュリティ規則は、人口統計カテゴリの規則によってオーバーライドされるのである。

【００５３】

ここで図４を参照すると、アプリケーション１４０の初期プログラム・ロードおよび動作を例示する方法３００が示してある。（たとえば、管理者が）アプリケーション１４０を起動すると、この方法に入る。ステップ３０２で、データ・リポジトリ抽象化コンポーネントのロード／取り出しを行う。複数のデータ・リポジトリ抽象化コンポーネントが利用可能な一実施形態では、取り出すべきデータ・リポジトリ抽象化コンポーネントは、アプリケーション１４０によって指定する。ステップ３０４で、データ・リポジトリ抽象化コンポーネントのツリー構造ごとにループに入る。ステップ３０６で、ツリー構造にある最厳格規則の識別を行う。これを行うには、そのツリーを最下位ノードから最上位ノードへと「たどって」上に行き（"walking" up）、関連付けられているセキュリティ規則のある最も上位のノードを識別する。最厳格規則の識別が済むと、アプリケーション１４０により、その規則がアクセス型であるかどうかを（ステップ３０８で）判断する。そうである場合、ステップ３１０で、その規則に関連付けられたフィールドを、ユーザ固有アクセス型セキュリティ・リスト３１２に入れる。次いで、方法３００はステップ３０４へと戻って、そのデータ・リポジトリ抽象化コンポーネントにある次のツリー構造のための処理を始める。

【００５４】

ステップ３０８の答えが「いいえ」である場合、処理はステップ３１４へと進み、ここで方法３００は、その規則が値型の規則であるかどうかを判断する。そうである場合、この規則に関連付けられているフィールドを、ステップ３１６でユーザ固有値型セキュリティ・リスト３１８に入れる。次いで、方法３００はステップ３０４へと戻って、データ・リポジトリ抽象化コンポーネントにある次のツリー構造のための処理を始める。

【００５５】

ステップ３１４の答えが「いいえ」である場合、処理はステップ３２０へと進み、ここでその規則に関連付けられたフィールドを適切なセキュリティ・リストに入れる。ステップ３２０は、方法３００を他の型のセキュリティ規則に拡張することに相当する。次いで、方法３００はステップ３０４へと戻って、データ・リポジトリ抽象化コンポーネントにある次のツリー構造のための処理を始める。

【００５６】

方法３００は、データ・リポジトリ抽象化コンポーネントにあるツリー構造のそれぞれを処理してしまうまで続く。すると、アプリケーション１４０が要求の受理を始める準備ができる。

【００５７】

図５に、アプリケーション１４０のログオン検証アルゴリズム１６０の実行するログオン検証方法４００の一実施形態を示している。ログオン検証方法４００は、アプリケーシ

ョン１４０の使用のためにユーザがサイン・オンするたびに実行される。ステップ４０２で、ログオン検証アルゴリズム１６０は、サイン・オンを行っているユーザに関するアクセス型セキュリティ・リスト３１２を取り出す。ステップ４０４で、検証方法４００は、セキュリティ・リスト３１２のフィールドごとにループを開始する。ステップ４０６で、検証アルゴリズム１６０は、ユーザ・データ（すなわち、データベース１５６の物理データにあるもの）の対応するフィールド（すなわち、処理中のセキュリティ・リスト・フィールドに対応するフィールド）内のエントリについて検査する。対応するエントリが見当たらない場合（ステップ４０８）、処理中のフィールドをセキュリティ・リスト３１２から取り除く。こうして、取り除いたフィールドは、ランタイム中、セキュリティ目的では考慮しないことにするのである。対応するフィールドが見つかった場合（ステップ４０８）、特にアクションを取る必要はなく処理はステップ４０４へと戻って、セキュリティ・リスト３１２の次のフィールドの処理を始める。セキュリティ・リスト３１２の各フィールドの処理が済むと、ログオン検証メソッド４００はステップ４１２へと進む。

【００５８】

ステップ４１２で、サイン・オンを行っているユーザについて値型セキュリティ・リスト３１８を検索する。セキュリティ・リストのフィールドごとに実行するループに、ステップ４１４で入る。ステップ４１６で、検証アルゴリズム１６０がユーザ・データ（すなわち、データベース１５６の物理データ内のもの）の対応するフィールド（すなわち、処理中のセキュリティ・リスト・フィールドに対応するフィールド）内のエントリについて検査を行う。対応するエントリが見当たらない場合（ステップ４１８）、処理中のフィールドをセキュリティ・リスト３１８から取り除く。そうでない場合（すなわち、エントリがステップ４１８で見つかった場合）、対応するユーザ固有エントリの値を値型セキュリティ・リスト３１８に（または何か他のセキュリティ・データ構造に）格納する。どちらの場合も、処理はステップ４１４へと戻る。セキュリティ・リスト３１８の各フィールドを処理してしまうと、ログオン検証アルゴリズム１６０を抜け、ここでユーザは問い合わせの発行を開始してよいことになる。

【００５９】

図６～７には、ランタイム・コンポーネント１５０の動作の一実施形態を例示する、例示的なランタイム方法５００を示している。特に、ランタイム方法５００の諸態様は、セキュリティ・アルゴリズム１５１（図１）によってインプリメントすることができる。方法５００にステップ５０２で入るのは、ランタイム・コンポーネント１５０が入力として（図２～３に示す抽象的問い合わせ２０２などの）抽象的問い合わせのインスタンスを受け取るときである。ステップ５０４で、ランタイム・コンポーネント１５０は、抽象的問い合わせのインスタンスを読み込み、構文解析（parse）し、個々の選択基準および所望の結果フィールド（result fields）を見出す。ステップ５０６で、ランタイム・コンポーネント１５０、その抽象的問い合わせに存在する各問い合わせ選択基準ステートメントの処理のためにループに入り、それにより、具体的問い合わせ（ConcreteQuery）のデータ選択部分を作成する。一実施形態では、選択基準は（論理フィールドについて）フィールド名、比較演算子（＝、＞、＜など）、および値の式（そのフィールドを比較する相手）からなる。ステップ５０８で、ランタイム・コンポーネント１５０は、その抽象的問い合わせの選択基準からのフィールド名を使用して、データ・リポジトリ抽象化１４８内のフィールドの指定のルック・アップを行う。上で触れたように、フィールド仕様は、そのフィールドに関連付けられている物理データをアクセスするために使用するアクセス・メソッドの定義を含む。さらに、フィールド仕様は、セキュリティ情報１６２を含むことができる。したがって、ランタイム・コンポーネント１５０は、その取り出した問い合わせフィールド仕様に対して、その特定のユーザ（すなわち、その問い合わせを発行しているユーザ）についてのセキュリティ規則があるかどうかを判断する（ステップ５１０）。すなわち、ランタイム・コンポーネント１５０は、アクセス型セキュリティ・リスト３１２および値型セキュリティ・リスト３１８を参照して、そのフィールド仕様に対応するフィールド名の存在を判断する。どちらのリストにもそのフィールド仕様に関するセキュリテ

ィ規則が見当たらない場合、処理はステップ５１６へと進む。セキュリティ規則があった場合には、ランタイム・コンポーネント１５０は、その規則の指定するセキュリティ・アクションを取らなければならないかどうかを判断する（ステップ５１２）。アクセス型制限の場合、規則の指定するセキュリティ・アクションを取らなければならないのは、そのフィールド仕様の論理フィールドに対応するフィールド・エントリが、ステップ５１０でアクセス型セキュリティ・リスト３１２内に見つかった場合であり、これはそのフィールド・エントリの値によらない。値型制限の場合は、規則の指定するセキュリティ・アクションを取らなければならないのは、分析中の論理問い合わせコンポーネント／基準内の論理フィールド値に対応するフィールド値が、ステップ５１０で値型セキュリティ・リスト３１８内に見つかった場合である。ステップ５１２の答えが「はい」である場合は、ランタイム・コンポーネント１５０は、そのアクションを取らなければならないことを（ステップ５１４で）記録する。次いで、処理はステップ５１６へと進む。

【００６０】

次いで、ランタイム・コンポーネント１５０は、処理中の論理フィールドに関する具体的問い合わせ寄与（Concrete Query Contribution）を作成する。本明細書で定義するように、具体的問い合わせ寄与とは、データ選択の実行に使用する具体的問い合わせの、現在の（current）論理フィールドに基づく部分である。具体的問い合わせは、ＳＱＬやＸＱｕｅｒｙなどで表現した問い合わせであり、所与の物理データ・リポジトリ（たとえば、リレーショナル・データベースまたはＸＭＬリポジトリ）のデータと一貫性がある。したがって、具体的問い合わせは、図１に示すＤＢＭＳ１５４の表す、物理データ・リポジトリからデータを見出し、取り出すのに使用される。次いで、現在のフィールド用に生成した具体的問い合わせ寄与を、具体的問い合わせステートメント（ConcreteQuery Statement）に追加する（ステップ５１８）。次いで、方法５００はステップ５０６へと戻って、抽象的問い合わせの次のフィールドのための処理を始める。したがって、ステップ５０６で入ったプロセスを、その抽象的問い合わせないのデータ選択フィールドごとに繰り返して、それにより、実行すべき最終的な問い合わせに追加の内容の寄与を行う。

【００６１】

具体的問い合わせのデータ選択部分を作成した後、ランタイム・コンポーネント１５０は、問い合わせ実行の結果として返すべき情報を識別する。上で説明したように、一実施形態では、抽象的問い合わせは、本明細書で結果仕様（result specification）と呼ぶ、問い合わせ実行の結果として返すべき抽象的フィールドのリストを定義する。抽象的問い合わせ内の結果仕様は、フィールド名および整列基準からなる。一実施形態では、結果仕様はセキュリティ情報１６２をも含み、これは表ＩＩを参照して上で説明したセキュリティ情報と同一にまたは類似に定義することができる。結果仕様を扱うために、方法５００はステップ５２０でループに入って、結果フィールド定義を生成中の具体的問い合わせに追加する。ステップ５２２で、ランタイム・コンポーネント１５０は、データ・リポジトリ抽象化コンポーネント１４８内の（その抽象的問い合わせの結果仕様からの）結果フィールド名のルック・アップを行い、次いでデータ・リポジトリ抽象化コンポーネント１４８から結果フィールド定義を取り出して、現在の論理結果フィールドに関して返すべきデータの物理ロケーションを識別する。ステップ５２４で、ランタイム・コンポーネント１５０は、処理中の特定の結果フィールド定義に関してこの特定のユーザについてセキュリティ規則が存在するかどうかについて判断する。すなわち、ランタイム・コンポーネント１５０がアクセス型セキュリティ・リスト３１２を参照して結果仕様に対応するフィールド名の存在を判断する（値型セキュリティ・リスト３１８は、結果の場合、値が利用可能でないため、検査を行わない）。結果仕様に関するセキュリティ規則が、リスト３１２内に見当たらない場合は、処理はステップ５３０へと進む。そうでない場合、ランタイム・コンポーネント１５０は、その結果フィールドのために定義した規則で取らなければならないアクションが指定されているかどうかを判断する（ステップ５２６）。これは、ステップ５１２に関して説明したのと同様のしかたで実行することができる。ステップ５２６の答えが「はい」である場合、ランタイム・コンポーネント１５０は、そのアクションを

取らなければならないことを記録する（ステップ５２８）。どちらの場合にも、処理はステップ５３０へと進む。

【００６２】

ステップ５３０で、ランタイム・コンポーネント１５０は、論理結果フィールドに対する（返すべきデータの物理ロケーションを識別する具体的問い合わせの）具体的問い合わせ寄与を作成する。次いで、ステップ５３２で、具体的問い合わせ寄与を具体的問い合わせステートメントに追加する。

【００６３】

抽象的問い合わせ内の結果仕様のそれぞれの処理が済むと、ランタイム・コンポーネント１５０は、（上で説明したやり方で何かアクションが記録されているかどうかに従って）何かセキュリティ・アクションを取らなければならないかどうかを判断する（ステップ５３４）。そうしなくてよい場合、問い合わせをステップ５４２で実行する。しかし、セキュリティ・アクションを取らなければならない場合は、記録されているアクションを検討して（ステップ５３６）、最厳格のアクションを判断する。一実施形態では、各アクションには、厳しいほど大きくなる数値を割り当ててある。ステップ５３８で、ランタイム・コンポーネント１５０は、そのアクションが致命的（すなわち、問い合わせを実行させない）かどうかを判断する。そうでない場合、問い合わせをステップ５４２で実行する。アクションが致命的である場合は、問い合わせを実行せず、ユーザにセキュリティ違反（security violation）を通知する（ステップ５４０）。そうして、方法５００は終わる。

【００６４】

セキュリティ・アルゴリズム１５１によって実行する上述の問い合わせ検証は、一般に、２つの時点のどちらかで実行できることに留意されたい。一実施形態では、検証は、問い合わせ条件のそれぞれの作成次第起きることにしてよい。そのようなアプローチでは、ユーザへのセキュリティ違反の通知が迅速化されるはずである。しかし、このアプローチの不利な点は、実施中のセキュリティ制限の破壊を試みるユーザの実験（たとえば、問い合わせを修正しながら徐々に増やす）を容易にしてしまうことである。このため、代替アプローチでは、問い合わせ検証の実行を、ユーザが問い合わせ全体を入力し発行してしまってからに限っている。後者のアプローチでは、ランタイム・コンポーネント１５０は、違反がないかを検査する前に問い合わせの全体図が得られることにもなり、それにより、ユーザの意図は単一の条件の場合より、より明確になる。

【００６５】

値型制限についてのランタイム方法５００は、表ＩＩの「Last Name」フィールドおよびデータ・リポジトリ抽象化に関する上の例を参照して例示することができる。ユーザがメアリー・マグーン（MaryMcGoon）であると仮定すると、ラスト・ネームの列（column）にある「McGoon」という値に対するエントリが、ログオン（ログオン検証方法３００）中に値型セキュリティ・リスト３１８に入れられているはずである。ここで、メアリー・マグーンが、McGoonという値を「LastName」フィールドに含む論理問い合わせを発行し、ランタイム・コンポーネント１５０が、アクセス型セキュリティ・リスト３１２と値型セキュリティ・リスト３１８（メアリー・マグーンに対して、この人のログオン時に生成済み）のどちらかが「LastName」フィールドに関するセキュリティ規則を含むかどうかを検査する（ステップ５１０および５２４）。値型セキュリティ・リスト３１８が「Last Name」フィールドに関するセキュリティ規則を含むと判断した後、セキュリティ・アルゴリズム１５１により、その規則では、試みられたアクセスに応答してアクションが要求されているかどうかを判断する（ステップ５１２および５２６）。本例では、分析中の論理的問い合わせ基準が「McGoon」という値を含み、その規則がメアリー・マグーンというユーザに関するアクションを指定する値型規則であるので、アクションが要求されていることになる。表ＩＩのデータ・リポジトリ抽象化コンポーネントによって指定された（また値型セキュリティ・リスト３１８に記録された）アクションは、「NoAction」である。試みられたアクセスがアクションを要求する場合には、セキュリティ・アルゴリズム１５１で、セキュリティ情報１６２で定義するアクションを取らなければならないことが記録される

（ステップ５１４および５２８）。

【００６６】

本例では、指定したセキュリティ・アクション（これは、メアリー・マグーンの場合では、NoActionである）が、指定したユーザが問い合わせを特定の値で発行したときに実行される。もちろん、セキュリティ規則の論理をＮＯＴ演算子を用いて構成することもできることが当業者には理解されよう。たとえば、セキュリティ規則を、ユーザ（たとえば、メアリー・マグーン）が問い合わせ内に他の誰かの氏名（name）を入力した場合には実行してログを記録する（RunAndLog）ように構成してよいが、同じユーザが自分の氏名を入力した場合には、何らアクションを取らない（NoAction）ように構成することができ、この逆も可能である。

【００６７】

メアリー・マグーンの例を続けて、メアリーの記録が、この人がＨＩＶ検査を実施してもらったことを示していると仮定する。したがって、アクセス型セキュリティ・リストは、（ログオン検証方法３００の結果として）ＨＩＶ検査フィールドのための列を含むはずである。さらに、メアリーが、ＨＩＶ検査フィールドを参照する問い合わせを発行すると仮定する。

【００６８】

ランタイム・コンポーネント１５０は、アクセス型セキュリティ・リスト３１２と値型セキュリティ・リスト３１８（メアリー・マグーンに対して、この人のログオン時に生成済み）が「HIV Test」フィールドに関するセキュリティ規則を含むかどうかを検査する（ステップ５１０および５２４）。アクセス型セキュリティ・リスト３１２が「HIVTest」フィールドに関するセキュリティ規則を含むかどうかを判断した後で、セキュリティ・アルゴリズム１５１は、その規則が試みられたアクセスに応答してアクションを要求するかどうかを判断している（ステップ５１２および５２６）。これは、アクセス型制限の場合には、問い合わせ内に指定されているフィールドの値は関係がなく、したがって、ＨＩＶ検査フィールドがアクセス型セキュリティ・リスト３１２にあるだけでセキュリティ規則が存在しておりそれを実施しなければならないことが示されるためである。本例のそれぞれにおいて、規則が「アクション」を要求することの判断は、単にある規則が指定されていること、およびその規則を実施するための条件が満たされていることの判断に過ぎないことに留意されたい。もちろん、これまでの実施形態で触れたように、その規則を、「No Action」規則（たとえば、表ＩＩ、０３８行を参照されたい）として、その「何もしないアクション」（NoAction）を取るべきことを指定することもできる。本例では、取るべきアクションは「LogAndDoNotRun」（ログに記録し実行しない）である。

【００６９】

一実施形態では、アクセス型セキュリティ・リスト３１２および値型セキュリティ・リスト３１８は単にパフォーマンスを高めるために提供しているのに過ぎないことを明確にしておきたい。別の実施形態では、リスト３１２、３１８は使用せず、ランタイム・コンポーネント１５０が、問い合わせを検証するために、データ・リポジトリ抽象化コンポーネント１４８に直接にアクセスする。

【００７０】

ステップ５１６および５３０に従って、論理フィールドのための具体的問い合わせ寄与（ConcreteQuery Contribution）を作成するための方法６００の一実施形態を、図８を参照して説明する。ステップ６０２で、方法６００は、現在の論理フィールドに関連付けられているアクセス・メソッドが単純アクセス・メソッドであるかどうかを問い合わせる。そうである場合、具体的問い合わせ寄与を、物理データ・ロケーション情報に基づいて作成し（ステップ６０４）、次いで処理は上に説明した方法５００に従って続く。そうでない場合、処理はステップ６０６へと進んで、現在の論理フィールドに関連付けられているアクセス・メソッドがフィルタされた（filtered）アクセス・メソッドであるかどうかを問い合わせる。そうである場合、具体的問い合わせ寄与を、一部の物理データ・エンティティに関する物理データ・ロケーション情報に基づいて作成する（ステップ６０８）。ス

テップ６１０で、具体的問い合わせ寄与の拡張を、物理データ・エンティティに関連付けられているデータからサブセットを作る（subset）のに使用する追加の論理（フィルタ選択）によって行う。次いで、処理は上で説明した方法５００に従って続く。

【００７１】

アクセス・メソッドがフィルタされたアクセス・メソッドではなかった場合、処理はステップ６０６からステップ６１２へと進んで、ここで方法６００は、そのアクセス・メソッドが合成アクセス・メソッドであるかどうかを問い合わせる。そのアクセス・メソッドが合成アクセス・メソッドである場合、合成フィールド式にあるサブ・フィールド参照のそれぞれに対する物理データ・ロケーションを、ステップ６１４で見出し、取り出す。ステップ６１６で、合成フィールド式の物理フィールド・ロケーション情報を、合成フィールド式の論理フィールド参照に代入し、これにより、具体的問い合わせ寄与を生成する。次いで、処理は上で説明した方法５００に従って続く。

【００７２】

アクセス・メソッドが合成アクセス・メソッドでなかった場合、処理はステップ６１２からステップ６１８へと進む。ステップ６１８は、本発明の諸実施形態として企図されている他のどのようなアクセス・メソッドのタイプをも表わす。しかし、全部に満たない利用可能なアクセス・メソッドが実施される実施形態が企図されていることを理解されたい。たとえば、ある特定の実施形態では、使用するのは単純アクセス・メソッドだけである。別の実施形態では、使用するのは単純アクセス・メソッドおよびフィルタされたアクセス・メソッドだけである。

【００７３】

上で説明したように、論理フィールドで、下位の物理データとは異なるデータ・フォーマットが指定されている場合には、データ変換の実行が必要となることがある。一実施形態では、初期変換の実行を、それぞれのアクセス・メソッドごとに、方法６００に従って論理フィールドに対する具体的問い合わせ寄与を作成するときに行う。たとえば、この変換は、ステップ６０４、６０８、および６１６の一部として、またはその直後に実行することができる。それに続く物理データのフォーマットから論理フィールドのフォーマットへの変換の実行は、ステップ５４２で問い合わせを実行した後になる。もちろん、論理フィールド定義のフォーマットが下位の物理データと同じである場合には、変換は必要ではない。

【００７４】

以上の例示では、問い合わせを発行するユーザに対する表の中の列と、別の人物に対する同じ列の間の関連付けを確立することを、セキュリティ条件は前提としている。すなわち、たとえば、値型制限では、ユーザ固有の値（たとえば、ラスト・ネーム）と別の人物に関する（そのユーザがアクセスしようと試みている）同じ値との比較が要求される。もちろん、他の実施形態では、セキュリティ制限を、（その問い合わせを発行している特定のユーザの視点から）他のどのような「慎重さを要する」（sensitive）情報も含むように拡張することができる。たとえば、ユーザが前の配偶者のラスト・ネームにアクセスしないようにすることが望ましいこともあろう。正確にどの値／フィールドを制限するかは、アプリケーション１４０のセキュリティ・アルゴリズムの中で指定することができる。

【００７５】

一実施形態では、セキュリティ制限を他のどのような「慎重さを要する」情報も含むように拡張するという以上の態様を、フィールド相関（field correlation）によって達成することができる。フィールド相関により、特定のフィールドに関するセキュリティを別のフィールド内の値と関係させることができるようになる。たとえば、「LastName」フィールドを「address」フィールドと関連付けて、その同じ住所に住んでいる、またはその問い合わせを発行中のユーザがその住所に住んでいたのと同じ時に、その以前の住所にいた、他の人物の氏名を取り出すことができる。すると、これらの氏名があると、そのユーザの氏名の履歴（namehistory）を検査して、氏名の広範囲のリストを生成し、それにそのユーザがアクセスしないようにすることができる。別の例示としては、データベースに

家系情報（genealogicalinformation）があると、規則を書いて家計図をたどり、過去Ｎ世代にわたって複雑な氏名のリストを作成することができる（ここで、たとえば、Ｎは整数とする）。また別の例示としては、ある医者の今の（active）患者のリストがあると（データ・ウェアハウスから容易に利用可能である）、他のすべての氏名に関して、その医者にある程度のセキュリティ制限を課すことができる。

【００７６】

上で説明したように、セキュリティ・アクションには試みられたアクセスのログの記録を含むものがある。より一般には、セキュリティ規則が呼び出されたときには、どのような形（variety）の応答も取ることができることが当業者には理解されよう。たとえば、システム管理者への（たとえば、電子メールによる）通知を発行するのでもよい。

【００７７】

上で触れたように、データ・リポジトリ抽象化コンポーネント１４８は、単に様々な利点を提供する一実施形態を例示するものに過ぎない。一態様では、諸利点の達成を、アプリケーション問い合わせ仕様と下位のデータ表現との緩やかな結合を定義することによって行う。ＳＱＬを使用する際にそうであるように、アプリケーションを特定の表、列およびリレーションシップ情報でエンコードするのではなく、アプリケーションでは、データ問い合わせ要件の定義をより抽象的に行い、次いで、このデータ問い合わせ要件をランタイム時に特定の物理データ表現にバインドする。本発明の緩やかな問い合わせ／データ結合により、要求側エンティティ（たとえば、アプリケーション）を機能させることが、下位のデータ表現が変更された場合、または要求側エンティティを、開発したときに使用したものに比べまったく新しい物理データ表現とともに使用することになる場合でも、可能になる。所与の物理データ表現を変更または再構成する場合には、対応するデータ・リポジトリ抽象化を更新して下位の物理データ・モデルに行われた変更を反映させる。同じ組の論理フィールドが問い合わせによる使用のために利用可能であり、単に物理データ・モデル内の異なるエンティティまたはロケーションにバインドされていたのに過ぎない。その結果、抽象的問い合わせインターフェースに書かれている要求側エンティティは、対応する物理データ・モデルがかなりの変更を受けた場合でも、変更なしで機能し続ける。要求側エンティティが開発したときに使用したものに比べまったく新しい物理データ表現とともに使用することになる場合には、新しい物理データ・モデルの実装を、同じ技術（たとえば、リレーショナル・データベース）を用いながら、異なる情報を命名し編成する方策（たとえば、異なるスキーマ）に従って行うことができる。この新しいスキーマは、アプリケーションの要求する１組の論理フィールドへの対応付けを単純、フィルタ済み、および合成フィールド・アクセス・メソッド技法を用いて行うことのできる情報を含むことになる。あるいは、新しい物理表現では、類似の情報を表現するのに代替的な技術を使用することができる（たとえば、リレーショナル・データベース・システムの使用に対するＸＭＬベースのデータ・リポジトリの使用）。どちらの場合でも、抽象的問い合わせインターフェースを使用するように書かれている既存の要求側エンティティを、容易に、新しい物理データ表現を使用するように移行させる（migrate）ことは、問い合わせ内で参照されるフィールドを、新しい物理データ・モデル内のロケーションおよび物理表現と対応付ける代替的なデータ・リポジトリ抽象化の提供のもとで可能となる。

【００７８】

エンド・ユーザに関して述べると、データ・リポジトリ抽象化により、関係するデータを公開し、選択された内容へのアクセスをさせない、データ・フィルタリング機構が提供される。しかし、データ・リポジトリ抽象化は、単に本発明の一実施形態に過ぎないことを理解されたい。より一般には、本発明はユーザ／データ依存性（user-data dependency）に応じて問い合わせの実行（または非実行）に対する必要を提供するどのようなやり方でも実装される。すなわち、問い合わせ実行は、エンド・ユーザおよびその問い合わせの実行後にアクセスされる／返されるはずの特定のデータに依存させてある。

【００７９】

しかし、本発明のセキュリティ機能および機構を、データ・リポジトリ阻害コンポーネ

ントとは別個に実装できることは、当業者には容易に理解されることを強調しておきたい。たとえば、従来のリレーショナル・データベースの状況で、ある実施形態では、表および列をセキュリティ規則に関連付ける情報を格納するための追加のデータベース表が数多く提供される。セキュリティ規則の適用は、表／列内の情報にアクセスしようとする試みに対して行われる。この場合、セキュリティ・コンポーネントの役目は、問い合わせ言語（たとえば、ＳＱＬ）で書かれた問い合わせを分析すること、および適用される１組のセキュリティ規則の決定を、問い合わせ術語内部で参照される、または特定の問い合わせに対する結果の組のメンバとして返される１組の表／列に基づいて行うことである。

【００８０】

本明細書における特定の値、定義、プログラミング言語および例への参照は、例示の目的に過ぎないことに留意されたい。したがって、本発明は、どのような特定の例示および例によっても限定されるものではない。さらに、本発明の諸態様は選択（SELECTION）操作に関して説明しているが、追加（ADD）、変更（MODIFY）、挿入（INSERT）、削除（DELETE）など、よく知られている操作を含めて、他の入出力操作も企図するものである。もちろん、一部のアクセス・メソッドでは、その特定のアクセス・メソッドを利用するフィールドを用いて定義できる抽象的問い合わせ関数のタイプに制限が生じることもある。たとえば、合成アクセス・メソッドを含むフィールドは、変更、挿入、および削除の実行可能なターゲットではない。

【００８１】

以上は本発明の諸実施形態を対象としているが、本発明の他のおよびさらなる実施形態を、本発明の基本的な範囲から逸脱することなく考案することができ、本発明の範囲は以下の特許請求の範囲によって定めるものとする。

【図面の簡単な説明】

【００８２】

【図１】コンピュータ・システムの一実施形態の図である。

【図２】本発明の一実施形態のソフトウェア・コンポーネントの論理／物理ビュー（logical/physicalview）である。

【図３】抽象的問い合わせおよび抽象化のデータ・リポジトリの論理ビューである。

【図４】アプリケーションの開始および動作を示す流れ図である。

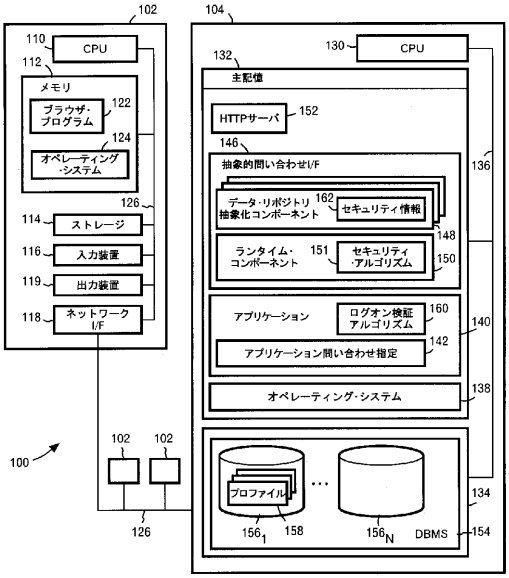
【図５】ユーザ・ログオン検証シーケンスを示す流れ図である。

【図６】ランタイム・コンポーネントの動作を示す流れ図である。

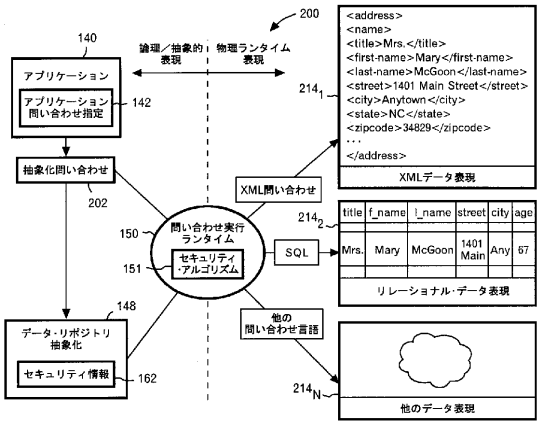
【図７】ランタイム・コンポーネントの動作を示す流れ図である。

【図８】ランタイム・コンポーネントの動作を示す流れ図である。

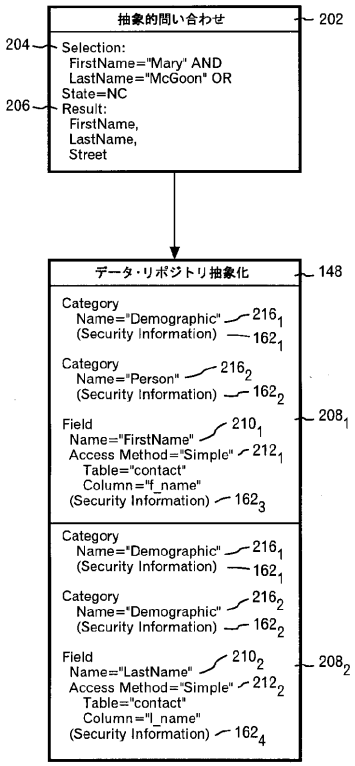
【図１】


102
104
110
CPU
112
メモリ
ブラウザ・プログラム
122
124
オペレーティング・システム
126
114
ストレージ
116
入力装置
119
出力装置
118
ネットワークI/F
130
CPU
132
主記憶
HTTPサーバ
152
146
抽象的問い合わせI/F
データ・リポジトリ抽象化コンポーネント
162
セキュリティ情報
148
ランタイム・コンポーネント
151
セキュリティ・アルゴリズム
150
136
アプリケーション
ログオン検証アルゴリズム
160
アプリケーション問い合わせ指定
142
140
オペレーティング・システム
138
100
102
102
126
プロファイル
156₁
158
156N
DBMS
134
154

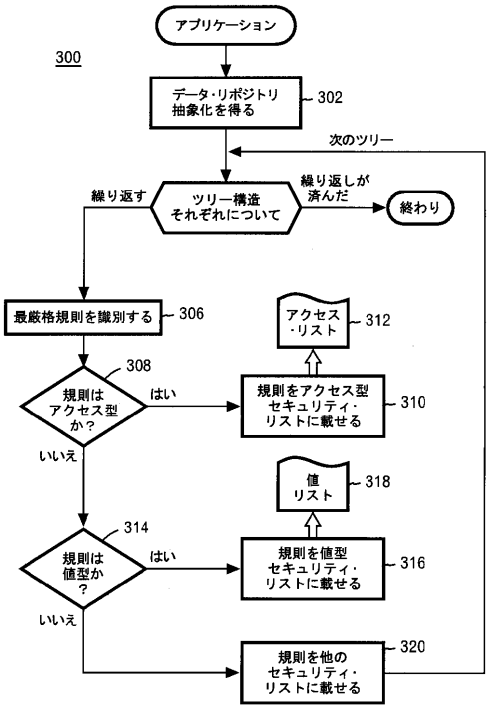
【図２】


200
140
論理／抽象的表現
物理ランタイム表現
アプリケーション
アプリケーション問い合わせ指定
142
抽象化問い合わせ
202
XML問い合わせ
SQL
他の問い合わせ言語
問い合わせ実行ランタイム
150
151
セキュリティ・アルゴリズム
148
データ・リポジトリ抽象化
セキュリティ情報
162
214₁
<address>
<name>
<title>Mrs.</title>
<first-name>Mary</first-name>
<last-name>McGoon</last-name>
<street>1401 Main Street</street>
<city>Anytown</city>
<state>NC</state>
<zipcode>34829</zipcode>
</address>
XMLデータ表現
214₂
title
f_name
l_name
street
city
age
Mrs.
Mary
McGoon
1401 Main
Any
67
リレーショナル・データ表現
214N
他のデータ表現

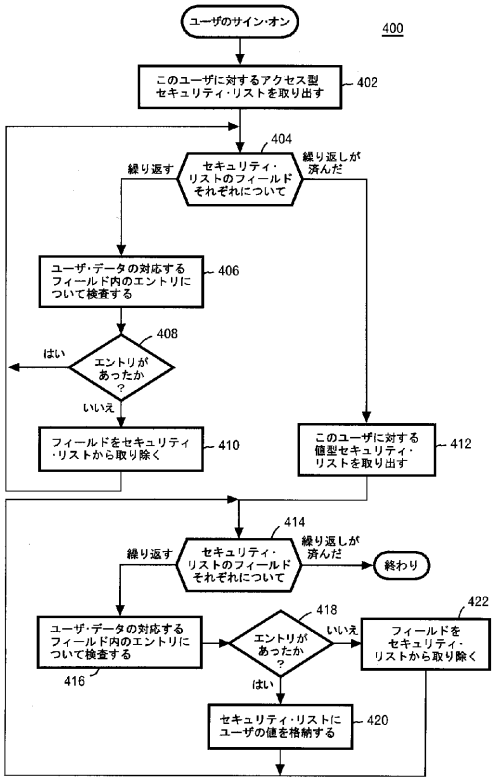
【図３】


抽象的問い合わせ
202
204
Selection:
FirstName="Mary" AND
LastName="McGoon" OR
State=NC
206
Result:
FirstName,
LastName,
Street
データ・リポジトリ抽象化
148
Category
Name="Demographic"
(Security Information)
216₁
162₁
Category
Name="Person"
(Security Information)
216₂
162₂
Field
Name="FirstName"
Access Method="Simple"
Table="contact"
Column="f_name"
(Security Information)
210₁
212₁
162₃
208₁
Category
Name="Demographic"
(Security Information)
216₁
162₁
Category
Name="Demographic"
(Security Information)
216₂
162₂
Field
Name="LastName"
Access Method="Simple"
Table="contact"
Column="l_name"
(Security Information)
210₂
212₂
162₄
208₂

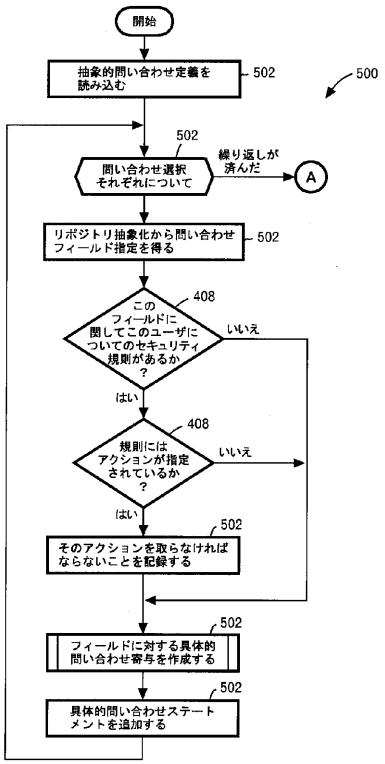
【図４】


アプリケーション
300
データ・リポジトリ抽象化を得る
302
次のツリー
繰り返す
ツリー構造それぞれについて
繰り返しが済んだ
終わり
最厳格規則を識別する
306
アクセス・リスト
312
308
規則はアクセス型か？
はい
規則をアクセス型セキュリティ・リストに載せる
310
いいえ
値リスト
318
314
規則は値型か？
はい
規則を値型セキュリティ・リストに載せる
316
いいえ
規則を他のセキュリティ・リストに載せる
320

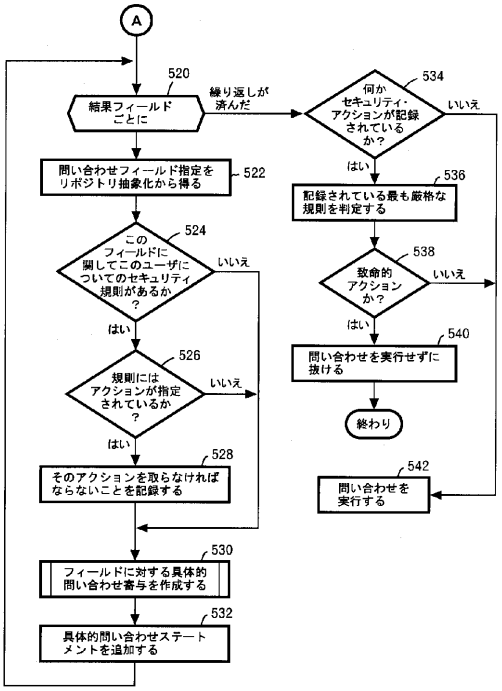
【図５】


ユーザのサイン・オン
400
このユーザに対するアクセス型セキュリティ・リストを取り出す
402
404
セキュリティ・リストのフィールドそれぞれについて
繰り返す
繰り返しが済んだ
ユーザ・データの対応するフィールド内のエントリについて検査する
406
408
エントリがあったか？
はい
いいえ
フィールドをセキュリティ・リストから取り除く
410
このユーザに対する値型セキュリティ・リストを取り出す
412
414
セキュリティ・リストのフィールドそれぞれについて
繰り返す
繰り返しが済んだ
終わり
ユーザ・データの対応するフィールド内のエントリについて検査する
416
418
エントリがあったか？
いいえ
はい
フィールドをセキュリティ・リストから取り除く
422
セキュリティ・リストにユーザの値を格納する
420

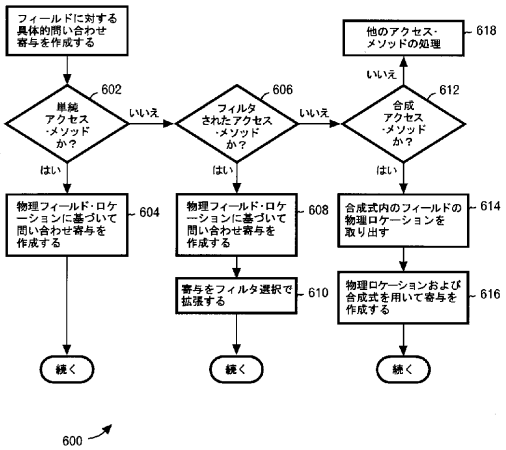
【図６】


開始
抽象的問い合わせ定義を読み込む
502
500
502
問い合わせ選択それぞれについて
繰り返しが済んだ
A
リポジトリ抽象化から問い合わせフィールド指定を得る
502
408
このフィールドに関してこのユーザについてのセキュリティ規則があるか？
いいえ
はい
408
規則にはアクションが指定されているか？
いいえ
はい
502
そのアクションを取らなければならないことを記録する
502
フィールドに対する具体的問い合わせ寄与を作成する
502
具体的問い合わせステートメントを追加する

【図７】


A
520
結果フィールドごとに
繰り返しが済んだ
534
何かセキュリティ・アクションが記録されているか？
いいえ
はい
問い合わせフィールド指定をリポジトリ抽象化から得る
522
536
記録されている最も厳格な規則を判定する
524
このフィールドに関してこのユーザについてのセキュリティ規則があるか？
いいえ
はい
538
致命的アクションか？
いいえ
はい
526
規則にはアクションが指定されているか？
いいえ
はい
540
問い合わせを実行せずに抜ける
終わり
528
そのアクションを取らなければならないことを記録する
542
問い合わせを実行する
530
フィールドに対する具体的問い合わせ寄与を作成する
532
具体的問い合わせステートメントを追加する

【図８】


フィールドに対する具体的問い合わせ寄与を作成する
602
単純アクセス・メソッドか？
いいえ
はい
606
フィルタされたアクセス・メソッドか？
いいえ
はい
612
合成アクセス・メソッドか？
いいえ
はい
618
他のアクセス・メソッドの処理
物理フィールド・ロケーションに基づいて問い合わせ寄与を作成する
604
物理フィールド・ロケーションに基づいて問い合わせ寄与を作成する
608
合成式内のフィールドの物理ロケーションを取り出す
614
寄与をフィルタ選択で拡張する
610
物理ロケーションおよび合成式を用いて寄与を作成する
616
続く
続く
続く
600

フロントページの続き

(74)代理人 100086243
弁理士 坂口 博
(72)発明者 デッティンガー、リチャード、ディーン
アメリカ合衆国５５９０１ ミネソタ州ロチャスター ノースウエスト ケンジントン・レーン
５３０５
(72)発明者 スティーブンス、リチャード、ジョゼフ
アメリカ合衆国５５９５５ ミネソタ州マントービル ２５２番アベニュー ６１４３２

審査官 深沢 正志

(56)参考文献 特開２０００－０３５９４９（ＪＰ，Ａ）
特開２００１－３４４１４７（ＪＰ，Ａ）
特開２００１－０９２８４４（ＪＰ，Ａ）
特表２００５－５２４１３８（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 12/00
G06F 21/00 - 21/24
G06F 17/30