(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2008-541226
(P2008-541226A)
(43)公表日　平成20年11月20日(2008.11.20)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 21/24	(2006.01)	G06F	12/14	520A	5B017
G06F 12/00	(2006.01)	G06F	12/00	537A	5B082

審査請求 未請求　予備審査請求 未請求　（全 20 頁）

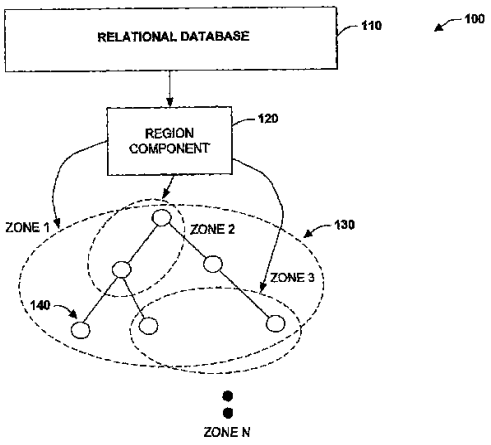
(21)出願番号　特願2008-509998 (P2008-509998)
(86)(22)出願日　平成18年3月9日 (2006.3.9)
(85)翻訳文提出日　平成19年11月5日 (2007.11.5)
(86)国際出願番号　PCT/US2006/008416
(87)国際公開番号　WO2006/118662
(87)国際公開日　平成18年11月9日 (2006.11.9)
(31)優先権主張番号　11/122,299
(32)優先日　平成17年5月4日 (2005.5.4)
(33)優先権主張国　米国 (US)

(71)出願人　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　９８０５２－６３９９　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　チークェン　リ
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

最終頁に続く

(54)【発明の名称】リージョンベースのセキュリティ

(57)【要約】
本発明は、階層関係を有するデータベースオブジェクトにリージョンベースのセキュリティを付与するシステムおよび方法に関する。一態様では、データベースセキュリティおよび管理を容易にするシステムが実現される。システムは、オブジェクト間に階層関係を有する複数のオブジェクトを格納するデータベースコンポーネントを備える。リージョンコンポーネントは、オブジェクトの部分集合に対するセキュリティゾーンを定義し、セキュリティデータをその部分集合に対応付け、そこでは、セキュリティゾーンは、オブジェクト間の階層関係から独立しているか、または減結合されているか、または切り離されている。



【特許請求の範囲】
【請求項１】
データベースのセキュリティおよび管理を容易にするシステムであって、
オブジェクト間に階層関係を有する複数の前記オブジェクトを格納するデータベースコンポーネントと、
前記オブジェクトの部分集合に対する１つまたは複数のセキュリティゾーンを定義し、セキュリティデータを前記部分集合に対応付けするリージョンコンポーネントであって、前記セキュリティゾーンは、前記オブジェクト間の前記階層関係から独立しているリージョンコンポーネントを備えることを特徴とするシステム。
【請求項２】
前記リージョンコンポーネントは、オブジェクトドメインからセキュリティドメインへの変換を行うことを特徴とする請求項１に記載のシステム。
【請求項３】
前記リージョンコンポーネントは、前記セキュリティゾーンのうちの少なくとも１つを定義する少なくとも１つのセキュリティディスクリプタを含むことを特徴とする請求項２に記載のシステム。
【請求項４】
前記リージョンコンポーネントは、前記セキュリティドメイン内のリージョン間の継承セキュリティをサポートすることを特徴とする請求項２に記載のシステム。
【請求項５】
前記リージョンコンポーネントは、セキュリティ変更の分析に基づいてセキュリティゾーンの展開またはセキュリティゾーンの折り畳みをサポートすることを特徴とする請求項１に記載のシステム。
【請求項６】
前記リージョンコンポーネントは、検出されたセキュリティ変更に基づき少なくとも３つのリージョンによりセキュリティリージョンを展開することを特徴とする請求項５に記載のシステム。
【請求項７】
前記セキュリティ変更は、アクセス制御エントリ（ＡＣＥ）により検出されることを特徴とする請求項６に記載のシステム。
【請求項８】
前記アクセス制御エントリは、明示的な、または暗黙のセキュリティ変更を表すことを特徴とする請求項７に記載のシステム。
【請求項９】
さらに、オブジェクトアイテムをセキュリティディスクリプタ識別子に関連付けるテーブルを備えることを特徴とする請求項１に記載のシステム。
【請求項１０】
さらに、セキュリティディスクリプタ識別子をセキュリティディスクリプタの内容に対応付けるテーブルを備えることを特徴とする請求項９に記載のシステム。
【請求項１１】
さらに、ルートノード、子、ノード、コンテナアイテム、および非コンテナアイテムのうちの少なくとも１つを含むことを特徴とする請求項１に記載のシステム。
【請求項１２】
さらに、前記セキュリティゾーンの間でセキュリティ変更を伝播するコンポーネントを備えることを特徴とする請求項１に記載のシステム。
【請求項１３】
さらに、前記リージョンコンポーネントまたは前記データベースを対話操作するための少なくとも１つのインターフェースを備えることを特徴とする請求項１に記載のシステム。
【請求項１４】

前記インターフェースは、セキュリティ取得関数、ディスクリプタ取得関数、セキュリティ設定関数、保持リンク追加関数、保持リンク削除関数、および実効セキュリティ取得関数を含むことを特徴とする請求項１３に記載のシステム。

【請求項１５】

さらに、セキュリティドメイン内でセキュリティオブジェクト関係を定義するセキュリティ階層行を含むことを特徴とする請求項１に記載のシステム。

【請求項１６】

請求項１に記載の前記コンポーネントを実装するためのコンピュータ可読命令を格納することを特徴とするコンピュータ可読媒体。

【請求項１７】

オブジェクトデータベースセキュリティのための方法であって、

オブジェクトドメイン内でデータベースオブジェクトを定義することと、

セキュリティドメイン内でセキュリティコンポーネントを定義することと、

前記オブジェクトドメインと前記セキュリティドメインとの間の対応付けを行うことと、

前記対応付けを使用して、前記データベースオブジェクトのセキュリティリージョンを定義することとを含むことを特徴とする方法。

【請求項１８】

さらに、前記セキュリティドメイン内の変更を検出した後、少なくとも３つのセキュリティリージョンを生成することを含むことを特徴とする請求項１７に記載の方法。

【請求項１９】

さらに、前記セキュリティドメイン内において前記セキュリティコンポーネントの継承メカニズムを備えることを含むことを特徴とする請求項１７に記載の方法。

【請求項２０】

データベースオブジェクトセキュリティを容易にするシステムであって、

データベースオブジェクトを格納する手段と、

前記オブジェクトについてセキュリティリージョンを設定する手段と、

前記セキュリティリージョンを前記オブジェクトに対応付ける手段と、

前記セキュリティリージョンに少なくとも一部は基づき前記オブジェクトにアクセスする手段とを備えることを特徴とするシステム。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、一般に、コンピュータシステムに関するものであり、より具体的には、古典的なオブジェクト継承階層におけるデータの伝播および格納に関する要件を緩和するために、セキュリティをオブジェクトの部分集合に、その部分集合に対するリージョンディスクリプタ（ｒｅｇｉｏｎ　ｄｅｓｃｒｉｐｔｏｒ）に基づいて付与するシステムおよび方法に関するものである。

【背景技術】

【０００２】

現代的な商用データベース設計は、大量のデータを格納し、管理し、操作する方法を含む、データに関する込み入った多くの考慮事項を伴う。このようなデータは、多くの場合、様々なオブジェクト間の継承プロパティを与えるオブジェクトツリーなどにおける他のデータとの入り組んだ関係を含む。これらの種類の関係があるため、このようなデータを管理する場合に、データベースおよびコンポーネントの効率的設計が複雑なものになることが多い。例えば、データベース設計プロセスの一態様は、リレーショナルデータベース管理システムによってデータが格納される方法を理解することにある。ユーザに対し情報を効率的に、また正確に提供するために、データベースプログラム側で、別々のテーブルに格納されている異なるサブジェクトに関するファクト（データ）にアクセスする必要がある。例えば、１つのテーブルには、従業員に関するファクトのみを格納し、他のテーブ

ルには、売上高に関するファクトのみを格納し、さらに他のテーブルには、企業に関する他の何らかの事柄を格納することができる。データを使用するときに、これらのファクトは自動的に結合され、様々な方法で提示される。例えば、ユーザは、従業員に関するファクトおよび売上高に関するファクトを結合したレポートを印刷することができる。

【０００３】

一般に、データベースを設計する場合、情報は、ライブラリ内の別々のサブジェクトなど何らかの順序で分割され、次いでデータベースプログラムがサブジェクトがどのように関係付けられるかを決定する。これらのプログラムは、多くの場合、構造化照会言語（ＳＱＬ）などの共通データベース言語を使用するリレーショナルデータベースのクエリを含む。このような言語をデータに適用する前に、通常は、どの型のデータが重要であり、そのようなデータをどのように編成すべきかについて何点か決定される。例えば、これらの決定は、どのデータを格納するかを決定するためのデータベースのスコープを確定することを含むことができる。次いで、情報を「Ｅｍｐｌｏｙｅｅｓ」または「Ｏｒｄｅｒｓ」などの別々のサブジェクトに分割するために必要なテーブルを決定する。すると、それぞれのサブジェクトは、データベース内の１つのテーブルとなる。他の態様としては、それぞれのテーブルにどのような情報を格納すべきかを決定するために必要なそれぞれのフィールドを決定することが挙げられる。テーブル内の情報のそれぞれのカテゴリは、フィールドと呼ばれ、テーブル内の列として表示される。例えば、Ｅｍｐｌｏｙｅｅｓテーブル内の１つのフィールドを、Ｌａｓｔ　Ｎａｍｅとし、他のフィールドをＨｉｒｅ　Ｄａｔｅとすることが可能である。他の考慮事項として、一方のテーブル内のデータが他のテーブル内のデータとどのような関係にあるかを決定するなど、関係の決定が挙げられる。設計者は、多くの場合、必要に応じて、関係を明確にするために、複数のフィールドをテーブルに追加するか、または新規テーブルを作成する。

【０００４】

データベースを設計する際に遭遇するであろうありがちな落とし穴がいくつかある。これらの問題があるため、データの使用および維持が難しくなることがある。このようなものとして、すべてが同じサブジェクトに関係しているわけではない多数のフィールドを１つのテーブルに詰め込むことなどが挙げられる。例えば、１つのテーブルに、顧客に関係するフィールドだけでなく、売上高情報を含むフィールドも入れる場合である。また、多くの場合、それぞれのテーブルに１つのサブジェクトのみに関係するデータを格納する場合に効率的なものとなる。他の場合には、多数のレコードにおいてレコードに適用できないためフィールドが故意にブランクのままにされたときにオーバーヘッドが生じる。これは、通常、フィールドが他のテーブルに属していることを意味する。冗長性は、多数のテーブルがあり、その多くが同じフィールドを有している場合に生じるもう１つの問題である。例えば、１月の売上高と２月の売上高に対するテーブル、または近場の顧客および遠隔地の顧客に対するテーブルを分離する場合であり、同じ種類の情報が格納され冗長性を有する。そのため、一技術は、１つのテーブル内の単一サブジェクトに関係するすべての情報を集約することである。

【発明の開示】

【発明が解決しようとする課題】

【０００５】

データベースのテーブルおよびフィールドの設定および設計の方法が複雑であることに加えて、他の事項についても考慮されなければならない。これらの事項としては、それぞれのテーブルおよびフィールドに対しデータセキュリティをどのように行うべきかという問題が挙げられる（例えば、ファイルに誰がまたは何がアクセスできるかなどのセキュリティ）。これは、階層オブジェクトなどのデータベース内に格納されている複雑な構造体にセキュリティを与える方法を含む。古典的には、セキュリティの考慮は、そのようなオブジェクトに対する継承階層内で伝播されており、階層内のそれぞれのアイテムは、複数のアイテムのうちの１つが変更された場合に更新される必要がある。しかし、リレーショナルデータベースのテーブル行を使用して階層オブジェクトを格納する実装では、セキュ

リティ情報またはデータをそれぞれのオブジェクト上にどのように設定し、継承モデルに基づいてセキュリティデータをその子オブジェクトにどのように初期値として入れるかというよくある問題に直面する。

【課題を解決するための手段】

【０００６】

以下では、本発明のいくつかの態様の基本的な内容を理解できるように、発明の開示を簡単に説明する。この発明の開示は、本発明の概要を広範囲にわたって述べたものではない。この発明の開示は、本発明の鍵となる／決定的な要素を示したり、本発明の範囲を定めることを目的としていない。後で述べるより詳細な説明の前置きとして、本発明のいくつかの概念を簡略化した形式で述べることのみを目的とする。

【０００７】

本発明は、オブジェクト間に階層関係を有する複数のデータベースオブジェクトにリージョンベースのセキュリティを付与するシステムおよび方法に関する。階層内に存在するオブジェクトの部分集合にセキュリティ情報を対応付けるリージョンコンポーネントを用意し、階層から独立している１つまたは複数のセキュリティゾーンを作成する。これにより、リージョンまたはゾーン内に存在するオブジェクトはデータベース処理要件を緩和するセキュリティ属性を共有することができる（例えば、セキュリティデータを更新するノードを少なくする）。一般に、古典的なデータベースアーキテクチャでは、多くの場合、リレーショナルデータベースのテーブル行を使用して、階層オブジェクトを格納するが、これによりさらに、関係するセキュリティディスクリプタをそれぞれのオブジェクトに設定し、さらに継承モデルに基づきそれぞれの子オブジェクトにセキュリティディスクリプタを初期値として入れることができる。これにより、それぞれのオブジェクト更新にかかる処理時間は長引く一方であるが、リージョンベースの考慮を持ち込むことにより緩和される。

【０００８】

リージョンは、同じまたは類似のセキュリティディスクリプタを共有するオブジェクトの集合体（連続的ツリー内にある必要はない）とすることができる。１つのオブジェクト上のセキュリティディスクリプタが更新されると、そのオブジェクトが属しているリージョンは、分割されるか、または折り畳まれる。例えば、リージョンは、子オブジェクト上の異なるセキュリティディスクリプタが変更の結果得られた場合に分割することができるが、変更により他のリージョンのセキュリティディスクリプタと同じセキュリティディスクリプタが得られる場合に他のリージョン内に折り畳むことができる。それぞれのオブジェクトが直接そのオブジェクト自身のセキュリティディスクリプタを所有する代わりに、リージョンがセキュリティディスクリプタを所有するようにすると、他のオブジェクト上のセキュリティディスクリプタに影響を及ぼす可能性のあるオブジェクト上のセキュリティディスクリプタが変更されたときのオブジェクト更新の回数が劇的に減少する。

【０００９】

一般に、リージョンは、古典的には、階層オブジェクトモデル内のオブジェクトのサブツリーとして定義される。本発明の場合、リージョンは、同じセキュリティディスクリプタを共有するオブジェクトの集合として定義され、これにより、同じセキュリティディスクリプタを共有するオブジェクトは、同じサブツリーの下になくてもよくなる。この間接作用により、オブジェクトのセキュリティディスクリプタを操作する効率的なプロセスが実現されうる。したがって、リージョンベースのセキュリティは、本質的に、オブジェクトドメインをセキュリティディスクリプタドメインに変換し、セキュリティディスクリプタ演算をセキュリティディスクリプタドメインに対し直接的に、またデータベース処理全体を軽減する階層とは無関係に、実行する。

【００１０】

前記の関係する目的を達成するために、本発明のいくつかの例示されている態様について、以下の説明および付属の図面に関して本明細書で説明される。これらの態様は、本発明を実施できる様々な方法を示しており、すべて、本発明の対象となることを意図されて

いる。本発明の他の利点および新規性のある特徴は、図面を参照しつつ本発明の以下の詳細な説明を読むと明白になるであろう。

【発明を実施するための最良の形態】

【００１１】

本発明は、階層関係を有するデータベースオブジェクトにリージョンベースのセキュリティを付与するシステムおよび方法に関する。本発明では、オブジェクト毎に別のセキュリティディスクリプタを更新するのではなく、リージョンという概念を導入し、これにより、与えられたオブジェクトのセキュリティは階層とは反対にリージョンとの関連付けから導出される。これは、個別のオブジェクト記述を必要とし、継承階層により課されたセキュリティを有する古典的アーキテクチャとは対照的である。この方法では、多くのオブジェクトがそれぞれのリージョンについてより大域的な規模で定義することができる類似のセキュリティ属性を共有することが可能であるため、データベース処理および格納は保存されうる。一態様では、データベースセキュリティおよび管理を容易にするシステムが実現される。システムは、オブジェクト間に階層関係を有する複数のオブジェクトを格納するデータベースコンポーネントを備える。リージョンコンポーネントは、オブジェクトの部分集合に対するセキュリティゾーンを定義し、セキュリティデータをその部分集合に対応付け、そこでは、セキュリティゾーンは、オブジェクト間の階層関係から独立しているか、または減結合されているか、または切り離されている。

【００１２】

本出願で使用されているように、「コンポーネント」、「システム」、「オブジェクト」、「ゾーン」などの用語は、コンピュータ関連のエンティティ、つまりハードウェア、ハードウェアとソフトウェアの組合せ、ソフトウェア、または実行中のソフトウェアのいずれかを指すことを意図されている。例えば、コンポーネントとして、限定はしないが、プロセッサ上で実行されているプロセス、プロセッサ、オブジェクト、実行可能ファイル、実行スレッド、プログラム、および／またはコンピュータなどがある。例えば、サーバ上で実行されているアプリケーションとサーバは両方ともコンポーネントであってよい。１つまたは複数のコンポーネントは、１つのプロセスおよび／または実行スレッド内に常駐することができ、またコンポーネントは、１台のコンピュータにローカルとして配置され、および／または２台以上のコンピュータ間に分散されることも可能である。また、これらのコンポーネントは、様々なデータ構造体が格納されている様々なコンピュータ可読媒体から実行することが可能である。コンポーネントは、１つまたは複数のデータパケットを有する信号などに従って、ローカルおよび／またはリモートプロセスを介して通信することができる（例えば、ローカルシステム、分散システム内の他方のコンポーネントと相互にやり取りする一方のコンポーネントからのデータ、および／または信号を介して他のシステムとインターネットなどのネットワークを介して相互にやり取りするデータ）。

【００１３】

最初に図１を参照すると、オブジェクトセキュリティシステム１００は、本発明の一態様に従って例示されている。システム１００は、１つまたは複数のオブジェクトセキュリティゾーン１３０を定義する（１つまたは複数の）リージョンコンポーネント１２０と関連付けられているリレーショナルデータベース１１０（例えば、ＳＱＬまたは他の種類のデータベース）を備える。一般に、オブジェクト階層の個別のノード（例えば、参照番号１４０の階層の１つのオブジェクトを見る）は、オブジェクトセキュリティ変更が行われたときに個別に更新されない。むしろ、セキュリティポリシーが、１３０のそれぞれのセキュリティゾーンに従ってリージョンコンポーネント１２０により割り当てられる。それぞれのオブジェクトを個別に更新するのではなく、オブジェクトをセキュリティゾーン１３０に対応付けることにより、データベース１１０における読み書きオペレーションの回数を減らすことができる。そのため、リージョンコンポーネント１２０は、それぞれのオブジェクトが更新される継承階層からのセキュリティポリシー対応付けをオブジェクトのゾーンが類似のセキュリティポリシーを共有するオブジェクトのセキュリティドメインに変換する。このようにして、古典的な継承階層内のそれぞれの個別オブジェクトを更新す

るのとは反対にセキュリティゾーン１３０の縮小部分集合を単に更新するだけでオブジェクトのセキュリティポリシーが変更されたときに、セキュリティ更新の小さな部分集合を伝播されうる。継承の概念は、システム１００内でポリシーを伝播するために使用されうるが、しかし、ツリー内のオブジェクト間の従来の継承と異なり、継承はセキュリティゾーン１３０の間にある。そのため、継承は、オブジェクトドメインではなくセキュリティドメイン内でモデル化されたコンポーネント同士の間で生じる。このことは、それぞれのオブジェクトに対するセキュリティ対応付けは、個別のオブジェクト１４０毎に明示的に開始するのではなく、オブジェクトとその関連付けられたゾーン１３０との間で行われることを意味する。したがって、リージョンコンポーネント１２０は、セキュリティを識別されたオブジェクトの１つのリージョンに付与し、本質的に、階層内のすべてのオブジェクトにセキュリティ変更を伝播する従来のオブジェクト階層を減結合するか、引き離すか、またはこの階層から独立している。

【００１４】

一般に、データベース１１０内のアイテムに、セキュリティディスクリプタに対する（識別子）ＩＤを割り当てることができる。データベースは、ＳＤＩＤ（セキュリティディスクリプタＩＤ）と呼ばれる列を有する［Ｔａｂｌｅ！Ｉｔｅｍ］を含む。このＳＤＩＤは、例えば、隠しＳＱＬシステムテーブル内に格納され、保持されるセキュリティディスクリプタの一意的なＩＤである。システムテーブルは、パブリックビュー（例えば、Ｓｙｓ．Ｓｅｃｕｒｉｔｙ＿Ｄｅｓｃｒｉｐｔｏｒ）を介して公開することができる。以下のテーブルは、セキュリティディスクリプタを基本オブジェクトモデル内に差し込むか、または関連付ける方法を簡単に示したものである。

【００１５】

【表１】

［Ｔａｂｌｅ！Ｉｔｅｍ］：アイテムをセキュリティディスクリプタＩＤに関連付ける。

_ItemID	...	_SDId	...

【００１６】

【表２】

［Ｓｙｓ．Ｓｅｃｕｒｉｔｙ＿Ｄｅｓｃｒｉｐｔｏｒ］：ＩＤをセキュリティディスクリプタの内容に対応付ける。

Sd_id	型	SecurityDescriptor	...

【００１７】

セキュリティディスクリプタＩＤ（ＳＤ　ＩＤ）をオブジェクトアイテムに効率よく割り当てるために、ＳＤリージョン技術は、一部は、大半のオブジェクトアイテムは同じセキュリティディスクリプタを共有する傾向を有するという監察結果に基づく。ＡＳＤリージョンは、同じまたは類似のＳＤ　ＩＤを共有するアイテムの集合（従来のシステムのように連続的である必要はない）である。典型的には、上に示されている［Ｔａｂｌｅ！ｉｔｅｍ］内のすべてのアイテムは、異なるＳＤリージョンにグループ化することができる。ＳＤリージョンの関係は、１つのＳＤリージョンのＳＤが上述のセキュリティドメイン内の他のＳＤリージョンのＳＤから継承することができるような方法で確立することがで

きる。本質的に、対応するオブジェクトアイテムツリーに相当するが、以下の図２および３に関して示されるようにノード数の少ない、ＳＤリージョンツリーが確定される。したがって、ＳＤリージョンツリーは、アイテムのＳＤを効率よく更新するために使用することができる。典型的には、セキュリティアイテムツリーが作成されるときに、２つのＳＤリージョンが作成され、ＳＤはツリー内の実質的にすべてのアイテムに割り当てられる。そのため、一方のＳＤリージョンはルートアイテム（明示的ＳＤが定義される場合）に対するものであり、他方のＳＤリージョンはそれぞれのコンテナアイテムに対するものであり、最後のＳＤは非コンテナアイテムに対するものである。

【００１８】

次に図２および３を参照すると、例示的なセキュリティドメイン変換２００および３００は、本発明の一態様に従って例示されている。図２の２００では、オブジェクトツリーのノードが例示されており、２１０の黒色ノードはルートアイテムであり、２２０の灰色ノードは、コンテナアイテムであり、２３０の白色ノードは、非コンテナアイテムである。アイテムのセキュリティディスクリプタ（ＳＤ）が更新される場合（例えば、ＳＤの所有者、グループ、アクセス制御リストなどを変更することにより）、アイテムが属しているＳＤリージョンは、２４０に例示されているように３つのサブグループまたは部分集合に分割されること可能である。セキュリティ変更は、一般に、明示的または暗黙の形式とすることができるアクセス制御エントリ（ＡＣＥ）と呼ばれるデータを通じて行われる。明示的なＡＣＥがアイテムのＳＤに加えられたときに、新しいＳＤリージョンをこのアイテムの周りに生成することができる。この場合、３つのＳＤリージョンが生成され、内訳はアイテム（明示的なＡＣＥが追加される場合）それ自体に対し１つ、そのコンテナの子に１つ、および非コンテナの子に１つである。図３を参照すると、非伝播明示的ＡＣＥが３１０でアイテム上のＳＤに追加された場合に、より複雑な状況が例示されており、この場合、５つの新しいリージョンが３２０に例示されているようにアイテムの周りに作成される。この場合、アイテム自体（明示的ＡＣＥが追加される場合）３３０に対し１つのリージョン、その直接的なコンテナの子に１つのリージョン、３５０のその直接的非コンテナの子に１つのリージョン、３６０のその非直接的コンテナの子に１つのリージョン、および３７０の非直接的非コンテナの子に１つのリージョンが作成される。

【００１９】

図２および３を要約して言うと、アイテムのＳＤが明示的に更新された場合に新規リージョンを作成できるということである（継承を通してではなく）。一般に、ＳＤに対し行われる更新に応じて、３または５つの新しいリージョン（他の量も可能）が作成される。非伝播ＡＣＥが追加された場合に５つのＳＤリージョンが作成され、３つのＳＤリージョンは、一般に他の場合に作成される。例えば、ＳＤが非継承プロパティ（ほとんどの場合において非継承ＡＣＥ）をＲｏｏｔ　Ｉｔｅｍとして含むアイテムを仮定する。上記のように、コンテナ型Ｒｏｏｔ　Ｉｔｅｍは、ＳＤにおける明示的ＡＣＥの型に応じて３または５つのリージョンを所有することができる。非コンテナは、ＳＤが明示的プロパティを持っている場合にそれ独自のＳＤリージョンを持つことができる。Ｒｏｏｔ　ＩｔｅｍのＳＤの明示的プロパティすべてが削除された場合、このＲｏｏｔ　Ｉｔｅｍが所有するＳＤリージョンは、その親アイテムのＳＤ内に折り畳むことができ、そうして、個別のオブジェクトセキュリティ更新を軽減する。それぞれのＳＤリージョンは、以下の例などのＳｅｃｕｒｉｔｙ＿Ｈｉｅｒａｃｈｙテーブル内の行として表すことができる。

【００２０】

【表３】

［Ｔａｂｌｅ！Ｓｅｅｎｒｉｔｙ＿Ｈｉｅｒａｃｈｙ］：ＳＤ継承関係を格納し、同じセキュリティディスクリプタを共有するアイテムを確定する。

_SDIdParent	_SDId	_RootItemID	_IsContainer	_Scope

【００２１】

上記テーブルの列はＳＤリージョンのＩＤである＿ＳＤＩｄ、継承されたセキュリティプロパティの元になっているＳＤのＩＤである＿ＳＤＩｄＰａｒｅｎｔフィールド、明示的ＳＤが定義されるアイテムのＩＤである＿ＲｏｏｔＩｔｅｍＩＤフィールド、ＳＤがコンテナに適用される場合に１、非コンテナに適用される場合に０である＿ＩｓＣｏｎｔａｉｎｅｒフィールド、および以下のように符号化される＿Ｓｃｏｐｅフィールドを含むことができる。０：ＳＤがＲｏｏｔ　Ｉｔｅｍに適用される。１：ＳＤがＲｏｏｔ　Ｉｔｅｍの子に適用される。２：ＳＤがＲｏｏｔ　Ｉｔｅｍの直接的子に適用される。３：ＳＤがＲｏｏｔ　Ｉｔｅｍの非直接的子に適用される。

【００２２】

データベースがブートストラップされるときに、必要ならば３つの既定のセキュリティディスクリプタを作成することができる、つまり、最上位のＲｏｏｔ　Ｉｔｅｍに対し１つのディスクリプタ、すべてのコンテナの子に対し１つのディスクリプタ、およびすべての非コンテナの子に対し１つのディスクリプタを作成することができるということに留意されたい。したがって、さらに、最上位のＲｏｏｔ　Ｉｔｅｍ上に３つのＳＤリージョンを作成することができる。典型的には、ボリューム内にその後作成されるすべてのアイテムは、これらのＳＤのうちの１つを既定のＳＤとして持つことができる。明示的なＡＣＥがアイテムに加えられたときに、新しいＳＤリージョンを上述のように生成することができる。

【００２３】

図４は、本発明の一態様による例示的なセキュリティインターフェース４００を示している。上述のリージョンベースの考慮事項を対話操作するために、様々なセキュリティインターフェース４００を備えることができる。以下では、適用することができるインターフェースの実施例を２、３説明する。これらは、以下でさらに詳しく説明するように、４１０でセキュリティデータを取り出すためのインターフェース、４２０でセキュリティ情報を設定するためのインターフェース、およびリンクを保持するためのインターフェースを備えることができる。以下のコード断片は、これらのインターフェース４００の一部に対するパブリック宣言の一実施例である。

【００２４】

【表４】

```
Public sealed class ItemSecurity
{
        public ItemSecurity( Guid itemId )

        public string GetSDDLSecurity()

        public GenericSecurityDescriptor GetSecurity()

        public void SetSDDLSecurity( string sd, SECURITY_INFORMATION
        si )

        public void SetSecurity( GenericSecurityDescriptor gsd ,
                                        SECURITY_INFORMATION si )

        public string GetUserEffectiveSecurity()

        public void AddHoldingLink( Guid itemId )
        public void RemoveHoldingLink( Guid itemId )
}
```

【００２５】

以下は、セキュリティインターフェース４１０から４３０について簡単な説明を行う。

public string GetSDDLSecurity()－ＳＤＤＬ文字列形式でアイテムのセキュリティディスクリプタ全体を取り出す。継承された、明示的なアクセス制御リストを含む。

public GenericSecurityDescriptorGetSecurity()－Ｍａｎａｇｅｄ　ＡＣＬｓクラスＧｅｎｅｒｉｃＳｅｃｕｒｉｔｙＤｅｓｃｒｉｐｔｏｒの形式でアイテムのセキュリティディスクリプタ全体を取り出す。

public void SetSDDLSecurity (string sd, SECURITY_INFORMATION si)－アイテム上にセキュリティディスクリプタを設定する。この関数は、継承されたＡＣＥを無視する。これは、親および他の保持リンクから継承されたＡＣＥを再生成する。所有者、グループ、制御フラグ、または明示ＡＣＥを設定するために呼び出すことができる。

ＳＥＣＵＲＩＴＹ＿ＩＮＦＯＲＭＡＴＩＯＮは、セキュリティディスクリプタのどの部分を更新するかを指定する。

public void SetSecurity (GenericSecurityDescriptor gsd, SECURITY_INFORMATION si)－アイテム上にセキュリティディスクリプタを設定する。Ｍａｎａｇｅｄ　ＡＣＬｓクラスを入力パラメータとして受け取る。

public void AddHoldingLink (Guid itemId)－新しい保持リンクをアイテムに追加したときにアイテム上のセキュリティディスクリプタを更新する。

public void RemoveHoldingLink (Guid itemId)－新しい保持リンクをアイテムから削除したときにアイテム上のセキュリティディスクリプタを更新する。

public string GetUserEffectiveSecurity()－現在のセキュリティコンテキストに関連するＡＣＥを含むアイテム上のセキュリティディスクリプタを取り出す。

【００２６】

図５は、本発明の一態様によるリージョンコンポーネント処理５００を例示している。５１０で、リージョン定義が行われる。これらは、同じＳＤを共有するアイテムの集合であるセキュリティディスクリプタ（ＳＤ）リージョンを含む。アイテムの集合は、連続ツリーを形成しなくてよい。セキュリティ階層（ＳＨ）行は、後述の［Ｔａｂｌｅ！Ｓｅｃｕｒｉｔｙ＿Ｈｉｅｒａｃｈｙ］テーブル内の１つの行である。それぞれのＳＤリージョンは、テーブル内にＳＨ行を持っていなければならない。

【００２７】

【表５】

_ParentSDId	_SDId	_RootItemId	_IsContainer	_Scope
SD0	SD1	ItemId	0	3

【００２８】
上記テーブル内の行は、ＳＤリージョンに対応するＳＨ行として参照される。このテーブル内の行は、アイテムの集合（単一アイテムでもよい）が同じセキュリティディスクリプタ（上の実施例の中のＳＤ１）を共有することを示す。このアイテムの集合は、共通ルート（ＩｔｅｍＩｄ）、共通型（コンテナまたは非コンテナ）、およびスコープにより定義される。スコープは、異なるオペレーティングシステムのセキュリティモデルをサポートするためのオプションである。
【００２９】
５２０で、リージョンのマージおよび作成の考慮事項について説明される。この点に関して、以下の条件の下で新しいＳＤ領域を１つ作成することができる。
１．非コンテナアイテム上で行われるＳＤ変更。
３つの新しいＳＤリージョンは、以下の条件の下で作成することができる。
１．コンテナアイテム上で行われるＳＤ変更。
２．ＳＤ変更は、非伝播ＡＣＥを含まない。
５つの新しいＳＤリージョンは、以下の条件の下で作成することができる。
１．コンテナアイテム上で行われるＳＤ変更。
２．ＳＤ変更は、非伝播ＡＣＥを含む。
【００３０】
ＳＤリージョンは、以下の条件の下でマージすることができる。
１．親ＳＤは、子ＳＤをフラッシュすることによりＳＤ継承を強制するか、または、
２．明示的ＡＣＥが、ＳＤから削除される。
【００３１】
５３０で、図６に関して説明されている以下のアルゴリズムにおいて使用することができる様々な概念が提示されている。これらの概念は以下を含む。
＿Ｉｔｅｍまたは＊－オペレーションが適用される現在のアイテムシステム。
ＳＤＩｄ（ｘ）またはＳＤＩｄ－アイテムｘ上のセキュリティディスクリプタのｓｄｉｄ。
ＳＤＩｄ＿ＮＣ（ｘ）またはＳＤＩｄ＿ＮＣ－ＳＤＩｄは、アイテムｘの非コンテナ子オブジェクトに適用される。
ＳＤＩｄ＿Ｃ（ｘ）またはＳＤＩｄ＿Ｃ－ＳＤＩｄは、アイテムｘのコンテナ子オブジェクトに適用される。
ＳＤＩｄ＿ＮＣ２（ｘ）またはＳＤＩｄ＿ＮＣ２－ＳＤＩｄは、アイテムｘの直接的非コンテナ子オブジェクトに適用される。
ＳＤＩｄ＿Ｃ２（ｘ）またはＳＤＩｄ＿Ｃ２－ＳＤＩｄは、アイテムｘの直接的コンテナ子オブジェクトに適用される。
ＳＤＩｄ＿ＮＣ３（ｘ）またはＳＤＩｄ＿ＮＣ３－ＳＤＩｄは、アイテムｘの非直接的非コンテナ子オブジェクトに適用される。
ＳＤＩｄ＿Ｃ３（ｘ）またはＳＤＩｄ＿Ｃ３－ＳＤＩｄは、アイテムｘの非直接的コンテナ子オブジェクトに適用される。
ＳＨＲｏｗ（ｘ，ｉ，ｊ）－＿ＲｏｏｔＩｔｅｍＩｄ＝ｘ、＿ＩｓＣｏｎｔａｉｎｅｒ＝ｉ、＿Ｓｃｏｐｅ＝ｊである［Ｔａｂｌｅ！Ｓｅｃｕｒｉｔｙ＿Ｈｉｅｒａｃｈｙ］テーブル内の行。
UpdateItemSD (OldSDId, NewSDId, RootItem, IsContainer, Scope)－現在のＳＤＩｄ＝ＯｌｄＳＤＩｄであり、祖先がＮｅｗＳＤＩｄのスコープ内のＲｏｏｔＩｔｅｍである

型（ＩｓＣｏｎｔａｉｎｅｒ）のすべてのアイテムのＳＤＩｄを更新する。

【００３２】

UpdateSDBlob (SDId)－その子のＳＤＩｄがこのＳＤＩｄを含むサイクルを形成しない場合にこのＳＤＩｄおよびその子のセキュリティディスクリプタの内容を更新する。例えば、保持リンク（ＳＤ０を持つ）が、［Ｔａｂｌｅ！Ｓｅｃｕｒｉｔｙ＿Ｈｉｅｒａｃｈｙ］テーブル内のそれ専用の行を持たないファイルアイテム（ＳＤ１を持つ）に加えられたときに、３つの行（ＳＤ０，ＳＤ１，＿Ｉｔｅｍ，０，０）、（ＳＤ１，ＳＤ０，＿Ｉｔｅｍ，０，１）、（ＳＤ１，ＳＤ０，＿Ｉｔｅｍ，１，１）が作成される。ここで、このアイテムの子アイテムに対しＳＤ０を再利用することで、［Ｔａｂｌｅ！Ｉｔｅｍ］テーブル内の更新回数を著しく減らす。

UpdateSDId (SDId, SDId New)－＿ＳＤＩｄ＝ＳＤＩｄである［Ｔａｂｌｅ！Ｓｅｃｕｒｉｔｙ＿Ｈｉｅｒａｃｈｙ］内の現在のアイテムの行を更新して＿ＳＤＩｄ＝ＳＤＩｄ Ｎｅｗに設定する。

UpdateParentSDId (SDIdPar, SDIdPar New)－ＰａｒｅｎｔＳＤＩｄ＝ＳＤＩｄＰａｒである［Ｔａｂｌｅ！Ｓｅｃｕｒｉｔｙ＿Ｈｉｅｒａｃｈｙ］内の現在のアイテムの行を更新して＿ＰａｒｅｎｔＳＤＩｄ＝ＳＤＩｄＰａｒ Ｎｅｗに設定する。

CreateNewSD (SDId)－現在のＳＤから新しいＳＤを作成し、さらに変更を加える（ＡＣＥを追加／削除、保持リンクを追加／削除）。

【００３３】

図６は、本発明の一態様による例示的なリージョン処理アルゴリズム６００を示している。この態様では、少なくとも３つの別々の、または組み合わせたアルゴリズム６００を使用して、リージョンプロセスを実行することができる。これらは、６１０のセキュリティディスクリプタの設定、６２０の保持リンクの追加、および６３０の保持リンク削除アルゴリズムを含む。セキュリティディスクリプタの設定６１０に関しては、少なくとも以下を含むオブジェクト上のセキュリティディスクリプタを変更する様々な方法がある。

・ 非継承可能明示的ＡＣＥを追加／削除する。

・ このアイテムおよびその子すべてに適用される継承可能明示的ＡＣＥを追加／削除する。

・ その子にのみ適用される継承可能明示的ＡＣＥを追加／削除する。

・ このアイテムおよびその直接的子のみに適用される継承可能明示的ＡＣＥを追加／削除する。

・ 子コンテナにのみ適用される継承可能明示的ＡＣＥを追加／削除する。

・ 子オブジェクトにのみ適用される継承可能明示的ＡＣＥを追加／削除する。

・ 特定の型のオブジェクトにのみ適用される継承可能明示的ＡＣＥを追加／削除する。

・ セキュリティディスクリプタの所有者を変更する。

・ セキュリティディスクリプタのグループを変更する。

・ セキュリティディスクリプタ制御フラグを変更する。

ｉ．ＡＣＥの継承を停止する。

ｉｉ．ＡＣＥの継承を開始する。

ｉｉｉ．このアイテムにのみ適用される他の制御フラグを制御する。

【００３４】

６２０で、保持リンクがアイテムに適用されたときに、このアイテム上のセキュリティディスクリプタは、保持リンクが継承可能ＡＣＥを持つかどうか、このアイテム上のＳＤのＳＥ＿ＤＡＣＬＥ＿ＰＲＯＴＥＣＴＥＤフラグがオンになっているかどうかに応じて、変更される場合も、変更されない場合もある。しかし、［Ｔａｂｌｅ！Ｓｅｃｕｒｉｔｙ＿Ｈｉｅｒａｃｈｙ］テーブルは、更新されなければならない。保持リンクがアイテム上に追加されたとき、アイテムが指定された行をまだ持っていない場合に、そのアイテムに対する３つの新しい行が［Ｔａｂｌｅ！Ｓｅｃｕｒｉｔｙ＿Ｈｉｅｒａｃｈｙ］テーブル内に追加されなければならない。［Ｔａｂｌｅ！Ｉｔｅｍ］テーブル内の更新を減らすた

めに、（ＳＤ０，ＳＤ１，＊，０，０）、（ＳＤ１，ＳＤ０，＊，０，１）、（ＳＤ１，ＳＤ０，＊，１，１）の形式を使用して、行を作成することができるが、ただし、ＳＤ０は、保持リンクのターゲットアイテムの古いＳＤＩｄであり、ＳＤ１は、ターゲットアイテムの新しいＳＤＩｄである。この方式により、［Ｔａｂｌｅ！Ｉｔｅｍ］テーブル内のソースアイテムを更新するだけでよくなる。

【００３５】

この方式に基づき、明示的非継承可能ＡＣＥが後でこのアイテムに追加される場合、［Ｔａｂｌｅ！Ｉｔｅｍ］テーブル内の更新を実行しない。６３０で、削除すべき保持リンク上のセキュリティディスクリプタのＳＤＩｄがＳＤＩｄ＿ＨＤであると仮定することができる。保持リンクを削除する場合、ＳＤリージョンは、折り畳むことができ、そのため［Ｔａｂｌｅ！Ｓｅｃｕｒｉｔｙ＿Ｈｉｅｒａｃｈｙ］内の行をマージすることができる。

【００３６】

図７は、本発明の一態様によるデータベースオブジェクトのセキュリティのための例示的なセキュリティリージョンプロセス７００を示している。説明を簡単にするために、方法が図に例示され、一連の、または多数の活動として記述されているが、本発明は活動の順序によって制限されるわけではなく、本発明により、いくつかの活動はその図に示されているここで説明しているのと異なる順序で、および／または他の活動と同時に実行することも可能であることが理解され、認識されるであろう。例えば、当業者であれば、代替えとして方法を一連の相互に関連のある状態またはイベントとして状態図などの中に表されることが可能であることを理解し、認識するであろう。さらに、本発明により、方法を実装するために例示されているすべての活動が必要なわけではない。

【００３７】

図７の７１０に進むと、データベース内のそれぞれのオブジェクトに対するセキュリティディスクリプタは、階層内の潜在的更新を考慮して更新される（セキュリティに関して）それぞれのオブジェクトに対する要件を取り除くことにより古典的なオブジェクト階層から減結合または引き離される。７２０において、１つまたは複数のセキュリティディスクリプタを使用して、データベース内に常駐するオブジェクトに対しオブジェクトリージョンを定義する。上述のように、これは、リージョンの類似のセキュリティデータにサブスクライブするセキュリティリージョンまたはオブジェクト部分集合を定義するために、類似の、または異なるオブジェクトツリーからのオブジェクトセキュリティデータを折り畳むか、またはマージすることを含むことができる。また、このようなリージョンデータは、そのリージョンに属する他のオブジェクトとの結果として得られる関係を含めて、データベースの行内に定義することができる。７３０において、オブジェクトセキュリティポリシーは、データベース内の選択されたリージョンに従って設定される。上記のように、アクセス制御エントリ（暗黙／明示）の種類およびオブジェクト階層内のセキュリティ変更の場所に応じて、様々なセキュリティリージョンをこのような設定から作成することができる。７４０では、古典的なオブジェクトドメインと本発明のセキュリティドメインとの間で変換が行われ、データベース内のセキュリティ変更を伝播する。これは、セキュリティ変更がそのオブジェクトに対して要求されたときに与えられたオブジェクトの周りにリージョン部分集合を作成することを含むことができる（例えば、セキュリティ変更の種類に応じて３または５つのリージョンを作成する）。

【００３８】

図８を参照すると、本発明の様々な態様を実装するための例示的環境８１０はコンピュータ８１２を含んでいる。コンピュータ８１２は、処理ユニット８１４、システムメモリ８１６、およびシステムバス８１８を備える。システムバス８１８は、限定はしないが、システムメモリ８１６を含むシステムコンポーネントを処理ユニット８１４に結合する。処理ユニット８１４は、様々な市販プロセッサがあるがそのうちのどれでもよい。デュアルマイクロプロセッサおよびその他のマルチプロセッサアーキテクチャも、処理ユニット８１４として採用されうる。

【００３９】
システムバス８１８は、メモリバスまたはメモリコントローラ、周辺機器バスまたは外部バス、および／または、限定はしないが、１１ビットバス、Ｉｎｄｕｓｔｒｉａｌ Ｓｔａｎｄａｒｄ Ａｒｃｈｉｔｅｃｔｕｒｅ（ＩＳＡ）、Ｍｉｃｒｏ－Ｃｈａｎｎｅｌ Ａｒｃｈｉｔｅｃｔｕｒｅ（ＭＳＡ）、Ｅｘｔｅｎｄｅｄ ＩＳＡ（ＥＩＳＡ）、Ｉｎｔｅｌｌｉｇｅｎｔ Ｄｒｉｖｅ Ｅｌｅｃｔｒｏｎｉｃｓ（ＩＤＥ）、ＶＥＳＡ Ｌｏｃａｌ Ｂｕｓ （ＶＬＢ）、Ｐｅｒｉｐｈｅｒａｌ Ｃｏｍｐｏｎｅｎｔ Ｉｎｔｅｒｃｏｎｎｅｃｔ（ＰＣＩ）、Ｕｎｉｖｅｒｓａｌ Ｓｅｒｉａｌ Ｂｕｓ（ＵＳＢ）、Ａｄｖａｎｃｅｄ Ｇｒａｐｈｉｃｓ Ｐｏｒｔ（ＡＧＰ）、Ｐｅｒｓｏｎａｌ Ｃｏｍｐｕｔｅｒ Ｍｅｍｏｒｙ Ｃａｒｄ Ｉｎｔｅｒｎａｔｉｏｎａｌ Ａｓｓｏｃｉａｔｉｏｎバス（ＰＣＭＣＩＡ）、およびＳｍａｌｌ Ｃｏｍｐｕｔｅｒ Ｓｙｓｔｅｍｓ Ｉｎｔｅｒｆａｃｅ（ＳＣＳＩ）をはじめとする利用可能な各種バスアーキテクチャを使用するローカルバスなど数種類のバス構造のうちのいずれでもよい。

【００４０】
システムメモリ８１６は、揮発性メモリ８２０および不揮発性メモリ８２２を含む。起動時などにコンピュータ８１２内の要素間の情報転送を行うための基本ルーチンを含む基本入出力システム（ＢＩＯＳ）は、不揮発性メモリ８２２に格納される。例えば、限定はしないが、不揮発性メモリ８２２には、読み取り専用メモリ（ＲＯＭ）、プログラム可能ＲＯＭ（ＰＲＯＭ）、電気的プログラム可能ＲＯＭ（ＥＰＲＯＭ）、電気的消去可能ＲＯＭ（ＥＥＰＲＯＭ）、またはフラッシュメモリなどがある。揮発性メモリ８２０には、外部キャッシュメモリとして動作するランダムアクセスメモリ（ＲＡＭ）がある。例えば、限定はしないが、使用可能なＲＡＭには、同期ＲＡＭ（ＳＲＡＭ）、ダイナミックＲＡＭ（ＤＲＡＭ）、同期ＤＲＡＭ（ＳＤＲＡＭ）、Ｄｏｕｂｌｅ Ｄａｔａ Ｒａｔｅ ＳＤＲＡＭ（ＤＤＲ ＳＤＲＡＭ）、Ｅｎｈａｎｃｅｄ ＳＤＲＡＭ（ＥＳＤＲＡＭ）、Ｓｙｎｃｈｌｉｎｋ ＤＲＡＭ（ＳＬＤＲＡＭ）、およびＤｉｒｅｃｔ Ｒａｍｂｕｓ ＲＡＭ（ＤＲＲＡＭ）など様々な形態のものがある。

【００４１】
コンピュータ８１２は、さらに、取り外し可能／取り外し不可能な揮発性／不揮発性コンピュータ記憶媒体も備える。図８は、例えばディスク記憶装置８２４を示している。ディスク記憶装置８２４は、限定はしないが、磁気ディスクドライブ、フロッピー（登録商標）ディスクドライブ、テープドライブ、Ｊａｚドライブ、Ｚｉｐドライブ、ＬＳ－１００ドライブ、フラッシュメモリカード、またはメモリスティックを含む。さらに、ディスク記憶装置８２４は、記憶媒体を、単独で備えることも、また限定はしないが、コンパクトディスクＲＯＭデバイス（ＣＤ－ＲＯＭ）、ＣＤ書き込み可能ドライブ（ＣＤ－Ｒドライブ）、ＣＤ書き換え可能ドライブ（ＣＤ－ＲＷドライブ）、またはデジタル多用途ディスクＲＯＭドライブ（ＤＶＤ－ＲＯＭ）などの光ディスクドライブを含む他の記憶媒体と組み合わせて備えることもできる。ディスク記憶装置８２４をシステムバス８１８に接続しやすくするために、通常、インターフェース８２６などの取り外し可能または取り外し不可能インターフェースを使用する。

【００４２】
図８は、ユーザと適当な動作環境８１０内の説明されている基本コンピュータ資源との媒介手段として動作するソフトウェアを説明していることは理解されるであろう。このようなソフトウェアとして、オペレーティングシステム８２８がある。オペレーティングシステム８２８は、ディスク記憶装置８２４に格納されることができ、コンピュータシステム８１２の資源の制御および割り当てを行う働きをする。システムアプリケーション８３０は、システムメモリ８１６またはディスク記憶装置８２４に格納されているプログラムモジュール８３２およびプログラムデータ８３４を通じてオペレーティングシステム８２８により資源の管理を利用する。本発明は、様々なオペレーティングシステムまたはオペレーティングシステムの組合せで実装できることは理解されるであろう。

【００４３】

ユーザは（複数の）入力デバイス８３６を使用してコマンドまたは情報をコンピュータ８１２に入力する。入力デバイス８３６は、限定はしないが、マウスなどのポインティングデバイス、トラックボール、ペン、タッチパッド、キーボード、マイク、ジョイスティック、ゲームパッド、衛星放送受信アンテナ、スキャナ、ＴＶチューナーカード、デジタルカメラ、デジタルビデオカメラ、Ｗｅｂカメラなどを含む。これらの入力デバイスやその他の入力デバイスは、（複数の）インターフェースポート８３８を介してシステムバス８１８を通じて処理ユニット８１４に接続する。例えば、（複数の）インターフェースポート８３８には、シリアルポート、パラレルポート、ゲームポート、およびユニバーサルシリアルバス（ＵＳＢ）がある。（複数の）出力デバイス８４０は、（複数の）入力デバイス８３６と同じ種類のポートのうちいくつかを使用する。したがって、例えば、ＵＳＢポートは、コンピュータ８１２に入力し、コンピュータ８１２からの情報を出力デバイス８４０に出力するために使用されることができる。出力アダプタ８４２が備えられており、特別なアダプタを必要とする他の出力デバイス８４０のうちモニタ、スピーカ、およびプリンタなどいくつかの出力デバイスがあることを示している。出力アダプタ８４２は、例えば、限定はしないが、出力デバイス８４０とシステムバス８１８とを接続する手段となるビデオおよびサウンドカードを含む。他のデバイスおよび／またはデバイスのシステムは（複数の）リモートコンピュータ８４４などの入出力機能を備えることに留意されたい。

【００４４】

コンピュータ８１２は、（複数の）リモートコンピュータ８４４などの１つまたは複数のリモートコンピュータへの論理接続を使用してネットワーク接続環境で動作させることができる。（複数の）リモートコンピュータ８４４は、パーソナルコンピュータ、サーバ、ルータ、ネットワークＰＣ、ワークステーション、マイクロプロセッサベースの機器、ピアデバイス、またはその他の共通ネットワークノードなどとすることができ、通常は、コンピュータ８１２に関係する上述の要素の多くまたはすべてを含む。簡単のため、メモリ記憶デバイス８４６のみ（複数の）リモートコンピュータ８４４とともに例示されている。（複数の）リモートコンピュータ８４４は、ネットワークインターフェース８４８を通じてコンピュータ８１２に論理的に接続され、通信接続８５０を介して物理的に接続される。ネットワークインターフェース８４８は、ローカルエリアネットワーク（ＬＡＮ）およびワイドエリアネットワーク（ＷＡＮ）などの通信ネットワークを含む。ＬＡＮ技術には、Ｆｉｂｅｒ Ｄｉｓｔｒｉｂｕｔｅｄ Ｄａｔａ Ｉｎｔｅｒｆａｃｅ（ＦＤＤＩ）、Ｃｏｐｐｅｒ Ｄｉｓｔｒｉｂｕｔｅｄ Ｄａｔａ Ｉｎｔｅｒｆａｃｅ（ＣＤＤＩ）、Ｅｔｈｅｒｎｅｔ（登録商標）／ＩＥＥＥ ８０２．３、Ｔｏｋｅｎ Ｒｉｎｇ／ＩＥＥＥ ８０２．５などがある。ＷＡＮ技術には、限定はしないが、２地点間接続リンク、統合デジタル通信網（ＩＳＤＮ）などの回線交換ネットワークとその変種、パケット交換ネットワーク、およびデジタル加入者回線（ＤＳＬ）などがある。

【００４５】

（複数の）通信接続８５０とは、ネットワークインターフェース８４８をバス８１８に接続するために使用されるハードウェア／ソフトウェアのことである。通信接続８５０はわかりやすくするためにコンピュータ８１２内に示されているが、コンピュータ８１２の外部にあってもかまわない。ネットワークインターフェース８４８の接続に必要なハードウェア／ソフトウェアには、例えば、通常の電話グレードのモデム、ケーブルモデム、およびＤＳＬモデムを含むモデム、ＩＳＤＮアダプタ、およびＥｔｈｅｒｎｅｔ（登録商標）カードなどの内部および外部技術がある。

【００４６】

図９は、本発明との相互やり取りが可能なコンピューティング環境例９００の概略ブロック図である。システム９００は、１つまたは複数のクライアント９１０を備える。（複数の）クライアント９１０は、ハードウェアおよび／またはソフトウェア（例えば、スレッド、プロセス、コンピューティングデバイス）とすることができる。システム９００は、さらに、１つまたは複数のサーバ９３０も備える。サーバ９３０も、ハードウェアおよ

び／またはソフトウェア（例えば、スレッド、プロセス、コンピューティングデバイス）とすることができる。サーバ９３０は、例えば、本発明を採用することにより変換を実行するスレッドを置くことができる。クライアント９１０とサーバ９３０との間で可能な通信の１つは、２つまたはそれ以上のコンピュータプロセス間で伝送されるように適合されたデータパケットの形であってよい。システム９００は、（複数の）クライアント９１０と（複数の）サーバ９３０との間の通信を容易に行えるようにするために採用されることができる通信フレームワーク９５０を含む。（複数の）クライアント９１０は、（複数の）クライアント９１０のローカルにある情報を格納するために使用されることができる１つまたは複数のクライアントデータストア９６０に動作可能なように接続される。同様に、（複数の）サーバ９３０は、（複数の）サーバ９３０のローカルにある情報を格納するために使用されることができる１つまたは複数のサーバデータストア９４０に動作可能なように接続される。

【００４７】

上述した内容は、本発明の複数の実施例を含む。もちろん、本発明を説明するためにコンポーネントまたは方法の考えられるすべての組合せを説明することは不可能であるが、当業者であれば、本発明の他の多くの組合せおよび置換が可能であることを理解できるであろう。したがって、本発明は、付属の請求項の精神と範囲内に収まるすべてのそのような変更、修正、および変更形態を包含することが意図されている。さらに、「含む、備える（ｉｎｃｌｕｄｅ）」という言い回しが詳細な説明または請求項で使用されている範囲において、「備える、含む、からなる（ｃｏｍｐｒｉｓｉｎｇ）」が使用された場合に請求項の中で接続語として解釈されるのでこのような用語は「備える、含む、からなる（ｃｏｍｐｒｉｓｉｎｇ）」という用語と同様の使い方で包含的であることが意図される。

【図面の簡単な説明】

【００４８】

【図１】本発明の一態様によるオブジェクトセキュリティシステムを例示する概略ブロック図である。

【図２】本発明の一態様による例示的なセキュリティドメイン変換を例示する図である。

【図３】本発明の一態様による他のセキュリティドメイン変換を例示する図である。

【図４】本発明の一態様による例示的なセキュリティインターフェースを示す図である。

【図５】本発明の一態様によるリージョンコンポーネント処理を例示する図である。

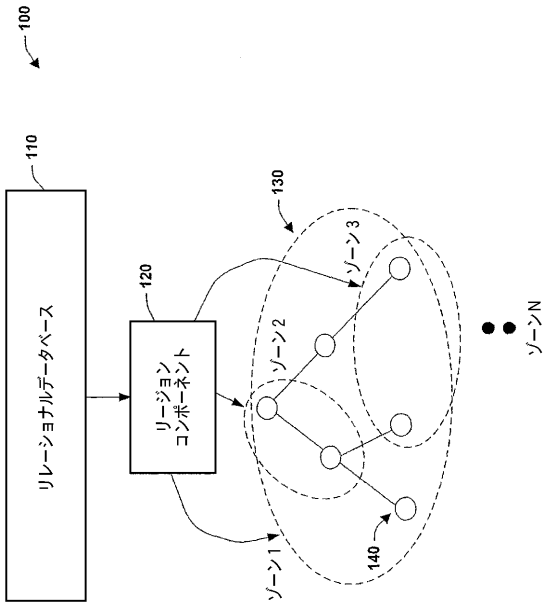
【図６】本発明の一態様による例示的なリージョン処理アルゴリズムを示す図である。

【図７】本発明の一態様によるセキュリティリージョンプロセスを例示する図である。

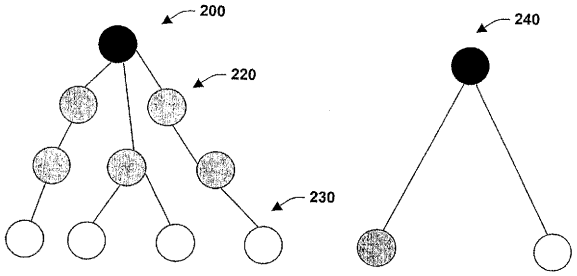
【図８】本発明の一態様による好適な動作環境を例示する概略ブロック図である。

【図９】本発明との相互やり取りが可能なコンピューティング環境例の概略ブロック図である。

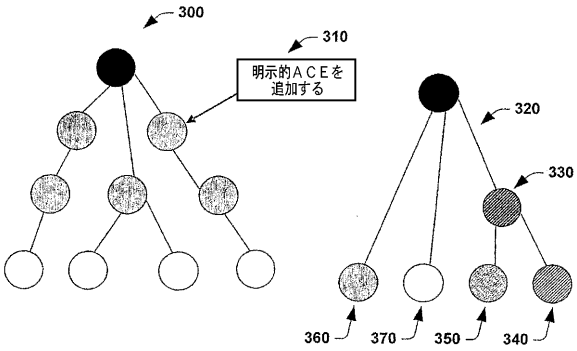
【図１】


100
110
リレーショナルデータベース
120
リージョン
コンポーネント
130
ゾーン3
ゾーン2
ゾーンN
ゾーン1
140

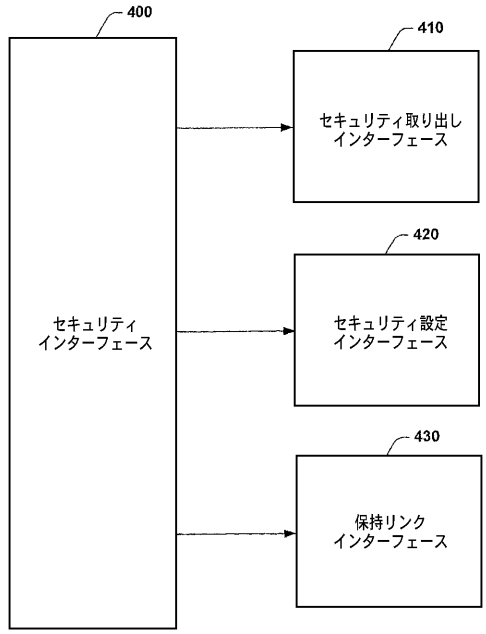
【図２】


200
220
230
240

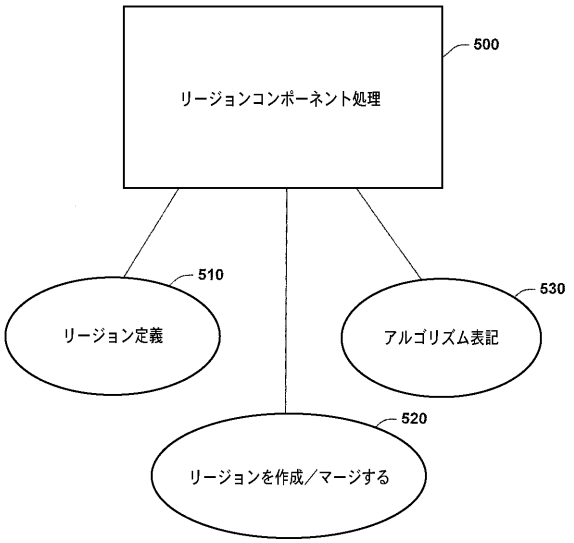
【図３】


300
310
明示的ＡＣＥを
追加する
320
330
360
370
350
340

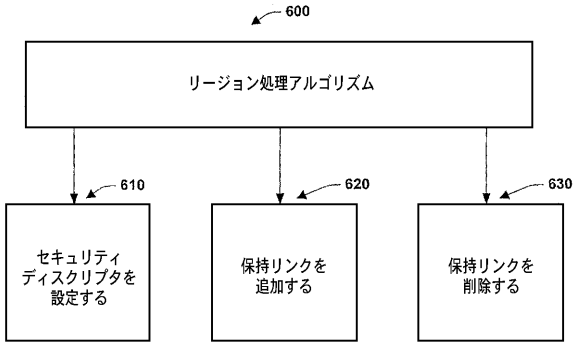
【図４】


400
セキュリティ
インターフェース
410
セキュリティ取り出し
インターフェース
420
セキュリティ設定
インターフェース
430
保持リンク
インターフェース

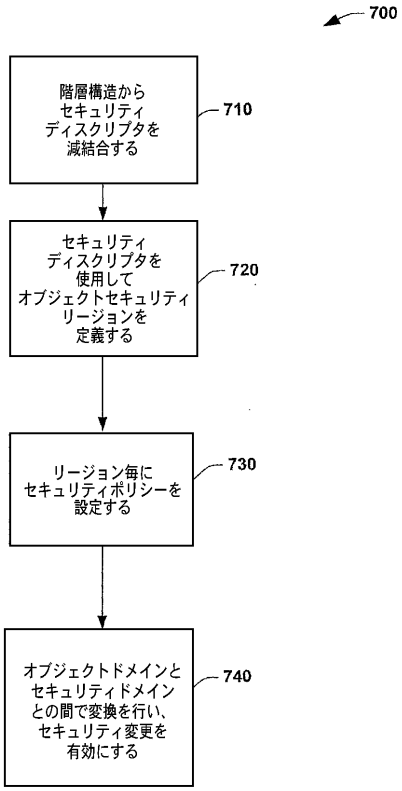
【図５】


500
リージョンコンポーネント処理
510
リージョン定義
530
アルゴリズム表記
520
リージョンを作成／マージする

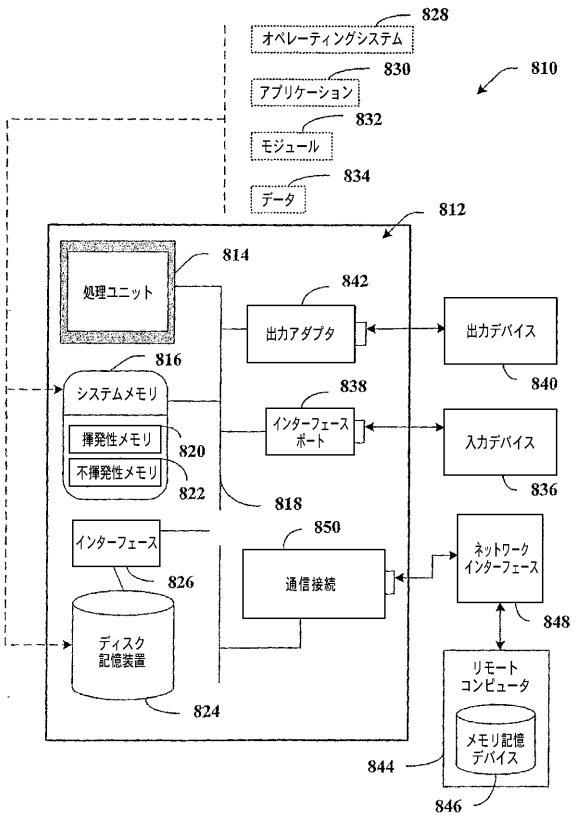
【図６】


600
リージョン処理アルゴリズム
610
620
630
セキュリティディスクリプタを設定する
保持リンクを追加する
保持リンクを削除する

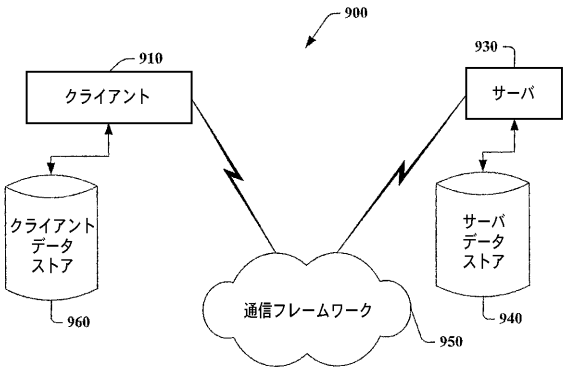
【図７】


700
階層構造からセキュリティディスクリプタを減結合する
710
セキュリティディスクリプタを使用してオブジェクトセキュリティリージョンを定義する
720
リージョン毎にセキュリティポリシーを設定する
730
オブジェクトドメインとセキュリティドメインとの間で変換を行い、セキュリティ変更を有効にする
740

【図８】


828
オペレーティングシステム
830
アプリケーション
832
モジュール
834
データ
810
812
814
処理ユニット
842
出力アダプタ
出力デバイス
840
816
システムメモリ
揮発性メモリ
820
不揮発性メモリ
822
818
838
インターフェースポート
入力デバイス
836
850
インターフェース
826
通信接続
ネットワークインターフェース
848
ディスク記憶装置
824
リモートコンピュータ
メモリ記憶デバイス
844
846

【図９】


900
910
クライアント
930
サーバ
クライアントデータストア
960
通信フレームワーク
950
サーバデータストア
940

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 06/08416

A. CLASSIFICATION OF SUBJECT MATTER
IPC (8) - G06F 17/20 (2007.01)
USPC - 707/9
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
USPC - 707/9

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC: 707/1; 726/2; 726/3

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
PubWEST (USPT, PGPB, EPAB, JPAB) - terms: database$, objects, files, security, descriptor, query, structured, language, table, tree
Google - terms: database, objects, descriptor, security

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6,202,066 B1 (BARKLEY et al.) 13 March 2001 (13.03.2001), see col 6, ln 35-47, col 8, ln 24-60, col 9, ln 48-67 and col 10, ln 5-62.	1-20
A	US 6,381,605 B1 (KOTHURI et al.) 30 April 2002 (30.04.2002), abstract and col 5-8.	1-20

☐ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search
27 May 2007 (27.05.2007)

Date of mailing of the international search report
25 SEP 2007

Name and mailing address of the ISA/US
Mail Stop PCT, Attn: ISA/US, Commissioner for Patents
P.O. Box 1450, Alexandria, Virginia 22313-1450
Facsimile No. 571-273-3201

Authorized officer:
Lee W. Young
PCT Helpdesk: 571-272-4300
PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (April 2005)

フロントページの続き

(81)指定国 AP(BW,GH,GM,KE,LS,MW,MZ,NA,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM),EP(AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HU,IE,IS,IT,LT,LU,LV,MC,NL,PL,PT,RO,SE,SI,SK,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AT,AU,AZ,BA,BB,BG,BR,BW,BY,BZ,CA,CH,CN,CO,CR,CU,CZ,DE,DK,DM,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,HR,HU,ID,IL,IN,IS,JP,KE,KG,KM,KN,KP,KR,KZ,LC,LK,LR,LS,LT,LU,LV,LY,MA,MD,MG,MK,MN,MW,MX,MZ,NA,NG,NI,NO,NZ,OM,PG,PH,PL,PT,RO,RU,SC,SD,SE,SG,SK,SL,SM,SY,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,YU,ZA,ZM,ZW

(72)発明者 タンモイ デュッタ
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

Ｆターム(参考) 5B017 AA01 BA06 CA16
5B082 EA11