(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5427599号
(P5427599)
(45)発行日　平成26年2月26日(2014.2.26)　(24)登録日　平成25年12月6日(2013.12.6)

(51)Int.Cl.　F I
G06F 21/62 (2013.01)　G06F 21/24 163A
G06F 21/24 165G

請求項の数 6　(全 25 頁)

(21)出願番号　特願2009-297522(P2009-297522)
(22)出願日　平成21年12月28日(2009.12.28)
(65)公開番号　特開2011-138298(P2011-138298A)
(43)公開日　平成23年7月14日(2011.7.14)
審査請求日　平成24年2月20日(2012.2.20)

(73)特許権者　000102728
株式会社エヌ・ティ・ティ・データ
東京都江東区豊洲三丁目３番３号
(74)代理人　110001634
特許業務法人　志賀国際特許事務所
(72)発明者　桝本　圭
東京都江東区豊洲三丁目３番３号　株式会社エヌ・ティ・ティ・データ内

審査官　岸野　徹

最終頁に続く

(54)【発明の名称】アクセス制御設定装置、方法及びコンピュータプログラム

(57)【特許請求の範囲】
【請求項１】
アクセス制限が個別に指定されていないアクセス行為を許可する方式でのアクセス制御設定装置であって、
それぞれ、アクセス制限が設定される可能性のある複数のリソースの各々を特定するためのリソース情報と、各リソースについて既に個別にアクセス制限が設定されているかどうかを表す設定状況情報と、前記リソース情報のうちの所定の表示装置に表示されたリソースを特定する情報である表示状況情報と、を保持する保持手段と、
前記リソース情報及び前記設定状況情報を所定の表示装置に表示させるとともに、当該表示されたリソースを特定する情報を前記表示状況情報として前記保持手段に記憶させ、前記保持されているアクセス制限が設定される可能性のあるすべてのリソース情報から、前記保持手段に記憶された前記表示状況情報で特定されるリソースのリソース情報を差し引くことにより、未だ表示がなされておらず且つアクセス制限が個別に設定されていないリソースを特定する制御手段と、
を有する、アクセス制御設定装置。
【請求項２】
前記複数のリソースの各々が、それぞれ、他のリソースと階層的に関連付けられ、且つ、所定のグループに分類可能な状態で存在し、分類されたときはグループ単位でアクセスの制限内容の設定がなされるものであり、
前記リソース情報は、他のリソースと階層的に関連付けられた前記複数のリソースにお

ける階層構造を表す情報と各々のリソースのＩＤとを含む第１情報、及び、各々のリソースがどのグループに分類されたかをそのグループ名と共に表す第２情報で構成されており、

前記保持されている設定状況情報は、グループ毎に非パス表現形式で表現されるものであり、

前記制御手段は、前記リソース情報及び前記設定状況情報を表示する際に、前記第１情報により特定される各階層のリソースのＩＤとそのリソースが属するグループ名とをリンクさせ、さらに、前記第２情報により特定される各グループ名とそのグループに設定された設定状況情報とをリンクさせることにより、前記非パス表現形式の設定状況情報をパス表現形式のものに変換する、

請求項１記載のアクセス制御設定装置。

【請求項３】

前記制御手段は、個々のリソース、階層化又はグループに分類されたリソースのリソース情報及び前記設定状況情報を、前記表示装置に既に表示されたリソースと未だ表示されていないリソースとで異なる態様で前記表示装置に表示させる、

請求項２記載のアクセス制御設定装置。

【請求項４】

前記制御手段は、設定されているアクセス制限の内容が同一となる複数のリソース又はグループのリソース情報については同じ態様で前記表示装置に表示させる、

請求項３記載のアクセス制御設定装置。

【請求項５】

アクセス制限が個別に指定されていないアクセス行為を許可する方式でのアクセス制御によって、それぞれ、アクセス制限が設定される可能性のある複数のリソースに対するアクセス制御内容を設定する装置が実行する方法であって、

前記複数のリソースの各々を特定するためのリソース情報と、各々のリソースについて既に個別にアクセス制限が設定されているかどうかを表す設定状況情報と、前記リソース情報のうちの所定の表示装置に表示されたリソースを特定する情報である表示状況情報と、を所定のメモリに保持するステップと、

前記リソース情報及び前記設定状況情報を所定の表示装置に表示させるとともに、当該表示されたリソースを特定する情報を前記表示状況情報として前記所定のメモリに記憶させ、前記保持されているアクセス制限が設定される可能性のあるすべてのリソース情報から、前記所定のメモリに記憶された前記表示状況情報で特定されるリソースのリソース情報を差し引くことにより、未だ表示がなされておらず且つアクセス制限が個別に設定されていないリソースを特定するための制御を行うステップとを有する、

アクセス制限の設定漏れ検出方法。

【請求項６】

アクセス制限が個別に指定されていないアクセス行為を許可する方式でのアクセス制御内容を設定するコンピュータプログラムであって、

コンピュータを、それぞれ、アクセス制限が設定される可能性のある複数のリソースの各々を特定するためのリソース情報と、各リソースについて既に個別にアクセス制限が設定されているかどうかを表す設定状況情報と、前記リソース情報のうちの所定の表示装置に表示されたリソースを特定する情報である表示状況情報と、を保持する保持手段；

前記リソース情報及び前記設定状況情報を所定の表示装置に表示させるとともに、当該表示されたリソースを特定する情報を前記表示状況情報として前記保持手段に記憶させ、前記保持されている前記アクセス制限が設定される可能性のあるすべてのリソース情報から、前記保持手段に記憶された前記表示状況情報で特定されるリソースのリソース情報を差し引くことにより、未だ表示がなされておらず且つアクセス制限が個別に設定されていないリソースを特定する制御手段；として機能させる、

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、例えばセキュリティを強化するためにアクセス制御機能を具備したオペレーティングシステム（以下、「セキュアＯＳ（Secure OS）という）において、そのセキュアＯＳに対するアクセス制御の設定を行う際に、設定漏れを容易に検出するための情報処理技術に関する。
【背景技術】
【０００２】
セキュアＯＳのアクセス制御方式として、アクセスを許可したい行為を設定ファイルに記述し、記述されていない行為についてはアクセスを拒否するホワイトリスト方式と、その逆で、アクセスを拒否したい行為を設定ファイルに記述し、記述されていない行為はアクセスを許可するブラックリスト方式とが存在する。
【０００３】
ホワイトリスト方式において、ＯＳカーネル上で動作するプログラムは、設定ファイルに記述されたアクセス許可ルール以外の振る舞いができないため、不正アクセスやプログラムの誤動作が発生した際でも、それらの動作を最小限に制限することができる。そのため、近年のセキュアＯＳには、ホワイトリスト方式が比較的多く採択されている。
【０００４】
しかしながら、ホワイトリスト方式は、プログラムの全ての振る舞いについてアクセス許可しないと、正当なプログラムであっても正常動作しない。そのため、アクセス制御の設定にあたっては、プログラムの振る舞いを考慮して、設定作業を行う必要があり、通常はプログラムの振る舞いまで関知していない設定者にとっては、設定作業に多大な時間がかかり、負担となっている。
【０００５】
そこで、設定者の負担を軽減するために、例えば、プログラムをしばらく動作させ、アクセスした履歴をログに出力し、そのログをセキュアＯＳの設定に変換することが行われている（特許文献１参照）。また、例えば、プログラムのソースコードやバイナリから、プログラムがアクセスすべき箇所を探索し、探索結果をセキュアＯＳの設定に変換することも行われている（特許文献２参照）。
【０００６】
しかし、特許文献１に記載のような技術にあっては、プログラムをどれだけ動かせば充分なログを取得できるのかを把握できず、また、プログラムを動かすこと自体に時間がかかるといった問題がある。また、特許文献２に記載のような技術にあっても、様々な言語で記述されたプログラムから、プログラムのアクセス箇所を全て抽出するのは困難である。
以上の説明から明らかなように、ホワイトリスト方式を改善するものでは、その問題の全面的な解決には至らない。
【０００７】
一方、ブラックリスト方式は、ホワイト方式とは逆に、当初はプログラムの全ての振る舞いについてアクセスが許可されており、必要最小限の振る舞いについて随時アクセス制御をする。そのため、プログラムの全振る舞いを意識することなく、アクセス制御の設定をすることができ、設定者の負担は比較的小さい。
しかしながら、ブラックリスト方式においては、ＯＳのアクセス対象となるリソースの数は数十万にもなるため、設定漏れの起こるおそれがある。
【０００８】
ブラックリスト方式の問題点を改善するためには、設定漏れを有効に検出する必要があるが、ブラックリスト方式では、アクセス許可されていても、それが、設定漏れなのか、設定者が確認のうえ許可したものなのか判断がつかない。
【先行技術文献】
【特許文献】

【０００９】
【特許文献１】特開２００７－１０９０１６
【特許文献２】特開２００５－６３２２４
【発明の概要】
【発明が解決しようとする課題】
【００１０】
本発明の課題は、セキュアＯＳのアクセス制御の設定において、設定漏れを有効に検出するアクセス制御設定装置を提供することにある。
【課題を解決するための手段】
【００１１】
本発明は、上記課題を解決するために、アクセス制御設定装置、アクセス制限の設定漏れ検出方法及びコンピュータプログラムを提供する。
本発明のアクセス制御設定装置は、アクセス制限が個別に指定されていないアクセス行為を許可する方式でのアクセス制御設定装置であって、それぞれ、アクセス制限が設定される可能性のある複数のリソースの各々を特定するためのリソース情報と、各リソースについて既に個別にアクセス制限が設定されているかどうかを表す設定状況情報と、前記リソース情報のうちの所定の表示装置に表示されたリソースを特定する情報である表示状況情報と、を保持する保持手段と、前記リソース情報及び前記設定状況情報を所定の表示装置に表示させるとともに、当該表示されたリソースを特定する情報を前記表示状況情報として前記保持手段に記憶させ、前記保持されているアクセス制限が設定される可能性のあるすべてのリソース情報から、前記保持手段に記憶された前記表示状況情報で特定されるリソースのリソース情報を差し引くことにより、未だ表示がなされておらず且つアクセス制限が個別に設定されていないリソースを特定する制御手段と、を有するものである。
このアクセス制御設定装置によれば、すべてのリソース情報から既にアクセス制限が設定されたリソース及び表示されたリソースのリソース情報が特定されるので、本来アクセス制限が設定されるべきリソースに対する設定漏れを有効に防止することができる。
【００１２】
ある実施の態様では、前記複数のリソースの各々が、それぞれ、他のリソースと階層的に関連付けられ、且つ、所定のグループに分類可能な状態で存在し、分類されたときはグループ単位でアクセスの制限内容の設定がなされるものであり、前記リソース情報は、他のリソースと階層的に関連付けられた前記複数のリソースにおける階層構造を表す情報と各々のリソースのＩＤとを含む第１情報、及び、各々のリソースがどのグループに分類されたかをそのグループ名と共に表す第２情報で構成されており、前記保持されている設定状況情報は、グループ毎に非パス表現形式で表現されるものである。そして、前記制御手段は、前記リソース情報及び前記設定状況情報を表示する際に、前記第１情報により特定される各階層のリソースのＩＤとそのリソースが属するグループ名とをリンクさせ、さらに、前記第２情報により特定される各グループ名とそのグループに設定された設定状況情報とをリンクさせることにより、前記非パス表現形式の設定状況情報をパス表現形式のものに変換する。
これにより、複数のリソースを階層構造で表現できるとともに、表示装置に表示されるパス表現形式の設定状況情報は、既にアクセス制限が設定済であるか、あるいは設定不要のものということになるので、表示をもって設定の有無が確認されることになり、設定漏れをより有効に防止することができる。
【００１３】
他の実施の態様では、前記制御手段は、個々のリソース、階層化又はグループに分類されたリソースのリソース情報及び前記設定状況情報を、前記表示装置に既に表示されたリソースと未だ表示されていないリソースとで異なる態様で前記表示装置に表示させる。また、前記制御手段は、設定されているアクセス制限の内容が同一となる複数のリソース又はグループのリソース情報については同じ態様で前記表示装置に表示させる。
これにより、アクセス制限の設定の有無及び設定内容を視覚的に把握できるようになり

、アクセス制御設定装置の操作性を高めることができる。

【００１４】

本発明のアクセス制限の設定漏れ検出方法は、アクセス制限が個別に指定されていないアクセス行為を許可する方式でのアクセス制御によって、それぞれ、アクセス制限が設定される可能性のある複数のリソースに対するアクセス制御内容を設定する装置が実行する方法であって、前記複数のリソースの各々を特定するためのリソース情報と、各々のリソースについて既に個別にアクセス制限が設定されているかどうかを表す設定状況情報と、前記リソース情報のうちの所定の表示装置に表示されたリソースを特定する情報である表示状況情報と、を所定のメモリに保持するステップと、前記リソース情報及び前記設定状況情報を所定の表示装置に表示させるとともに、当該表示されたリソースを特定する情報を前記表示状況情報として前記所定のメモリに記憶させ、前記保持されているアクセス制限が設定される可能性のあるすべてのリソース情報から、前記所定のメモリに記憶された前記表示状況情報で特定されるリソースのリソース情報を差し引くことにより、未だ表示がなされておらず且つアクセス制限が個別に設定されていないリソースを特定するための制御を行うステップとを有することを特徴とする。

【００１５】

本発明のコンピュータプログラムは、アクセス制限が個別に指定されていないアクセス行為を許可する方式でのアクセス制御内容を設定するコンピュータプログラムであって、コンピュータを、それぞれ、アクセス制限が設定される可能性のある複数のリソースの各々を特定するためのリソース情報と、各リソースについて既に個別にアクセス制限が設定されているかどうかを表す設定状況情報と、前記リソース情報のうちの所定の表示装置に表示されたリソースを特定する情報である表示状況情報と、を保持する保持手段；前記リソース情報及び前記設定状況情報を所定の表示装置に表示させるとともに、当該表示されたリソースを特定する情報を前記表示状況情報として前記保持手段に記憶させ、前記保持されているアクセス制限が設定される可能性のあるすべてのリソース情報から、前記保持手段に記憶された前記表示状況情報で特定されるリソースのリソース情報を差し引くことにより、未だ表示がなされておらず且つアクセス制限が個別に設定されていないリソースを特定する制御手段；として機能させるコンピュータプログラムである。

【発明の効果】

【００１６】

本発明によれば、表示装置において未だ表示されず、且つ、アクセス制限が設定されていないリソースが特定されるので、本来アクセス制限が設定されるべきリソースに対する設定漏れを有効に防止することができる。

【図面の簡単な説明】

【００１７】

【図１】本実施形態のアクセス制御設定装置と端末装置とを含む情報処理システムの全体構成図。

【図２】アクセス制御設定装置及び端末装置の機能ブロック図。

【図３】マッピングファイルの例を示す説明図。

【図４】（ａ），（ｂ）はアクセス制御ルールの例を示す説明図。

【図５】アクセス制御設定装置の動作時の全体処理図。

【図６】対象範囲を定める処理（Ｓ１）の詳細手順説明図。

【図７】表現形式の変換処理（Ｓ３）の詳細手順説明図。

【図８】表現形式の変換処理（Ｓ３）の詳細手順説明図。

【図９】表現形式の変換処理（Ｓ３）の詳細手順説明図。

【図１０】設定情報の生成処理（Ｓ４）の詳細手順説明図。

【図１１】設定漏れ検出処理（Ｓ５）の詳細手順説明図。

【図１２】初期情報入力画面の例を示す説明図。

【図１３】マッピングオブジェクトのパターン説明図。

【図１４】オブジェクト生成の概念図。

【図１５】（ａ），（ｂ）はアクセス制御詳細設定画面の例を示す説明図。
【図１６】ラベルオブジェクトとパスオブジェクトの関係を示す概念図。
【図１７】ラベルオブジェクトとパスオブジェクトの関係を示す概念図。
【図１８】ラベルオブジェクトとパスオブジェクトの関係を示す概念図。
【図１９】ラベルオブジェクトとパスオブジェクトの関係を示す概念図。
【図２０】リソースの階層構造を示す説明図。
【図２１】リソースの階層構造を示す説明図。
【図２２】相互確認に関する設定画面の例を示す説明図。
【発明を実施するための形態】
【００１８】
以下、本発明の実施の形態例を説明する。本実施形態は、セキュアＯＳを搭載した端末装置に対するブラックリスト方式によるアクセス制御内容の設定の有無の確認、制御内容の変更、更新等をリモートで行うアクセス制御設定装置の例を挙げる。
【００１９】
［全体構成］
図１は、本実施形態のアクセス制御設定装置１００と、このアクセス制御設定装置１００によりリモート操作される端末装置２００とを含む情報処理システムの全体構成図であり、特徴的な部分のみを示してある。端末装置２００は、アクセスポリシ２４０に従ってアクセス制御内容が設定されるセキュアＯＳ２１０を搭載している。このセキュアＯＳ２１０は、非パス表現形式の一例となるラベルベースの制御方式によって、リソースに対するアクセス制御が設定されるものである。
【００２０】
アクセス制御設定装置１００と端末装置２００は、例えば、ＳＳＨ（Secure SHell）などによって双方向の通信可能に接続されている。本例では、端末装置２００によるアクセスが制限されるリソースは、ＨＤＤ（Hard disk drive）などの外部デバイスにおいて、それぞれ、他のリソースと階層的に関連付けられ、且つ、所定のグループに分類可能な状態で存在するものとして説明する。これらのリソースは、例えばプログラム、ファイル、ディレクトリ等であり、グループに分類されたときはそのグループ単位でアクセス制御の設定がなされるものである。
【００２１】
[アクセス制御設定装置]
まず、アクセス制御設定装置１００の構成例を説明する。図２は、本実施形態におけるアクセス制御設定装置の機能ブロック図である。
アクセス制御設定装置１００は、ＣＰＵ（Central Processing Unit）、ＲＡＭ（Random Access Memory）、ＲＯＭ（Read Only Memory）、ネットワークカード、入出力インタフェース等を備えた処理装置１１０と、ディスプレイ等の表示装置１２０と、キーボード等の入力装置１３０と、端末装置２００又は外部デバイスと接続するための接続インタフェース１４０とを備えている。
処理装置１１０は、コンピュータの一種であり、ＣＰＵが本発明のコンピュータプログラムを読み込んで実行することにより、端末装置２００のアクセス制御設定に関する機能、具体的には、入力受付部１１１、情報取得部１１２、設定情報更新部１１３、表現形式変換部１１４、表示制御部１１５、及び、出力制御部１１６として動作する。
【００２２】
入力受付部１１１は、入力装置１３０から、アクセス制御の設定に関わる情報、例えば現在の設定内容の確認指示情報、設定内容の変更情報または新たな指定情報、設定漏れの検出指示等の入力を受け付ける。
【００２３】
情報取得部１１２は、端末装置２００から、セキュアＯＳ２１０がアクセス可能なすべてのリソースを特定するためのリソース情報、例えば、各リソースが何処にどのような状態で存在するかを表す情報と、各リソースについて既にアクセス制限が設定されているか

どうかを表す設定状況情報とを取得する。リソース情報は、具体的には、階層構造を表す情報と各々のリソースのＩＤとを含む第１情報、及び、各々のリソースがどのグループに分類されているかをそのグループ名と共に表す第２情報で構成されている。設定状況情報は、設定済か未設定か、設定されている場合はどのような設定内容かを各々のグループについてラベル表現形式で表現されるものである。

【００２４】

設定情報更新部１１３は、図示しないメモリ制御機構との協働により、情報取得部１１２で取得したリソース情報及び設定状況情報をＲＡＭの所定領域に保持する。また、入力装置１３０から変更情報又は設定情報を受け付けたときは、受け付けた内容に従って、保持されている設定状況情報を更新する。更新する際には、更新前の設定状況情報をＲＡＭに保持し、更新後の設定状況情報に代えて更新前の設定状況情報に復帰可能にする。

【００２５】

表現形式変換部１１４は、保持されているリソース情報及び設定状況情報のうち、リソース情報に含まれる第１情報により特定される各階層のリソースのＩＤと、第２情報により特定されるそのリソースが属するグループ名とをリンクさせ、さらに、各グループ名と、設定状況情報により特定されるそのグループに設定されたアクセス制御の内容とをリンクさせることにより、当該アクセス制御の内容をラベル表現形式からパス表現形式のものに変換する。

【００２６】

表示制御部１１５は、表示装置１２０への情報の表示制御を行う。表示される情報は、ラベル表現形式からパス表現形式に変換されたリソースの状態を表す画面、各リソースに対して設定されたアクセス制御の内容を表す画面、初期設定を含む各種設定画面等である。リソース及び各リソースの設定状況情報を表す画面において表示される情報は、表現形式変換部１１４でパス表現形式に変換されているので、表示されることをもってアクセス制御の内容が確認されたことと同様となる。

【００２７】

表示制御部１１５は、設定者の指示に従い、設定漏れの検出を容易にするための処理の制御を行う。この制御は、具体的には、ＲＡＭに保持されているすべてのリソース情報を読み出し、読み出したリソース情報から、既にアクセス制限が設定されているリソース及び表示装置１２０に表示されたリソースのリソース情報を差し引くことにより、未だ表示がなされておらず且つアクセス制限が設定されていないリソースを特定するための制御である。既にアクセス制限が設定されているリソースかどうかは、設定状況情報より特定することができる。表示されたリソースかどうかは、例えば、表示済みのリソースについては色を変える等の処理を行うことで特定が可能である。

【００２８】

出力制御部１１６は、ＲＡＭに保持されている設定状況情報が更新されたときは、更新された設定状況情報を端末装置２００に出力するための制御を行う。

【００２９】

［端末装置］

次に、端末装置２００の構成を詳しく説明する。端末装置２００のセキュアＯＳ２１０において、ＯＳカーネル２１１の内部にはアクセス制御モジュール２１２が存在する。アクセス制御モジュール２１２は、ＯＳ起動時に、アクセスポリシ２４０からマッピングファイル２２０とアクセス制御ルール２３０を読み込み、アクセス制御を開始するものである。マッピングファイル２２０は、ファイル、ディレクトリ等のリソースが、それぞれ、どのラベルに割り当てられるべきかを記述したファイルであり、アクセス制御ルール２３０は、どのプロセスが、どのリソースに対して、どのようなアクセスが可能なのかを示すファイルである。

【００３０】

マッピングファイル２２０の内容例を図３に示す。リソースへのアクセス行為には「アクセス主体」と「アクセス客体」が関与するが、アクセス主体となりうるプログラムも、

アクセス客体となるファイルやディレクトリも一つのマッピングファイルに記述される。なお、アクセス主体となりうるのはプログラム(プロセス)のみであり、アクセス客体にはプログラムがアクセスするファイルやディレクトリのほか、プログラムによって起動されるプログラムも含まれる。

【００３１】

この例のマッピングファイルの１行目、２行目、４行目はアクセス客体に関するラベル定義の例であり、１行目の「/var(/.*)? system_u:object_r:var_t:s0」は、「ラベルvar_tが、/var以下全てのリソースに割り当てられる」ことを意味する。２行目の「/var/run/*.pid system_u:object_r:var_run_t:s0」は、「ラベルvar_run_tが、/var/run/*.pidの正規表現にマッチするリソースに割り当てられる」ことを意味する。４行目の「/usr(/.*)? system_u:object_r:var_t:s0」は、「ラベルvar_tが、/usr以下全てのリソースに割り当てられる」ことを意味する。１行目と４行目の定義の結果、/varと/usr以下の全てのリソースが１個のラベルvar_tで指定されたことになる。

この例の３行目はアクセス主体になりうるプログラムに関するラベル定義の例であり、「/usr/sbin/httpd system_u:object_r:httpd_exec_t:s0」は、「ラベルhttpd_exec_tが、/usr/sbin/httpdに割り当てられる」ことを意味する。

【００３２】

アクセス制御ルール２３０の内容例を図４（ａ），（ｂ）に示す。

アクセス制御ルール２３０には、アクセス主体がアクセス客体にどのようなアクセスをしてよいかを記述する。また、アクセス主体が新たにプログラムを起動した場合や新たにファイルを作成した場合、起動されたプログラムや作成されたファイルに付与されるラベルを定義する。また、ユーザ用のプロセスはログインデーモン（プログラム）が生成するので、やはりアクセス制御ルールにおいてラベルを定義する。

なお、ブラックリスト方式においてもホワイトリスト方式と同様のアクセスポリシファイルを利用するが、ブラックリスト方式においては、全てのリソースについてあらゆるアクセス許可が基本となるので、このための設定を行う。

【００３３】

図４（ａ）のアクセス制御ルールの５行目の「allow httpd_t var_t:file read;」は、「アクセス主体httpd_tは、アクセス客体var_tのfileにreadしかできない」ことを意味する。なお、var_tにはファイルの他、ディレクトリやプログラムが属する場合もあり得るが、このうちfileのreadだけに制限したことを意味する。

６行目の「type_transition usr_t httpd_exec_t:process httpd_t;」は、「アクセス主体usr_tがアクセス客体httpd_exexc_tを実行した結果起動したプロセスは、ラベルhttpd_tに属する」ことを意味する。７行目の「type_transition httpd_t var_run_t:file httpd_var_run_t:」は、「アクセス主体httpd_tがアクセス客体 var_run_t に作ったファイルは、ラベルhttpd_var_run_tに属する」ことを意味する。

１行目から３行目では「属性」と、その「属性」を有するラベルを定義している。「属性」はラベルをまとめるグループのようなもので、ファイルやディレクトリがまとめられたラベルを、さらにまとめるために使用する。１行目～３行目の例では、ラベル「var_t」とラベル「usr_t」が同じ属性「sysadm_type」を有することを意味している。

このように定義されたグループ情報を利用して、例えば４行目の「allow masumoto_t sysadm_type :file read;」のように、ブラックリスト方式の前提となる広範囲のアクセス許可を効率的に行う。

図４（ｂ）はアクセス主体がユーザプロセスの場合のラベル定義の例であって、ここではユーザ root がログインした直後のプロセスにラベル「my_root_t」が付与された例を示す。

【００３４】

従来のアクセス制御の設定は、マッピングファイル２２０とアクセス制御ルール２３０の双方を参照することによって可能となっていた。すなわち、図３のマッピングファイル２２０の１行目の「/var(/.*)? system_u:object_r:var_t:s0」（var_tとは、/var以下

全てのことである）と、図４（ａ）のアクセス制御ルール２３０の５行目の「allow httpd_t var_t:file read;」（httpd_tは、var_tのfileにreadしかできない）とによって、「httpd_tは、/var以下全てのfileにreadしかできない」との設定がなされていた。

【００３５】

［アクセス制御設定装置の動作例］

次に、アクセス制御設定装置１００の動作例を説明する。

図５は、全体的な動作手順説明図である。図５を参照すると、アクセス制御設定装置１００が端末装置２００に接続され、処理装置１１０が起動すると、処理装置１１０は、動作環境を整え、端末装置２００から現在のリソースの情報及びアクセスポリシ２４０の内容を表す情報を取得し、ＲＡＭに保持するとともに、表示装置１２０に初期情報入力画面を表示させ、アクセス制御の対象プログラムを定める処理を行う（ステップＳ１）。

初期情報入力画面は、リソースへのアクセスを制限する主体を定めるための情報の入力を促す画面である。本例では、アクセス主体はどのプログラムか、及びどのユーザ又はどのプログラムがそれを起動したのか、という内容を指定できるようにしている。このような指定を設定者が入力装置１３０を通じて入力すると、処理装置１１０は、指定された内容を受け付け、以後の処理のためにＲＡＭに保持する（ステップＳ２）。

【００３６】

処理装置１１０は、ＲＡＭに保持された情報をもとに、アクセスポリシの表現形式の変換処理を行う（ステップＳ３）。本例では、ラベル表現形式のアクセスポリシをパス表現形式のものに変換する処理を行う。そして、変換された表現形式のアクセスポリシの内容を表示装置１２０に表示させる。

【００３７】

表示画面をみた設定者が、制御内容の新たな指定又は変更の情報を入力すると、処理装置１１０は、これらの情報を受け付け、設定情報の生成処理を行う（ステップＳ４）。具体的には、受け付けた指定又は変更の情報に従って、それ以前に端末装置２００より取得してＲＡＭに保持した情報を更新し、更新された情報を端末装置２００に出力するための制御を行う。情報を更新するときは更新内容に従って表示装置１２０における表示態様を変化させる。なお、情報を更新する際には更新前の情報を保持しておき、更新後の情報に代えて何時でも更新前の情報に復帰可能にする。

【００３８】

設定情報の生成処理（ステップＳ４）をリソース毎に繰り返し実行した後、処理装置１１０は、設定漏れ検出処理を行う（ステップＳ５）。この処理は、未だ表示がなされておらず且つアクセス制限が設定されていないリソースを特定することにより行う。

【００３９】

その後、処理装置１１０は、ステップＳ１で取得したアクセス主体情報と、ステップＳ４で取得したアクセス制御設定情報（アクセス客体情報と、パーミッション情報）を、端末装置２００のアクセスポリシ２４０に書き込む処理を行い、処理を終える（ステップＳ６）。

なお、アクセスポリシのパス表現形式への変換は、設定者が指定した範囲で、リソース毎、階層毎又はグループ毎に情報を読み出して行う。例えばあるディレクトリを指定することにより、そのディレクトリに属するファイルのみの情報を読み出す。これにより、全階層のすべての情報を読み込む必要がなくなり、処理量が節約されるので、設定に要する時間が短縮される。

【００４０】

次に、図５の全体処理における各処理の手順を詳細に説明する。

［ステップＳ１の詳細手順］

対象プログラムを定める処理（ステップＳ１）の詳細手順を図６に示す。

まず、図１１に例示される初期情報入力画面３００を表示装置１２０に表示させる（ステップ２１０）。図１１において、「ＡＰ」はアクセス制御の対象となるプロセスを表す。「ＡＰのパス」は、そのＡＰがどこに存在するどのプログラムであるかを指定するパス表

現領域３１０である。「ＡＰを起動するユーザ（３２０）」か、「ＡＰを起動するプログラムのパス（３３０）」は、どちらか一方が必須入力となる。これらは、パス表現領域３１０の入力内容とあわせて、「以降設定するアクセス制御内容は、領域３２０で指定されたユーザか、領域３３０で指定されたプログラムが、領域３１０で指定したプログラムを実行したときに生成されるプロセスに対して有効である」という意味になる。

【００４１】

ユーザ指定領域３２０への指定を検出した場合、処理装置１１０は、入力されたユーザのＩＤをキーとして、端末装置２００が保持するアクセス制御ルール２３０を参照し、そのユーザに付与されるラベルを取得する（ステップＳ２３０）。例えば、ユーザ名「root」が入力されたとする。処理装置１１０は、例えば、図４（ｂ）のアクセス制御ルール２３０の「root:root:s0-s0:c0.c1023」の記述を参照し、「root」の設定を検出する。検出された設定が「user root roles {my_root_r system_r} level s0 range s0-s0:c0.1023」であった場合、「my_root_r」をキーとして、例えば「my_root_r:my_root_t」の結果を検出し、ラベル「my_root_t」を得る。

【００４２】

一方、プログラム指定領域３３０への指定を検出した場合、処理装置１１０は、アクセス制御ルール２３０及びマッピングファイル２２０から、そのプログラムが実行されたときに付与されるラベルを取得する（ステップＳ２４０）。

例えば、ＡＰを起動するプログラムのパスとして「/etc/init.d/httpd」が指定されたとする。処理装置１１０は、図３のマッピングファイル２２０における「/etc/init.d/http--system_u:object_r:initrc_exec_t:s0」を参照する。「/etc/init.d/httpd」にはラベル「initrc_exec_t」が付与されていることがわかるので、「initrc_exec_t」をキーとしてアクセス制御ルール２３０を検索し、例えば「type_transition XXX_t initrc_exec_t:process YYY_t;」のように「initrc_exec_t」を「:」の直前に含むルールを検出し、ＡＰを起動するプログラム「/etc/init.d/httpd」のラベル「YYY_t」を得る。

このように、各領域３１０～３３０への指定により、「ユーザ○○／プログラム○○が実行したＡＰについてのアクセス制御の設定を行う」という内容の初期情報が設定者より入力され、ステップＳ２により保持されることになる。

【００４３】

［ステップＳ３の詳細手順］

表現形式の変換処理（ステップＳ３）の詳細手順を図７乃至図９に示す。

以下では、主にルートディレクトリ"/"に属するリソースの場合を例にとって説明する。

【００４４】

処理装置１１０は、まず、マッピングファイルを取得して、マッピングオブジェクトを生成する（ステップＳ３１０）。個々のマッピングオブジェクトは、例えば図３におけるマッピングファイルの一行一行に対応し、ラベルオブジェクトと同様にラベル情報と属性情報を有する他、パス情報と正規表現情報を有する。

マッピングファイルは、通常、高々数千行なので、マッピングオブジェクトはマッピングファイル全体を一度に読み込んで生成してもよい。マッピングオブジェクトの属性情報はマッピングファイルの定義を書き写したものなので、マッピングオブジェクトではアクセス制御設定対象となるリソース個々の属性を特定できない。そこで、具体的なアクセス制御設定作業の対象となるリソースと直接関連付けて生成されるラベルオブジェクトを利用して、設定対象リソースの属性を特定する。

【００４５】

一方、名前と属性情報のみからなるラベルオブジェクトだけでは、そのラベルはパスに直すとどこのことかを特定できないため、パス情報を参照するためのマッピングオブジェクトが必要となる。

マッピングオブジェクトは、例えば、図３のマッピングファイル２２０の「/var(/.*)? system_u:object_r:var_t:s0」の記述に基づくと、マッピングオブジェクトには、ラベ

ル名は「var_t」、パスは「/var」、正規表現は「(/.*)?」、属性は「正規表現が表すリソースの属性」が記述されることとなる。また、設定者が新しいラベルを作成した場合には、マッピングオブジェクトが新たに作成される。マッピングオブジェクトは、処理の簡易化のため、処理装置１１０で定義したマッピング設定とデフォルトのマッピング設定と、ラベル名の付与ルールを区別する等して、分けて保持するものとしてもよい。

【００４６】

マッピングオブジェクトは、様々なパターンを有する。このことを、図１２を参照して説明する。図１２において、パターン１におけるマッピングオブジェクト７１０は、デフォルトのマッピング設定のオブジェクトである。正規表現がパス表現から分離できないため、マッピングオブジェクト７１０の正規表現の欄は空欄となっており、属性は「dir（ディレクトリ）」である。パターン２のマッピングオブジェクト７２０は、処理装置１１０でディレクトリに対して新しいラベルを割り当てた場合の例である。パスには新しいラベルを定義したリソースのパスが記述される。正規表現は「(/.*)?」となり、属性は「dir（ディレクトリ）」である。パターン３のマッピングオブジェクト７３０は、処理装置１１０でファイルに対して新しいラベルを割り当てた場合の例である。パスには新しいラベルを定義したリソースのパスが記述され、属性は「file（ファイル）」である。パターン４のマッピングオブジェクト７４０は、処理装置１１０で、あるディレクトリ中に動的に作成されるファイルに、親ディレクトリとは異なるラベルを割り当てた場合の例である。パスには親ディレクトリのフルパスが記述され、正規表現にはファイル名のパターンが記述されている。パターン５は、パターン４とほぼ同様の例で、正規表現を定義せず、「*（何でもよい）」とする例である。

【００４７】

次に、処理装置１１０は端末装置２００から、リソースの階層構造（ディレクトリツリー：パス表現）に関する情報と、リソースのラベル情報に関する情報を取得する。

具体的には、端末装置２００に対して、リモート操作によりコマンド“ls /”を実行することにより、例えば、“/”以下のファイル、ディレクトリ等のリソースの一覧を取得する（ステップＳ３１５）。

また、端末装置２００に対し、コマンド“ls－Z /”を実行することにより、例えば“/”以下のファイル、ディレクトリ等のリソースに対するラベルの一覧を取得する（ステップＳ３２０）。その際、上述したように、処理量を減らすために、実際に表示装置１２０での表示に必要な範囲、例えば、“ /”の１段だけ下位の階層のリソースについてのみ、リソース一覧とラベル一覧を取得し、その他のリソースについては、その下位のリソースが展開される都度、リソース一覧とラベル一覧を取得するものとする。このようにすると、実際に展開され、表示装置１２０に表示され、設定者に視認される等の必要な範囲でのみ処理を行うため、処理量を分散することができ、一回あたり処理の負担を減らすことができる。

【００４８】

次に、処理装置１１０は、ステップＳ３１５とステップＳ３２０で取得した情報に基づき、パス表現用のオブジェクト（以下、「パスオブジェクト」）とラベル表現用のオブジェクト（以下、「ラベルオブジェクト」）を生成する。また、６種類のパーミッションオブジェクトを生成する（ステップＳ３２５）。

【００４９】

ステップＳ３２５におけるオブジェクト生成の概念を図１４に示す。図１４の左側は、パスオブジェクト４１０～４１４であり、それぞれ、ファイル、ディレクトリ等のリソースの「名前」、ＯＳのディレクトリツリーの親子関係を示す「親参照」情報、ラベルオブジェクトとの対応付けを示す「旧参照」、「新参照」情報が記述される。

「旧参照」には既存のラベルとの対応関係を示すための情報が記述される。「新参照」は、「旧参照」と同様にラベルとの対応関係を示すためのものであるが、設定者が新しいラベルを割り当てた場合に、その新しいラベルを参照先として記述する。このように、「旧参照」とは別に「新参照」を記述するため、一旦新しいラベルを割り当てた後であって

も、「新参照」を削除し、「旧参照」の記述に基づいて、元の状態に容易に戻すことができる。

【００５０】

パスオブジェクト４１０～４１４は、それぞれ、ラベルオブジェクト５１０～５１２と関連付けられる。ラベルオブジェクト５１０～５１２は、ラベル表現用のオブジェクトであり、ラベルの「名前」と「属性」とを示している。ラベルオブジェクト５１０～５１２は、それぞれ、パーミッションオブジェクト６１０～６１２に関連付けられる。パーミッションオブジェクト６１０～６１２は、アクセス制御の一例となる制限内容を示すオブジェクトである。例えば「ＲＷＸＣ」は「読み取り（Read）、書き込み（Write）、実行（eXecute）、ファイル作成（fileCreate）について制限なし」を意味し、ブラックリスト方式におけるデフォルトの設定である。これに対して「ＲＷ」は「実行（eXecute）、ファイル作成（fileCreate）の制限」を、「Ｒ」は「書き込み（Write）、実行（eXecute）、ファイル作成（fileCreate）の制限」を意味する。

制限内容を示すオブジェクトを「パーミッションオブジェクト」と呼ぶことは、一見、適切でないが、見方を変えれば「Ｒ」は「読み取りだけ許可」、「ＲＷ」は「読み取りと書き込みだけ許可」を意味するので、このように呼ぶこととする。

パーミッションオブジェクトには、この他に「すべて制限」、「ＲＷＣ」、「ＲＷＸ」のパターンがある。パスオブジェクトがラベルオブジェクトを介してパーミッションオブジェクトと関連づけられることにより、後述するように、図２０や図２１のようなツリー構造表示において、パスリソースはパーミッションタイプごとに（ＲＷＸＣ情報と共に）色分けして表示される。

【００５１】

図７に戻り、処理装置１１０は、各パスオブジェクトの「旧参照」が、そのパスリソースに割り当てられたラベルを参照するように、パスオブジェクトとラベルオブジェクトを関連付ける（ステップＳ３３０）。

次に、図８のステップＳ３３５に進み、処理装置１１０は、ステップＳ３２０で取得したラベル一覧の中に、アクセス制御設定装置１００により設定されたラベル（以下、「ラベルＡ」という。また、ラベルＡの定義したラベルオブジェクトを「ラベルオブジェクトＡ」という。）があるか否か判定する（ステップＳ３３５）。ラベルＡがない場合（ステップＳ３３５：Ｎｏ）、図９のステップＳ３４０に進み、処理装置１１０はさらにマッピングオブジェクトを参照して、図１２のパターン４または５に該当する、プログラムによって動的に生成・削除されるファイルとラベルの定義したマッピングオブジェクト（以下、「マッピングオブジェクトＢ」という。また、マッピングオブジェクトＢを定義したラベルを「ラベルＢ」という。）がないか検索する。マッピングオブジェクトＢがなければ（Ｓ３４０：Ｎｏ）、表現形式変換処理（ステップＳ３）を終了する。

【００５２】

端末装置２００について初めてアクセス制御設定装置１００を使用した場合、当然、ラベルＡもマッピングオブジェクトＢも発見されることはない。この場合、表現形式変換処理（ステップＳ３）終了後、設定情報生成処理（ステップＳ４）のためにユーザに提示されるリソースのツリー構造表示は、全てのリソースについてＲ、Ｗ、Ｘ、Ｃが全て許可された初期状態としてユーザに提示される。

【００５３】

ここで、いったんステップＳ３を離れ、先に設定情報の生成処理（ステップＳ４）について説明する。

［ステップＳ４の詳細手順］

処理装置１１０は、ステップＳ３の処理により、表現形式の変換処理を終えると、設定情報を生成するための処理を行う。設定情報の生成処理（ステップＳ４）の詳細手順を図１０に示す。

ステップＳ３の処理を終えると、処理装置１１０はリソースをツリー構造で表現した、「アクセス制御設定画面」をユーザに提示する。各リソースについて種々のパーミッショ

ン情報が得られた場合、アクセス制御設定画面に表示されるリソースは、例えば図２０や図２１のように、パーミッション情報ＲＷＸＣとともに色分けされて表示される。
アクセス制御設定画面を提示された設定者は、アクセス制御の内容を設定（あるいは変更）したいファイルやディレクトリを指定してクリックする（ステップＳ４１０）。
クリック内容を解読した処理装置１１０は、図１５（ａ）に示すアクセス制御詳細設定画面９００を表示装置１２０に表示させる（ステップＳ４２０）。
設定者が、設定画面９００に従い、「このパスのアクセス許可」の設定領域９０１に提示されたパーミッションの中から一つを選び、ＯＫボタン９０５を押し、パーミッションの設定を行う（ステップＳ４３０）。選んだ内容をキャンセルする場合は、キャンセルボタン９０６を押す。なお、ブラックリスト方式における「アクセス許可」とは許可行為以外の「制限」を意味し、例えば設定領域９０１のラジオボタン「ｒｗｃ」の選択は「実行（eXecute）」の制限を意味する。
「同じアクセス許可のパス」の表示領域９０２には、同じアクセス許可のリソースのパスが表示される。表示領域９０２にリストされたリソースとは異なるアクセス制御の内容を設定するために、新しいラベルを割り当てる場合には、チェックボックス９０３を選択する。指定したパスの下位のリソースに同様のアクセス許可を設定する場合には、チェックボックス９０４を選択する。
【００５４】
図１０に戻り、設定者が設定操作を行うと、処理装置１１０は、設定者によって選択されたリソース名を読み取り、該当するパスオブジェクトを参照する（ステップＳ４４０）。さらに、そのパスオブジェクトからラベルオブジェクトを参照する（ステップＳ４５０）。
以下、図１６乃至図１９を参照し、パスオブジェクトとラベルオブジェクトのタイプごとに場合分けして、アクセス制御設定作業と各オブジェクトの関係を説明する。
パスオブジェクトの新参照がいずれのラベルオブジェクトも参照していない場合、処理装置１１０はパスオブジェクトの旧参照にしたがってデフォルトラベルのラベルオブジェクトを参照し、ステップＳ４３０で選ばれたパーミッションに基づき、パーミッションオブジェクトのリスト構造に該当するラベルオブジェクトを追加する（図１６（ａ）参照）。この結果、ラベルオブジェクトとパーミッションオブジェクトが関連付けられる（ステップＳ４６０）。
【００５５】
設定者が新しいラベルの割り当てを選択していた場合、処理装置１１０は新しいラベルオブジェクトを生成し、パスオブジェクトの新参照が当該新ラベルオブジェクトを参照するようにする。そして、ステップＳ４３０で選ばれたパーミッションに基づき、パーミッションオブジェクトのリスト構造に新ラベルオブジェクトを追加することにより、ラベルオブジェクトとパーミッションオブジェクトを関連付ける（ステップＳ４６０）（図１６（ｂ）参照）。なお、新しいラベルは一定の命名規則に則って付与し、デフォルトマッピングファイルに記載されたラベルと区別できるようにする。
パスオブジェクトの新参照が参照する新ラベルのラベルオブジェクトがすでに存在する場合、処理装置１１０は当該ラベルオブジェクトとパーミッションオブジェクトの関連付けを、ステップ４３０で選ばれたパーミッションの内容に更新する（図１６（ｃ）参照）。
【００５６】
なお、「ディレクトリ○○に新しいラベル○○を割り当てる」とは、通常、「ディレクトリ○○以下の全てのリソースに対して新ラベル○○を割り当てる」という意味になる。しかし、ディレクトリ○○配下のリソースを示すパスオブジェクトの全てについて、いちいち新参照が新ラベル○○のラベルオブジェクトを参照するように設定する必要はない。子リソースのラベルは、親リソースに従うことを原則とすることで、親をたどっていけば任意の子リソースのラベルを把握することができるからである。
【００５７】

設定者による設定作業の結果、図１７に示されるように、パスオブジェクトには３つのタイプが生まれることになる。

タイプ１は、新ラベルが付与されず、旧参照だけがデフォルトラベルを参照するパスオブジェクトである（図１７のタイプ１参照）。

タイプ２は、新ラベルが付与され、旧参照はデフォルトラベルを、新参照は新ラベルを参照するパスオブジェクトである（図１７のタイプ２参照）。

タイプ３は、新ラベルが付与されたリソースの子リソースを表すパスオブジェクトであって、新参照が新ラベルオブジェクトを参照することはしないが、親リソースを介して実質的に新ラベルを参照するパスオブジェクトである（図１７のタイプ３参照）。

【００５８】

パスオブジェクト～ラベルオブジェクト～パーミッションオブジェクトについて必要な関連付けを終えると、アクセス制御設定装置１００は、画面上に、ディレクトリツリー構造を再描画する（ステップＳ４７０）。

図２０は、ステップＳ３の処理の結果、ルートディレクトリ"/"とその直下のディレクトリからなるツリー構造が、パーミッション制限なしの状態で設定者に提示され、設定者がディレクトリ８０２、８０４、８０６にパーミッション制限設定を施したものである。

図２０において、ディレクトリはリソースの集合を表している。図示の例では、ディレクトリ８００を最上位とする階層構造となっており、ディレクトリ８０４には、さらに下位層のディレクトリが存在することを示す「＋」表示のボタン（画像）８５１が示されている。ディレクトリ８１３には下位層のディレクトリ８１４が関連付けられている。「－」表示のボタン（画像）８５２を押すことにより、ディレクトリ８１４が表示画面から消え、「－」表示が「＋」表示に変わる。

ディレクトリ８０２は所定の色付けがなされ、且つ、「ＲＸ」が表示されている。前述したように、「Ｒ」は「読み取り許可（Read）」、「Ｘ」は「実行許可（eXecute）」であるから、ディレクトリ８０２には、「書き込み」と「ファイル作成」の禁止が設定されていることがわかる。また、ディレクトリ８０４には、別の色が付され、且つ、「ＲＷ」が設定されている。「Ｒ」は「読み取り許可（Read）」、「Ｗ」は、「書き込み許可（Write）」であるから、ディレクトリ８０４には、「ファイル作成」と「実行」の禁止が設定されていることがわかる。同様に、ディレクトリ８０６には、読み取り許可のみが設定されていることがわかる。

【００５９】

図２０の「＋」表示のボタン（画像）８５１をクリックすると、表示装置１２０の表示画面は、図１９から図２０のようにディレクトリ８０４の下位層のディレクトリが表示され、ボタン（画像）８５１は「＋」から「－」に変わる。図２１の例では、ディレクトリ８０４に属する複数のディレクトリ８１５～８２５のうち、「ＰＴＳ」により識別されるディレクトリ８２４を除いて、他のすべてのディレクトリ８１５～８２２、８２４、８２５が同じ色で表されることから、設定されている制御内容が同じであることが一目で理解でき、また、制御内容は「ＲＷ」とあるから「ファイル作成」と「実行」の禁止が設定されていることがわかる。

【００６０】

ここで、設定者が、例えば図２１において、ディレクトリ８０４に設定を行うと、同じラベルに関連付けられている下位層のディレクトリ８１５～８２２、８２４～８２５も、連動して色及び設定内容の表示が変更される。そのため、設定者は、その設定操作の過程で、どのディレクトリに対して同じラベルが関連づけられており、どのような設定内容となっているかを知ることができるため、マッピングファイル２２０及びアクセス制御ルール２３０を参照する必要がなく、設定作業が容易になる。

【００６１】

このように、本実施形態では、一度に、全てのリソースのツリー構造とマッピングファイル２２０及びアクセス制御ルール２３０の情報とを関連付けると膨大な処理量となり、時間がかかるため、下位層のディレクトリを展開し、表示装置１２０に表示するときには

じめて、必要な範囲で、マッピングファイル２２０及びアクセス制御ルール２３０の情報を収集し、ラベルの関連付けを行うようにしている。これにより、設定者の操作単位で、ラベルの関連付けを行う結果、一度に大量の処理を行う必要がなくなる。

また、一旦収集した情報を、内部のメモリに記録しておけば、情報収集のため再度同じ処理を行う必要がなくなる。

【００６２】

このように色づけを行う中での例外は、例えば、図４（ａ）の３行目「type_transition httpd_t var_run_t:file httpd_var_run_t:」で示す表現である。あるプロセスが、あるラベルに割当てられたディレクトリ以下に、ファイルやディレクトリを作成した場合、別ラベルを割当てるというものである。つまり、事前に新ラベルを一つ作成しておき、決められたラベルが付与されるディレクトリ以下にファイルを作成する場合、通常親のラベルを踏襲するが、今回は作成した新ラベルを割当てるということである。この場合、図１５の（ｂ）で示す画面で設定を行う。９０７は９０１と同じ用途であるが、９０８については、作成するファイルやディレクトリの正規表現で表されたパスを入力する。図１３の７５０で示されるように’＊’（何でも良い）という表記も可能である。

図１５の（ｂ）においてＯＫボタンを押した直後は、新ラベルを生成し、指定されたパーミッションオブジェクトと関連付けるが、Ｓ４１０で指定したファイルやディレクトリに新ラベルを割当てればよいわけではなく、Ｓ４１０で指定したディレクトリ以下に存在するファイルやディレクトリに対して９０７で指定した正規表現がマッチングするか調べ、逐次新ラベルを割当てる必要がある（Ｓ４８０）。

【００６３】

［ステップＳ３の詳細手順］

ここで再び図７乃至図９に戻り、過去にアクセス制御設定装置１００による設定がなされていた場合の、表現形式変換処理（ステップＳ３）について説明する。

本実施形態による設定作業のためには、パスオブジェクトが前述のとおり、図１６に示される３つのタイプで表現されていなければならない。しかし、いったん設定作業を終えて設定内容をアクセスポリシに反映してしまうと、リソースと直接関連するラベル情報は新ラベルに置き換わり、“ｌｓ－Ｚ”コマンドでラベル情報を読み込むだけでは、タイプ１（旧参照のみがデフォルトラベルオブジェクトを参照）は再現されるが、タイプ２（旧参照がデフォルトラベルオブジェクトを、新参照が新ラベルオブジェクトを参照するもの）とタイプ３（タイプ２の子ノード）のパスオブジェクトが再現されない。以下、タイプ２とタイプ３のパスオブジェクトを再現する手順を、図７乃至９、図１８及び図１９を用いて説明する。

【００６４】

ステップＳ３２０で取得したラベルが、過去にアクセス制御設定装置１００により設定されたラベル"myLabel_t"であった場合でも、処理装置１１０はまず旧参照に当該新ラベルのラベルオブジェクトを参照させる（ステップＳ３３０）。この結果、タイプ２およびタイプ３のパスオブジェクトはそれぞれ図１８（ａ）及び（ｂ）のようになる。

次に、図８に移り、処理装置１１０は、ステップ３２０で取得したラベルがアクセス制御設定装置１００が付与したものか否かを判定する。アクセス制御設定装置１００が生成したラベルＡがあった場合（ステップＳ３３５：Ｙｅｓ）、処理装置１１０はさらにラベルオブジェクトＡと関連付けられたパスオブジェクトが、過去に新ラベルを割り当てた親ノードなのか、その子ノードなのか判定する（ステップＳ３４５）。パスオブジェクトの名前がマッピングオブジェクトのパスと一致すれば親ノードであり、一致しなければ子ノードである。

まずタイプ２（図１８（ａ））について見ると、"/boot"はマッピングオブジェクト（図１８（ｃ））のパス"/boot"と一致するので親ノードであり、この場合は当該パスオブジェクトの新参照にも同じ（新ラベルの）ラベルオブジェクトを参照させる（図１８（ｄ）：ステップＳ３５０）。

【００６５】

一方、タイプ３（図１８（ｂ））では、"/boot/xxx"はマッピングオブジェクト（図１８（ｃ））のパス"/boot"と一致しないので子ノードであり、この場合は新参照によるラベルオブジェクトの参照は行わせない（図１８（ｂ）のまま変わらない：ステップＳ３５０）。

ついで、処理装置１１０は、新ラベルを参照するパスオブジェクトの旧参照と、その親ノードのパスオブジェクトの旧参照とを一致させる（親に従わせる：ステップＳ３５５）。すなわち、図１９（ａ）のように"/boot"の親ディレクトリ"/"のパスオブジェクトの旧参照がデフォルトラベルオブジェクト"boot_t"を参照している場合、タイプ２は図１９（ｂ）のようになり、タイプ２（図１７のタイプ２）が再現される。

また、旧参照のみが新ラベルオブジェクトを参照していたタイプ３（図１８（ｂ））の場合は、親ノード（"/boot"）の旧参照がデフォルトラベルオブジェクト"boot_t"を参照することになったので、図１９（ｃ）のようになり、やはりタイプ３（図１７のタイプ３）が再現される。

【００６６】

以上の処理を経て、処理装置１１０は、過去にアクセス制御設定装置１００による設定がなされていた場合、表現形式変換処理に必要なパスオブジェクトの再現に成功する。

【００６７】

設定作業が一度終了した後は、パーミッション情報はアクセス制御ルールに反映されている。したがって、続いて処理装置１１０は、アクセス制御設定装置１００によって付与されたラベルＡをキーにしてアクセス制御ルールを検索し、ラベルＡに関するパーミッション情報を取得する（ステップＳ３６０）。処理装置１１０は得られたパーミッション情報に基づいてラベルオブジェクトとパーミッションオブジェクトを関連付ける（ステップＳ３６５）。

以上の結果、処理装置１１０は図１４に示したような、パスオブジェクト、ラベルオブジェクト、パーミッションオブジェクトの再構成を完了し、ラベルオブジェクトを介して結合されたパーミッションオブジェクトの種類にしたがって、パスオブジェクトが指示するリソースに色づけを行う（ステップＳ３７０）。

【００６８】

次に図９に移る。プログラムによって動的に生成・削除されるファイルやディレクトリについては、新たなラベルオブジェクトを生成してもパスオブジェクトからは旧参照にも新参照にも参照させない。実在パスについてパスオブジェクト、ラベルオブジェクト、パーミッションオブジェクトについて関連付ける処理を終えると、処理装置１１０はマッピングオブジェクトの中に動的に生成・消滅するファイルＢとラベルＢの定義がないか検索する（ステップＳ３４０）。

動的に生成・消滅するファイルＢとラベルＢの定義がある場合（ステップＳ３４０：Ｙｅｓ）、処理装置１１０はさらにパスオブジェクトを確認し、動的なファイルＢがある場合は（ステップＳ３７５：Ｙｅｓ）、対応するラベルＢをキーにアクセス制御ルールを検索し、ラベルＢに関するパーミッション情報を取得する（ステップＳ３８０）。処理装置１１０は得られたパーミッション情報に基づいてラベルオブジェクトＢとパーミッションオブジェクトを関連づけ（ステップＳ３８５）、パーミッションオブジェクトの種類にしたがって、動的リソースに色付けを行う（ステップＳ３９０）。

【００６９】

このようにして、本実施形態のアクセス制御設定装置１００は、過去になした設定情報から3つのタイプのパスオブジェクトを再現して表現形式変換処理を行い、リソースをツリー構造で表現した「アクセス制御設定画面」をユーザに提示する。過去にアクセス制御設定がなされていた場合、アクセス制御設定画面に表示されるリソースは、例えば図２０や図２１のように、パーミッション情報ＲＷＸＣとともに色分けされて表示される。

【００７０】

［ステップＳ５の詳細手順］

設定漏れ検出処理（ステップＳ５）の詳細手順を図１１に示す。

処理装置１１０は、設定者の設定漏れ検出指示を受け付けると、ラベルオブジェクトから処理装置１１０が保持するラベル一覧（ラベルリスト１）を生成する（Ｓ５１０）。ラベルオブジェクトは、アクセス制御設定者がツリー構造を展開することにより収集されたものと、アクセス制御設定者が新たに定義したラベルの集合なので、ラベルリスト１はアクセス制御設定者が確認済みのラベル一覧である。ここで、アクセス制御設定者がツリー構造を展開することにより収集されたものには、既にアクセス制限が設定されているものと、単に表示装置に表示されただけのものが含まれる。

一方で、処理装置１１０は、マッピングファイル２２０に記録されている全てのラベル情報を取得してラベル一覧（ラベルリスト２）を生成する（Ｓ５２０）。ラベルリスト２は、ツリー構造展開の過程で処理装置１１０が取得し、すでにアクセス制御設定者が確認したものも含む。

そして、処理装置１１０はラベルリスト１とラベルリスト２をマージしたラベル一覧（ラベルリスト３）を生成する（Ｓ５３０）。ラベルリスト３は、アクセス制御設定対象となりうる全ラベル一覧となる。

さらに、処理装置１１０はラベルリスト３からラベルリスト１を差し引いたラベル一覧（ラベルリスト４）を生成し（Ｓ５４０）、ラベルリスト４を設定者に提示する（Ｓ５５０）。すなわち、アクセス制御設定対象となりうる全ラベル一覧（ラベルリスト３）からアクセス制御設定者が確認済のラベル一覧（ラベルリスト１）を差し引くこととなり、ラベルリスト４は全ラベル一覧のうち、アクセス制御設定者が未確認のラベル一覧となる。

【００７１】

ブラックリスト方式では、ブラックリスト方式のデフォルト状態である「ＲＷＸＣ全て許可」のまま残されたリソースが、確認の上で「全許可」としたものか、未確認のために「全許可」となっているのか容易に判別できない、という問題があった。本発明のブラックリスト方式に基づくアクセス制御設定装置によれば、設定者は設定作業完了後、未確認のラベルのみを把握することができるので、ブラックリスト方式の設定作業が容易、かつ確実なものとなる。

【００７２】

設定者は、処理装置１１０が提示したラベルリスト４を確認し、設定漏れを確認する。設定者が設定漏れを発見した場合（ステップＳ６：Ｎｏ）、処理装置１１０は設定者の指示を受けて設定情報生成処理（ステップＳ４）に戻る。

設定者が設定漏れを発見しなければ、処理装置１１０は設定者の指示を受けて設定情報をマッピングファイルとアクセス制御ルールに反映させて、全体処理を終了する（ステップＳ７）。

【００７３】

このように、本実施形態によるアクセス制御設定装置１００は、端末装置２００から、リソースの情報とラベル表現形式で設定されたアクセス制御の内容を表す情報を取得し、取得した情報のうち、各階層のリソースのＩＤとそのリソースが属するグループ名とをリンクさせ、さらに、各グループ名と、そのグループに設定されたアクセス制御の内容とをリンクさせることにより、当該アクセス制御の内容をパス表現形式の代表例である階層構造のものに変換し、変換されたアクセス制御の内容を表示装置１２０に表示させるとともに、アクセス制御の設定内容が同一となる複数のリソースについては同一の態様で表示させるようにしたので、階層構造の表現形式には詳しいが、ラベル表現形式への理解が十分でない設定者が、容易に既存の設定内容を把握することができる。

これにより、過大なアクセス制限が設定されていたり、あるリソースに対しては必要以上に厳しいアクセス制限がなされているなど、ゆる過ぎたり厳しすぎたりする設定がなされている状況の把握が容易になり、端末装置２００のセキュアＯＳを使用する際の初期学習効果を軽減することができる。また、ラベル表現形式でアクセス制御の設定を行った設定者がいなくなった場合でも、階層構造の表現形式に慣れた者が容易に既存の設定内容を把握できるという利点も生じる。

【００７４】

また、すべてのリソース情報から、既にアクセス制限が設定されているリソース及び表示装置１２０に表示されたリソースのリソース情報を差し引くことで設定漏れのリソースを検出するようにしたので、ブラックリスト方式をアクセス制御に適用したときの利点を活かしつつ、残っていた設定漏れなのか、設定者が確認のうえ許可したものなのか判断がつかないという課題を解決することができる。

【００７５】

なお、設定漏れの検出に際しては、検出されたリソースについては、設定者による設定処理を待つことなく、自動的にアクセス禁止とする処理を加えるようにしても良い。このようにすれば、例えばセキュアＯＳがアクセス可能なリソースとしてＵＳＢメモリを設定する場合、設定時にそのＵＳＢメモリが存在しない場合であっても、設定をし忘れることが無くなり、より安全なアクセス制御が可能になるという効果が得られる。

【００７６】

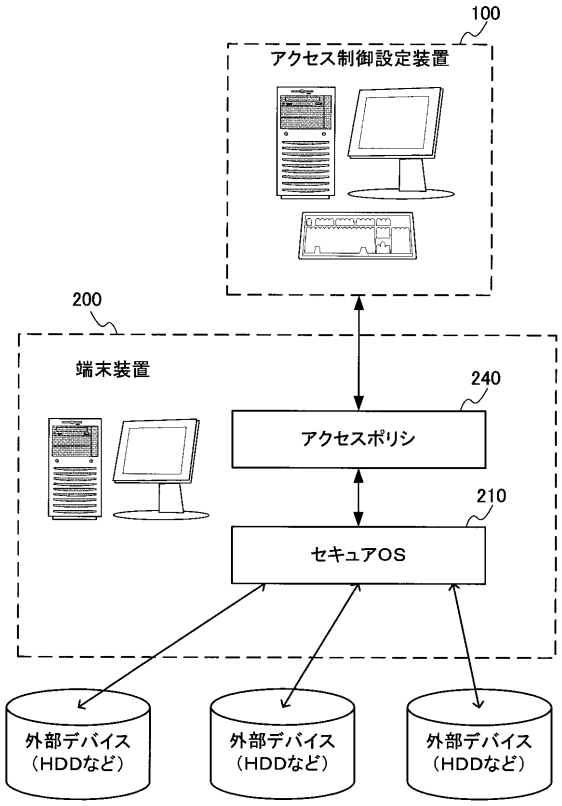
なお、上記実施例ではファイルやフォルダ等のリソースにＲ、Ｗ、Ｘ、Ｃの記号が表示されている場合、それぞれ当該リソースについて読み取り、書き込み、実行、ファイル作成が「許可」されていることを意味したが、逆に、ファイルやフォルダ等のリソースにＲ、Ｗ、Ｘ、Ｃの記号が表示されている場合、それぞれ当該リソースについて読み取り、書き込み、実行、ファイル作成が「禁止」されている状態を表すものとしてアクセス制御設定装置を構成してもよい。

【符号の説明】

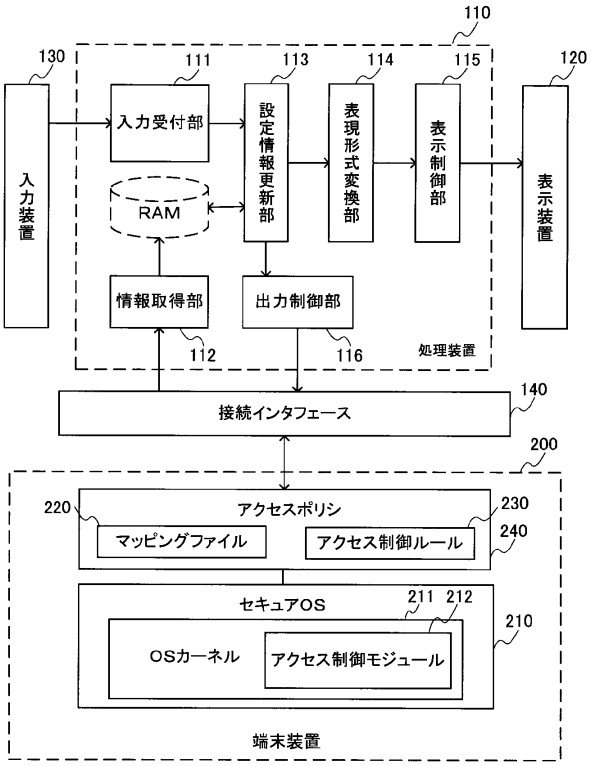
【００７７】

１００・・・アクセス制御設定装置、１１０・・・処理装置、１１１・・・入力受付部、１１２・・・情報取得部、１１３・・・設定情報更新部、１１４・・・表現形式変換部、１１５・・・表示制御部、１１６・・・出力制御部、１２０・・・表示装置、１３０・・・入力装置、１４０・・・接続インタフェース、２００・・・端末装置、２１０・・・セキュアＯＳ、２１１・・・ＯＳカーネル、２１２・・・アクセス制御モジュール、２２０・・・マッピングファイル、２３０・・・アクセス制御ルール、２４０・・・アクセスポリシ、３００・・・初期情報入力画面、４１０～４２６・・・パスオブジェクト、５１０～５２８・・・ラベルオブジェクト、６１０～６１５・・・パーミッションオブジェクト、７１０～７６０・・・マッピングオブジェクト、８００～８２５・・・ディレクトリ、８５１～８５２・・・ボタン（画像）、９００・・・詳細設定用画面、１０００・・・相互確認用の設定画面。

【図１】



【図２】



【図３】

220

```
.....
/var(/.*)?  system_u:object_r:var_t:s0
/var/run/*.pid  system_u:object_r:var_run_t:s0
/usr/sbin/httpd  system_u:object_r:httpd_exec_t:s0
/usr(/.*)?  system_u:object_r:var_t:s0
/etc/init¥.d/http--system_u:object_r:initrc_exec_t:s0
/opt/java -- system_u:object_r:var_t:s0
.....
```

【図４】

(a)

230

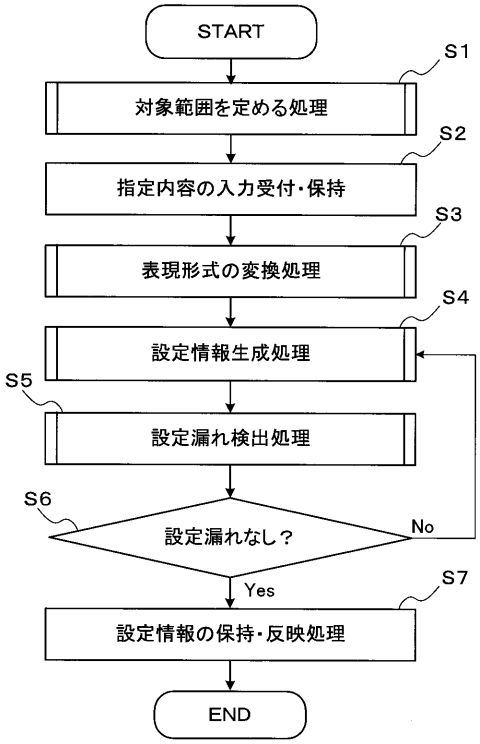
```
attribute sysadm_type;
type var_t sysadm_type;
type usr_t sysadm_type;
.....
allow masumoto_t sysadm_type :file read;
.....
.....
allow httpd_t var_t:file read;
type_transition usr_t httpd_exec_t:process httpd_t;
type_transition httpd_t var_run_t:file httpd_var_run_t:
.....
```

(b)

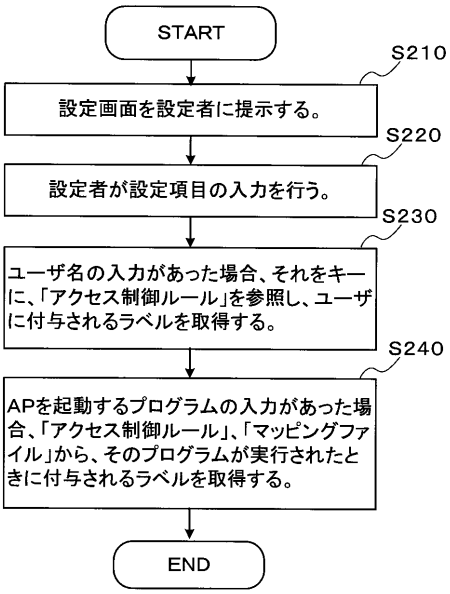
230

```
.....
1) root:root:s0-s0:c0.c1023
2) user root roles { my_root_r system_r } level s0 range s0-
s0:c0.c1023;
3) my_root_r: my_root_t
.....
```

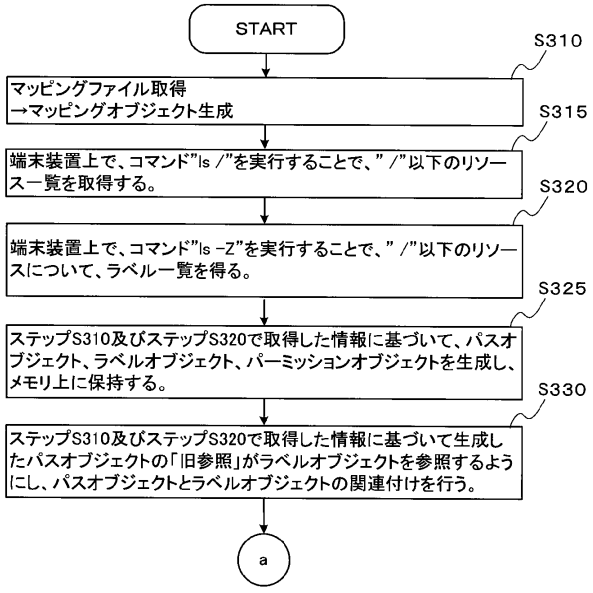
【図５】


START
S1
対象範囲を定める処理
S2
指定内容の入力受付・保持
S3
表現形式の変換処理
S4
設定情報生成処理
S5
設定漏れ検出処理
S6
設定漏れなし？
No
Yes
S7
設定情報の保持・反映処理
END

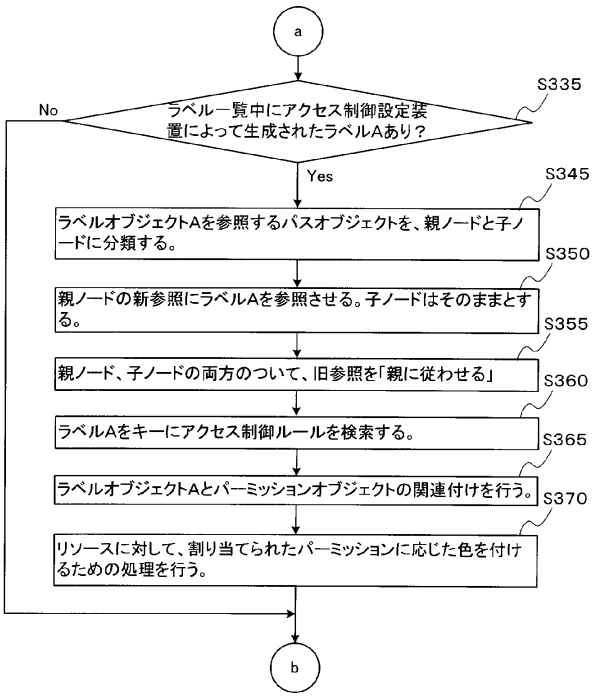
【図６】


START
S210
設定画面を設定者に提示する。
S220
設定者が設定項目の入力を行う。
S230
ユーザ名の入力があった場合、それをキーに、「アクセス制御ルール」を参照し、ユーザに付与されるラベルを取得する。
S240
APを起動するプログラムの入力があった場合、「アクセス制御ルール」、「マッピングファイル」から、そのプログラムが実行されたときに付与されるラベルを取得する。
END

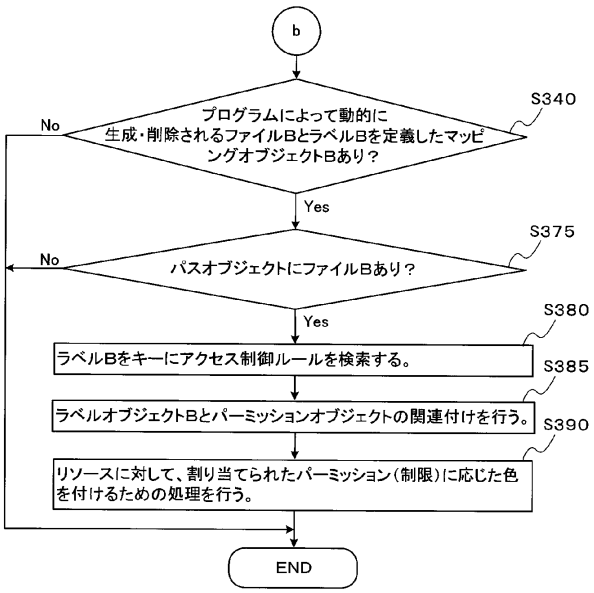
【図７】


START
S310
マッピングファイル取得
→マッピングオブジェクト生成
S315
端末装置上で、コマンド"ls /"を実行することで、" /"以下のリソース一覧を取得する。
S320
端末装置上で、コマンド"ls -Z"を実行することで、" /"以下のリソースについて、ラベル一覧を得る。
S325
ステップS310及びステップS320で取得した情報に基づいて、パスオブジェクト、ラベルオブジェクト、パーミッションオブジェクトを生成し、メモリ上に保持する。
S330
ステップS310及びステップS320で取得した情報に基づいて生成したパスオブジェクトの「旧参照」がラベルオブジェクトを参照するようにし、パスオブジェクトとラベルオブジェクトの関連付けを行う。
a

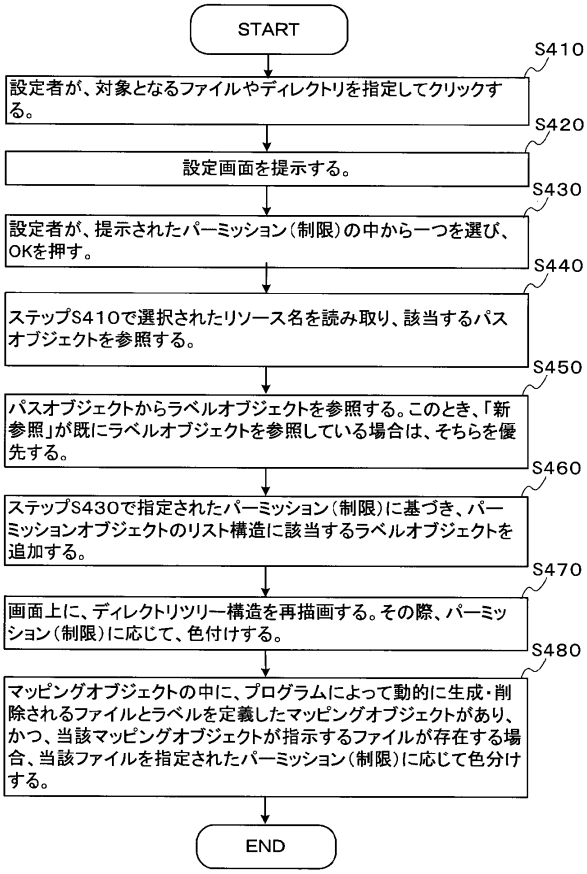
【図８】


a
S335
No
ラベル一覧中にアクセス制御設定装置によって生成されたラベルAあり？
Yes
S345
ラベルオブジェクトAを参照するパスオブジェクトを、親ノードと子ノードに分類する。
S350
親ノードの新参照にラベルAを参照させる。子ノードはそのままとする。
S355
親ノード、子ノードの両方のついて、旧参照を「親に従わせる」
S360
ラベルAをキーにアクセス制御ルールを検索する。
S365
ラベルオブジェクトAとパーミッションオブジェクトの関連付けを行う。
S370
リソースに対して、割り当てられたパーミッションに応じた色を付けるための処理を行う。
b

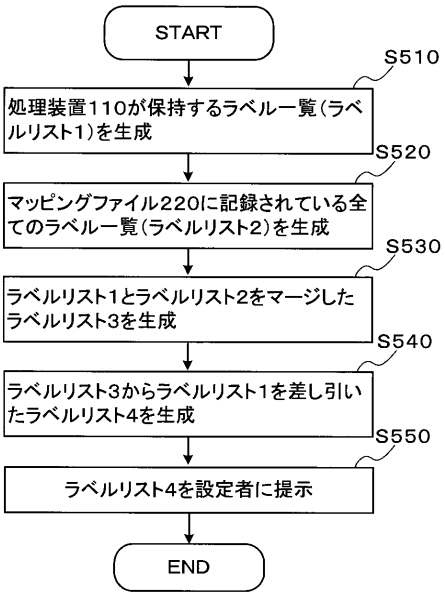
【図９】



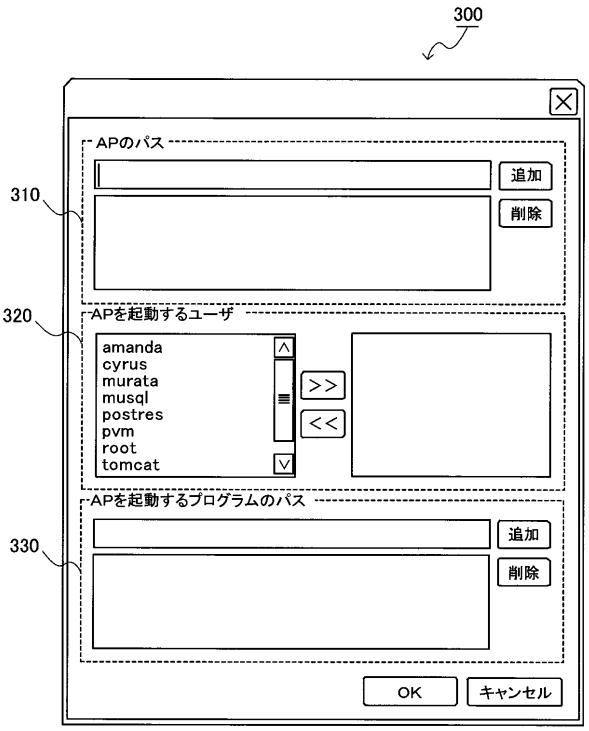
【図１０】



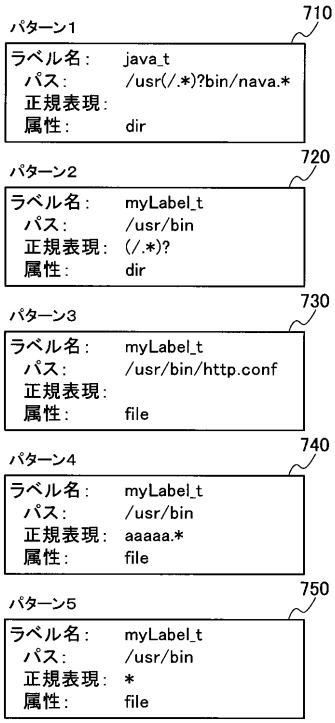
【図１１】



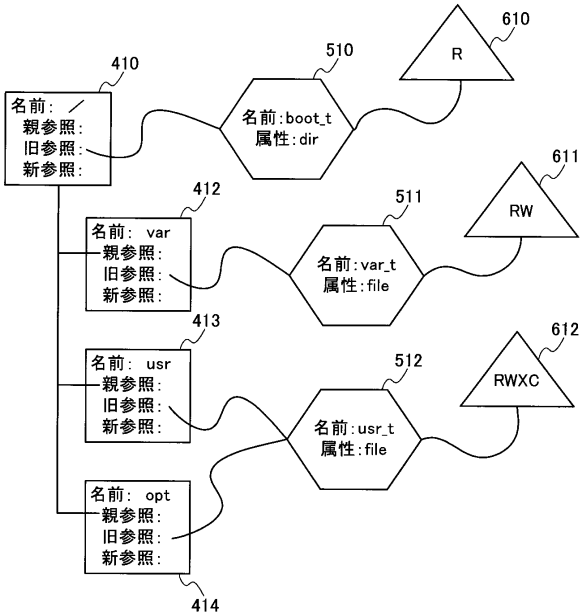
【図１２】



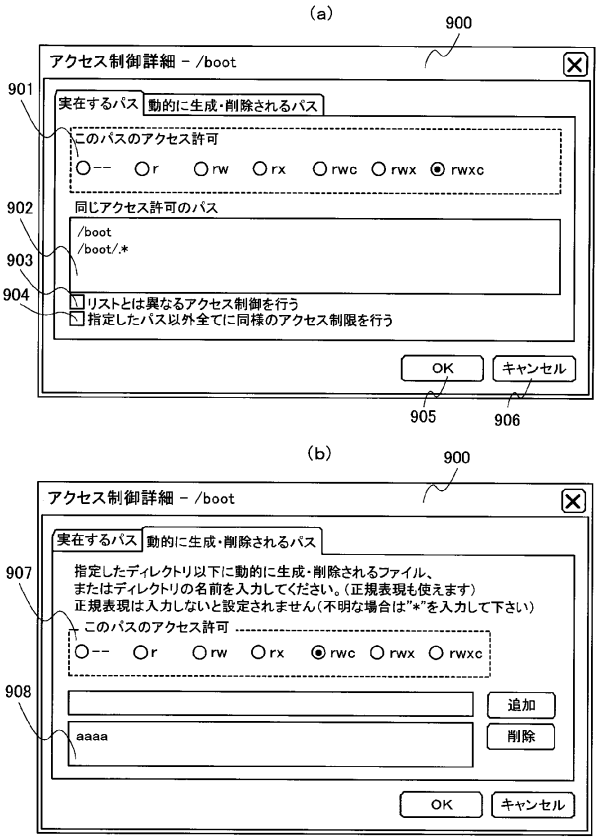
【図１３】



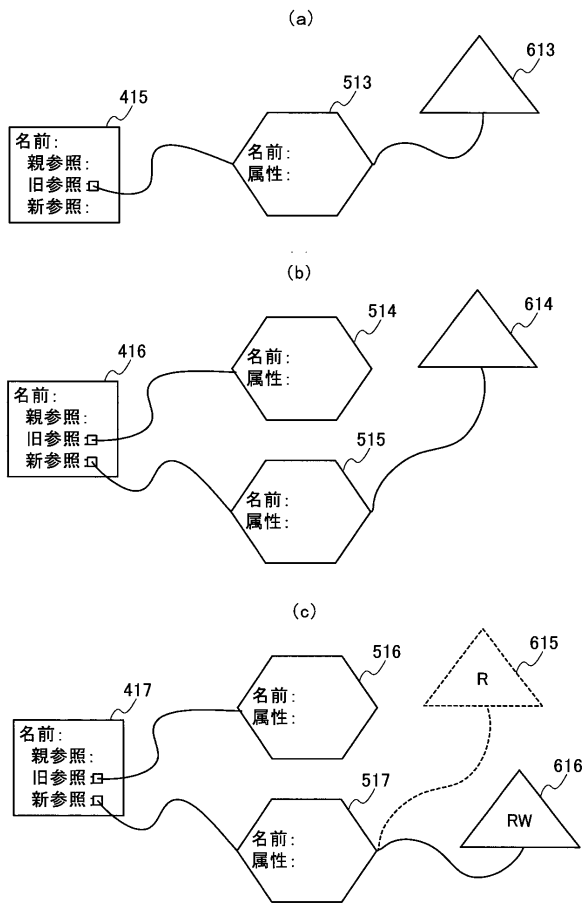
【図１４】



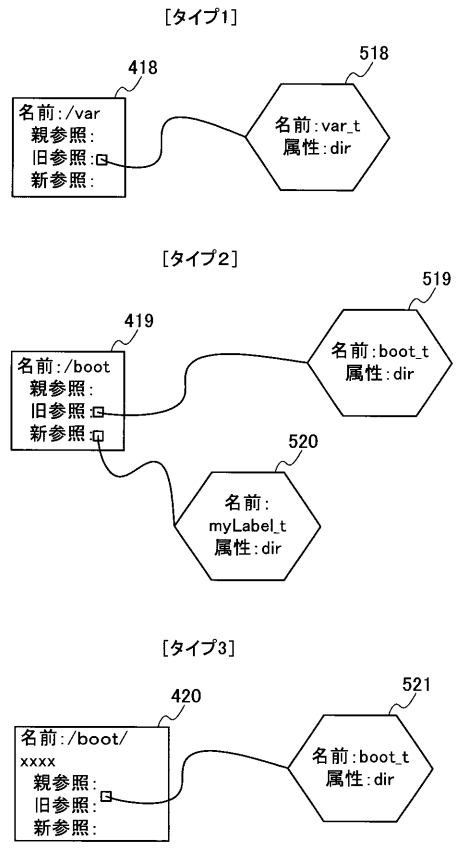
【図１５】



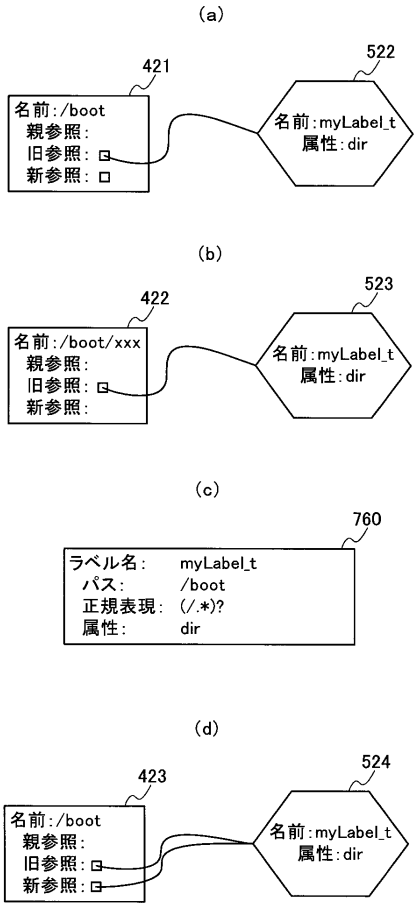
【図１６】



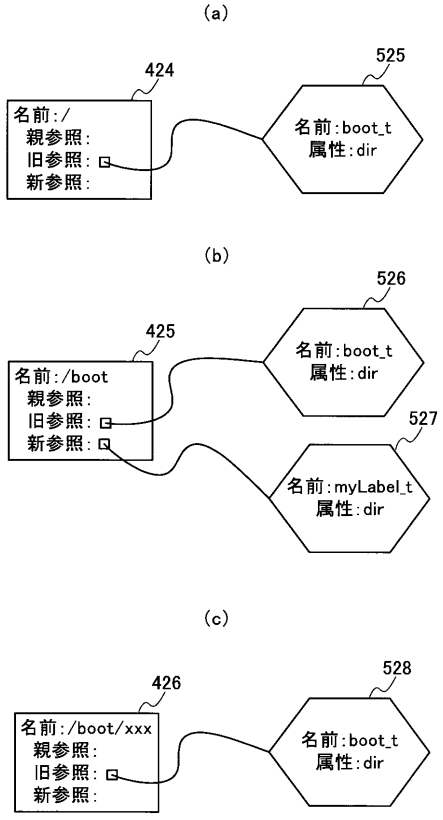
【図１７】



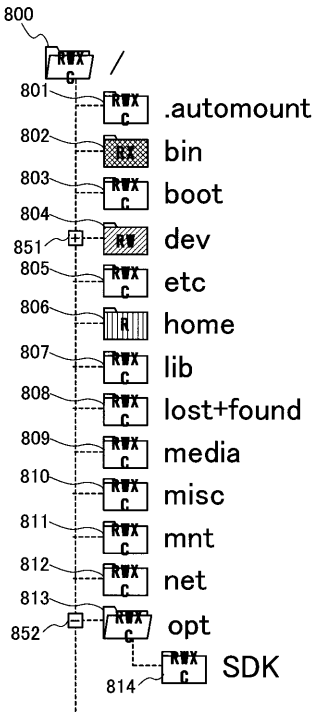
【図１８】



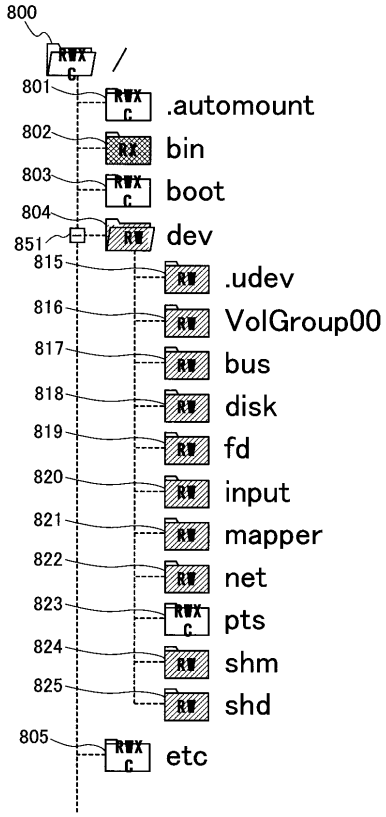
【図１９】



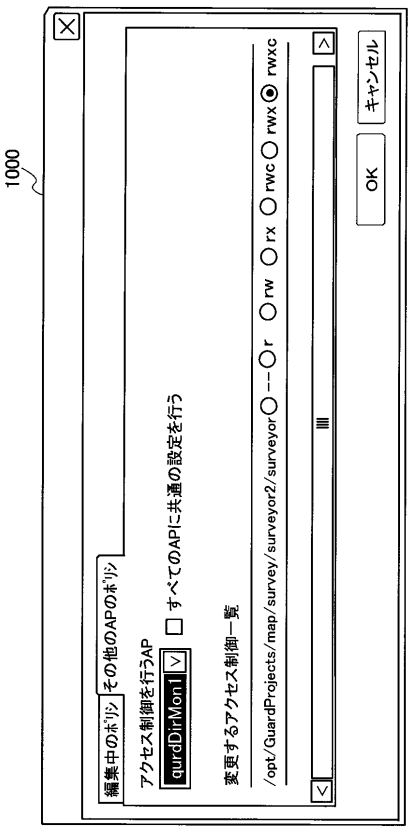
【図２０】



【図２１】


800
/
801
.automount
802
bin
803
boot
804
dev
851
815
.udev
816
VolGroup00
817
bus
818
disk
819
fd
820
input
821
mapper
822
net
823
pts
824
shm
825
shd
805
etc

【図２２】


1000
編集中のﾎﾟﾘｼｰ
その他のAPのﾎﾟﾘｼｰ
アクセス制御を行うAP
qurdDirMon1
すべてのAPに共通の設定を行う
変更するアクセス制御一覧
/opt/GuardProjects/map/survey/surveyor2/surveyor
r
rw
rx
rwc
rwx
rwxc
OK
キャンセル

フロントページの続き

(56)参考文献 特開２００５－２６７２３７（ＪＰ，Ａ）
特開２００５－０６３２２３（ＪＰ，Ａ）
特開２００５－２０９０６８（ＪＰ，Ａ）
特開２００３－１６２４４９（ＪＰ，Ａ）
特開２００２－３４２１４３（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　２１／６２