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

特許第4889575号
(P4889575)

(45)発行日　平成24年3月7日(2012.3.7)　(24)登録日　平成23年12月22日(2011.12.22)

| (51)Int.Cl. | | F I | | |
|---|---|---|---|---|
| G06F 21/24 | (2006.01) | G06F | 21/24 | 165D |
| G06F 21/04 | (2006.01) | G06F | 21/04 | 185 |
| G06F 9/44 | (2006.01) | G06F | 9/44 | 530S |

請求項の数 6　(全 9 頁)

(21)出願番号　特願2007-154249 (P2007-154249)
(22)出願日　平成19年6月11日 (2007.6.11)
(65)公開番号　特開2008-305336 (P2008-305336A)
(43)公開日　平成20年12月18日 (2008.12.18)
審査請求日　平成21年7月21日 (2009.7.21)

(73)特許権者　000004226
日本電信電話株式会社
東京都千代田区大手町二丁目３番１号
(74)代理人　100083806
弁理士　三好　秀和
(74)代理人　100120455
弁理士　勝　治人
(72)発明者　森　航哉
東京都千代田区大手町二丁目３番１号　日本電信電話株式会社内
(72)発明者　依田　育生
東京都千代田区大手町二丁目３番１号　日本電信電話株式会社内

審査官　宮司　卓佳

最終頁に続く

(54)【発明の名称】アクセス許可設定方法、アクセス許可設定装置およびアクセス許可設定プログラム

(57)【特許請求の範囲】
【請求項１】
コンピュータが行う、インスタンスへのアクセス許可を設定するアクセス許可設定方法であって、
前記インスタンスに、一意に識別可能な識別情報を付与する付与ステップと、
前記インスタンスの識別情報と、当該インスタンスへのアクセスを許可するアプリケーションの情報とを記憶部に登録する登録ステップと、
アプリケーションからインスタンスが呼び出された際に、当該インスタンスの識別情報と呼出元のアプリケーションの情報とが前記記憶部に登録されている場合は当該アプリケーションからインスタンスへのアクセスを許可し、前記記憶部に登録されていない場合は当該アプリケーションからインスタンスへのアクセスを禁止する制御ステップと、を行うこと
を特徴とするアクセス許可設定方法。
【請求項２】
請求項１記載のアクセス許可設定方法であって、
前記インスタンスの識別情報は、OSGi Frameworkのservice.pidであること
を特徴とするアクセス許可設定方法。
【請求項３】
インスタンスへのアクセス許可を設定するアクセス許可設定装置であって、
前記インスタンスに一意に識別可能な識別情報を付与する付与手段と、

前記インスタンスの識別情報と、当該インスタンスへのアクセスを許可するアプリケーションの情報とを記憶手段に登録する登録手段と、
アプリケーションからインスタンスが呼び出された際に、当該インスタンスの識別情報と呼出元のアプリケーションの情報とが前記記憶手段に登録されている場合は当該アプリケーションからインスタンスへのアクセスを許可し、前記記憶手段に登録されていない場合は当該アプリケーションからインスタンスへのアクセスを禁止する制御手段と、を有すること
を特徴とするアクセス許可設定装置。
【請求項４】
請求項３記載のアクセス許可設定装置であって、
前記インスタンスの識別情報は、OSGi Frameworkのservice.pidであること
を特徴とするアクセス許可設定装置。
【請求項５】
コンピュータが実行する、インスタンスへのアクセス許可を設定するアクセス許可設定プログラムであって、
前記コンピュータに、
前記インスタンスに一意に識別可能な識別情報を付与する付与ステップと、
前記インスタンスの識別情報と、当該インスタンスへのアクセスを許可するアプリケーションの情報とを記憶部に登録する登録ステップと、
アプリケーションからインスタンスが呼び出された際に、当該インスタンスの識別情報と呼出元のアプリケーションの情報とが前記記憶部に登録されている場合は当該アプリケーションからインスタンスへのアクセスを許可し、前記記憶部に登録されていない場合は当該アプリケーションからインスタンスへのアクセスを禁止する制御ステップと、を実行させること
を特徴とするアクセス許可設定プログラム。
【請求項６】
請求項５記載のアクセス許可設定プログラムであって、
前記インスタンスの識別情報は、OSGi Frameworkのservice.pidであること
を特徴とするアクセス許可設定プログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、アプリケーションからインスタンスへのアクセス許可の設定を行う技術に関する。
【背景技術】
【０００２】
特許文献１には、宅内のネットワークに接続された機器やセンサなどを連携させて、各種のサービスを実現する機器制御装置（ホームゲートウェイ装置）が記載されている。
【０００３】
また、非特許文献１には、複数のホームサービスを１つのプラットホーム上に多重化して、複数のホームサービスを１つのホームゲートウェイ装置を用いて実現することが記載されている。なお、非特許文献１のホームゲートウェイ装置では、Java（登録商標）VM（Virtual Machine）上で動作するOSGi（Open Services Gateway Initiative） Frameworkにより、ネットワークに接続された機器、センサなどのデバイスを制御する。
【特許文献１】特開２００５－２３４７３３号公報
【非特許文献１】“生活シーンに応じて進化するガス機器を中心とした総合生活支援サービスの実験開始”、[online]、［平成１９年３月１２日検索］、インターネット＜ＵＲＬ：http://www.ntt.co.jp/news/news05/0508/050805.html＞
【発明の開示】
【発明が解決しようとする課題】

【０００４】
非特許文献１のホームゲートウェイ装置では、デバイスを制御するインスタンスを、Java（登録商標）VM上に生成している。インスタンスは、メモリ上に存在するソフトウェアを構成するプログラムの一部であって、必要に応じてクラスから動的に生成される。クラスは、インスタンスの設計図にあたるものであり、プログラムを構成するファイルとして記憶装置上に存在する。また、Java（登録商標）の場合、再利用性を高めるために、関連性の高い複数のクラスがまとめられ、パッケージが構成される。
【０００５】
Java（登録商標）においては、クラスやパッケージには永続的なＩＤがあるが、インスタンスはJava（登録商標）VMの起動後に動的に生成されるために永続的なＩＤを有しない。そのため、非特許文献１のホームゲートウェイ装置では、Java（登録商標）のアクセスコントロールモデルを用いてクラス単位またはパッケージ単位でのアクセスコントロールを行うことができるが、インスタンス単位でのアクセスコントロールを行うことは困難である。このため、同一のクラス定義により生成される複数のインスタンスを、それぞれ特定のアプリケーションのみがアクセスできるようにすることができない。
【０００６】
本発明は上記事情に鑑みてなされたものであり、本発明の目的は、インスタンス単位のアクセスコントロールを可能とするアクセス許可設定方法、アクセス許可設定装置およびアクセス許可設定プログラムを提供することにある。
【課題を解決するための手段】
【０００７】
本発明は、コンピュータが行う、インスタンスへのアクセス許可を設定するアクセス許可設定方法であって、前記インスタンスに一意に識別可能な識別情報を付与する付与ステップと、前記インスタンスの識別情報と、当該インスタンスへのアクセスを許可するアプリケーションの情報とを記憶部に登録する登録ステップと、アプリケーションからインスタンスが呼び出された際に、当該インスタンスの識別情報と呼出元のアプリケーションの情報とが前記記憶部に登録されている場合は当該アプリケーションからインスタンスへのアクセスを許可し、前記記憶部に登録されていない場合は当該アプリケーションからインスタンスへのアクセスを禁止する制御ステップと、を行う。
【０００８】
また、本発明は、アクセス許可を設定するアクセス許可設定装置であって、前記インスタンスに一意に識別可能な識別情報を付与する付与手段と、前記インスタンスの識別情報と、当該インスタンスへのアクセスを許可するアプリケーションの情報とを記憶手段に登録する登録手段と、アプリケーションからインスタンスが呼び出された際に、当該インスタンスの識別情報と呼出元のアプリケーションの情報とが前記記憶手段に登録されている場合は当該アプリケーションからインスタンスへのアクセスを許可し、前記記憶手段に登録されていない場合は当該アプリケーションからインスタンスへのアクセスを禁止する制御手段と、を有する。
【０００９】
また、コンピュータが実行する、インスタンスへのアクセス許可を設定するアクセス許可設定プログラムであって、前記コンピュータに、前記インスタンスに一意に識別可能な識別情報を付与する付与ステップと、前記インスタンスの識別情報と、当該インスタンスへのアクセスを許可するアプリケーションの情報とを記憶部に登録する登録ステップと、アプリケーションからインスタンスが呼び出された際に、当該インスタンスの識別情報と呼出元のアプリケーションの情報とが前記記憶部に登録されている場合は当該アプリケーションからインスタンスへのアクセスを許可し、前記記憶部に登録されていない場合は当該アプリケーションからインスタンスへのアクセスを禁止する制御ステップと、を実行させる。
【発明の効果】
【００１０】

本発明によれば、インスタンスに一意に識別可能な識別情報を付与し、当該識別情報を用いてアクセスの可否を判断するため、インスタンス単位でのアクセスコントロールを可能とすることができる。すなわち、各インスタンスを、特定のアプリケーションからのみアクセスさせることができる。

【発明を実施するための最良の形態】

【００１１】

以下、本発明の実施の形態について、図面を参照して説明する。

【００１２】

図１は、本発明の一実施形態のホームゲートウェイ装置を示す図である。

【００１３】

図示するホームゲートウェイ装置１は、通信部１１、記憶部１２、入力部１３、表示部１４および制御部１５を備えた汎用的なコンピュータシステムであって、これら各部１１～１５は、内部バス１６を介して接続されている。

【００１４】

通信部１１は、ホームネットワーク３に接続され、当該ホームネットワーク３を介して機器２と通信を行う。

【００１５】

なお、説明の便宜上、図１には機器２のみがホームネットワーク３に接続されているが、複数の機器がホームネットワーク３に接続されていてもよい。

【００１６】

記憶部１２は、ハードディスク等の記録媒体により構成され、ホームゲートウェイ装置１自体を制御する制御プログラム、図２で説明するソフトウェアなどが記憶される。

【００１７】

入力部１３は、各種ボタンやキーボードなどの入力装置により構成され、ホームゲートウェイ装置１の各種設定を受け付ける。

【００１８】

表示部１４は、パワーランプや液晶表示装置などの表示装置によって構成され、入力部１３が受け付けた各種設定やホームゲートウェイ装置１の状態を表示する。

【００１９】

制御部１５は、ＣＰＵ(Central Processing Unit)、ＲＯＭ（Read Only Memory）、ＲＡＭ（Random Access Memory）などにより構成される。ＣＰＵが、記憶部１２からメモリ上に読み込まれた各種プログラム（制御プログラム、ソフトウェア等）を実行することにより、ホームゲートウェイ装置１全体が制御され、ホームゲートウェイ装置１の各機能が実現される。

【００２０】

図２は、本実施形態のホームゲートウェイ装置１が、機器２を制御するためのソフトウェア（バンドル、プログラム群）のソフトウェア構成図である。なお、これらのソフトウェアは、Java（登録商標）VM上で動作するOSGi Framework上で動作する。また、図２は、OSGi Frameworkの仕様に従って示したものである。

【００２１】

本実施形態では、１つの物理デバイス（機器）２に対して、当該物理デバイス２を制御するプログラムであるデバイスインスタンス１０４が１つ生成される。アプリケーション１０７は、デバイスインスタンス１０４を利用して物理デバイス２を制御し、ユーザにサービスを提供する。

【００２２】

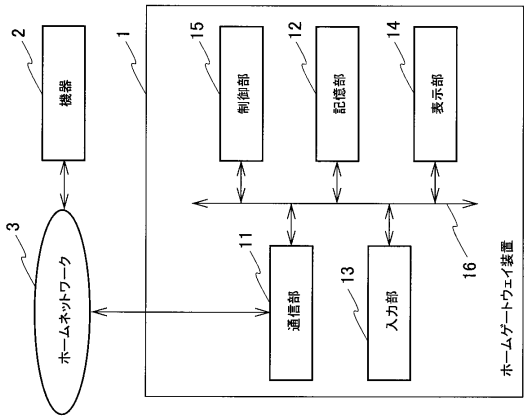
図２において、デバイスドライバ１０３（付与手段）は、物理デバイス２の種別毎に設けられ、対応する種別の物理デバイス２を発見するとともに、当該物理デバイス２を制御するプログラムである。なお、デバイスドライバ１０３は、複数のクラスから構成される。また、デバイスドライバ１０３は、発見した物理デバイス２に対応するインスタンスとして、デバイスインスタンス１０４をOSGi Frameworkに登録する。

【００２３】
デバイスインスタンス１０４（制御手段）は、物理デバイス２毎に設けられ、対応する物理デバイス２に対して所定の処理を行う。また、デバイスインスタンス１０４は、後述するJava（登録商標）VMのアクセスコントローラの機能を利用する。
【００２４】
割当管理１０５（登録手段）は、デバイスインスタンス１０４毎に、当該デバイスインスタンス１０４を利用可能なアプリケーション１０７が指定された割当情報を管理するとともに、割当情報サービス１０６をOSGi Frameworkに登録する。また、割当管理１０５は、割当情報に基づいて後述するパーミッションを生成し、Java（登録商標）VMに登録する。
【００２５】
割当情報サービス１０６は、割当情報を取得または保存するためのインタフェースをデバイスドライバ１０３に提供する。
【００２６】
アプリケーション１０７は、インタフェースであるデバイスインスタンス１０４を呼び出して、当該デバイスインスタンス１０４を用いて物理デバイス２を制御する。
【００２７】
以上説明したホームゲートウェイ装置１において、アプリケーション１０７とのインタフェースであるデバイスインスタンス１０４は、割当情報で指定されたアプリケーション１０７からの呼び出しのみを受け付けて、その他のアプリケーション１０７からの呼び出しは拒否する。
【００２８】
具体的には、デバイスドライバ１０３は、デバイスインスタンス１０４の生成時に指定するservice.pid（識別情報）を、当該デバイスインスタンス１０４の識別子とする。そして、デバイスドライバ１０３は、生成したデバイスインスタンス１０４のservice.pidと、当該インスタンスへのアクセスを許可するアプリケーションとを対応付けた割当情報を、割当管理１０５に登録する。
【００２９】
なお、service.pidは、OSGi Frameworkのサービスオブジェクトであるインスタンスを一意に識別可能なものである。また、OSGi Frameworkの仕様上、デバイスドライバ１０３は、インスタンスの生成および消滅を超えて、インスタンスが意味するものが同じ限り、継続して同じservice.pidを付与する。
【００３０】
図３は、本実施形態のホームゲートウェイ装置１が行うアクセスコントロール処理を説明するための説明図である。
【００３１】
デバイスドライバ１０３は、service.pidを指定してデバイスインスタンス１０４のインスタンスを生成する。そして、デバイスドライバ１０３は、インスタンス生成時にOSGi Frameworkに指定したservice.pidと、当該インスタンスを利用可能な所定のアプリケーション１０７の識別子とを、割当情報サービス１０６を介して割当管理１０５に登録（格納）する（Ｓ１）。なお、当該インスタンスを利用可能なアプリケーション１０７は、ユーザが指定することなどにより、あらかじめ定められているものとする。
【００３２】
OSGi Frameworkを用いる場合には、アプリケーション１０７の識別子として、BundleLOCATIONなどを用いることが可能である。また、これ以外に、電子署名を用いることも可能である。
【００３３】
割当管理１０５は、デバイスドライバ１０３により割当情報が登録されると、当該割当情報に基づいてパーミッション（Permission）１２１を生成し、当該パーミッション１２１をJava（登録商標）VMに登録する（Ｓ２）。

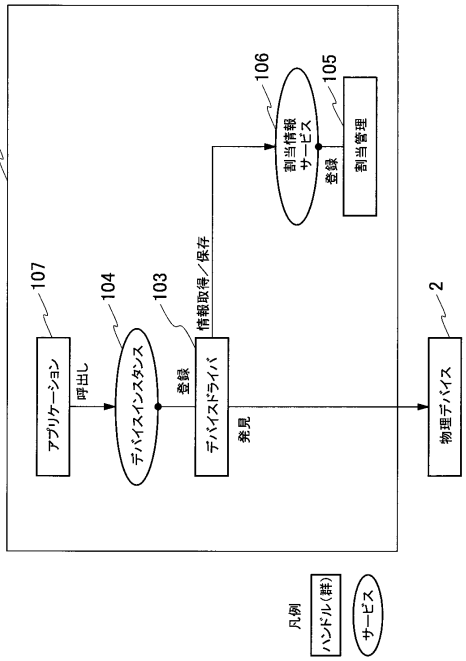
【００３４】
OSGi Frameworkを用いる場合の具体的な登録方法としては、割当管理１０５は、割当情報に設定されたアプリケーション１０７のBundleLOCATIONと、デバイスインスタンス１０４のservice.pidとが対になったパーミッション１２１を生成する。そして、割当管理１０５は、生成したパーミッション１２１をOSGi Frameworkの図示しないパーミッションアドミン（Permission Admin）を用いて、Java（登録商標）VMのポリシー１１３（記憶部）に登録する。
【００３５】
このようにして、Java（登録商標）VMにパーミッション１２１が登録される。
【００３６】
次に、アプリケーション１０７は、ユーザの指示を受け付けて、対象となる物理デバイス２に対応するデバイスインスタンス１０４を呼び出す（Ｓ３）。アプリケーション１０７は、デバイスインスタンス１０４を呼び出す際に、Java（登録商標）VMの機能により自身の識別子がデバイスインスタンス１０４に通知される。
【００３７】
なお、ユーザは、図示しない表示装置に表示されたＧＵＩ（Graphical User Interface）を操作することなどにより、アプリケーション１０７に対する指示を入力する。
【００３８】
アプリケーション１０７から呼び出されたデバイスインスタンス１０４は、処理を開始する前に、Java（登録商標）VMの機能であるアクセスコントローラ（AccessController）１１１に、パーミッションのチェックを要求する（Ｓ４）。すなわち、デバイスインスタンス１０４は、自身に設定されたservice.pidと、アプリケーション１０７の識別子とを含むチェック要求１２２を、アクセスコントローラ１１１に送出する。
【００３９】
アクセスコントローラ１１１は、Java（登録商標）VMの機能を用いて、要求元のデバイスインスタンス１０４を呼び出したアプリケーション１０７が、当該デバイスインスタンス１０４にアクセスする権利があるか否かのパーミッションチェックを行い、チェック結果を返す。
【００４０】
すなわち、アクセスコントローラ１１１は、ポリシー１１３の中にチェック要求１２２に含まれるデバイスインスタンス１０４のservice.pidおよびアプリケーション１０７の識別子のパーミッションが存在する場合は、呼出元のアプリケーション１０７はアクセスする権利があると判別する。一方、ポリシー１１３に存在しない場合は、呼出元のアプリケーション１０７はアクセスする権利がないと判別する。
【００４１】
要求元のデバイスインスタンス１０４は、アクセスコントローラからチェック結果を取得する。呼出元のアプリケーション１０７がアクセス権を有する（パーミッションが与えられている）チェック結果の場合、デバイスインスタンス１０４は、物理デバイス２に対して所定の処理を行う。一方、呼出元のアプリケーション１０７がアクセス権を有しない（パーミッションが与えられていない）チェック結果の場合、デバイスインスタンス１０４は、処理を中断し、エラーを呼出元のアプリケーション１０７に返す。
【００４２】
以上説明した本実施形態のホームゲートウェイ装置１では、インスタンスの生成時に、当該インスタンスを一意に識別可能な識別情報（service.pid）を付与するとともに、当該識別情報とアクセスを許可するアプリケーションとを対応付けたパーミッションを登録する。そして、アプリケーションからインスタンスが呼び出されたときに、インスタンスの識別情報を用いてパーミッションのチェックを行い、アクセスが許可されているアプリケーションから場合にのみ処理を実行する。これにより、本実施形態では、インスタンス単位でのアクセスコントロールを行うことができる。すなわち、各インスタンスを、特定のアプリケーションからのみアクセスさせることができる。

【００４３】
また、このようなインスタンス単位でのアクセスコントロールを応用して、火災、地震、侵入検知などの緊急事態が発生した際に、特定のアプリケーションのみに物理デバイス２の制御権を強制的に移して、危機管理を行うことができる。
【００４４】
具体的には、割当管理１０５は、緊急事態の情報を受信し、または、緊急事態を検知した場合、緊急事態の種別に応じて制御権を移したい物理デバイス２（デバイスインスタンス１０４）の割当情報を取得する。そして、割当管理１０５は、OSGi Frameworkのパーミッションアドミンを用いて、当該デバイスインスタンスのパーミッションをJava（登録商標）VM から削除し、インスタンスとアプリケーションの割当を解除する。
【００４５】
そして、割当管理１０５は、緊急事態の種別に対応して設定されているアプリケーションに、制御権を移したい物理デバイス２に対応するデバイスインスタンスのパーミッションを設定する。
【００４６】
これにより、例えば、火災を検知した場合、火災サービス（アプリケーション）のみに家電機器（物理デバイス）の制御を許可し、他のサービスからの窓やドアを閉めたり電灯を点灯させるといった競合する制御を排除した上で、火災サービスによる危機対応を優先させることができる。
【００４７】
なお、本発明は上記実施形態に限定されるものではなく、その要旨の範囲内で数々の変形が可能である。例えば、上記実施形態では、アプリケーションから呼び出されたインスタンスは、パーミッションチェックを依頼する際に、自身に設定されたservice.pidを用いることとした。しかしながら、service.pidは、インスタンス生成時にデバイスドライバが通知してもよく、また、インスタンスがOSGi Frameworkから取得するようにしてもよい。
【００４８】
また、本実施形態では、インスタンスを識別する情報としてservice.pidを用いたが、インスタンスを識別できる情報であれば他の情報を用いることとしてもよい。
【図面の簡単な説明】
【００４９】
【図１】本発明の実施形態のホームゲートウェイ装置のブロック図である。
【図２】本発明の実施形態のホームゲートウェイ装置のソフトウェア構成図である。
【図３】アクセスコントロール処理を説明するための説明図である。
【符号の説明】
【００５０】
１　ホームゲートウェイ装置
１１　通信部
１２　記憶部
１３　入力部
１４　表示部
１５　制御部
１６　内部バス
１０３　デバイスドライバ
１０４　デバイスインスタンス
１０５　割当管理
１０６　割当情報サービス
１０７　アプリケーション
２　機器（物理デバイス）
３ホームネットワーク

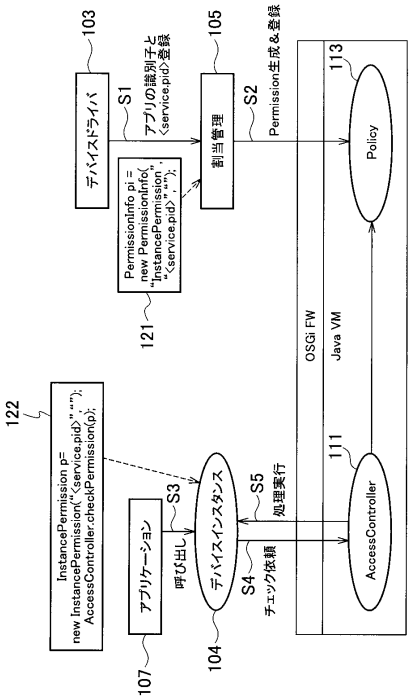
【図１】


2
機器
1
15
制御部
12
記憶部
14
表示部
3
ホームネットワーク
11
通信部
13
入力部
16
ホームゲートウェイ装置

【図２】


1
106
割当情報サービス
105
登録
割当管理
107
アプリケーション
呼出し
104
デバイスインスタンス
103
登録
情報取得／保存
デバイスドライバ
発見
2
物理デバイス
凡例
ハンドル(群)
サービス

【図３】


103
デバイスドライバ
S1
アプリの識別子と<service.pid>登録
105
割当管理
S2
Permission生成＆登録
113
Policy
PermissionInfo pi = new PermissionInfo( "InstancePermission", "<service.pid>", "*");
121
OSGi FW
Java VM
122
InstancePermission p= new InstancePermission("<service.pid>","*"); AccessController.checkPermission(p);
111
AccessController
S3
呼び出し
S5
処理実行
S4
チェック依頼
107
アプリケーション
104
デバイスインスタンス

フロントページの続き

(56)参考文献 特開２００６－３３０８３５（ＪＰ，Ａ）
特開２００６－２１６０６１（ＪＰ，Ａ）
特開２００７－０１８３６９（ＪＰ，Ａ）
特開２００５－０６３４３５（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　２１／０４
Ｇ０６Ｆ　９／４４
Ｇ０６Ｆ　２１／２４