M_008

情報家電サービスを対象としたアクセスコントロール方式

An Access Control Method for The Services using Home Information Appliances

森 航哉† 川幡 太一‡ 依田 育生†
Koya Mori Taichi Kawabata Ikuo Yoda

１．はじめに

近年、ホームネットワークの発達とともに、複数の情報家電製品を連携させて利用できるようなサービスが提供され始めている。しかし現在では１つの機器を複数のサービスに使い回すことはまれであり、今後は、複数の情報家電を複数のサービスで利用できるような世界の実現が期待される。既に UPnP[1]や ECHONET[2]など、標準化された家電制御プロトコルが普及し始めており、次はこれらのプロトコルを利用して様々な機器を制御する汎用的なプラットフォームが必要とされるだろう。

我々は、このような汎用プラットフォームとして、OSGi[3]準拠のホームゲートウェイを家庭内に設置し、その上で機器にアクセスするデバイスドライバと、それらを使ってサービスを提供するソフトウェアを動作させることを考えている[4]。このように複数のサービスから共通に利用できるドライバを備えたプラットフォームを整えることで、より低コストで多様なサービスをユーザに提供できると考えられる（図１）。

このプラットフォームを構築する上で重要になるのは、機器を一元的に管理することである。ここで言う管理には、大きく次の３つの意味が含まれる。１）機器の発見、２）サービスに対する機器の割当設定、３）サービスから機器へのアクセスコントロール、である。このような管理を行うことで、サービス毎に制御できる機器を制限し、セキュリティを担保しつつ、機器制御の競合を防ぐことが可能となる。

このうち最初の機器の発見については、各デバイスドライバの機能に任せ、本方式では、主に後の２つの機能の実現方法を提案する。

２．従来技術と課題

各サービスはドライバを介してしか機器にアクセスできないと仮定すると、上記の技術は、サービスからデバイスドライバへのソフトウェア間の呼び出し関係の設定と制御という、アクセスコントロールの問題に行き着く。OSGi では、プログラムの配布単位であるバンドル毎に、Java の Security システムの Permission[5]を設定することによりアクセスコントロールを実現している。

しかし、Java の標準的な Permission や、OSGi Framework で標準に備わっている ServicePermission の仕組みだけでサービスからデバイスへのアクセスコントロールを行うには、いくつかの課題がある。

第一に、ServicePermission の仕組みでは、クラス単位でしかアクセスコントロールできない。UPnP や ECHONET などの標準プロトコルに対応した汎用のデバイスドライバでは、異なる機器が、同一のクラスの異なるインスタンスとして表現され、インスタンス単位でのアクセスコントロールが必要である。

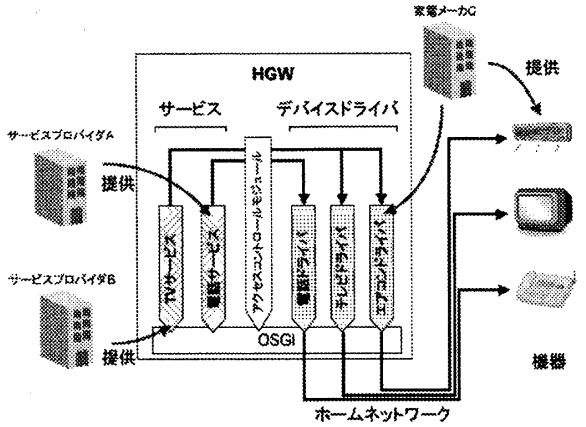

図１　汎用プラットフォームの概要

第二に、あらかじめ各サービスに設定する Permission の内容を決定することは難しい。ホームネットワーク上に存在する機器は、各家庭によって異なる。さらにそれぞれの機器をどのサービスに利用させるかも、各家庭によって異なる。従って、Permission をあらかじめ決定することは出来ず、各家庭において動的に設定する必要がある。

第三に、Permission を後から動的に設定するにあたっては、各サービスが制御可能である機器の種類、すなわちデバイスのクラスを知る必要がある。つまり、サービスが要求する Permission の情報を、設定者にわかりやすく示す必要がある。

本研究では、上記の３つの課題を解決する方法を提案する。

３．提案方式

3.1　アクセスコントロールシステムの構成

ServicePermission ではクラス単位でのアクセスコントロールしか出来ないため、インスタンス単位のアクセスコントロールを行う仕組みを別途用意する。具体的には、サービスが使用してよいデバイスインスタンスの対応表であるアクセスコントロールリスト（ ACL ）[6]を持つアクセスコントロールモジュールを新たに設け、各デバイスのインスタンスのメソッドがサービスから呼び出される際に、対応の有無を確認する（図２）。

このために、各デバイスドライバは OSGi の Service Factory の機能を用いて、呼出し元の各サービスに異なるデバイスインスタンスを払い出し、それぞれのインスタンスには払い出し先のサービスの識別子を持たせる。そして、デバイスインスタンスのメソッドがサービスから呼び出さ

† 日本電信電話（株）
‡ ＮＴＴコミュニケーションズ（株）

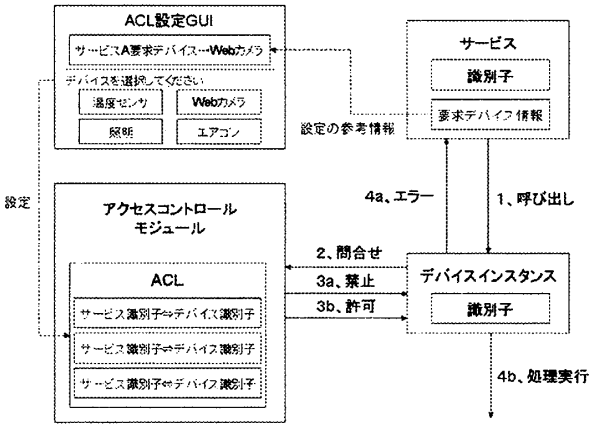

図2　アクセスコントロールシステムの概要

れると、本来の処理を実行する前に、呼び出し元のサービスの識別子と、自分自身の識別子とを対にして、アクセスコントロールモジュールに問い合わせる。アクセスコントロールモジュールは、ACL の中から、問い合わせ内容と同一の識別子の対応を検索し、有れば当該サービスからデバイスへの呼び出しが許可され、デバイスインスタンスは本来の処理を実行する。許可されなかった場合は、例外を発生させるなどの処理を行い、本来の処理は行わない。

上記のような仕組みを導入することで、インスタンス単位でのアクセスコントロールが可能になり、例えばキッチンの Web カメラはセキュリティサービスからの制御を受け付けるが、寝室の Web カメラは受け付けない、などの細かな設定ができ、より利便性が増す。

3.2　Permission の動的な設定と反映

家庭によって存在する機器は異なり、またサービスの利用方法も異るため、各家庭の環境に合わせて Permission を設定する必要がある。このために提案手法では、上記のアクセスコントロールモジュールが持つ ACL を編集する GUI を提供する。

本提案方式では、ACL の内容を分かりやすくユーザに提示するために、デバイスインスタンスから UPnP の Friendly Name のようなユーザが理解できる名前を取り出す仕組みを持つ。またユーザは部屋の名前などを付与してより分かりやすい名前を独自に付けることもできる。この際に重要なことは、機器は固定的ではなく任意のタイミングで追加／削除されたり場所を移動したりするため、機器の固有識別情報などを活用して名前情報に一貫性を持たせることである。本提案方式では、機器の電源 OFF などにより、デバイスインスタンスが消滅した場合でも、ACL の内容を消さずに保存しておき、デバイスインスタンスが再び生成された場合には、ACL の内容が再び有効になるようにしている。

また ACL の編集にあたっては、その内容をリアルタイムにアクセスコントロールに反映しなくてはならない。提案方式では、デバイスインスタンスから問い合わせがあるたびに ACL の確認を行うため、変更の即時の反映が可能である。

3.3　Permission の要求情報の提示

上記のように ACL をユーザが GUI で設定するにあたっては、各サービスがどの種類の機器をどのような目的で使用するのかを提示することで、設定作業を効率化することができる。

このために本提案方式では、各サービスを実装するバンドルに、使用する機器と目的の情報が入った独自の Permission (InstancePermission) を設定する。InstancePermission は、name 部にサービスが要求するプロトコル名、インスタンスを特定するフィルタ情報、使用目的を持ち、ACL 設定 GUI は InstancePermission を読み出してユーザにデバイスの使用目的や割り当て可能なデバイスインスタンスの提示を行う。

これにより、サービスが利用できないデバイスを割り当ててしまうのを防ぎ、あらかじめ種類を絞ったデバイスの集合の中からユーザに選択させることが可能になる。

4．まとめ

家庭内で様々なサービスを快適に利用するためには、各サービスから共通して利用できるデバイスドライバと、デバイスドライバへのアクセスコントロールを行うことが重要である。我々が提案するプラットフォームは OSGi 準拠であり、OSGi ではアクセスコントロールに Java の Permission の仕組みが利用できる。

本提案方式では、インスタンス単位でのアクセスコントロールを導入し、デバイスインスタンスへの名前付け、および、サービスの要求する機器の絞込みと使用目的の提示によって使いやすい ACL 設定 GUI を実現できた。

本方式は複数のサービス間での競合を防ぐと共に、ユーザの意図に応じて個々のサービスがアクセスする機器を詳細に制限することができ、セキュリティの向上やプライバシーの保護に役立つものと思われる。

参考文献

[1]UPnP Forum, http://www.upnp.org/
[2]Echonet Consortium, http://www.echonet.gr.jp/
[3] OSGi, http://www.osgi.org/
[4] 小林英嗣, 小河原成哲, 依田育生, “ホームネットワークにおける複数サービスの統括的制御システム, ” 2003 電子情報通信学会信学技報, no.115, pp.25-30, Mar.2003
[5]Scot Oaks, Java セキュリティ, O'REILLY, pp.69-174, 2001
[6]Ross Anderson, 情報セキュリティ技術大全, 日経 BP, pp54-55, 2002