情報処理学会論文誌　コンシューマ・デバイス & システム　Vol.3　No.3　20–29 (July 2013)

コンシューマ・デバイス論文

サービスゲートウェイ向けECHONET Liteバンドルの開発

寺岡 秀敏[1,a)]　今井 光洋[1]　小坂 忠義[1]　奈良 祐樹[1]　小田 輝[1]

受付日 2012年12月14日, 採録日 2013年4月26日

概要：国内で公知な標準インタフェースとして推奨されたECHONET^{TM} Liteに対応したサービスゲートウェイ（SGW）向けのミドルウェアをOSGi^{TM}のバンドルとして開発した．設計にあたり，マルチサービスを実現するSGW上にミドルウェアを実装するにあたっての課題およびECHONET Lite規格の実装上の課題を検討した．検討した課題に基づいて，OSGiフレームワークの機能を最大限活用して，アプリケーション開発工数の低減，省リソースとアプリケーション停止時間最小化の両立，任意のECHONET Lite機器特定の容易化，下位通信層への柔軟な対応を実現するECHONET Liteバンドルの構成を提案する．さらに，提案する構成のバンドルを実装して評価システムを構築し，前記要件の観点から提案するECHONET Liteバンドルの有効性を評価した．

キーワード：Service Gateway，HEMS，ECHONET Lite，OSGi

Development of ECHONET Lite Bundle for Service Gateway

Hidetoshi Teraoka[1,a)]　Mitsuhiro Imai[1]　Tadayoshi Kosaka[1]　Masaki Nara[1]
Kagayaki Oda[1]

Received: December 14, 2012, Accepted: April 26, 2013

Abstract: In this paper ECHONET Lite Bundle based on OSGi is proposed. ECHONET Lite is recommended by The Ministry of Economy, Trade and Industry as the standard interface for connecting electric appliances and a home energy management system (HEMS) in the home. In the first part of this paper the problems to implement ECHONET Lite Bundle as a middleware of service gateway are clarified. In the second part, the software structure to solve the problems is proposed. Using functions of OSGi effectively, the proposed method achieves (a) reduction of application development cost, (b) resource-saving and minimizing stop time of applications, (c) tracing ECHONET Lite devices easily, and (d) attaching various low layer communication interfaces easily. Finally, we implement the proposed method and evaluated the effectiveness of the ECHONET Lite Bundle.

Keywords: Service Gateway, HEMS, ECHONET Lite, OSGi

1. はじめに

家庭内の様々な機器を接続してサービスを提供するホームネットワークシステムについて，様々な取り組みがなされてきた．たとえば，ユーザの利便性を向上させるホームオートメーション（HA）機能や，防犯などのセキュリティサービスなどが専用の端末を利用して提供されている．このような状況で，近年，ホームネットワークを活用して家庭のエネルギーマネジメントを行う，ホームエネルギーマネジメントシステム（HEMS）が注目されている．HEMSにおいて，ネットワークで接続される家電機器や住宅設備は家庭ごとに異なっており，機器接続のプロトコル，伝送メディアも多様である．このため，共通の手段で情報取得や制御が行えるような標準技術の重要性が改めて認識されている．そのような標準技術として，国内ではECHONET Lite [1]が公知な標準インタフェースとして推奨され，平成23年度「エネルギー管理システム導入促進事業費補助金（HEMS導入事業）」においても当該規格の搭載が補助対象機器の要件となっている [2]．

[1] 株式会社日立製作所
Hitachi Ltd., Yokohama, Kanagawa 244–0817, Japan
[a)] hidetoshi.teraoka.rf@hitachi.com

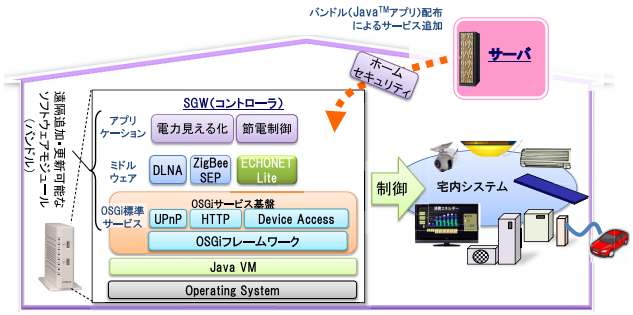

図 1　OSGi を利用したホームネットワークシステム概要
Fig. 1　Overview of Home Network System based on OSGi.

一方，これまで専用の端末上に実装されていた HA やセキュリティ，HEMS といった複数の機能を 1 つの端末（サービスゲートウェイ，以下 SGW）上に搭載するため，様々なサービスやデバイスに対応できるホーム ICT（Information and Communications Technology）実行基盤として OSGi™ が注目されている [3]．OSGi では，Java™ ベースのソフトウェアモジュール（バンドル）をネットワーク経由で動的に更新することができる．また，複数のバンドルが稼働しているときに，特定のバンドルだけを停止・再起動したり，更新したりすることが可能であり [4]，家庭ごとに環境や提供するサービスが異なっていても，必要なバンドルの組合せで適切なシステムを構築することが可能となる．**図 1** に，SGW に OSGi を利用したシステムの構成例を示す．ここでは，SGW は家庭内の各機器に制御指示を行うコントローラとなる．

そこで本研究では，国内の HEMS 標準プロトコルである ECHONET Lite に準拠したコントローラ用のミドルウェアを OSGi バンドルとして構成する方法について検討し，PC および SGW 上に実装して評価を行った．

2.　先行研究

ECHONET Lite 規格に先立ち標準化された ECHONET 規格では，ソフトウェア実装のための標準 API として，アプリケーション（AP）がミドルウェアを利用するための「基本 API 仕様」およびミドルウェアと下位通信層の間の「共通下位通信インタフェース仕様」が定義されており [5]，本仕様を参照した ECHONET 通信ミドルウェアバンドルが OSGi 上で開発されている [6]．また，OSGi 上で ECHONET バンドルを構成し PUCC（P2P Universal Computing Consortium）プロトコルと接続して ECHONET 機器の制御を行うシステムも提案されている [7]．これらの研究では，ECHONET を OSGi に適用する方法について検討されている．しかし，これらの研究はあくまで ECHONET を対象としており，ECHONET と ECHONET Lite の相違点については考慮されていない．たとえば，ECHONET では定義されている下位通信層が ECHONET Lite では定義されていないが，その対応方法は検討されていない．

さらに，ECHONET Lite 規格を Java 言語で実装したミドルウェアソフトとして，商用ベースの製品 [8] やオープンソースで公開されている実装 [9] がある．SGW 上でソフトウェアを実装する場合，リソースの制限やサービス停止時間といった課題があるが，これらの実装では，それらについては十分に考慮されていない．

そこで，本研究では，これらの先行研究をふまえ，OSGi 上の ECHONET Lite に準拠したコントローラ用のミドルウェアとして，ECHONET Lite ミドルウェアバンドルの検討および評価を行った．

3.　SGW の ECHONET Lite バンドルの要件

本稿では，以下の 4 つの要件を満足するソフトウェアを開発することを目標とする．

R1：AP の開発工数を低減できること．
R2：省リソースと AP の停止時間最小化を両立できること．
R3：機器の識別に必要な AP の実装量を低減できること．
R4：様々な下位通信層に柔軟に対応できること．

以下にこれらの要件に関する考察を述べる．

3.1　SGW 上のミドルウェア実装における要件

R1，R2 は，SGW 上にミドルウェアを実装するときの要件である．以下，各要件の詳細について述べる．

R1：AP の開発工数を低減できること

本提案のミドルウェアバンドルを用いる AP の主要機能は，ECONET Lite 機器を操作（制御および情報取得）して，デマンドレスポンスや見える化のようなサービスを提供することである．AP 開発者にとってはサービス提供以外の機能の開発工数をできるだけ低減できることが重要である．そのため，本提案のミドルウェアバンドルにおいては，複数の AP に共通的に利用される機器発見などの機能を適切に構成し，AP の開発工数をできるだけ低減できるようすることが課題となる．本研究では，従来 ECHONET で定義された Java API [5] を利用する場合と比較して AP の開発工数を低減できることを目標とする．

R2：省リソースと AP 停止時間最小化を両立できること

SGW 上のソフトウェア実装にあたっては，省リソースであることと，搭載された AP が提供するサービスを停止させないようすることが重要である．これは，SGW は，組込み機器として構成され，省リソースが要求される場合が多いこと，さらに，マルチサービスを実現する SGW においては，AP の要求により SGW そのものや ECHONET Lite ミドルウェアの無停止を求められるケースもあるためである．

以上から，ECHONET Lite バンドルでは，省リソースかつ AP の停止時間を最小化できる構成を実現することが課題となる．本研究では，ECHONET Lite の Java 実装 [9]

表 1　ECHONET Lite における機器識別のための情報
Table 1　Information for tracing ECHONET Lite devices.

識別情報	内容	課題
下位通信層における識別情報	IP アドレスまたは MAC アドレス	下位通信層のプロトコル・接続方法に依存
識別番号	オブジェクトをドメイン内で一意に識別するための情報（8〜16byte）	v1.01 では，0x00(未設定) も設定可能
個体識別情報	ドメイン内で各ノードを一意に識別可能とし，かつ機器の移動（サブネットの変更など）後も常に同一ノードは不変なものとして取扱い可能とするための情報（2byte）．初期値は任意（乱数など）．コントローラから変更可能	コントローラが複数存在する場合の競合回避方法などのガイドライン無し
メーカコード +製造番号	機器の製造者が一意であることを保証する番号	製造番号はオプション

と比較して省リソースな構成で停止時間を 0 秒にできることを目標とする．

3.2　ECHONET Lite 規格の実装上の要件

R3，R4 は，ECHONET Lite 規格を実装するにあたって，規格の課題に対応するための要件である．以下に，各要件の詳細について述べる．

R3：機器の識別に必要な AP の実装量を低減できること

ECHONET Lite を用いて接続された機器をユーザが利用するにあたって，プロトコルや下位通信層で定められた識別情報のままではどの機器がどれにあたるか分かりにくい．そのため，たとえば「リビングのエアコン」といった機器名称などの人間に分かりやすい識別情報と紐づけ，ネットワークの構成や機器のアドレス情報などが変更されてもそのまま対応づけができる必要がある．たとえば，UPnP（Universal Plug and Play）では UUID（Universally Unique Identifier）と機器名称を紐づけておけば，IP アドレスが変わっても接続された機器が一意に識別できる．一方，ECHONET Lite 規格では機器を永続的に一意に識別するためのいくつかの方法は提供されているが，どの情報を利用可能にするかはメーカの実装依存となっており，すべての ECHONET Lite 機器で統一的に利用できない．**表 1** に ECHONET Lite 規格において機器識別に利用できる情報と当該情報を利用するにあたっての課題を示す．文献 [10] では本課題について検討し，個体識別番号あるいはメーカコードと製造番号の組合せのどちらかを使うと一意に識別（トレース）可能という考察がなされている．しかし実際はいずれの方法も課題があり，すべての機器で統一的に利用できる方法がないことから，コントローラ用のミドルウェアは，AP が ECHONET Lite 機器を容易に一意に識別する仕組みを提供することが課題となる．

本研究では，Java API [5] に基づく実装と比較して，AP

図 2　2 つのサブネットにまたがる HEMS の例
Fig. 2　Example of HEMS that has 2 subnets.

が操作対象の機器選択のために必要な実装量を低減できるような機能を提供することを目標とする．

R4：様々な下位通信層に柔軟に対応できること

ECHONET Lite ではプロトコルの軽量化とグローバルな標準仕様を利用したいという要望に基づいて OSI 参照通信レイヤ 4 層以下の下位通信層は規格範囲外とされた．しかし一方で下位通信層に規定がないことは，伝送メディアや下位通信層の実装の相違による相互接続性の問題につながる可能性がある．ECHONET Lite の下位通信層の候補としては UDP/IP/Ethernet 以外にも複数の通信層（920 MHz 帯無線を利用する ZigBee^{TM} IP など）が想定されている [11]．ECHONET Lite を使用するシステムでどの下位通信層を採用するかはユーザまたは機器ベンダの選択によるため，家ごと，機器ごとに多様な構成となってしまう．また，**図 2** に示すように SGW は複数の下位通信層にまたがる機器と接続するというユースケースも考えられる．そのため，様々な機器と接続する可能性のある SGW は，複数の下位通信層に柔軟に対応できる構成である必要がある．ここでいう柔軟とは，下位通信層の入れ替えまたは追加のために，ECHONET Lite バンドルの改変が不要であること，および，ECHONET Lite バンドルを停止することなく下位通信層の追加または入れ替えができることを意味する．下位通信処理部の実装方法によっては，家ごとの環境に合わせて複数のバージョンのバンドルが必要となり，開発効率や保守性に問題が生じる．また，新しい機器を導入し，新しい下位通信層に追加対応する場合にはバンドルの更新が必要になり，その間 AP を停止させてしまう可能性がある．そこで，本研究では，バンドルを改変せず，かつバンドルおよび AP を停止せずに下位通信層の追加や変更が可能な構成を実現することを目標とする．

4.　ECHONET Lite バンドルの設計

ECHONET Lite バンドルおよび関連するバンドルについて，**図 3** に示す構成を提案する．

本提案の構成では，AP は機器操作 I/F 抽象化バンドル，ECHONET Lite I/F 実装ドライババンドルを介して，ECHONET Lite バンドルを利用する．機器操作 I/F 抽象化バンドルおよび ECHONET Lite I/F 実装ドライババンドルは，機器オブジェクトごとに準備し，機器オブジェクトのプロパティ定義はそれぞれの ECHONET Lite I/F 実装ドライババンドルが保持する．プロパティ定義とは，

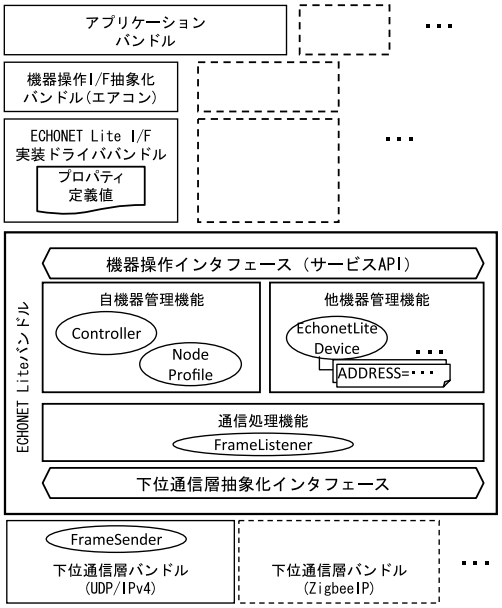

図 3　ECHONET Lite バンドルとその周辺構成

Fig. 3　Structure of ECHONET Lite bundle and peripheral bundles.

ECHONET Lite において，具体的に何をどう制御するかをやりとりするための定義情報である．ECHONET Lite バンドルは，自機器管理機能，他機器管理機能，通信処理機能から構成する．自機器管理機能では，コントローラの機器オブジェクトおよびノードプロファイルオブジェクトのインスタンスを管理する．また，他機器管理機能は，発見した機器オブジェクトのインスタンスを管理する．これらのインスタンスを利用する場合は，機器操作インタフェースを利用する．ネットワークへの電文送出と受信を行う下位通信層バンドルは，下位通信層の種類ごとに準備し，下位通信層に依存する処理を実装する．電文の送受信は，ECHONET Lite バンドルの通信処理部の受信用インスタンスと下位通信層バンドルの送信用インスタンスが，下位通信層抽象化インタフェースを介して行う．以下に，前述の要件と，提案するバンドル構成についての詳細を述べる．

4.1　AP 開発工数低減への対応

AP 開発工数低減の観点から，ECHONET Lite バンドルが提供すべき機能として，他機器管理機能，自機器管理機能，通信処理機能を提案する．

本提案のバンドルを搭載する SGW の基本的なユースケースとしては以下のようなステップが考えられる．

(a) ネットワーク上に接続された機器を探索・発見する．
(b) 発見した機器から操作対象を選択する．
(c) 選択機器に対して，操作要求を発行する．
(d) 要求に応じて ECHONET Lite 電文を構成し，ネットワークに送出する．
(e) 機器からの応答や通知を受信し，電文を解析する．
(f) 前記解析結果に基づいて要求の成否を判定し，結果に応じた処理を行う．
(g) 他機器からの ECHONET Lite 電文による要求を受信し，応答する．

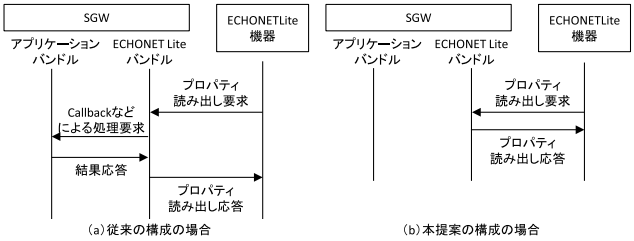

図 4　要求受信時のシーケンス比較

Fig. 4　Comparison of sequence at receiving a request.

AP は，上記 (a) から (f) のステップが実行できれば，「使用電力をスマートメータから取得し，表示する」ことや「取得した室温や使用電力に基づいてエアコンの設定温度を変更する」などのサービス実現が可能になる．

上記 (a) から (f) のステップのうち，(b)，(c)，(f) のステップ，すなわち，ネットワークに接続された機器のうち，どれを選択してどのような要求を行い，その結果に応じてどのような処理を行うかについては，AP ごとに異なる．一方で，(a)，(d)，(e) は複数の AP が共通的に必要とする機能である．このうち，(d)，(e) はプロトコルを実装したミドルウェアが通常提供すべき機能であるため，本提案においても通信処理機能として提供する．また，(a) に関し，ECHONET Lite 規格は，プラグアンドプレイにより機器の発見ができるという特徴を有している．しかし，機器発見の具体的な手順については規定されていないため AP 開発者が個別に設計および実装を行う必要がある．そのため，本提案のバンドルにおいて，手順 (a) の処理を他機器管理機能として提供する．

また，本提案のバンドルを搭載する SWG はコントローラであるため，他の機器から情報を取得されることや制御されること（前記 (g) に相当）は想定する必要がない場合が多い．一方，ECHONET コンソーシアムによる規格適合性認証 [12] では，「送信電文は適切か」「受信電文を正しく解釈し，適切に自機器・自ノードオブジェクトの処理ができているか」という被制御機器としての項目が中心となっている．これを実現するためには，自機器・自ノードオブジェクトの管理が必要になる．このような自機器・自ノードオブジェクトの管理はミドルウェアの範囲外として，AP 側での対応が必要とされる場合が多い [5]．そこで，本提案のバンドルでは，(g) に対応し，被制御機器としての役割を担う自機器管理機能を提供する．**図 4** に電文受信時のシーケンスの比較を示す．

以下に，各機能の詳細を述べる．

通信処理機能

(d)，(e) に相当する電文生成や解釈といったプロトコル処理に加え，不正な電文を発行しないためのチェックや，

不正な電文を受信した場合の異常処理などを行う．

他機器管理機能

前述の (a) ECHONET Lite 機器の探索・発見を行う．また，発見した機器を AP から利用可能な OSGi サービス（EchonetLiteDevice サービス）として OSGi フレームワークに登録する（**図 5**）．さらに，応答状態を監視し，応答がなくなった機器は OSGi サービスから削除する．AP は，EchonetLiteDevice サービスが提供する**機器操作インタフェース（サービス API）**を利用することで，発見した機器を操作したり，イベントを受信したりすることができる．ECHONET Lite バンドルは，機器操作インタフェースによる AP からの要求（API コール）に従ってネットワークに電文を送出し，対応する応答を API の戻り値やイベントとして AP に通知する．

以上のように，本提案では SGW のプラットフォームとして，プラグアンドプレイによる機器発見および機器管理機能や，発見した機器に対する操作インタフェースを AP に提供する．これにより，AP は所望の機器を制御するためには OSGi サービスの検索・取得のみ実装すればよく，独自の制御アルゴリズムなど，ユーザにサービスを提供するための機能の実装に注力することができると考える．

自機器管理機能

コントローラとして搭載が必要な機器オブジェクトスーパークラスおよびノードプロファイルクラスの必須プロパティを管理し，読み出し要求に対する応答，書き込み要求に対する管理値の更新や状態変更通知を行い，被制御機器としての (g) に相当する機能を提供する．これは，前述のとおり AP は接続された機器の操作が主眼であるため，自機器・自ノードオブジェクトの管理を最小限にできることが望ましいためである．本機能により，AP は自機器について特別な管理をしなくても，ECHONET Lite 規格でいうコントローラとしての最小限の要求を満足し，サブネット内の他機器からは，1 つの ECHONET Lite「コントローラ」機器として認識される．本機能により，AP 開発者は ECHONET Lite コントローラ機器としての認証取得のための工数を低減できる，と考える．

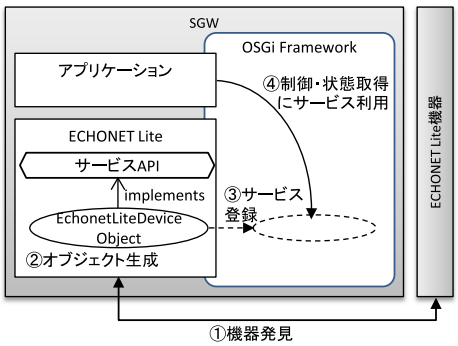

図 5　OSGi を利用した機器登録・利用方法

Fig. 5　Device search and use method using OSGi.

4.2　省リソースと AP 停止時間の最小化への対応

本提案の構成では，省リソース化と AP 停止時間の短縮のため，ECOHNET Lite で定義された機器の ECHONET プロパティ定義の値は保持せず，外部でバンドル化して保持することとする．

ECHONET Lite では多様な機器操作のための情報が定義されており，これは他の規格と比較した場合の ECHONET 規格の強みである．一方で，これらを最初からフル実装した場合，定義値の数だけプログラムサイズが大きくなってしまい，大きなリソースが必要となる．これを解決し，省リソースを実現する 1 つの方法として，必要最小限の機器オブジェクト定義のみ実装し，必要に応じてソフトウェアの更新を行って順次追加する方法がある．しかしこの場合，SGW またはソフトウェアの再起動が必要となり，無停止が要件である AP には都合が悪い．

また，前述の機器操作インタフェースはこの定義値をどこで管理するかによって 2 通りの定義方法が考えられる．すなわち，ミドルウェアのレイヤで定義値を管理して操作ごとの API（setTemperature() など）を定義し AP は ECHONET 規格の定義値を意識しないようにする方法**図 6** (a) と，AP が ECHONET 規格を意識し定義値を管理してプロパティの値を渡す全操作に共通の API（setProperty() など）を定義する方法図 6 (b) である．

表 2 に，省リソースと AP 停止時間の観点から各方法を比較した結果を示す．

比較の結果，省リソースと AP 停止時間の最小化を両立させるためには，プロパティの定義値をバンドル外で管理し，必要に応じて順次追加していく方法がよいと考えられる．ただし，このままでは，更新対象の AP は更新時に停止する必要があることと，AP 開発者は ECHONET Lite プロトコルの詳細定義を意識する必要が生じてしまうとい

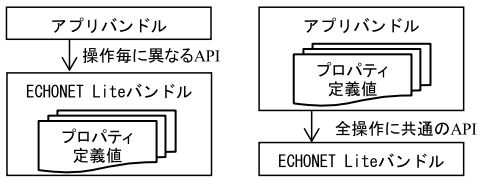

図 6　機器操作インタフェースの提供方法

Fig. 6　Implementation methods of service API.

表 2　機器操作インタフェース定義方法の比較

Table 2　Comparison of implementation methods of device interface.

実装方法		省リソース	停止時間
(a)	フル実装	×	○
	順次追加	○	×
(b)	フル実装	×	○
	順次追加	○	△*1

*1 更新対象以外の AP には影響しない．

う課題がある．

そこで，本バンドルを利用する AP 周辺の構成として，図 7 の (b)’ の構成を提案する．

具体的には，AP 向けインタフェースとして別途操作内容を抽象化したインタフェース（機器操作 I/F 抽象化バンドル）を定義し，これを実装した機器操作 I/F ドライババンドルを用意する．プロパティ定義値をこのドライババンドルで保持し，EchonetLiteDevice サービスを呼び出すことにより，開発者は ECHONET Lite 規格の詳細を知らなくても AP を開発できるようになる．さらに EchonetLiteDevice サービスを Device Access [13] に準拠させておく．これによって，新しい種別の機器を ECHONET Lite バンドルが発見したときに自動的に対応するドライバを検索，およびダウンロードできる仕組みが容易に構成できる．Device Access サービスは，機器を表現する Device サービスと，機器に接続するための実装を担う Driver サービスなどから構成され，これらを結合する機能を提供する．また，適切な Driver サービスがフレームワーク内に存在しない場合，サーバからこれを取得する機能も提供する．ここでは，EchonetLiteDevice サービスを Device サービス，機器操作 I/F ドライバを Driver サービスとすることで，新規機器に対応するドライバの自動取得を実現できる．

以上の構成とすることにより，各家庭の機器状況に応じた最小限の構成とし，省リソースを実現しつつ，機器追加時の AP 停止時間を最小限にすることができると考える．

4.3　機器の識別に必要な AP の実装量低減への対応

本提案の構成では，機器識別のために必要な AP の実装量低減を実現するため，AP が機器の識別情報として OSGi においてサービス検索に用いるフィルタ文字列を利用し，提案の ECHONET Lite バンドルは検索に必要な情報をあらかじめ機器から取得しておく構成とする．

前述のとおり，ECHONET Lite 規格では，機器を一意に識別する統一的な方法が規定されていない．このため，AP は，機器を一意に識別に機器ごとに異なる情報を用いる必要が生じる可能性がある．たとえば，機器 A は識別番号での識別を想定して，識別番号に有意な値が設定されるが，機器 B は MAC アドレスでの識別を想定して識別番号は 0 であるといったケースが考えられる．この場合，ユーザ（または設置者）は機器外装に印刷された識別番号や MAC アドレスを使って機器登録を行い，ユーザフレンドリな情報（機器名称など）との紐づけを行うというユースケースが考えられる．このようなユースケースを実現するためには AP は，ユーザが登録した情報に応じて機器から必要な情報を取得し，かつそれをなんらかのデータベース（DB）で管理する必要が生じる．この際，機器を識別するための情報が一意ではないため，情報登録や検索の方法が複雑になってしまう可能性がある．

一方，OSGi フレームワークでは，登録するサービスにサービスプロパティと呼ばれる付加情報を設定しておくことで，登録されたサービスの検索および追跡にフィルタ機能を利用することができる．このフィルタは LDAP フィルタ [14] と呼ばれるフィルタであり，複数の条件を含む複雑な検索式を構成することができる．そこで，本バンドルを利用する AP が，EchonetLiteDevice サービスを取得する際に，このフィルタ機能を活用することができるように機器を一意に識別するために利用される可能性のある情報をあらかじめ取得し，サービスプロパティとして登録しておくこととする（図 8）．

このように ECHONET Lite 機器の情報を OSGi のサービスとして管理することで，OSGi フレームワークが提供するサービスレジストリの仕組みを機器管理 DB として利用できる．さらに，機器管理 DB のインデックスとして ECHONET プロパティ情報を利用できるようにすることで，AP が機器を容易に識別できるようになると考えられる．すなわち，AP は，機器ごとに異なる可能性のある情報の取得や管理・比較などの処理を実装する必要はなくなり，機器を表す ECHONET Lite サービスを取得するためのフィルタ文字列を識別情報として管理するだけでよい．

4.4　様々な下位通信層への柔軟な対応

ECHONET Lite バンドルでは下位通信層との結合を疎なものにするため，下位通信層とのインタフェースとして

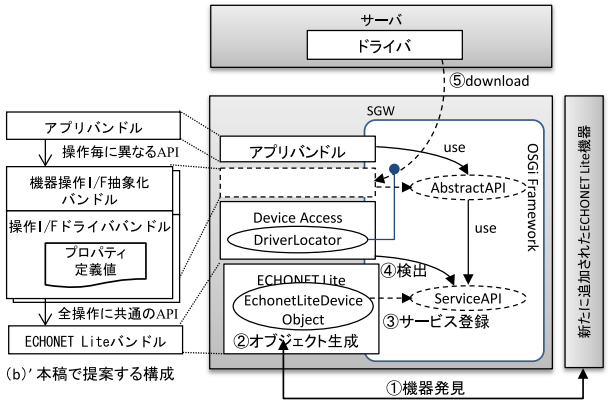

図 7　機器操作インタフェースの実装方法とそれを利用した抽象化インタフェースの構成

Fig. 7　Proposed implementation method of service API and the structure of abstract interface using that.

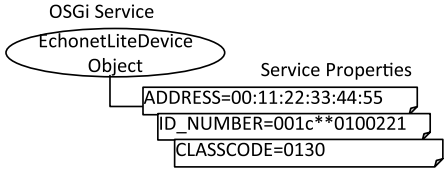

図 8　EchonetLiteDevice サービスのプロパティ構成

Fig. 8　The structure of properties of EchonetLiteDevice service.

下位通信層の構成に依存しない抽象的なネットワークインタフェース（**下位通信層抽象化インタフェース**）のみ規定する．すなわち，下位通信層を実現するソフトウェアはECHONET Lite バンドルの外部で前記インタフェースを実装したバンドル（下位通信層バンドル）とする構成をとる（**図 9**）．通信用リソース（ソケットや COM ポート）の確保，各種プロトコルやデバイス実装に依存したパケット化などは，下位通信層バンドルにおいて対応する．これにより，下位通信層と ECONET Lite ミドルウェアバンドルを切り離し，異なる物理層のネットワークインタフェースが複数存在する場合も対応可能にする．本構成によって，下位通信層を変更・追加する場合でも ECHONET Lite バンドルの修正を行うことなく，新しく定義された下位通信層に対応することができる．

さらに，OSGi のバンドル追跡機能を利用して，下位通信バンドルの新規登録を追跡することで，ECHONET Lite バンドルを停止することなく，新しく追加された下位通信バンドルを利用して新しい下位通信層に接続された機器を発見・制御することが可能となると考えられる．

また，新規追加されたネットワークに対応した下位通信層バンドルをサーバからダウンロードして追加することも可能となる．この場合のシーケンス例を**図 10** に示す．たとえば，SGW の USB ポートに ZigBee IP スタックを搭載したドングルを挿入した際，対応した下位通信バンドルをダウンロードして実行する仕組みを作り込んでおくことで，以降，当該インタフェース（USB-ZigBee IP）を経由した機器の操作が可能になる．

以上の構成により，本提案のバンドルを更新せず異なる下位通信層を選択・追加できるとともに，SGW のように連続運転が要求される機器においても，機能を停止することなく新しい下位通信層を導入することが可能になる．

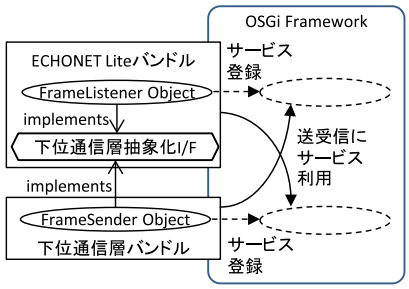

図 9　下位通信層バンドル構成

Fig. 9　The structure of low layer communication interface bundle.

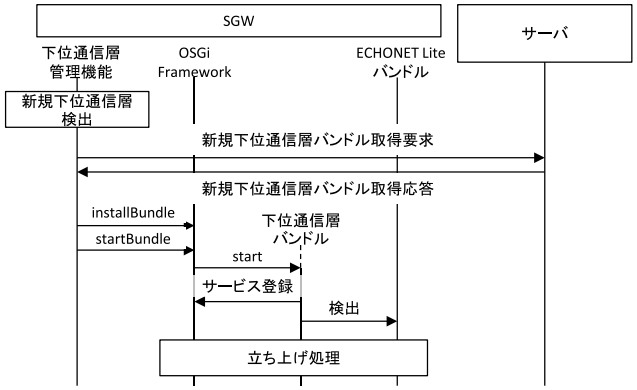

図 10　新規インタフェース追加時のシーケンス

Fig. 10　Sequence of attaching a new low layer communication interface.

5.　実装と評価

4 章で提案した ECHONET Lite バンドルを PC および SGW 実機上で試作し，評価を行った．**表 3** に評価に用いた機器の主要諸元を示す．また，**図 11** に評価用システムの概要を示す．

実装した試作ソフトウェアを ECHONET コンソーシアムの主催するプラグフェストにおいて他社実装と接続した．ECHONET Lite バンドル単体で機器発見し，OSGi フレームワークに接続された機器のサービスが登録されること，および取得した ECHONET プロパティが前記サービスのサービスプロパティとして登録されていることを確認した．これにより，実装したバンドルのプロトコル実装が相互接続性を確保できていることが確認できた．なお，PC と SGW 上では，下位通信層バンドルの一部を除いて実装の変更なくバンドルを動作させることができた．

5.1　SGW 上のミドルウェア実装における要件の評価

R1：AP の開発工数を低減できること

表 4 に，AP の開発工数低減を目的として ECHONET

表 3　試作機器仕様

Table 3　Specifications of prototype devices.

項目	SGW	PC
CPU	650MHz	2.7GHz
Memory	RAM:512MB, ROM:128MB	RAM:8GB HDD:280GB
LAN interface	4port(LAN), 1port(WAN)	1port
USB interface	2port	4port
OS	Linux	Windows7 64bit
Application Platform	JavaME (SuperJ Engine™)/ OSGi Framework (SuperJ Engine Framework™)	J2SE6 OSGi Framework (SuperJ Engine Framework™)

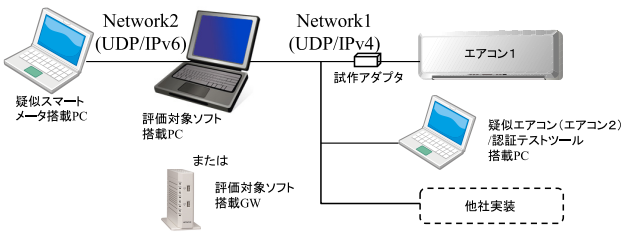

図 11　評価システム概要

Fig. 11　Overview of evaluation system.

表 4　共通機能の工数概算
Table 4　Production cost estimation of common functions.

機能	Step 数	工数(人月)
他機器管理機能	480	1.1
自機器管理機能	570	1.3

表 5　ECHONET Lite における自己認証試験項目数
Table 5　Number of items of self certification.

	全体	必須	受信	送信	
				自	他
フレーム処理	31	30	15	15	
オブジェクト処理	98	58	52	2	4
合計	149	88	68	20	

Lite バンドルに実装した他機器管理機能，自機器管理機能の概算ステップ数とステップ数から概算した工数を示す．ここに示すとおり，前述の構成とすることで，AP ごとにこれらの機能の実装をする必要がなくなり，文献 [5] のような AP にこれらの機能を実装する場合と比較して 1AP あたり約 2.4 人月の工数を低減できる．

また，自機器管理機能を利用することで，実装工数に加えて適合性認証に係る工数も低減できることを確認するため，認証項目についての評価を行った．**表 5** に ECHONET Lite における規格適合性認証の自己認証の項目数を示す．コントローラとして適合性認証を取得するためには，少なくとも 88 の必須項目の確認を行う必要がある．このうち，ほぼ 2/3 の項目が ECHONET Lite フレーム受信時の要求仕様に関するものである．前述のように本バンドルでは自機器管理機能を備えているため，AP は改めて何らかの実装を行うことなく，大半の項目に適合させることができる．電文送信に関しても，不正な電文を送信しない機能を提供することから，AP に依存して（AP 追加・変更時に）再テストが必要な項目は他機器への SET 要求送信に関する 4 つとなる．これらは，他機器オブジェクトに送信するプロパティの整合性確認および不正チェックを行う項目である．

他機器オブジェクトのプロパティ値の管理は，バンドル外で行うこととしたため，これらの認証項目については，本提案のバンドルでは担保できない．しかし，自機器管理機能を利用することによって 88 項目中 4 項目を除いて，バンドルの提供する機能で対応できるため，本バンドルを利用した場合，利用しない場合と比較して適合性認証に係る 84 項目分の開発工数を低減できる．

以上から，本提案の構成とすることで接続機器の管理などを AP が実装する構成 [5] と比較して，AP 開発工数を低減するという要件 R1 は満足できたといえる．

R2：省リソースと AP の停止時間最小化を両立できること

省リソースな実装であることを評価するため，実行モジュールのサイズを比較した．また，AP 停止時間を評価した．**表 6** に結果を示す．

表 6　モジュールサイズと AP 停止時間
Table 6　Size of software module and stop time of AP.

機器オブジェクト定義実装方法	ROM サイズ (KB)	AP 停止時間(秒)
(A)すべてを予め実装*1（従来の Java 実装）	× (約 1200:[9])	○ (0)
(B)必要に応じ追加*2（従来の Java 実装）	○ (約 95:[9])	× (1.7〜86)
(C)必要に応じ追加*2（本提案の実装）	○ (約 162)	○ (0)

*1 全機器オブジェクトを含む場合
*2 家庭用エアコンオブジェクトのみの場合

すべての機器オブジェクトおよびプロパティをあらかじめ実装している文献 [9] の構成 (A) では，モジュールサイズが約 1,200 KB と大きなものとなっている．一方，本提案 (C) では，家庭用エアコンクラスのみを含む場合の構成で，約 162 KB でモジュールを構成でき，省リソースであるといえる．文献 [9] において，対応するオブジェクトを家庭用エアコンのみにしてモジュールを構成した場合 (B)，約 95 KB と省リソースの構成にできるが，Java 実装においてソフトウェアを更新する際にはミドルウェアの停止が必要になるという課題が残る．

そこで，次に，従来の Java 実装 (B) において機能（機器オブジェクト定義）追加のときの AP 停止時間について見積もるため，システムの再起動に要する時間および，バンドルの更新（ここでは，ECHONET Lite バンドルが停止直前の状変通知を発行してから，再起動してインスタンス通知を発行するまでの）時間を計測した．その結果，システムの再起動に要する時間は約 86 秒，バンドルの更新に必要な時間は，1.7 秒であった．これらの結果から，通常の Java 実装でソフトウェアを更新するときの AP 停止時間は，バンドル更新時間 1.7 秒からシステム再起動 86 秒の範囲の時間と評価した．一方で，バンドルを追加するだけの (C) の場合，AP 停止時間は 0 秒で，新しい機能を追加することが可能であり，従来の Java 実装と比較して AP 停止時間の短縮を実現できる．

以上のとおり，提案の構成 (C) とすることで，従来の Java 実装 (A)，(B) と比較して，省リソースと AP 停止時間の最小化を両立させるという要件 R2 は満足できたといえる．

5.2　ECHONET Lite 規格の実装上の要件に対する評価

評価システムを利用し，以下のシナリオで要件 R3 と R4 についての評価を実施した．なお，評価用 GW は試作機で IPv6 未対応のため，評価は PC 上で実施した．

R3：機器の識別に必要な AP の実装量を低減できること

本シナリオにおいて，機器を特定する情報として異なる

<前提条件>
・ユーザ宅の既存ネットワーク 1 にはエアコン 1，エアコン 2 が接続
・エアコン 1，2 は DHCP でアドレスを取得
・SGW とエアコン 1，2 は UDP/IPv4 で通信
<シナリオ>
①ユーザは 2 つのエアコンを区別しやすいように名称を登録
　エアコン 1：リビングエアコン
　エアコン 2：寝室エアコン
②登録にあたって，ユーザが知ることができる情報
　エアコン 1：MAC アドレス
　エアコン 2：識別番号
③SGW 上の AP は，エアコンから室温を取得し，前記名称とともにリアルタイム(1 秒更新)でユーザに提示.
④ユーザ宅に新規にスマートメータ(SM)が導入され，新規ネットワーク（UDP/IPv6）に接続
⑤SGW は SM からの電力量をリアルタイムでユーザに提示.

表 7　OSGi のフィルタを利用した機器管理テーブル
Table 7　Device Management Table using OSGi Filter.

機器名称	検索フィルタ
リビングエアコン	(ADDRESS=00:11:22:33:44:55)
寝室エアコン	(ID_NUMBER=FE00000100****03)

識別情報を登録した 2 つの機器を正しく識別し，情報を取得できた．**表 7** に本シナリオにおける AP の機器管理テーブル例を示す．このようなフィルタ文字列を用いて機器操作サービスを取得することで，利用する識別情報が異なる場合でも，OSGi フレームワークからサービスを取得する手順に従って，10 ステップ程度の実装のみを行えば目的とする機器の識別・選択ができることを確認した．

本提案の機能を用いない文献 [5] のようなインタフェースの場合，AP は個別に機器ごとに異なる可能性のある識別情報の取得処理や管理を行わなければならない．これらの実装量は，本提案の他機器管理機能と同程度と仮定すると，480 ステップ程度となる．また，OSGi のフィルタ機能は約 1200 ステップで実装されている．一方で，本提案の手法では前述の機能を利用することによって，AP は OSGi を利用する際は必ず実装するサービス取得処理と簡単な識別情報の管理のみ実装すればよい．すなわち，前述の他機器管理機能およびフィルタ機能のすべてあるいはその一部を実装する必要がある従来の手法と比較して，少ない実装量で機器の識別ができる．以上から，文献 [5] に基づく実装と比較して，機器選択のために必要な実装量を低減できる機能を提供するという要件 R3 は満足できたといえる．

R4：様々な下位通信層に柔軟に対応できること

シナリオの④では，手動で IPv6 に対応した下位通信層バンドルを新規追加し，すでに接続されたエアコンからの情報取得が途切れないことを確認した．すなわち，本提案のバンドルを改変する追加工数 0 かつ，AP の停止時間 0 秒で，新しい下位通信層に対応できた．これにより，複数の異なる下位通信層が存在する環境でも ECHONET Lite バンドルを改変することなく，かつバンドル，AP を停止することなく下位通信層を追加する構成を実現するという要件 R4 は満足できたといえる．

6. おわりに

本研究では，国内の HEMS 標準プロトコルである ECHONET Lite に準拠したコントローラ用の OSGi バンドルについて検討し，PC および SGW 上に実装して評価を行った．設計にあたっては，マルチ AP を実現する SGW 上のミドルウェアとして実装するための課題および規格実装上の課題に着目して 4 つの要件を抽出し，これを基に ECHONET Lite バンドルおよびその周辺バンドルの構成を提案した．その際，OSGi フレームワークの持つ機能を最大限利用できるよう考慮した．その後，SGW 上に設計したバンドルを実装，評価し，提案する構成の有効性を確認できた．

今後は，本提案のバンドル構成におけるセキュア通信の実現の検討および，ECHONET Lite 機器と連携したアプリケーションの開発を行っていく．

参考文献

[1] ECHONET コンソーシアム ECHONET Lite 規格書 Ver1.01，入手先 ⟨http://www.echonet.gr.jp/spec/spec_v101_lite.htm⟩（参照 2012-12-13）.
[2] 平成 23 年度エネルギー管理システム導入促進事業（HEMS 導入事業）—対象機器の公募— 公募要領，入手先 ⟨http://sii.or.jp/hems/file/koubo.pdf⟩（参照 2012-12-13）.
[3] 丹　康雄：ホームネットワーク（OSGi，ECHONET）モデルに基づく家庭内エネルギーマネジメント，情報処理学会誌，Vol.51, No.8, pp.959–965 (2010).
[4] OSGi Alliance: OSGi Service Platform Core Specification, available from ⟨http://www.osgi.org/⟩ (accessed 2012-12-13).
[5] ECHONET コンソーシアム：ECHONET 規格書 Version 3.21，入手先 ⟨http://www.echonet.gr.jp⟩（参照 2012-12-13）.
[6] 大和ハウス工業株式会社：平成 21 年度スマートハウス実証プロジェクト報告書 第 2 章 テーマ 2-1：マッシュアップを促進するホームサーバ向け統合 API の開発実証およびテーマ 3-1：マルチベンダによる家電・設備機器統合コントロールシステムの開発 (Mar. 2010), 入手先 ⟨http://www.jipdec.or.jp/dupc/forum/eships/results/doc/h21project_report1-2.pdf⟩.
[7] Fukushima, K., Tanaka, Y., Kato, H. and Ishikawa, N.: Home Network System for Gas Appliances Using PUCC Technologies, *7th Consumer Communications and Networking Conference* (*CCNC*), pp.1–5, 9–12, IEEE (Jan. 2010).
[8] 日新システムズ EW-ENET Lite, 入手先 ⟨http://www.conss.co.jp/p-org/enetlite.html⟩（参照 2012-12-13）.
[9] SonyCSL: OpenECHO, available from ⟨https://github.com/SonyCSL/OpenECHO⟩ (accessed 2012-12-13).
[10] 宮本善則ほか：ECHONET Lite による蓄電池管理システムの開発，日本学術振興会産学協力研究委員会第 31 回インターネット技術第 163 委員会研究会.
[11] TTC TR-1043，ホームネットワーク通信インタフェース実装ガイドライン.
[12] ECHONET コンソーシアム：ECHONET 機器認証試

験仕様書，ECHONET Lite 規格 Ver.1.0*用，入手先 ⟨http://www.echonet.gr.jp⟩.
[13] OSGi Alliance: OSGi Service Platform Release4 Device Access Specification Version1.1, available from ⟨http://www.osgi.org/⟩ (accessed 2012-12-13).
[14] RFC1960, A String Representation of LDAP Search Filters, available from ⟨http://www.ietf.org/rfc/rfc1960.txt⟩ (accessed 2012-12-13).

1. ECHONET は，ECHONET コンソーシアムの登録商標です．
2. OSGi は，米国 OSGi Alliance の登録商標です．
3. Java は，Oracle Corporation およびその子会社，関連会社の米国およびその他の国における登録商標です．
4. ZigBee は，ZigBee Alliance, Inc. の登録商標です．
5. Linux は，Linux Torvals 氏の日本およびその他の国における登録商標または商標です．
6. Windows は米国 Microsoft Corporation の米国およびその他の国における登録商標です．
7. SuperJ Engine および SuperJ Engine Framework は株式会社日立ソリューションズの登録商標です．

寺岡 秀敏

1976 年生．2002 年京都大学大学院工学研究科修士課程修了．同年日立製作所入社．サービスゲートウェイおよび需要家向け Energy Management System に関連する研究に従事．

今井 光洋

1977 年生．2002 年慶應義塾大学大学院理工学研究科前期博士課程修了．同年日立製作所入社．サービスゲートウェイおよび需要家向け Energy Management System に関連する研究に従事．電子情報通信学会会員．

小坂 忠義

1967 年生．1993 年名古屋大学大学院工学研究科修士課程修了．同年富士通研究所入社．2007 年日立製作所入社．需要家向け Energy Management System に関連する研究に従事．電気学会会員．

奈良 祐樹

1975 年生．1997 年明治大学電子通信工学科学士課程修了．同年日立製作所入社．加入者伝送装置，サービスゲートウェイの設計・開発業務に従事．電子情報通信学会会員．

小田 輝

1987 年生．2011 年九州大学大学院理学府修士課程修了．同年日立製作所入社．サービスゲートウェイの設計・開発業務に従事．