

家電を Web で制御するサービスプラットフォーム

松倉隆一^{†1}

ホームネットワークに接続される家電や蓄電池などの機器を制御する HEMS やスマートハウスが注目されている。ホームネットワークは、白物家電の通信規格である ECHONET Lite など分野毎に通信規格が存在し、通信媒体も様々であることから構成が複雑になっている。この複雑さはアプリケーション開発の障壁となるため、機器とアプリケーションとの間の共通機能を整理し、統一的なインタフェースを持つプラットフォームを実現した。また、実際に住宅、店舗、校舎を含む 24 施設での検証を行っている。本稿では、実現したサービスプラットフォームの構成とその評価についてのべる。

A service platform to monitor and control home appliances as Web resources

RYUICHI MATSUKURA^{†1}

The Smart House and HEMS have gained the public attention, after it became available to monitor and control the home devices such as the home appliances and storage batteries. However, the home network is much complex. There are some communication protocol for each field of the devices and some communication media such as Ethernet, WiFi, PLC, and 920MHz wireless in the home network. Since this situation makes barrier for the developers, we studied and implemented the service platform providing the unified application interface that hides the complex home network. In this paper, we describe the architecture of the platform and the application to 24 buildings such as houses, shops, and school houses.

1. はじめに

生活環境の利便性向上にともない、住宅内に設置される家電や住宅設備は増加傾向にある。最近ではこれらをネットワークに接続し、新しいアプリケーションを実現する動きがでてきた。全体の消費エネルギー量を効率化するための HEMS (Home Energy Management System) や人にやさしい居住環境を実現するためのスマートハウスはその例であり、ICT が適用可能な新しい分野として認知されつつある。

従来、ホームネットワークに接続される機器としては PC が代表格であった。次第に DLNA による黒物家電 (AV 家電) や白物家電、最近では太陽光パネルや蓄電池などの創蓄エネ機器と呼ばれる住宅設備などが接続されるようになってきた。多くの機器は IP ベースの Ethernet や無線 LAN で接続されるが、白物家電では特定小電力無線や、RS-485 等のシリアルインタフェースも使われている。ホームネットワークに様々な機器が接続されるにつれて、ホームネットワークの構成が複雑になってきている。

一方、アプリケーション開発に関しては、Windows や Android, iOS 等の汎用 OS で動作する PC やスマートフォン、タブレット端末における開発とは異なり、機器固有の通信プロトコルを多数扱う必要ことが必須である。例えば、黒物家電や白物家電では独自に発展してきた通信プロトコルが存在し、また、RS-232C や RS-485 といったシリアルケーブルで接続する通信規格も広く使われている。これま

でのスマートハウス分野 (従来は情報家電等と呼ばれた分野) における開発は、ネットワークに機器を接続することが大きな課題であった。今後は、開発者がアプリケーションを簡単に開発できるような環境作りに、関心が徐々に移っていくと考えられる。

ホームネットワークに接続される家電を制御するアプリケーションを実現するには、機器やネットワークに関する他分野の知識が必要である。そのため、アプリケーションを誰でも開発できる状況にはなっていない。本稿では、こうした複雑なホームネットワークや家電機器のインタフェースを隠蔽し、共通の Web インタフェースで家電機器を制御するためのサービスプラットフォームについて述べる。以下、2 章では関連する研究を紹介し、3 章では実際に開発したサービスプラットフォームの構成について述べる。4 章で適用した例を紹介し、5 章でまとめを行う。

2. 関連研究

スマートハウスに関連する試みは 1980 年代後半の TRON 電脳住宅からあるが、共通プラットフォームの検討はようやく最近始まったところである。この章では関連する研究として、住宅を対象にしたプラットフォームと、家電を制御する M2M プラットフォームとの 2 つの視点から述べる。

2.1 ECHONET Lite[1]

ECHONET Lite が 2012 年に公知の標準規格として認知されるようになり、ECHONET Lite に対応した白物家電に対応したプラットフォームが提案されている。最初に

^{†1} 富士通(株)
Fujitsu Ltd.

ECHONET Lite を説明する。

ECHONET は、エコネットコンソーシアムが家電・設備系ホームネットワーク向け通信プロトコル仕様として策定しているものである。2000 年に策定された ECHONET と、2011 年に策定・公開された ECHONET Lite がある。ECHONET は、トランスポート層以下の伝送メディアとメディアによる仕様の差異を吸収するレイヤと機器を制御するための通信処理レイヤとからなっている。現在、広く利用されている ECHONET Lite は、ECHONET 仕様のうちトランスポート層以下は規定せずに IP アドレスもしくは MAC アドレスを利用するものであり、この下位レイヤの規定は情報通信技術委員会（TTC）が技術レポート TR-1043[2]として公開している。TR-1043 では、概ね IP 通信をベースとして、Ethernet、無線 LAN のほか、PLC や 920MHz 無線（ZigBee, WiSUN）等での接続について規定している。

ECHONET の特徴は、エアコンや蓄電池などの機器が持つ機能を機器オブジェクトと呼ぶ論理的なモデルである。このモデルは、機器が保持する情報やリモートで操作可能な制御項目（プロパティ）を論理モデルで定義している。実際のデバイスは、この機器オブジェクトのプロパティを変更することで、操作されるようになっている。この論理モデルは、それぞれの機器の各メーカーが協力して作成したものであり、メーカーが異なっても同じインタフェースで制御できることが特徴である。メーカーの特徴として独自の機能を入れる場合にはその独自機能を制御するインタフェースを追加可能になっており、メーカーに依存しない共通インタフェースとメーカーの特徴となる独自機能を同居させることが可能である。現在、機器オブジェクトとしては、センサ関連、空調関連機器、住宅関連機器、調理・家事関連機器、健康関連機器、管理・操作関連機器、AV 関連機器などが規定されている住宅向け機器だけでなく、小規模の店舗やオフィスで使われる業務用機器についても規定がある。

2.2 ECHONET Lite 対応プラットフォーム

ECHONET Lite は、通信インタフェースとして低性能の組み込み CPU で動作することを考慮して、電文はバイナリで表現されている。しかし、通信プロトコルが共通化されとはいえず、アプリケーション開発を促進するには、HTTP や Web インタフェースでの開発が強く望まれ、Web インタフェースから ECHONET Lite 機器を制御可能なプラットフォームが期待される。

大和ハウス工業が開発した住宅 API3 は、ホームサーバで動作するソフトウェアが ECHONET Lite に対応した Web インタフェースを提供する。スマートフォン等から HTTP で ECHONET Lite 機器のプロパティとその値を指定してホームサーバに送信するとその結果を XML で通知する。住宅 API ソフトウェアの背後には、ECHONET Lite 等の通信プロトコルを処理するモジュールがあり、このモジュールが

ECHONET Lite に変換して実際の機器を制御するようになっている。

増尾らのシステム[4]では、ECHONET Lite 機器を Web リソース化しており、Web サービスの枠組みで参照および制御を可能としている。具体的には、機器を制御する ECHONET Lite の電文を WSDL（Web Service Description Language）で記述し、SOAP で通信する。NAT がある環境においてもインターネット上にある HEMS から宅内の機器を制御できるように、双方向 Web サービスとして実現している。Ajax、ロングポーリング、WebSocket を使用する方式を比較して評価を行っている。

筆者らは ECHONET Lite 機器の制御情報を Broadband forum TR-069 で転送し、NAT を介して住宅内の家電を制御するプラットフォームを実現している[5][6]。ここでは、クラウドとホームゲートウェイ（HGW）を TR-069 で接続し、ECHONET Lite の制御情報を TR-069 における汎用的なデータ形式 TR-181 で通知しているのが特徴である。アプリケーションはクラウド上で実行され、住宅に設置されるホームゲートウェイ（HGW）を通じて家電を制御するプラットフォームを実現している。

2.3 M2M プラットフォーム

ホームネットワークに接続される家電等が人の手を返すことなく ICT により制御されるという点では、M2M で扱われる領域である。M2M プラットフォームについては様々な検討があるが、ここでは汎用的なプラットフォームについて説明する。汎用的なプラットフォームとしては、欧州の標準化団体である ETSI の TC M2M がある。ここで策定された仕様は、現在 oneM2M でも継続して議論が進められているが、概ね図 1 のような構成になっている[7]。

アプリケーションとデバイスとの接続方法は、ローカルなエリアネットワークがあるかどうかで分類される。ここで議論するホームネットワークはエリアネットワークにあたるので、エリアネットワークと広域ネットワーク（移動体ネットワーク、固定ネットワーク）を中継するゲートウェイが存在するケースである。プラットフォームとして

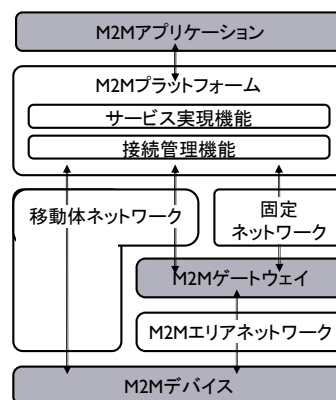


図 1 M2M アーキテクチャ

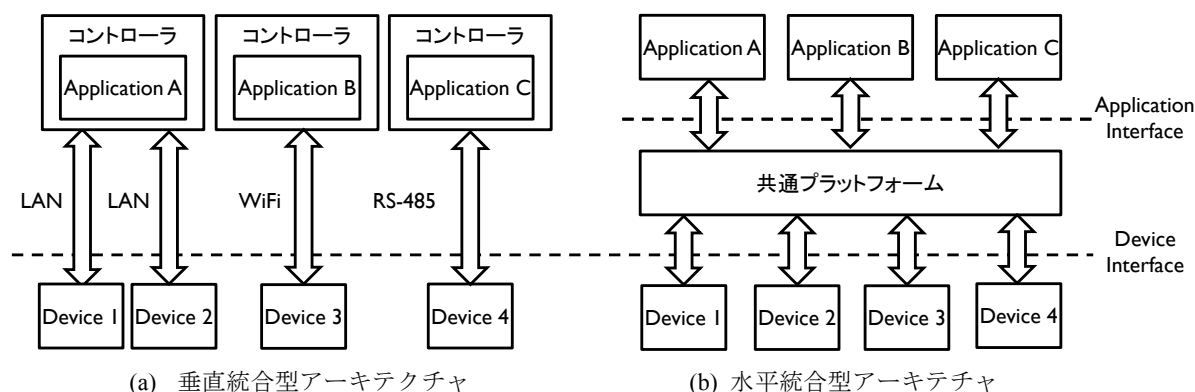


図 2 アプリケーションとデバイスとの接続方法

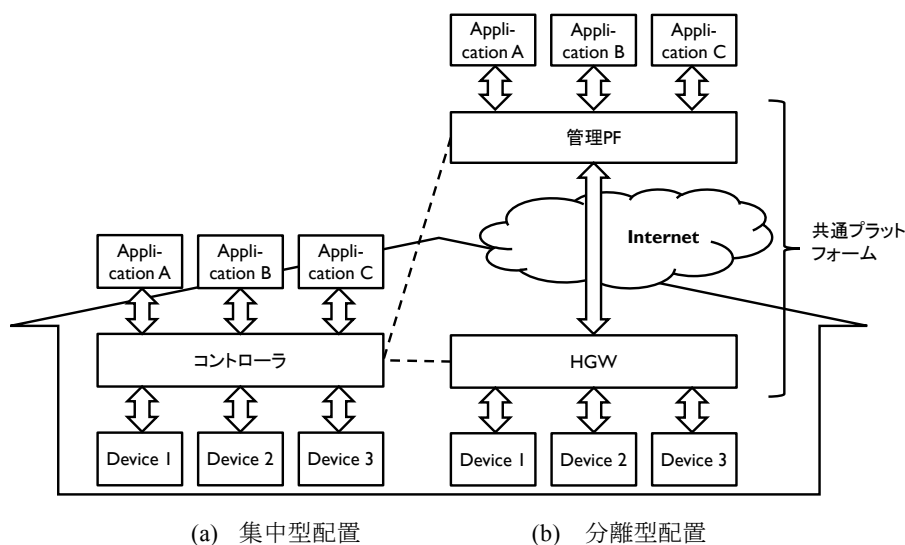


図 3 共通プラットフォームの配置

は、広域ネットワークの接続管理機能とデバイス管理や診断、認証などのセキュリティ機能と収集データのデータ分析やデータベース化などの M2M サービスを提供するための付加価値機能からなる。

2.4 プラットフォームの課題

ホームネットワークに接続される機器が様々な通信インタフェースを持つことから、機能の共通化に対するコストが高いことが課題になっている。近年では、多くのアプリケーションが Web インタフェースを利用して開発されていることから、ホームネットワーク向けサービスについて Web アプリケーション化を図ることが有効である。しかし、現在の機器インタフェースは Web インタフェースと隔たりがあるため、それぞれのインタフェース化のギャップを埋めるための共通プラットフォームの実現が必要と考えた。

機器のインタフェースの標準化が進み、統一化が進むと言え、現在は過渡期であることから、機器や接続インタフェースごとにアプリケーションが開発されるケースが多い。

図 2(a)のような構成で開発されている。これを図 2(b)のようにして、アプリケーション側のインタフェースを共通化するとともに、デバイス側のインタフェースについては、分野ごとに異なる通信インタフェースでの統合化するフレームワークを導入することが必要である。また、ホームネットワークとインターネットという異なるネットワークを接続することから、共通プラットフォームとしてはこれらを接続する仕組みが必要になる。例えば、ホームネットワークの入り口にあるブロードバンドルータで設定されるファイアウォールや NAT を通過する方法である。実際のネットワークで発生しうる問題をプラットフォームに備えることによって、開発におけるコスト削減を図ることが必要である。

3. サービスプラットフォーム

本章では、本稿で提案するサービスプラットフォームのアーキテクチャについて述べる。

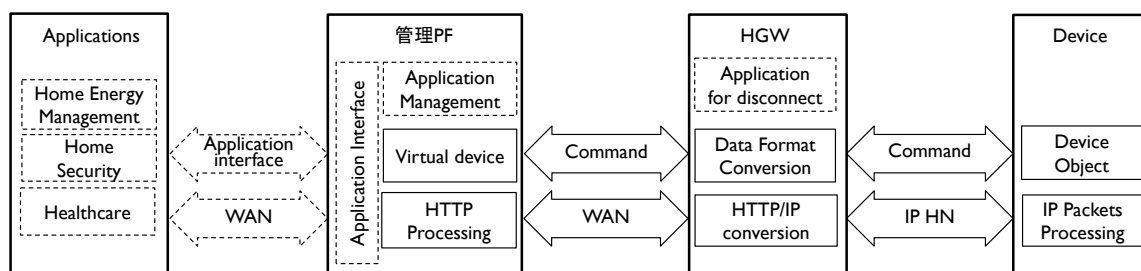


図 4 機器を制御する機能アーキテクチャ

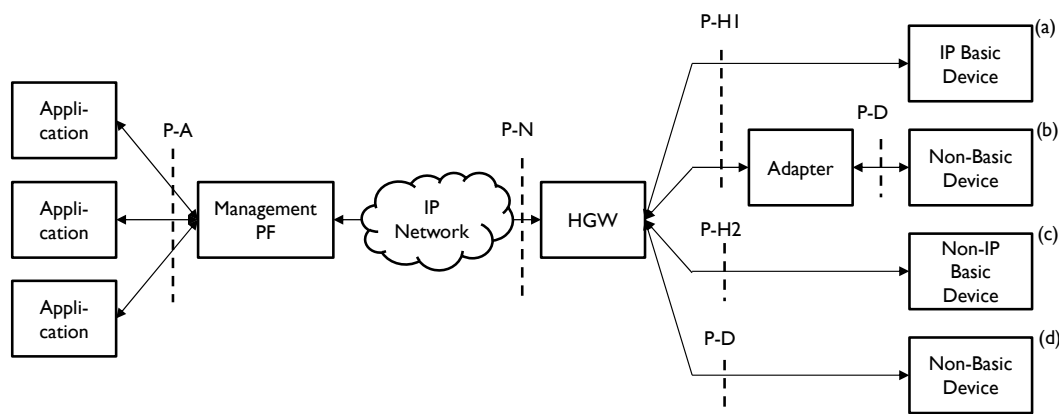


図 5 機器接続アーキテクチャと参照点

3.1 共通プラットフォーム

アプリケーションと機器との間に共通プラットフォームを配置することとして、アプリケーションインタフェースと機器インタフェースを定義することで、それぞれ標準的なインタフェースを提供する。この共通プラットフォームの実現方法は、図 3 に示すように集中型と分離型が考えられる。集中型はホームネットワーク内に設置されるホームコントローラから機器を制御するようなケースを想定しており、分離型では機器はホームネットワークに接続されるが、アプリケーションはインターネット上のサーバで提供するケースを想定している。そのため、分離型では共通プラットフォームの機能を HGW とインターネットに機能を分離して提供する。以下では、分離型を想定して説明するが、集中型の機能は分離型に含まれる。

共通プラットフォームの利点は以下のとおりである。

- ・機器の通信インタフェースから完全に独立したアプリケーションインタフェースが提供可能であり、Web インタフェースなど開発の容易な環境を提供できる。また、他の Web アプリケーションのとの連携が容易になる。
- ・共通プラットフォームがホームネットワークとインターネットという運用ポリシーの異なるネットワークを接続することにより、異種ネットワークを接続するときに発生する様々な問題を開発者が意識する必要がなくなる。
- ・共通 PF が、機器や HGW の認証、認可、ホームネットワークにおける通信暗号化等を実現することにより、サービス開発者はホームネットワーク内におけるセキュリテ

ィ機能を意識する必要がない。

- ・分離型では、接続される機器の数や通信頻度が多くなってもクラウド側で処理することが可能なため、HGW のハードウェア性能を処理量に合わせて高くする必要がない。

3.2 アーキテクチャ

図 4 に、アプリケーション、機器を含む全体のアーキテクチャを示す。機器に関しては、基準となる通信インタフェースとして、2.1 で述べた ECHONET Lite を参考にして Basic device として定義する。Basic device は、機器に備わっている機能を抽象的なデータモデルで表現される Device object と、この Device object のプロパティを、Set (制御)、Get (状態取得)、Inform (通知) する操作からなっている。Basic device のインタフェースは、IP パケット上に Set/Get/Inform の操作と具体的にやり取りされる情報<プロパティ、値>の組を送信することで実現する。これをコマンドと呼ぶことにする。欧州の KNX や米国の Smart Energy Profile (SEP) も ECHONET Lite と同じ思想で作られており、Basic device と同様のインタフェースを持っているため、海外でも広く利用可能であると考えている。

共通プラットフォームでは、Basic device の Device object を仮想デバイスとして保持し、この仮想デバイスのプロパティを参照・変更することで実際のデバイスを制御する。分離型アーキテクチャでは、この仮想デバイスを管理 PF に配置し、この仮想デバイスを Web リソースとして扱えるようにアプリケーションインタフェースを実現する。そのため、HGW ではホームネットワークとインターネットで

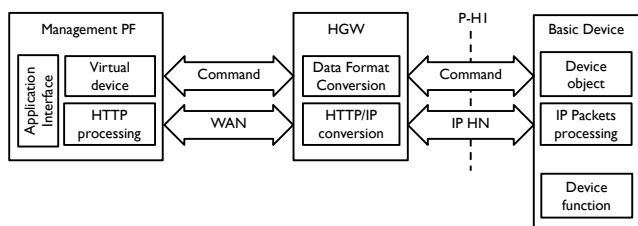


図 6 IP 通信 Basic device (P-H1)

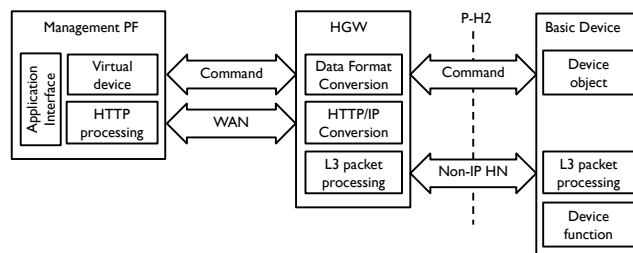


図 7 Non-IP 通信 Basic device (P-H2)

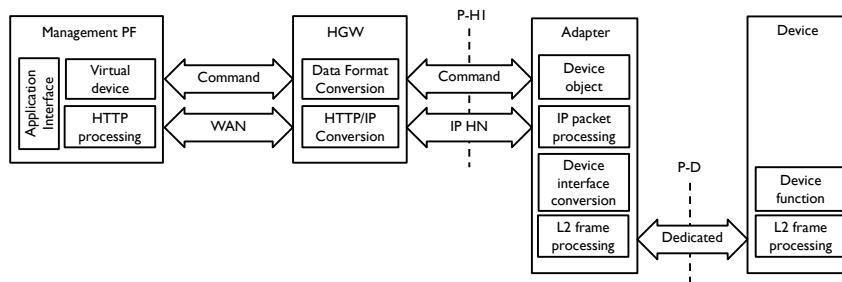


図 8 Non Basic device アダプタ接続 (P-D)

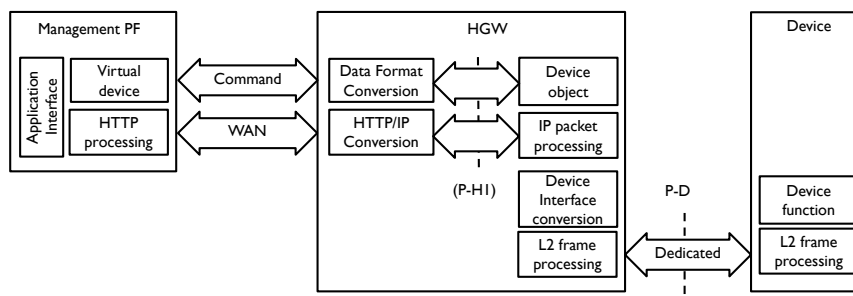


図 9 Non Basic device 直接接続 (P-D)

の通信プロトコルとデータ表現形式の変換を行う。データ表現に関しては、管理 PF では XML 形式で表現することが望まれるが、ハードウェアリソースに制約のある機器ではバイナリ形式で表現することも少なくない。そのため、HGW ではこの表現形式の変換と、ホームネットワークにおける IP パケットを HTTP に変換する機能を持たせることが必要である。管理 PF では複数のアプリケーションから同時に接続されるため、アプリケーション管理機能を持って機器からの通知を振り分けたり、受信したデータを保持したりする機能を持つ。

管理 PF に存在するバーチャル機器のプロパティを設定、参照することで、ホームネットワークに接続される機器の状態を参照し、制御することが可能である。アプリケーションインタフェースは、仮想デバイスを Web リソースとして扱えるようになっている。将来は、W3C で規格化を進めている HTML5 での対応を検討する予定である[8]。

3.3 Basic device 以外の通信方式

3.2 では Basic device の場合の動作について説明した。以下では、機器の接続方法についてさらに詳細に述べることとし、機器の種類、通信方式の違いによる接続方法について説明する。

想定する形態を図 5 に示す。機器が Basic Device である

かどうか、IP ベースの通信か否かで 4 種類の接続形態が考えられる。Basic device には IP 通信デバイスと、Non-IP 通信デバイスが存在する。IP 通信を行う IP 通信 Basic device は、バリエーションが少なく相互接続性が高い。この例として、ECHONET Lite がある。Non-IP 通信 Basic device は、ネットワーク層もしくはデータリンク層のパケットまたはフレームにより通信される操作を変換する必要がある。しかし、上位のフォーマットは IP 通信 Basic device と同じであるため、下位層のプロトコル変換を HGW で行うことにより機器は直接接続可能である。この例としては、KNX や SEP がある。これ以外に、Non Basic device があり、アダプタを使って Basic device として接続する形態と、デバイスがもつ独自のインタフェースのまま HGW に接続する 2 つ形態がある。

図 6 では、Basic device が接続されときの接続アーキテクチャを示している。これは図 5 で HGW とデバイスが P-H1 で接続されていたものである。ここでは、まず Basic device における管理 PF までの接続を説明する。

HGW とデバイス間の通信は、IP 通信を基本としているが、インターネット側については、セキュリティやファイヤウォール/NAT トラバース問題を考慮して HTTP での通信を基本とする。そのため、HGW では、IP 通信で転送

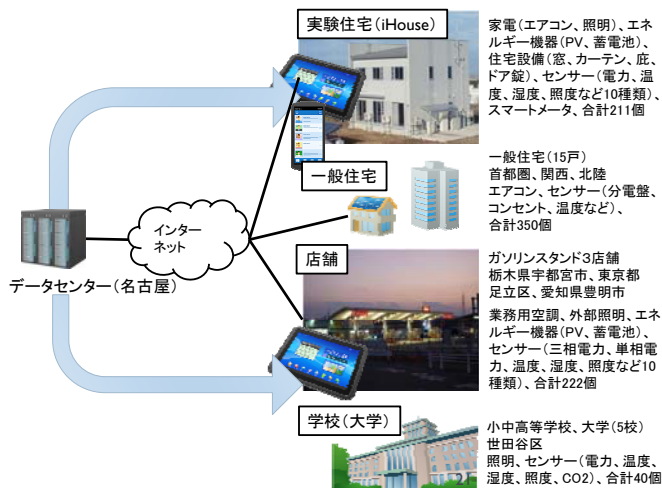


図 10 実証フィールド概要

されていたコマンドを HTTP で通信しやすい形式に変換する。XML や SOAP がその例である。管理 PF では機器の Device object に対応して仮想的な Virtual device として管理する。アプリケーションは管理 PF の Application Interface を通じて Virtual device を参照、変更することによって、ホームネットワークに接続される機器のプロパティを参照、変更することができる。

ただし、規格によってプロパティ種別が異なったり、同等の機能が違うプロパティ名で表現されることはありうる。したがって、HGW では共通プラットフォームの中で使用するプロパティ名を辞書として持つ必要があり、同種類のデバイスを扱う複数の規格を同時に使用する場合には、プロパティ名が衝突しないように定義することが必要になる。将来的には、これらの規格の間でプロパティ名称やデータモデルが統一化されることが期待される。

次は、HGW と Device が IP 通信ではなく、無線等の MAC フレームにコマンドをのせるケースである。このケースは、P-H2 での接続になる。このデバイスでは、図 6 における Basic device のように Device object は存在するが、通信イン

タフェースが IP ではないため、HGW でデータリンク層の通信機能を持ち、データリンク層から IP への通信の変換を行う。したがって、HGW 内のプロトコル変換機能で P-H1 と同等のインタフェースを持つことが可能となり、HGW からインターネット側への接続は 6 と同じ構成になる。例としては SEP がある。

次に、機器が論理モデルを持たず、機器がもつ内部機能 Device function を直接制御するインタフェースのみもつ場合の処理について説明する(図 8)。この場合には、アダプタで Device object を実現して、HGW とのインタフェースを P-H1 にする。P-D デバイスは、デバイス機能がもつインタフェースを、RS-485 等のシリアルケーブルや無線のフレームにのせて通信する。Basic Device では、Device function から Device object への変換を内部で行っているが、これをアダプタで実現し、アダプタと HGW との通信を P-H1 に合わせてしまうことで、図 6 のアーキテクチャに合わせるものである。図 9 はアダプタの機能を HGW 内部に持つケースであり、参照点 P-H1 が HGW の内部に存在する以外は図 8 と同じ形式になる。

4. 評価

3 章で述べたアーキテクチャに基づき、住宅や店舗、公共施設を利用した実験を行っている。本章では、この概要とそこでの課題について説明する。

4.1 実証フィールド構築

実験の全体像を図 10 に示す。実験フィールドとしては、実験住宅、一般住宅 15 戸、店舗 3 店、公共施設(学校 5 施設)の 24 施設である。実験住宅[9]は、エアコン・照明などの白物家電、太陽光パネル・蓄電池などのエネルギー機器、電動窓・電動カーテンなどの住宅設備、スマートメータ、電力・温湿度・照度などのセンサ 10 種類の合計 220 個の機器が ECHONET Lite で接続されている。一般住宅には、コンセント型の電力センサと一部の住宅には

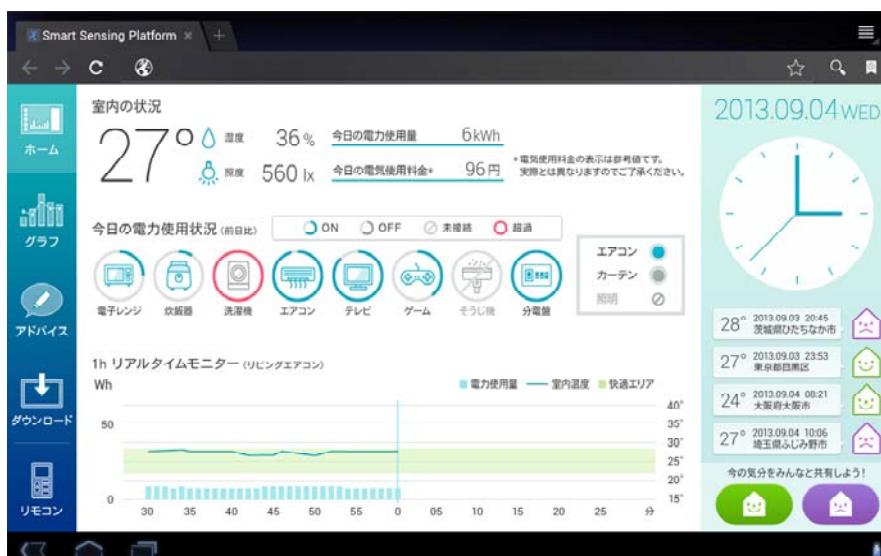


図 11 一般住宅向けアプリケーション画面

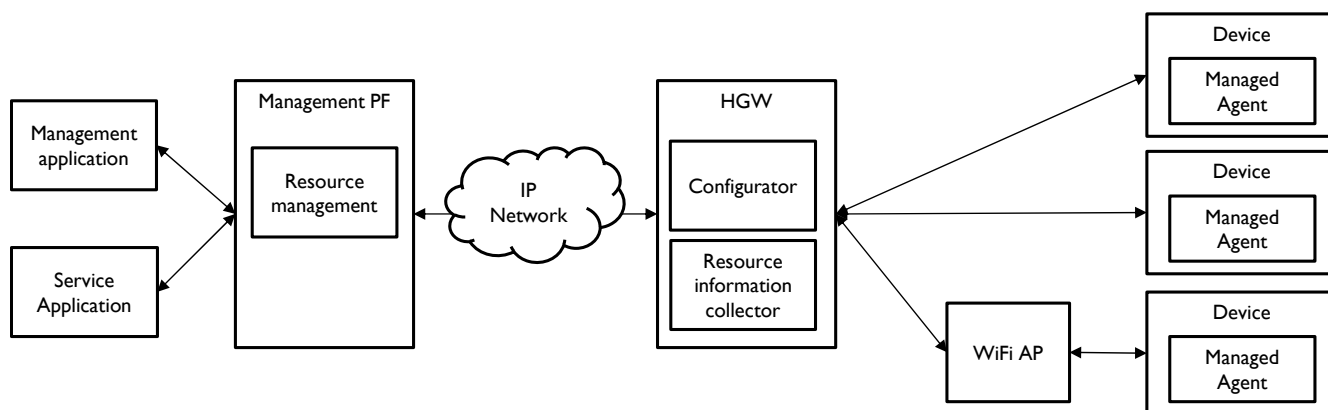


図 12 リソース管理機能の配置

ECHONET Lite 対応エアコン・照明が接続される。店舗に関しては、業務用空調や照明、電力センサを ECHONET Lite アダプタを接続し、ECHONET Lite 機器として接続行っている。公共施設は、電力センサの他に、CO2 センサや照度センサなど教室の環境情報を取得する機器を中心に設置している。これらの機器は、各施設に設置される HGW を経由してインターネットに接続され、名古屋にあるサーバに接続される。サーバには、管理プラットフォームが動作しており、アプリケーションとして HEMS や BEMS、学校向けの環境見える化が動作している。

アプリケーションの例として、図 11 に一般住宅で利用中のアプリケーション画面を示す。この画面中央の円グラフは、接続される電力センサで取得される家電機器の動作状況と消費電力が前日比で表示されている。画面下部のグラフは、省エネに大きく影響すると考えられるエアコンの商品電力と、室温の変化を重ねて表示している。画面左下にあるリモコンにより、エアコンや照明の制御ができるようになっている。

4.2 汎用機器の取り扱い

このアーキテクチャでは、PC やスマートフォンのような汎用的な用途に利用できる汎用機器は、機器として接続されない。これは、汎用機器にその機器の特徴となる Device object が定義されないためである。例えば、HEMS における消費電力見える化画面をブラウザで表示するとした場合、Web アプリケーションの表示端末ではあるが、アプリケーションから操作対象となるデバイスとはならない。

こうした汎用端末がこのアーキテクチャにおける機器となるには、端末に付属するセンサ、カメラ等の機能が Device object として端末上に実装され、HGW からその機能が参照もしくは変更できることが必要である。また、汎用機器上に実現されるアプリケーションを含めて機器として接続するためには、アプリケーションの内部データ構造を Device object として定義して、ECHONET Lite 等の規格としてメンテナンスされることが必要となる。アプリケーションが Device object として規格化されれば、ソフトウェアの再利用の一つの方法として考えることができる。

4.3 リソース管理

ホームネットワークサービスは、複数の通信プロトコル、様々な分野の機器が接続され、複雑に組み合わせられたシステムである。しかも、専門的な知識を持つ管理者は各住宅には存在しないため、遠隔からによる障害分析を行い、システムのメンテナンスが必要である。そのためには、トラブルを引き起こす設定ミスを防ぐ仕組みや、障害時にデバイスやネットワークの状況を把握する仕組みが必要である。

図 12 は、ホームネットワークにおけるリソース管理のための機能アーキテクチャである。ホームネットワーク内のデバイス、ネットワーク機器、デバイス上で動作するソフトウェアを HGW で管理している。HGW における機能は 2 つあり、ひとつは機器の設定を行う Configurator であり、HN におけるデバイスは Configurator と連携して自身の設定を行う Managed agent を搭載する。もうひとつの機能は resource information collector であり、デバイスで発生するアラート、内部状態情報やネットワーク機器の設定、トラフィック、ソフトウェアの情報を収集する。

Configurator は、UPnP や ECHONET Lite 等でサポートされるデバイス機器発見機能をきっかけに、デバイスのネットワーク情報、動作設定、必要なソフトウェアのダウンロードを行う。また、デバイスの新規追加や削除により、他のデバイスの設定を変更するときにも Configurator から一括で処理できる。一方で resource information collector は、デバイスの設定や内部状態を定期的に収集し、デバイスで発生する障害ときに、同時に他のデバイスでの設定やトラフィック情報等を参照して原因を解析するために利用する。

一方で、様々な機器が、複数のネットワークで接続されるホームネットワークにおいて、あらゆる組み合わせで事前の試験をすることは不可能である。したがって、ホームネットワークにおいては、障害が発生することある程度想定したシステム設計が必要である。

例えば、機器の通信処理部が障害に陥るのを防ぐために、定期的に再起動を行い、通信処理部の内部状態を初期化することは、長時間動作時の障害を回避する有効な手段である。このために、デバイスやネットワークの状況をプラッ

トフォーム側で管理することが重要になってくる。デバイス、アプリケーション、プラットフォームの役割分担を明確にし、重複した機能を開発しないとともに、重複する機能が異なる動作をすることがないようにする仕組みが必要となる。この領域においては、技術的な課題だけでなく、それぞれの開発者が協力し合いながら全体システムを提供できるようなエコシステムの構築も重要になると考えている。

5. まとめ

本稿では、ホームネットワークに接続される家電や住宅設備、センサを、Web インタフェースを通じて参照し、制御することを可能とするサービスプラットフォームについて述べた。白物家電、黒物家電等の様々な通信規格が混在する中で、それぞれの通信規格に依存しない Basic device に基づいたアプリケーションインタフェースを実現し、アプリケーションの開発コストを低減することが可能である。また、提案したプラットフォームでは Basic device に対応していない機器でも、Basic device に対応させるための変換機能をプラグインとして追加可能な仕組みを持っており、アプリケーションからは Basic device として扱うことができる。実証実験では、住宅、店舗、公共施設を含む 24 施設に 800 個超の機器を接続し、実際に Web インタフェースを通じて実用的なアプリケーションが動作することを確認した。

本稿で説明したアーキテクチャは、ITU-T において勧告化を進めている。

謝辞 本研究は総務省の「先進的 ICT 国際標準化事業 スマートコミュニティにおけるエネルギーマネジメント通信技術の実現」による委託を受けている。また、本研究に関して有益なコメントを頂いたプロジェクト関係者の皆様に深謝する。

参考文献

- 1) エコネットコンソーシアム：ECHONET Lite
<http://www.echonet.gr.jp/>
- 2) 情報通信技術委員会：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，TR-1043 v4.1(2013)
- 3) 大和ハウス工業：住宅 API WEB アクセス仕様書(2010)，
<http://www.daiwahouse.co.jp/lab/HousingAPI/>
- 4) 増尾剛，中村二郎，松岡茂登，長谷川剛，村田正幸，松田和浩：リアルタイム Web 技術による HEMS サービスクラウド化の検討，信学技報，Vol.112, No. 350, NS2012-117, pp.1-6 (2012)
- 5) 松倉隆一，角田潤，高橋英一郎：スマートコミュニティを実現するサービスプラットフォーム，信学会総合大会 (2013)
- 6) 矢野愛，角田潤，松倉隆一，金島正治：モデル住宅を利用した省エネルギー制御の実用化研究，建築学会シンポジウム (2013)
- 7) 藤田隆史，後藤良則，小池新：M2M アーキテクチャと技術課題，信学会誌，Vol.96, No.5, pp.305-312, (2013)
- 8) 芦村和幸，小松健作，一色正男：Web と機器を透過的につなぐ Multimodal Interaction フレームワークの実装，情処論文誌 コンシューマ・デバイス&システム，Vol.2, No.2, pp.19-28 (2012)
- 9) 金島正治，荻野修司：ホームネットワーク実証実験施設「i ハウス」の構築（その 1 実証施設の設計と熱特性把握），日本建築学会学術講演梗概集 D-2 (2010)