Vol.2015-ITS-61 No.1
Vol.2015-CDS-13 No.1
2015/5/21

クラウド型 HEMS サービス基盤の研究

内海和貴[†1]　岡本健司[†1]　村上隆史[†1]　宇佐美真[†1]　杉村博[†2]　一色正男[†2]

本論文では，HEMS を活用した Web サービスを開発するためのサービス基盤を提案する．サービス基盤を利用することで，開発者は ECHONET Lite の専門的な知識を使わずに Web 開発に用いる技術で HEMS サービスを開発することが出来る．実装したサービス基盤を利用して ECHONET Lite 機器を制御するサービスを開発し，動作することを確認した．

Studies of Cloud-HEMS Service Platform

UTSUMI KAZUKI[†1]　OKAMOTO KENJI[†1]
MURAKAMI TAKASHI[†1] USAMI MAKOTO[†1]
SUGIMURA HIROSHI[†2]　ISSHIKI MASAO[†2]

In this paper, we propose a service platform for developing Web services using HEMS, The developer can develop HEMS service in a technique to use for Web development without using the specialized knowledge of ECHONET Lite By utilizing service platform, Using a mounted service platform, it was confirmed to develop the service which controls ECHONET Lite equipment.

1. はじめに

日本国内の家庭部門が占める消費エネルギーの割合は，人口の増加に加えて家電製品の多様化や高機能化により増加の一途を辿っている．2012 年度の最終エネルギー消費では，家庭部門が 1973 年度と比較して約 2 倍のエネルギー消費量となっており，今後も増加することが予想される[1]．このような背景から，家庭の消費エネルギーを削減する手法として，HEMS（Home Energy Management System）の導入による省エネルギー効果が注目されている．

HEMS には電力の見える化による節電効果や機器の連携動作による新たなサービスの実現といったメリットがあり， これまで HEMS を活用した様々なサービスの研究，開発が盛んに行われてきた[1][2][3]．文献[4]ではホームネットワークシステムを利用した，一人一人のエンドユーザに最適化されたパーソナルリモコンの開発を容易化するためのアプリケーションフレームワークを提案しており，宅内環境に応じて背景やボタンをユーザ好みのインターフェイスに変更出来る．文献[5]では，ホームネットワークにおける家電連携サービス作成支援システムを提案しており，複数の家電製品を連携制御するサービスをエンドユーザが作成，編集，削除することが出来る．このように，多様化する宅内環境やライフスタイルに対応するために，エンドユーザ自らがサービスを開発するための手法が数多く提案されてきたが，先行研究では Android や iOS に代表されるネイティブアプリケーションやデスクトップアプリケーションといった環境に依存する実行形式を出力とする事が多く，宅外から利用可能な Web サービスを開発するための提案は少ない．

本研究では，国内における HEMS の標準通信規格である ECHONET Lite を用いて，HEMS を利用した Web サービスを開発するためのサービス基盤を提案する．実装したシステムを利用して開発したサービスが実行できる事を確認した．

2. システム構成

2.1 システム概要

提案するサービス基盤は，Web から宅内のホームネットワークを利用した家電機器の制御や状態取得を行うサービスを開発，実行するためのシステムである．サービス基盤は図 1 のように，サービスからの制御リクエストを取得するリクエスト取得モジュール，リクエストされた制御内容から ECHONET Lite 電文を生成するための ECHONET Lite コンバータ，制御電文をゲートウェイと通信するための制御電文通信モジュール，受信した制御電文を実行するゲートウェイ，そしてそれらをつなげる Application 部で構成される．Web からの利用を可能とするために，サービス基盤はクラウド上のサーバーで動作する．

†1 神奈川工科大学大学院　工学研究科　電気電子工学専攻
Kanagawa Institute of Technology
Graduate School of Engineering
Electrical and Electronic Engineering
†2 神奈川工科大学　創造工学部　ホームエレクトロニクス開発学科
Kanagawa Institute of Technology
Faculty of Creative Engineering
Department of Hone Electronics Development

Vol.2015-ITS-61 No.1
Vol.2015-CDS-13 No.1
2015/5/21

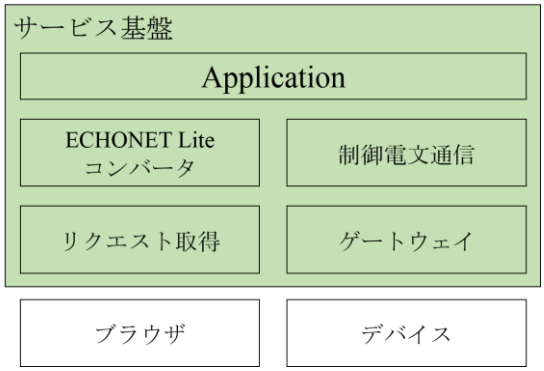

図 1　ソフトウェアスタック図
Figure 1　Software stack chart.

2.2 利用イメージ

サービス基盤の利用イメージを図 2 に示す．本研究で提案するサービス基盤は，ネットワークに接続可能な PC やスマートフォン，タブレットなどの端末から，ブラウザを開いてサービス基盤上のサービスにアクセスすることでサービスを利用できる．サービス開発は Web 開発と同様にテキストエディターや統合開発環境を用いて PC 上で行い，サービス基盤の動作するサーバー上にアップロードすることで Web サービスとして利用可能とする．

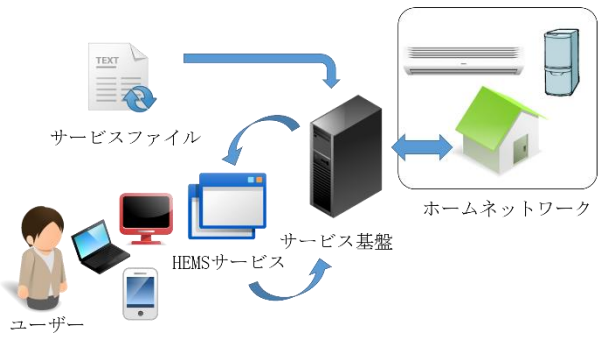

図 2　利用イメージ
Figure 2　An example of use.

2.3 要求仕様

上述した利用イメージを実現するためには，以下に示す要求仕様を満たす必要がある．

1)　Web からアクセス可能である
2)　API に準じたリクエストから ECHONET Lite の電文を生成することができる
3)　システム上でサービスを動作させることができる

要求 1 では，宅外からネットワークを介してサービスを利用するため，クラウドサーバーでサービス基盤を動作させる．

要求 2 では，ECHONET Lite 規格を用いた機器制御を，電文の構成やプロトコルに関する知識を必要とせずに機器制御を行うことができる．

要求 3 では，実装したプラットフォームの機能を使って機器制御を行うサービスを作成し，動作させることが出来る．

以下では，３つの要求仕様を満たすためのモジュールについて述べる．

2.4 モジュール

(1)　リクエスト取得モジュール

機器を制御するためのリクエストを取得し，制御の対象とする機器や制御する機能，設定する値などの情報を抽出する．

(2)　ECHONET Lite コンバータ

サービスが送信した制御内容を保有するリクエストと，実際に機器を制御することが出来る ECHONET Lite 電文とを相互に変換する．制御リクエストを受けた場合には，リクエストから制御の対象とする機器や制御の内容といった情報を取得し，電文を生成してゲートウェイに実行させる．情報の取得のように応答のある制御を行った場合には，応答として取得した ECHONET Lite の電文をサービス内で利用するために適当な文字列に置き換える．

(3)　制御電文通信

制御電文通信モジュールは，電文生成モジュールが生成した ECHONET Lite 規格の制御電文を，サービスを利用するユーザ宅内のゲートウェイへと送信する．また，制御電文を実行した機器から応答がある場合には，ゲートウェイからの応答電文を取得する．

(4)　ゲートウェイ

サービス基盤とホームネットワーク間での電文を送受信する．受信した制御電文はホームネットワーク内で実行し
情報を取得するリクエストの場合には応答を受け取りサービス基盤へ返信する．

3. 実装

3.1 サービス基盤

実装したサービス基盤の全体図を図 3 に示す．システムは Node.js を利用した Web サーバーとして構成されており，クラウドサーバー上にアップロードすることでサービス基盤として動作する．

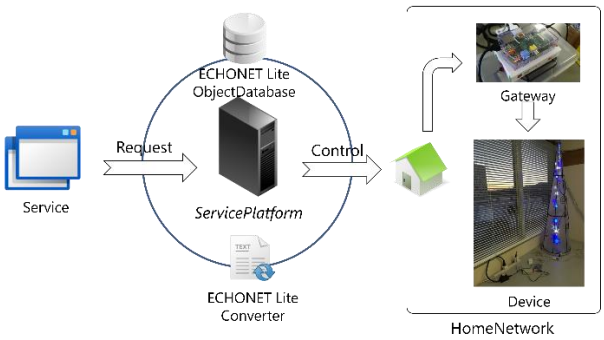

図 3　実装したシステムの全体図
Figure 3　General view of a system.

3.2 リクエスト取得モジュール

API は HTTP の GET リクエストを使用し，ECHONET Lite の電文構造や順序等についての知識を必要としない．

サービスの本体である HTML か，これに読み込まれる JavaScript ファイルに HTTP リクエストを送信するメソッドを記述し，HTML で記述したボタン等の UI と関連付けて実行することで機器制御リクエストを送信することが出来る．

API の使用に必要な情報は，制御したい機器の ECHONET Lite で規定された名称と，状態取得や制御の内容を表すプロパティ名だ．制御の場合には，加えて指定したプロパティに対して割り当てる値も指定する必要がある．例として，制御対象機器に「家庭用エアコン」，プロパティに「動作状態」を指定し，プロパティに対して「ON」を指定することで，エアコンの電源を入れるリクエストを送ることが出来る．図 4 に，JavaScript として記述する HTTP リクエストを送信するメソッドの一例を示す．

```
ON_btn.onclick = function() {
        $.getJSON(set,
        {
                DEOJ : "家庭用エアコン",
                EPC : "動作状態",
                EDT : "ON"
        },function(json){
                alert(json);
        });
}
```

図 4　制御リクエストの例
Figure 4　An example of Request.

3.3 ECHONET Lite コンバータ

ECHONE Lite オブジェクトのデータベースを作成し，機器制御リクエストで使用する機器名や制御内容を表す文字列と，ECHONET Lite で用いる数列とを相互に変換する．制御を行う場合は，リクエストに含まれる「機器名」や「制御内容」などの情報をデータベースから検索し，ECHONET Lite で用いる数列を抽出する．応答があった場合が逆の手順を踏むことで電文を文字列へ変換する．図 5 図 6 に，ECHONET Lite コンバータで用いたデータベースの構造を示す．

データベースの作成にあたっては，ソニーコンピュータサイエンス研究所の公開している ECHONET Lite オブジェクトデータベースをサービスプラットフォームで利用するために改変して作成した．

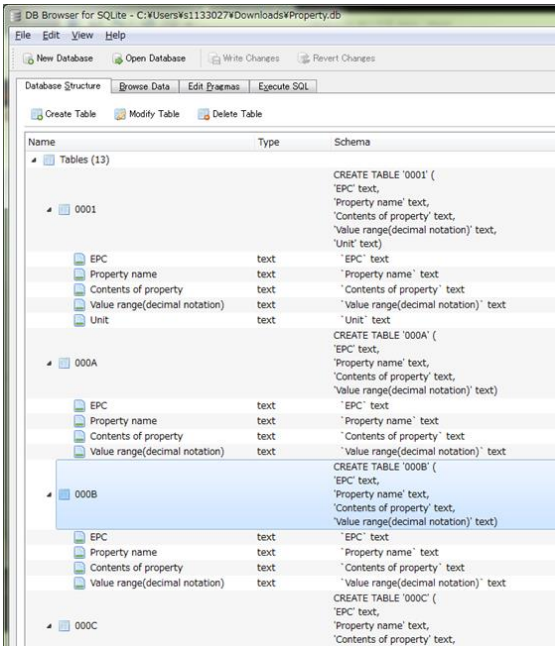

図 5　ECHONET Lite データベース構造①
Figure 5　Structure of ECHONET Lite Database①

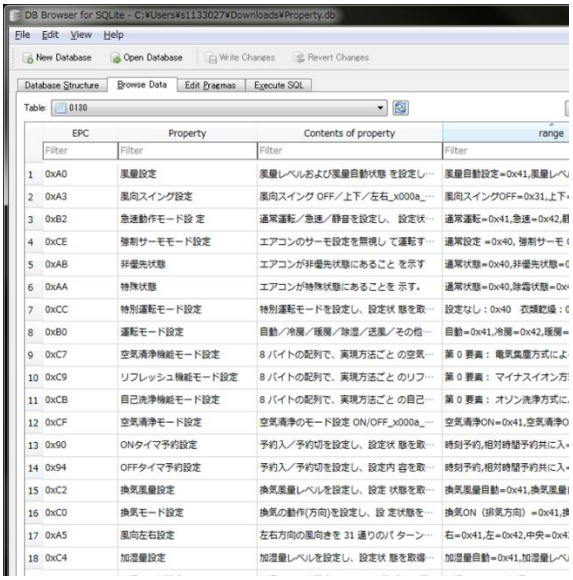

図 6　ECHONE Lite データベース構造②
Figure 6　The structure of ECHONET Lite Database②

3.4 制御電文通信

サービス基盤とホームネットワークに配置されるゲートウェイとは，Websocket による通信を行うことで電文のやり取りを行う．応答が期待される制御の場合には，送信した後に一定時間のブランクを設定してゲートウェイからの応答を待つが，応答を必要としない場合には送信が完了した時点でフローを終了する．図 7 にサービス基盤を利用したサービスから，指定機器の全てのプロパティを取得した場合に得られる応答の一例を示す．機器から

の応答は, サービス基盤で JSON 形式に変化されるため, サービス内のプログラムから利用しやすい形で情報を得ることが出来る.

ページ ex-session.herokuapp.com の記述:


```
{
  "Status": true,
  "Device": "013001",
  "User": "admin",
  "EPC": {
    "0x80": "0x31",
    "0x90": "0x42",
    "0xa0": "0x41",
    "0xb0": "0x44",
    "0xc0": "0x42",
    "0x81": "0x00",
    "0x82": "0x00004200",
    "0x84": "0x0001",
    "0x86": "0x050000170000000000",
    "0x96": "0x0000",
    "0x89": "0x0000",
    "0x8b": "0x00000f",
    "0x8e": "0x07dc0a08",
    "0x8f": "0x42",
    "0xff": "0x31"
```


図 7 応答の例

Figure 7 An example of response.

3.5 ゲートウェイ

サービス基盤とユーザの宅内ホームネットワークとを接続するためのゲートウェイを Node.js で動作するクライアントプログラムとして作成し，マイコンボードである「RaspberryPi model B」で稼働させた．図 8 にゲートウェイのデータフローを示す．ゲートウェイはインターネットが利用可能なネットワークに接続することで，Websocket を用いてサービス基盤との電文の送受信を行う機能を持つ．機器制御を行う場合は，UDP を用いたマルチキャスト通信によってホームネットワーク内の機器へサービスから受信した ECHONET Lite 電文を送信する．

制御の内容が特定の機器が持つ全てのプロパティ取得の場合には, ECHONET Lite で規定されるプロパティマップを利用する. プロパティマップとは, 機器自身が保有している利用可能な全てのプロパティを表す数列で，このプロパティマップを解析することで取得しなければいけないプロパティを把握することができる．これを元に状態取得の電文を動的に生成し実行することで，機器が持つ全てのプロパティ値を表す電文を応答として受け取ることが出来る.

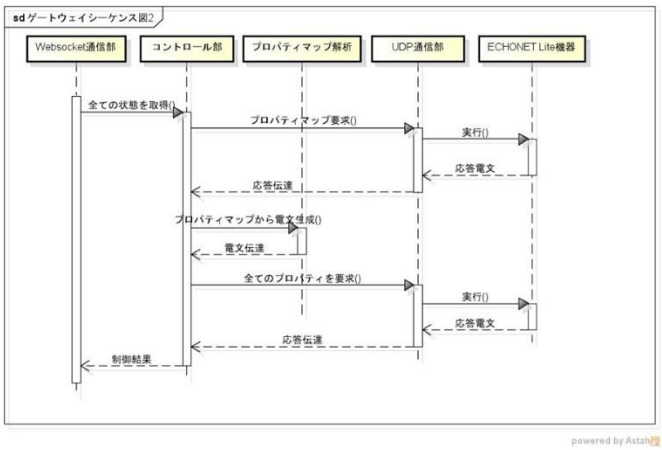

図 8 ゲートウェイのデータフロー

Figure 8 Dataflow of Gateway.

4. まとめ

本研究では，HEMS の標準通信規格である ECHONET Lite を用いて，家電機器を制御する Web サービスを開発するためのサービス基盤を提案した．機器の制御や状態の取得を，規格で定義される電文を作成せずに実行させることが出来る.

今後の課題としては，制御可能な機器の追加によるサービス範囲の拡大と，制御を実行するクライアントプログラムの汎用化が挙げられる．機器が増えることで実現可能なサービスの種類も増え，より高い快適性と利便性のあるサービスが生まれると考えられる.

参考文献

1) HEMS（ヘムス） - 住まいの設備と建材 – Panasonic http://sumai.panasonic.jp/hems/
2) フェミニティ倶楽部 http://feminity.toshiba.co.jp/feminity/about/index_j.html
3) クラウド HEMS http://www.sharp.co.jp/e_solution/hems/
4) 徳田啓介, 稲田卓也, 柗本真佑, 中村匡秀: ホームネットワークシステムのためのパーソナルリモコン開発フレームワーク, 電子情報通信学会技術研究報告, SS, ソフトウェアサイエンス, Vol.110, No.458, pp.7-12, 2011.
5) 関本純一, 中村匡秀, 井垣宏, 松本健一: ホームネットワークにおける家電連携サービス作成支援システムの開発, 電子情報通信学会技術研究報告, Vol.107, No.525, pp.289-294, 200