

研究論文

Web サービスとマルチデバイスの フレキシブルな連携方式の実現

渡部 智樹^{1,2,a)} 高嶋 洋一¹ 杉村 博² 一色 正男²

受付日 2014年4月24日, 採録日 2014年10月15日

概要: 家電やセンサなど, 身の回りの任意のデバイスの状態を任意の Web サービスと関連づけて, その状態の管理や制御が可能な仕組みを提案する. これまでに, HEMS などの規定のデバイスに限定して制御するサービスや, コンテキストの変化に応じてサービスを提供する仕組みは存在していたが, ユーザが任意のデバイスの状態を関連づけてサービス利用時にデバイスと連携させる提案はなかった. 本論文では, 提案方式について説明した後, 提案方式を実現するための設計指針をあげ, 具体的なアーキテクチャとプロトタイプにより実現可能性を示し, さらにサービスを提供するうえで課題となる各デバイスの管理・制御に関わる時間について実際の ECHONET Lite デバイスを用いて調査した結果を考察する.

キーワード: Web, 重畳表示, マルチデバイス, ECHONET Lite, デバイス制御

An Implementation of a Flexible Collaboration Method with a Web Service and Multiple Devices

TOMOKI WATANABE^{1,2,a)} YOUICHI TAKASHIMA¹ HIROSHI SUGIMURA² MASAO ISSHIKI²

Received: April 24, 2014, Accepted: October 15, 2014

Abstract: We propose a collaboration method that is able to manage so as to relate a web service which the user is browsing and several states of appliances around a user. There are some methods that are focused on only limited devices as HEMS or are using context of devices. However, our proposal realizes that a user associates a web service to some states of some appliances and will be able to use the service in the same situation as at that time. After explaining the proposed method, design criteria for implementing our method and the architecture are described. A prototype system is created and shows the feasibility. And we discussed about the processing time of the control and management of the actual several appliances.

Keywords: Web, overlay, multiple devices, ECHONET Lite, device control

1. はじめに

ECHONET Lite [1] や DLNA [2] の登場により, ネットワークにつながる家電が増え, スマートフォンやタブレットなどから状態確認や操作実行ができるようになった. 最近では Bluetooth や Zigbee などの省電力で通信可能なセン

サも実現されている. さらには, WoT (Web of Things) [3] や IoT (Internet of Things) [4] と呼ばれる「モノのインターネット」が注目され, 今後ますます身の回りの様々なモノがネットワークにつながり, それらを活用した多様なサービスが提供されるようになると思われる. 以降では, ネットワークにつながる家電やセンサ, モノを「デバイス」, これらの複数のデバイス群を「マルチデバイス環境」と称する.

このように, 多様なサービスが増加すると, デバイスは様々なサービスや複数のユーザから利用されるため, あるユーザにとって期待していた設定や状態が変更されている

¹ 日本電信電話株式会社 NTT サービスエボリューション研究所
NTT Service Evolution Laboratories, NTT Corporation,
Yokosuka, Kanagawa 239-0847, Japan

² 神奈川工科大学
Kanagawa Institute of Technology, Atsugi, Kanagawa 243-
0292, Japan

^{a)} watanabe.tomoki@lab.ntt.co.jp

事態が頻繁に発生すると予想される。たとえば、父親が映画を視聴するときには、カーテンを閉め、シーリングライトを消し、5.1chのスピーカで出力するといったマルチデバイス環境を期待する。一方で、子どもがアニメを視聴する際は、カーテンを開け、シーリングライトを点灯し、音声吹き替えのモード設定を期待するかもしれない。このように、同じデバイスであっても使う人や使うサービスによって期待する状態は異なる。

そのため、サービスを利用したりコンテンツを選択したりする際に、利用するデバイスが期待している状態であるかを確認し、期待と異なる場合には自動的に期待した状態となるように制御する、といった対応が考えられる。しかし、サービス提供者側でこれらを設定することは難しい。なぜなら、各世帯により家庭内に配置されたデバイスの数や種類、またサービスと連携して利用したいデバイスやその状態が異なり、さらには同一世帯でもユーザによってこれらの関連付け方は異なると考えられるためである。

従来、状況変化にともなうサービス提示 [5] や、サービス実行のために条件としてデバイス状態を利用する研究 [6] などが行われていた。しかし、任意のサービスと任意のデバイスの状態をユーザが設定・登録し、利用するという研究はなされていなかった。

そこで我々は、各ユーザがサービスごとにマルチデバイス環境を設定し、サービス利用時に登録したデバイスの状態を確認し制御する方式を提案する。なお、対象とするサービスとしては、HTML5 の勧告により今後の普及や移行が見込まれる Web を使ったサービスに注目する。

提案方式を実現するにあたり、要件の抽出と実現可能性の検証、実際のマルチデバイス環境における有効性評価が必要である。本論文では、実際のマルチデバイス環境においてシステムの実現可能性を確認することを目標とし、実装面と性能面から検証したので報告する。

以下、関連技術との差分、課題の抽出、そして提案方式の考え方を説明した後、提案方式を実現するための要件をあげ、具体的なアーキテクチャとプロトタイプにより実現可能性を示し、さらにサービスを提供するうえで課題となる各デバイスの管理・制御に関わる時間について実際の ECHONET Lite デバイスを用いて調査した結果を考察する。

2. 関連技術

サービスと複数のデバイスを関連付ける研究として、サービス提供者があらかじめコンテンツに組み込んで提供するアプローチと、ユーザが設定するアプローチがある。

前者にはたとえば HEMS (Home Energy Management System) [7] と呼ばれるシステムがある。これは家庭内にある家電やセンサをネットワークで接続し、主として消費電力をグラフなどにより可視化して PC で簡単に見えるよ

うにしたシステムである。PC の画面から見たり操作したりできるデバイスは既定のものに限られており、またユーザが他のサービスにこれらの家電の情報を関連付けることはできない。

一方後者のアプローチでは、ユーザが身の回りにあるデバイスを選択することができる。たとえば西垣らの研究 [5] では、家庭内のデバイスを効率良く操作するために、デバイスの状態をコンテキストとしてとらえ、操作実行の条件となるルールを簡潔に記述する提案がなされている。また前田らの研究 [6] では、連鎖する複数のコンテキストからサービス選択を制御する提案がなされている。

これらの研究はセンサやユーザ行動などの状態変化を起点としてデバイス制御やサービス提供などを目的としているが、我々の目的とするところは Web サービスの利用を起点としており、この点において異なる。すなわち、任意の Web サービスと任意のデバイスの状態をユーザが関連づけて動作させる仕組みはこれまで提案されていなかった。

3. 課題

市販のデバイスに付属された専用リモコンだけでなく、ソフトウェアによるリモコン、あるいは複数ユーザによる共同利用などのために、ユーザがサービスを利用しようとしたときに周囲のデバイスが期待する状態でない場合や、サービスの途中で想定外の動作をする場合がある。以下に具体的な例をあげる。

- 動画を見るときには、窓とカーテンを閉めて部屋を暗くし、エアコンをつけて視聴しようとしていたのに変更されてしまっている。
- 電子書籍での読書など集中したいとき、周りをできるだけ静かにしたいので、TV の電源を OFF (あるいは消音)、洗濯機やエアコン、ロボット掃除機を静音モードにしなければならない。
- ネットバンキングやオンライン学習を行っているとき、洗濯機や調理家電などの運転が終わり、作業を中断しなければならない。

これらの問題を解決するためには、それぞれのサービスにデバイスの自動制御や状態確認を行う仕組みを導入すれば実現できる。しかしながら、これらはサービス提供者側の問題でありユーザが望むあらゆるサービスに導入するのは難しい。また導入しようとしても各ユーザ(家庭)でどのようなデバイスをどのような状態に設定すべきかについてはユーザが決めるべきことであり、サービス提供者側で事前に設定できない。したがって、サービスとデバイスの状態をユーザの好みに合わせて関連付けられる仕組みが必要となる。

我々は、ネットワーク上のサービスとネットワークで制御可能なデバイスであれば、任意のサービス、任意のネットワークに適用することを目標とするが、本論文ではサービス

は Web によるサービス (アプリ), デバイスは ECHONET Lite に対応するデバイスを対象とする. その理由としては, 2014 年に HTML5 が W3C により勧告 [8] されるのを受けて, 今後 Web を使ったアプリ (Web アプリ) に移行していくと考えているためである. また ECHONET Lite は, 2011 年に HEMS における公知な標準インタフェースとして推奨され, また 2013 年には国際標準化規格として承認されたことから, 今後の普及が見込まれるためである.

以下, 上記の問題を解決する方式について提案する.

4. 提案方式

4.1 概要

上記に述べた課題を解決するために, 任意の Web サービスとそのときユーザが期待する任意のデバイス状態とを関連付けて, 再度同じ Web サービスを利用する際にデバイスの状態あるいは Web サービスの進行を制御可能とする方式を提案する. この方式を実現するために, ユーザが利用している Web サービスとそのときつながっているデバイスの状態を任意に選択し登録する機能, 登録している Web サービスを利用する際に, 登録情報と現在の状態との照合結果に基づいて, デバイスを制御したり, サービスの利用を制限したりするといった機能を実現する.

4.2 ユーザアンケート

提案方式の受容性を確認し, サービス提供の際に欠かせない設計指針を得るためにユーザアンケートを実施する.

4.2.1 実施概要

・目的:

提案方式の受容性確認とサービス提供における設計指針を得ること.

・回答者:

日常生活の中で複数の家電を利用し, PC やスマートフォン, タブレットなどで Web サービスを利用している人, 10 名 (20~50 代, 男女 5 名ずつ) を対象.

・回答方法:

目的とする以下の 2 つの質問 (1)(2) のほか, 「サービスを利用するときに周囲のデバイスで気になる状態があるか」, 「登録にかけてもよい時間はどのくらいか」, などの補足の質問を用意し, さらに回答後にインタビューを行った.

(1) サービス受容性:

提案方式のサービスイメージを説明し, 利用したい/利用したくない, のいずれかを回答する.

(2) 登録したいデバイス状態数:

5 つの Web サービス (動画視聴, 電子書籍の読書, ネットバンキング, オンライン学習, TV 電話) をイメージしてもらい, 登録したいデバイスの状態をあげてもらい. 状況をイメージしやすいように, 具体的な

表 1 ユーザアンケートによる登録したいデバイス状態数

Table 1 Number of device status that users want to resist according to user questionnaire.

Web サービス	登録したいデバイス状態数	
	平均(個)	標準偏差
長編動画視聴	5.6	3.6
電子書籍の読書	5.0	4.3
ネットバンキング	2.2	4.5
オンライン学習	4.3	5.3
TV 電話 (発着信)	4.0	3.4

家電の例 (照明, 洗濯乾燥機, エアコン, 風呂給湯器, 電子レンジなど) と ECHONET Lite で取得可能な状態例*1 (電源状態, 運転残り時間, 運転モードなど) を提示し, 合計 26 の状態から選択してもらう. そのほかに思いつくものがあれば自由に記入することも可とした.

4.2.2 アンケート結果

提案方式を使ったサービスについて, 回答者 10 名のうち 9 名が利用したいと回答し, 高い利用意向が見られた. 利用したくないと回答した 1 名は, 周囲に気になるデバイスがあったとしても特にそのデバイスを操作して状態を変えることはしないという意見であった.

表 1 は, 各回答者が登録したいとあげたデバイス状態数の平均と標準偏差である. 今回想定したサービスの利用シーンにおいては, ユーザが登録したいデバイスの状態数は 6 つ程度であることが分かった.

また, デバイス状態の登録にかけてもよい時間については, 平均 2.51 分 (標準偏差 1.94) であった. アンケート後のインタビューから, 初回だけなら登録に時間をかけてもよい, といった意見が聞かれた. このことから登録設定の作業に対してもユーザの受容性はあるといえる.

4.3 設計指針

提案方式を実現するための設計指針を示す.

- ① Web ブラウザから LAN 上に接続するすべてのデバイスの状態を確認できること
- ② ユーザが閲覧している Web サービスへの影響を最小限とすること
- ③ ユーザがサービスから離脱しない時間内に処理を完了すること

以下順に説明する.

まず ① について説明する. ユーザの端末がつながっている LAN 上のすべての ECHONET Lite デバイスを Web ブラウザで把握したり, その個々のデバイスの状態を確

*1 ECHONET Lite の家庭用エアコンでは, 動作状態, 運転モード設定, 温度設定値, ON タイマ予約設定など, 数十にわたる状態取得のコードが規定されている.

認したりするためには、ECHONET Lite のプロトコルを Web ブラウザ上で扱える必要がある。そのためにはプラグインにより機能拡張する方法もあるが、利用できる Web ブラウザが限定されてしまう。そのため、芦村ら [9] や松村ら [10] の研究で行われているように、Web ブラウザとデバイスとの間を仲介する仕組みを選択する方法が現実的である。

次に ② を説明する。ユーザは本来サービスを楽しむために Web を閲覧しているため、その行為をできるだけ阻害しないようにすべきである。ユーザが閲覧している Web 画面が別な画面に移動してしまうと閲覧中のサービスを継続できなくなってしまう可能性があるため、現在閲覧中の Web ページを遷移することなく実現する必要がある。

③ は、提案方式の処理に要する時間が長くなると、ユーザが不快感をいだき、提案方式だけでなく元のサービスさえも利用しなくなってしまう可能性があることに配慮したものである。具体的には、Web ブラウザ内の処理、ホームサーバ内の処理、各デバイスから状態を取得する処理、Web ブラウザとホームサーバ間の通信処理、といった処理に時間を要する。また 4.2 節のアンケートの結果から、デバイスに問い合わせる状態数は 6 つ程度であることが分かっている。そこで、この 6 つの状態数を取得する処理を想定し、ユーザがサービスから離脱しない時間内に上記の一連の処理が完了することを指針とする。

以上のように、① ② については実装面でのシステム要件、③ についてはシステムの性能面での要件となる。以降、システム要件と性能要件との 2 つに分けてその実現性を説明する。

4.4 システム要件とアーキテクチャ

本節では前節で述べた設計指針 ① ② に基づく方式の全体アーキテクチャを示す。まず前節の設計指針の検討からシステム構築要件として以下のように整理した。

1. Web ブラウザとデバイスとを仲介する構成で実現
2. 現在閲覧中の Web ページを遷移させないこと

これらを満たすアーキテクチャを図 1 に示す。

要件 1. を満たすため、家庭内 LAN 上にホームサーバを設置する。このホームサーバは、(1) Web との対話 I/F、(2) 家庭内デバイス群の状態確認、(3) 設定するデバイス状態の登録・管理と照合、の機能を備える。なお、各デバイスとの通信プロトコルは、プライベートな LAN 上の端末からアクセスされることが前提となっているため、ホームサーバは家庭内の LAN 上に配置される。

一方のユーザが利用している Web ブラウザには、(a) サービス・デバイス状態連携処理と、ホームサーバと通信して状態を登録する、(b) デバイス状態登録処理、登録している状態を照合する、(c) デバイス状態照合処理の機能を備

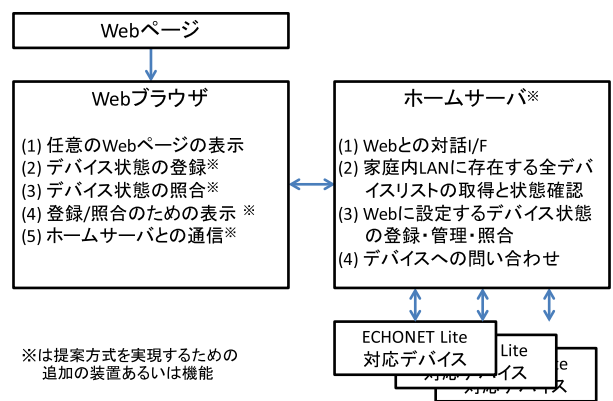


図 1 提案方式のアーキテクチャ
Fig. 1 A proposal architecture.

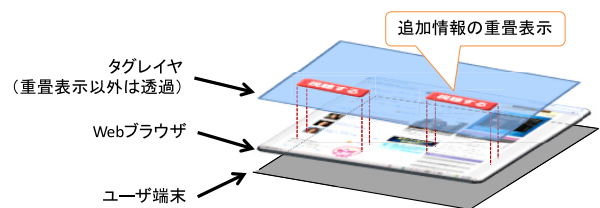


図 2 タグレイヤ重畳方式
Fig. 2 A concept image of “Taglayer” overlay.

えた実行モジュールを Web ページに組み込む必要がある。組み込む方法については後述する。

(a) の処理はユーザが Web ブラウザで表示している Web サービスの URL を取得し、それを Web サービスとしてデバイスの状態と関連付ける。このドメイン名での状態登録がなければ (b) の処理によりデバイス状態を登録し、登録済みであれば (c) の処理により登録されているデバイス状態を取得する。

(b) および (c) の処理においては、ユーザに登録する状態などを提示し入力を求める画面が必要である。そこで、要件 2. を満たすために、既存研究にて実現している「タグレイヤ重畳方式 [11]」を適用する。これは、任意の Web ページにユーザが関心を持つ情報を重畳して表示する技術である。このタグレイヤ重畳方式を用いて、デバイス状態の登録と照合の機能を実装する。タグレイヤ重畳技術について、ここでは概要のみ説明する。

タグレイヤ重畳技術の概念図を図 2 に示す。タブレットやスマートフォンなどの表示端末上で Web ブラウザを利用し、ユーザが閲覧中の Web ページに重畳して任意の情報を表示することができる。たとえば、テレビ番組に関する話題の Web ページにはそのテレビ番組を録画予約する表示ウィンドウを重畳して表示する。また、洗濯機の脱水運転が終了したら、引き続き乾燥運転を開始する操作ボタンと合わせて、天気が良ければ乾燥運転の電気代を表示して天日干しを促す省エネの工夫も取り入れ進められている [12]。

このタグレイヤ重畳方式は Web モジュール (JavaScript)

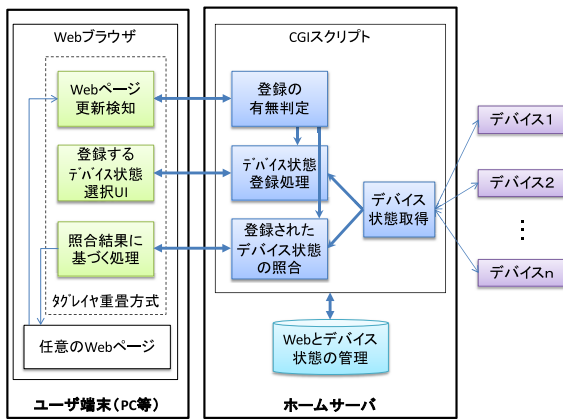


図 3 提案方式の機能構成

Fig. 3 A functional structure for proposal method.

で実装されており、元の Web ページにこのモジュールを追加することにより適用することができる。追加の方法としては、Web ブラウザの機能拡張とする方法もあるがスマートフォンやタブレットに適用するのは困難であるため、HTTP Proxy の機能を拡張することで実現する。

提案方式を利用するためには、閲覧中の Web ページにデバイス状態を登録する「登録機能」と、登録されている Web ページを表示する際に照合する「照合機能」の2つの機能が必要である。これら2つの機能を JavaScript で実装し、タグレイヤ重畳方式を使って閲覧中の Web ページに追加する。そして、Web ページの更新ごとに登録されているか否かを判定し、登録されていれば照合を、そうでなければ登録のための画面を、タグレイヤを使って表示する。デバイス状態の登録と照合の流れについて、図 3 を用いてそれぞれ説明する。

【照合の実行】

Web ブラウザ上で Web ページの更新が発生すると、その URL がホームサーバに登録されていればデバイス状態の照合の画面を、そうでなければ登録の画面を、Web ページ上に重畳して表示する。

【デバイス状態の登録】

各デバイスの現在の設定状態を取得し、登録画面を作成して CGI 応答により返却する。その中からユーザが選択したデバイスおよび設定状態の1つ以上の組合せと、そのときの Web の URL をホームサーバに送信し、ホームサーバに登録する。

【デバイス状態の照合】

受け付けた Web の URL をキーに登録されているデバイス状態を取得し、それぞれのデバイスに対し状態を確認する。その結果が登録されている状態と同じであれば OK を、そうでなければ NG とし、すべてが OK であるかを判定した結果を含めて CGI 応答として返却する。この結果を受信した追加モジュールは判定結果をもとに、サービス利用の許可などの制御を行う。

4.5 性能要件

設計指針 ③ で述べたように、提案方式に関わる処理の間、ユーザにとって不快感をいだかせないようにしなければならない。提案方式に関わる主な処理は図 1 に示した構成にあるとおり、(1) Web ブラウザ内の提案方式実行モジュール処理、(2) Web ブラウザとホームサーバ間の双方向の通信、(3) デバイス応答を含むホームサーバ内部処理の大きく3つの処理がある。ここでは(3)の処理に要する時間を測定し、提案方式の実現に影響しないことを確認する。その理由としては、(1)の処理は Web ブラウザ自体の内部処理あるいは PC やスマートフォンの仕様条件に起因すること、また(2)は有線 LAN だけでなく IEEE802.11g/n などにより無線 LAN の高速化が図られていることなどから、提案方式独自の処理にあたる(3)に着目する。

(3)の処理時間に大きく影響を与えるのは、問合せするデバイスの数と状態数である。なぜならデバイスの IP 制御は PC やホームサーバのそれとは違い、大容量あるいは頻繁なデータ送受信を想定していないため、処理性能が低い場合が多いためである。

そこで、実際の ECHONET Lite 対応デバイスを使って、デバイスへの問合せに要する時間を計測するとともに、ユーザアンケートから導いた登録したい状態数と比較することにより、実際のサービスに提案方式を導入してもユーザに不快感を与えないことを確認する。

5. 評価

本章では、現実的なハードウェア構成において、実現要件を満たすことが可能であることをプロトタイプの実機により確認する。さらに、実機のデバイスへの問合せ時間の調査と、ユーザアンケートによる登録したい状態数の結果とを比較し、提案方式がユーザに受け入れられることを示す。以降、検証に用いたプロトタイプについて説明し、このプロトタイプを用いた実機調査、およびユーザアンケートについて述べた後、考察を行う。

5.1 プロトタイプ

図 4 は、構築したプロトタイプシステムの構成図である。デバイス登録の処理は省略し、Web サービスに対してデバイス状態があらかじめ登録されている状態から動作を開始する。その理由は、4.2 節で述べたアンケートの結果、デバイス状態の登録にかけてもよい時間が約2分半と比較的長く、サービスを利用しようとしている場面の方がユーザの不快感をいただく恐れがあり優先すべきと考えたためである。実現構成としてホームサーバには、小型なシングルボードコンピュータである Raspberry Pi Type B [13] を利用した。Raspberry Pi の電力は 700 mA (3.5 W) であり比較的省電力であるため、家庭用に配置されるサーバとして適切であると考えた。ホームサーバおよびユーザ端末とな

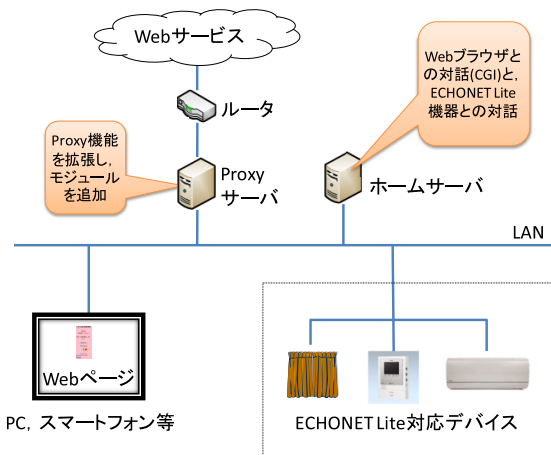


図 4 プロトタイプシステムの構成

Fig. 4 Structure of the prototype system.

表 2 プロトタイプシステムの仕様

Table 2 Specific of the prototype system.

ホームサーバ	
ハードウェア	Raspberry Pi Type B
OS	Linux raspberrypi 3.10.25+ #622 Raspbian "wheezy" 2012-09-18
HTTP サーバ	Apache/2.2.22
Web-CGI	Ruby1.9.3p194
ユーザ端末	
ハードウェア	Panasonic 社製 Let's Note CF-W8
OS	Windows 7 Professional
Web ブラウザ	Google Chrome 35.0.1916.27 beta-m

表 3 プロトタイプで使用する ECHONET Lite デバイス

Table 3 ECHONET Lite devices used in the prototype system.

デバイスタイプ	型番
冷蔵庫	GR-G51FXV
洗濯乾燥機	TW-Z96X1
エアコン	RAS-632NDR
コントローラ	BTR-4010AZ
エコキュート	HWH-FB372C
分電盤メータリング	HEM-EM31A
蓄電池	ENG-B6630A1
シーリングライト	LEDH-LT1

る PC の仕様は表 2 のとおりである。また、表 3 にあげた実機の ECHONET Lite デバイスを LAN に接続し、状態の登録と照合の確認を行った (図 5)。

5.2 動作検証

図 6 はタグレイヤ重畳方式により追加されたモジュールにより、一般の Web ページに提案方式のダイアログが重畳され、ダイアログ中の照合実行ボタンの押下後、ホームサーバに問い合わせしている画面である。このとき、ホーム



図 5 プロトタイプで利用したデバイス

Fig. 5 Devices used in the prototype system.



図 6 プロトタイプの動作画面 (照合中)

Fig. 6 A screen of the prototype during inquiring approval.

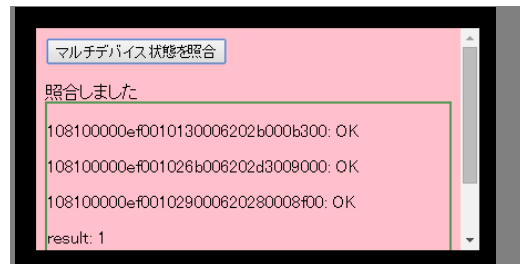


図 7 プロトタイプの動作画面 (結果表示の拡大)

Fig. 7 A screen of the prototype after receiving a result from the home server.

```

{"ref": "http://*****.html",
 "10810000ef0010130006202b000b300": "OK",
 "10810000ef001026b006202d3009000": "OK",
 "10810000ef001029000620280008f00": "OK",
 "result": "1"}
    
```

図 8 ホームサーバから受信した結果データ

Fig. 8 A received result data from the home server.

サーバでは今回用いた ECHONET Lite デバイスに状態を問い合わせ*2、登録された状態と同一であることを照合判定している。図 7 はホームサーバで判定した結果を表示した画面である。またこの表示の元になる、ホームサーバから受信した状態確認の結果を図 8 に示す。

*2 具体的には、エアコン (運転モード設定, 温度設定値), エコキュート (温度設定値, ON タイマ予約設定), シーリングライト (動作状態, 節電動作設定) の計 3 デバイス 6 状態を問合せしている。

以下、この結果の内容について説明する。まずフォーマットは汎用的に利用されている JSON 形式とした。“ref”は現在表示中の Web ページの URL である。それに続く、3つの“108100…”の項目は3つのデバイスに問い合わせた内容で、ECHONET Lite の電文をそのまま表している(青色部の“02”が同時に問い合わせる状態数、黄色・緑色部が状態数分の結果。詳しくはECHONET Lite仕様書[14]を参照)。この実行結果では、各デバイスに対し2つの状態を同時に問合せしている。そしてそれぞれの項目ごとに登録していた状態と一致するか否かの照合結果が“OK”/“NG”として表されている。これらがすべて“OK”であれば“result”は“1”，そうでなければ“0”となる。この実行結果では、すべて“OK”で“result”は“1”となっており、その結果をJavaScriptにより判断して、図7のように照合された旨のメッセージを表示している。ここでは確認用にこれらの表示をしているが、実際のサービスではJavaScriptで内部的に処理され、動画の視聴など元のサービスを実行する。また、1つでも“NG”があった場合は“result”が“0”となり、「照合されませんでした」というメッセージを表示し、画面の遷移を制限する。プロトタイプの実行例では動画再生のボタン(動画領域中央)の上に照合のダイアログを重畳表示しており、照合OKであればダイアログ消去、NGであればそのまま表示することにより再生の制御を行った。以上の一連の動作を確認し、提案方式による実現性を確認した。

5.3 性能評価

本節では、まずデバイスごとの状態問合せ時間について、実際のデバイスを用いて調査する。

5.3.1 計測方法

調査には、実際のECHONET Liteデバイスに対し前項で述べたプロトタイプのホームサーバのCGI処理プログラムを直接実行する。その処理時間をホームサーバ上のLinuxシェルコマンド“time”により計測し、5回の平均値をとる。また、ECHONET Liteデバイスから取得するデバイス数と状態数は、1回に問い合わせる状態数を1つから順にそのデバイスで取得可能な状態数まで1つずつ増やしながら繰り返し、それぞれの応答時間を計測した。

5.3.2 結果

今回用意したECHONET Liteデバイスのそれぞれの応答時間を表4と図9(デバイスごとの取得可能な最大状態数と平均応答時間)、図10(デバイスごとの問い合わせる状態数と平均応答時間)に示す。

デバイスごとに見ればほぼ比例の関係が見られる。一方で、デバイスの種類によって比例の傾きが異なっている。そこで最小二乗法によりデバイスごとの傾きと状態数1つの場合の応答時間を算出した(表5)。この値を参考に、デバイス状態数からおおむねの応答時間を予測することが可

表4 ECHONET Lite デバイスの応答時間*3

Table 4 Response time of ECHONET Lite devices.

デバイスタイプ	状態数 (Max)	平均応答時間(sec)	1状態あたりの応答時間(sec/状態)
コントローラ	2	1.34	0.67
冷蔵庫	13	3.51	0.27
エアコン	19	3.27	0.17
シーリングライト	19	2.91	0.15
洗濯乾燥機	21	5.13	0.24
分電盤	24	2.12	0.09
蓄電池	24	2.84	0.12
エコキュート	30	3.13	0.10

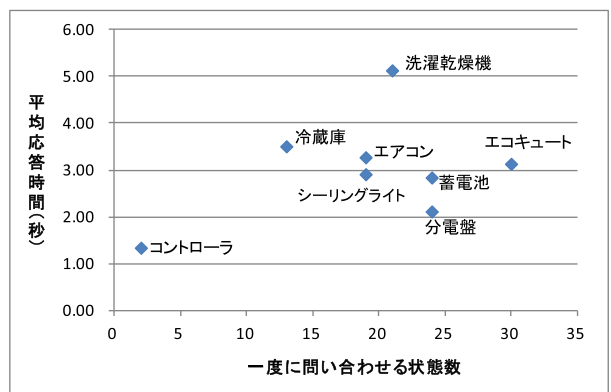


図9 問い合わせる状態数と平均応答時間

Fig. 9 Relations between requests for status and average of response time.

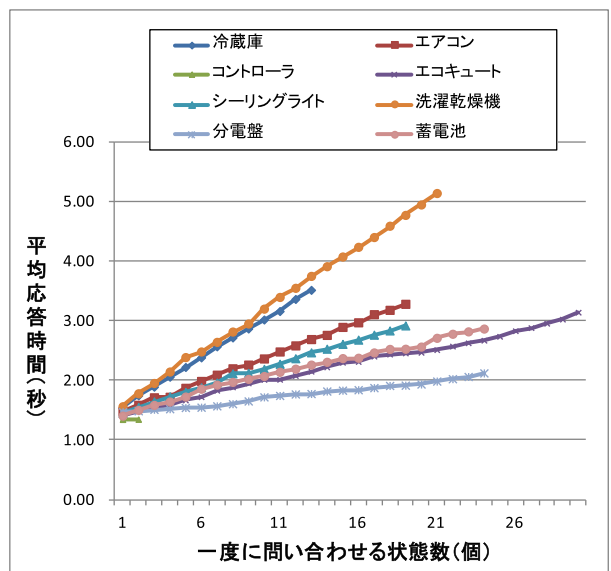


図10 問い合わせる状態数と応答時間

Fig. 10 Average of response time per a request for status.

能である。

*3 1つのデバイスタイプにつき、あるメーカーの1機種を用いて測定したものであり、他のメーカーの応答時間とは異なる場合がある。

表 5 状態数と応答速度の算出

Table 5 A response time of ECHONET Lite devices.

デバイスタイプ	Y 切片	傾き
コントローラ	1.337	0.002
冷蔵庫	1.447	0.153
エアコン	1.372	0.100
シーリングライト	1.397	0.080
洗濯乾燥機	1.421	0.176
分電盤	1.409	0.028
蓄電池	1.431	0.060
エコキュート	1.404	0.055
平均	1.402	0.082

5.4 考察

5.4.1 デバイス問合せの所要時間

アンケート結果から、登録したい状態数が6つ程度であることが分かった。そこで、最も時間を要する状況として、この6つの状態をそれぞれ異なるデバイスから1つずつ問い合わせ取得する場合の時間を算出してみる。表5で示した1状態取得あたりの平均応答時間の値(Y切片: 1.402 sec, 傾き: 0.082)を用いて6つの状態取得時間を算出すると、 $(1.402 + 0.082 \times 1) \times 6 = 8.904 \text{ sec}$ となる。すなわち、ユーザが求める状態数を得るために約9秒の時間がかかることになる。これに加えて、Webブラウザ内での処理時間、Webブラウザとホームサーバ間の通信時間なども含めて考慮しなければならない。この時間がWebページの更新時間だととらえると、ユーザインタフェースの分野で1つの指標とされている「10秒までならユーザの注意力は続く[15]」という基準を満たせるといえる。しかし、「3秒以内に表示されないとユーザはサービスから離れる[16]」という別の指標もあるため、さらなる時間の短縮が必要である。また、ここでは想定した5つのサービスでのアンケート結果から状態取得の時間を導いたが、それ以外のサービスでは6つ以上の状態取得が必要となり状態取得の時間が長くなる可能性がある。以上の考察から、(ア) デバイスの状態取得時間の短縮化と、(イ) ユーザがサービスから離脱しないための画面更新の2つの側面からの対処が可能であると考えられる。

まず(ア)については、ECHONET Lite プロトコルによるデバイスへの問合せ処理の並列化である。ECHONET Liteの規定では同時にアクセスすることは制限されていないため、ホームサーバ側でのスレッドによる並列処理により実現可能である。

(イ)については、既存のWebサービスで、応答の進捗状況などの情報を画面にしながら、バックグラウンドで状態を問い合わせる処理を実行する。表示する情報として、たとえば、電力使用量やユーザの直近の予定など有益な情報が望ましい。タグレイヤ重畳方式を用いて有益な情報を

サーバからのプッシュ方式で表示する方法については、著者らの先行研究[12]で実現しており、提案方式と組み合わせることが可能である。以上のことから、提案方式におけるデバイスへの問合せは、ユーザインタフェース分野の一部の指標は満たすものの十分ではなく、さらなる時間の短縮と表示の工夫が必要であることが分かった。そしてその対処方法として、デバイスへの問合せ処理の並列化と、バックグラウンド処理により問合せを行い、その間に有効な情報をユーザに提示する方法をあげた。これらの方法は実現可能性が高く、設計指針③のユーザがサービスから離脱しないように処理を完了することが可能であるといえる。

5.4.2 提案方式の効果

提案方式が様々なマルチデバイス環境に対応できることの効果について考察する。ただし、Webブラウザからデバイスへのアクセスはホームサーバでの機能などにより対応できていることとする。

提案方式を用いずに、Webサービスに合わせてデバイスの状態を変更するためには、各Webサービス提供者に対して各家庭のデバイスとその設定をユーザに登録してもらう仕組みが必要である。そして、そのうえでユーザは所有するデバイスを想起し、好みの状態に登録する。このとき、サービス提供者側で想定されていないデバイスやその状態には対応できない。つまり、サービス提供者側で各家庭につながるデバイスを網羅して準備し、日々更新しなければユーザの望むデバイスの状態に登録できない。また、すべてのWebサービスでこのようなサービスを提供することは現実的ではないため、ユーザが望むあらゆるWebサービスに対応するとはいえない。

一方提案方式では、ホームサーバで家庭内LANに存在するデバイスがECHONET Liteプロトコルにより把握できるため、あらゆるデバイスを網羅的に管理する必要はなく、サービス提供者にとって負担がない。また、Webサービス提供者の対応は不要であり、ユーザが望む任意のWebサービスに対応できる。

6. まとめと今後の課題

任意のWebサービスに複数のデバイスの状態をユーザが関連づけ、そのサービスとデバイスの状態の同一性を担保して、サービスを利用する仕組みを提案した。実現にあたり、提案方式の利用意向やデバイス状態の選択数をユーザアンケートにより抽出し、そのときユーザが利用しているWebサービスの継続性を維持するため、Webブラウザにおいてデバイス状態を確認できること、画面遷移しないこと、ユーザがサービスから離脱しないように処理を完了することを設計指針とした。それに従って、アーキテクチャを示し、プロトタイプにより実現性を確認した。さらに、ユーザアンケートで得られた登録したいデバイス状態数について実際のデバイスへの問合せ所要時間を測定し、ユーザイ

インタフェースの指標を用いて考察した。その結果、ユーザインタフェース分野の一部の指標は満たすものの十分ではなく、デバイスへの問合せ処理の並列化と、バックグラウンド処理中のユーザへの情報提示により、ユーザがサービスから離脱しないように処理を完了することが可能であると考察した。

今後は、登録のためのユーザ対話画面も含め、デバイス状態問合せ中の画面表示に関わるユーザビリティ評価を進める。

参考文献

- [1] エコネットコンソーシアム：ECHONET CONSORTIUM, 入手先 (<http://www.echonet.gr.jp/index.htm>).
- [2] dlina-for-industry: Digital Living Network Alliance, available from (<http://jp.dlina.org/>).
- [3] Duquennoy, S., Grimaud, G. and Vandewalle, J.-J.: The Web of Things: Interconnecting Devices with High Usability and Performance, *ICESS '09. International Conference on*, 2009, pp.323-330 (2009).
- [4] Luigi, A., Iera, A. and Morabito, G.: The internet of things: A survey, *Computer Networks*, pp.2787-2805 (2010).
- [5] 西垣弘二, 安本慶一, 柴田直樹, 伊藤 実: コンテキストに基づいた情報家電の連携を実現するためのフレームワークおよびルールベース言語の提案 (フレームワーク), 情報処理学会研究報告 UBI, Vol.112, pp.21-27 (2004).
- [6] 前田 潤, 福田健一, 木下和彦, 村上孝三: 情報の連想的結合によるコンテキストウェアサービス制御方式, 電子情報通信学会論文誌 B 通信 (1), pp.22-34 (2008).
- [7] HEMS (ECHONETLite) 認証センター: HEMS (ECHONETLite) 認証センター, 入手先 (<http://sh-center.org/>).
- [8] W3C HTML Working Group, available from (<http://www.w3.org/html/wg/>).
- [9] 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, コンシューマ・デバイス&システム, Vol.2, No.2, 情報処理学会論文誌論文誌トランザクション, pp.19-28 (2012).
- [10] 松村剛志, 安川健太ほか: Web とホームデバイスをつなぐプラットフォームによる新しいユーザ体験, 電子情報通信学会技術研究報告情報ネットワーク, 110, pp.181-186 (2011).
- [11] 渡部智樹, 青木良輔, 小林 透, 小林 稔, 一色正男: Web 閲覧と連動したアンビエントな家電操作方式の提案, 情報処理学会論文誌コンシューマ・デバイス&システム (CDS), Vol.2, No.2, pp.73-80 (2012).
- [12] Watanabe, T., Mochizuki, R., Kobayashi, T. and Isshiki, M.: HACCS: Home Appliance Control Concierge System: Extending Functions on Web Service, *COMPSAC 2013*, pp.208-213 (2013).
- [13] Model B | Raspberry Pi, available from (<http://www.raspberrypi.org/product/model-b/>).
- [14] APPENDIX ECHONET 機器オブジェクト詳細規定 Release E: エコネットコンソーシアム (2014).
- [15] Web サイトの応答時間, 入手先 (<http://www.usability.gr.jp/alertbox/20100621.response-times.html>).
- [16] CROSS-STYLE—ウェブデザイン講座「8秒ルールは今昔!? 進化した「3秒絶対ルール」とは?」, 入手先 (<http://www.cross-style.com/webdesign/006.html>).



渡部 智樹 (正会員)

1992年横浜国立大学電子情報工学科卒業。同年日本電信電話(株)入社。NTTヒューマンインタフェース研究所にて、放送通信連携に関わる大規模データ集配信技術の研究開発に従事。現在、NTTサービスエボリューション研究所に勤務し、家電制御に関わる研究開発に従事。電子情報通信学会会員。



高嶋 洋一

1990年横浜国立大学大学院電子情報工学専攻博士課程後期修了。博士(工学)。同年日本電信電話(株)入社。現在NTTサービスエボリューション研究所主任研究員。画像符号化、情報セキュリティに関する研究を経て、著作権保護システムやWeb誘導サービスの研究開発に従事。ITU-T等の標準化活動にも寄与。電子情報通信学会会員。



杉村 博 (正会員)

2007年神奈川工科大学大学院情報工学専攻博士前期課程修了。2012年同大学院同専攻博士後期課程修了。2012年神奈川工科大学スマートハウス研究センター特別研究員。2013年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助教。ホームネットワークやHEMSに、人工知能技術を応用する研究に従事。人工知能学会, 電気学会, IEEE各会員。



一色 正男 (正会員)

東京工業大学大学院理工学研究科修了。2009年より慶應義塾大学教授, 2012年より神奈川工科大学教授, スマートハウス研究センター所長。情報処理学会CDS研究会幹事(2010~2012年)。機械学会会員, ECHONETコンソーシアム2008運営委員長, 現フェロー。W3C Site Manager(2009~2014年)。経済産業省HEMSタスクフォース座長。HEMS認証支援センター長。