# アンコンシャス情報表示技術に関する研究

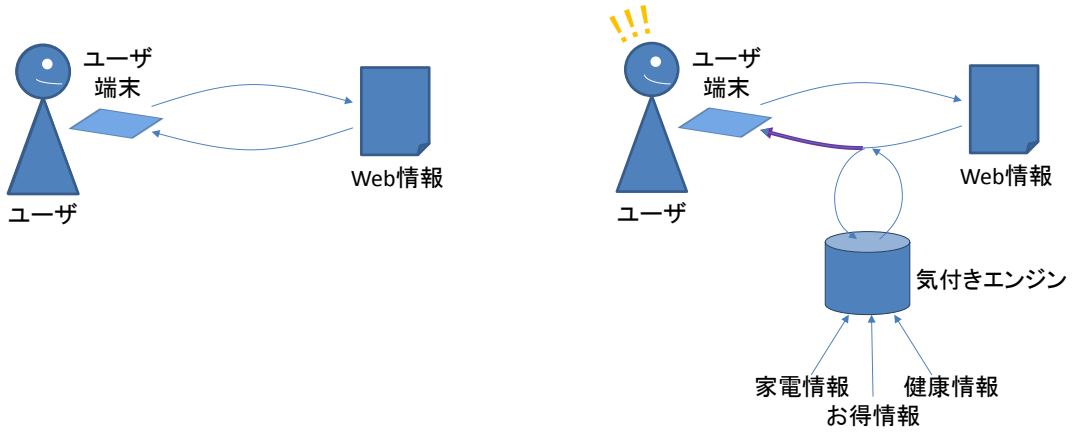
神奈川工科大学
電気電子工学専攻

渡部智樹

平成２６年度

# 概要

本研究は，インターネットと家電により人々の暮らしを豊かにする世界の実現を目指し，初めて社会実装可能なシステム基盤提供を目指した．現在，家電の状態確認や操作実行はユーザが意識してリモコンなどを操作する必要があり，意識していない時に役立つ情報（アンコンシャス情報）に気付かず，家電を適切に利用できていないという重大な損失課題があった．そこで，ユーザが日常利用している Web に着目し，アンコンシャス情報に気付かせる新たな方式として「気付き表示方式」を開発し提案した．




図．従来方式（左）と提案方式（右）の Web アクセス

本方式を実現するシステムを，周囲の家電情報とインターネットの情報を融合してアンコンシャス情報を生成する気付きエンジンと，アンコンシャス情報を Web 上に重畳表示する気付き表示エンジンの２つの構成要素を基盤設計することで実現した．社会実装可否を検討し，昨今次々と登場する新たなデバイスへの対応が実装規模や処理負荷の観点から難しくなると判断し，分離構成により気付きエンジンの処理負担を分散し，非力なユーザ端末でも多様なデバイスに対応可能とした．また，複数のユーザ端末が同一 LAN 内に存在するとデバイスへの重複アクセスが発生し，デバイス側の処理負担が発生するため，気付きエンジンをサーバとして機能させることで，複数のユーザ端末と複数のデバイスとを仲介する技術を設計し，それぞれ

の処理負荷低減を図った.

システム実装の基礎技術として，Web 表現技術(CSS)，Web 制御技術(JavaScript)，Web 文書解析技術(DOM)，Web 通信技術(HTTP)を利用した．特に，HTML5 技術を適用し，Web 文書解析の高度化拡張と Web 通信のリアルタイム化(WebSocket)を行い, 統合調整することで実現した．そして各種のプロトタイプ実装によりそれぞれの技術の採用可否を判断し，多様な場面にアンコンシャス情報を応用展開可能なシステム基盤として設計した．本論文では，家電の操作，複数の家電設定，家電状態の変化通知の３種のアンコンシャス情報に対して本基盤設計が具体的に対応できることを検証し，それぞれの要件に対するシステム評価について，以下の章立てにて述べた.

第１章では，アンコンシャス情報に着目した背景や目的について説明した.

第２章では，アンコンシャス情報表示技術＜設計指針＞について，実現するサービスの具体例を交えて，実現の方針とアプローチを示した．従来技術では任意の Web ページに組み込めなかったアンコンシャス情報を，HTTP 通信と DOM，CSS の各技術を拡張した家電情報重畳技術を研究開発することで実現し，さらにアンコンシャス情報の通信方式を WebSocket 化することでリアルタイムな表示を可能とした．これらの技術の実現にあたり，利用中の Web サービスを阻害しないという本研究独自の要件を挙げ, 以下の章にて実現技術を解説しながら可能性を示した.

第３章では，アンコンシャス情報表示技術＜Ⅰ．基本形・操作＞について実装検証した．まずシステムの基本形となるアーテキテクチャとして，Web ページ内の興味あるキーワードに気付かせ即座に家電を操作実行できる家電操作タグを重畳表示する家電情報重畳技術を確立し，プロトタイプを実現した．気付きエンジンは，ユーザが興味のあるキーワードをインターネットから取得し, 画面表示された領域内のキーワードにだけ家電操作タグを表示するようにキーワード抽出の機能を設計し研究した．この技術を実装したプロトタイプを構築し，評価実験により時間的有効性を示した.

第４章では，アンコンシャス情報表示技術＜Ⅱ．複数の家電設定＞について家電情報重畳技術を拡張する手法を開発し，実装検証した．利用する Web サービスごとに周囲の家電の状態

を登録・照合するマルチデバイス照合技術仕様を明らかにし，次回利用時に意識しなくても状態を再現する技術を確立した．複数の家電設定を再現するアンコンシャス情報を生成する技術を気付きエンジンに追加実装し，第３章の気付き表示エンジンを拡張したプロトタイプを構築した．複数家電へのアクセス時間の観点から最適化検討し，一度に取得する状態数と所要時間の関係を実機の家電８種を用いた計測実験により導き，利用中の Web サービスに影響しない仕様を示した．

第５章では，アンコンシャス情報表示技術＜Ⅲ．家電状態の変化通知＞について実装検証した．家電の状態変化に応じたアンコンシャス情報を表示することで，ユーザは家電の状態変化やなすべき行動を意識し続ける必要がなくなる．気付きエンジンを拡張し，状態変化にリアルタイムに気付きを与える HTTP 通信の WebSocket 化を図った．洗濯乾燥機の脱水運転完了時のプロトタイプを構築し，アンケートの結果，天気予報と乾燥運転の電気代が外干しを促す節電情報として役立つアンコンシャス情報であることを確認した．さらに Web サービスの長時間利用時も家電を意識しなくてすむ Web 要素通知技術を気付きエンジンに実装し，実現可能性を示した．

第６章では，結論として本研究で得られた成果を要約し，事業性の観点から技術的に普及可能であり，幅広い分野に応用することで様々な場面における活動を支援できることを示した．

以上のことから，本研究で開発した「アンコンシャス情報表示技術」は，実社会で実装可能な初めてのシステム基盤であり，既存サービスへの容易な展開基盤であることは，誰でもサービスを享受し得るという点で価値ある成果になった．特に，省エネや健康促進など日常生活において実行すべき適切な行動を意識することなく自然と実行できる有効な技術であることを明らかにした．そして，この技術が世界中に広がり，人々が豊かに暮らすために必要かつ有効であることを示した．

# 目次

# 第1章　序論

## 1.1 背景

昨今の家庭内へのネットワーク化は目覚ましいものがある．1つの側面は家電やモノのネットワーク化であり，もう1つの側面はスマートフォンやタブレットなどユーザが手元で扱うユーザ端末の増加である．これらについて順に説明する．

家庭内のネットワーク化の状況について，黒物家電（AV機器），白物家電（生活家電や環境センサ），IoT（Internet of Things）の順に説明する．まずTVやHDDレコーダなどの黒物家電においては，2011年7月の地上デジタル放送への完全移行により，番組の録画予約機能や他の機器との連携など，便利な機能が提供されるようになった．例えば，DLNA（Digital Living Network Alliance） [1]やAirPlay [2]といった規格に対応し，ネットワークを経由して映像や音楽を手軽に楽しめるようになった．次に，エアコンや洗濯機，あるいは温度計や湿度計といった白物家電やセンサの分野においても，ECHONET Lite [3]がHEMS（Home Energy Management System） [4] [5]の標準プロトコルとして2012年に日本国内で認定されたことを受けて対応家電の普及が進んでいる [6]．特にHEMSは，2011年の東日本大震災の影響により省エネや畜エネ，さらに創エネといった取り組み [7]が進められ，今後ますます普及が加速すると期待されている．さらには，IoT [8] [9]あるいはWoT（Web of Things） [10] [11]と呼ばれる「モノのインターネット」が着目されており，身の回りのあらゆるものがネットワークにつながり，そのネットワークを介して情報を取得したり制御を行ったりすることが将来的に可能になると言われている．例えば，ポットをネットワークにつないでその使用状況から生活の様子を遠隔地に住む家族に伝える製品 [12]が発売されている．また，玄関の扉やタンスの引き出しにセンサを取り付けてその動作から人の動きを捉え，外出する人に鉄道の遅延情報などを伝えるといった家具のインテリジェント化も提案されている [13]．このように，

今後ますます多種多様な機器やモノがネットワークにつながり，ユーザ端末からアクセスできるようになる．

一方，ユーザが利用するユーザ端末については，スマートフォンやタブレットの普及 [14] により，いつでもどこでも手元のユーザ端末から家電へのアクセスが可能となった．メールやブラウザ，SNS などのアプリは家中どこにいても利用され [15]，ユーザはスマートフォンを肌身離さず利用している．このようなユーザ端末を用いて，家電を自宅や外出先からリモコン操作するアプリなどが提供されている [16] [17] [18]．

以上に説明したように，ユーザ端末を使って様々な情報を取得し，周囲の家電やモノとのやりとりができる物理的な環境は整いつつある．

## 1.2 社会的な問題

インターネットの普及や端末の小型化などにより，人は様々な状況において情報を取得できるようになった．家にいるときでも Web やメール，SNS などを使って情報取得しており，ユーザ自ら主体的にアクセスして情報を得ることができる．しかし，ユーザが知るべき情報や知っておくとよい情報など，ユーザが意識していなかった情報にはたどり着けない．つまり，環境はネットワークにつながり様々な情報が取れるようになったが，主体的に取りにいかないと大事な情報を入手できない．

世界中で電力利用の削減と効率化が強く求められている．特に日本では 2011 年 3 月の東日本大震災の影響により，企業だけでなく一般ユーザにも節電の意識が高まっている [19]．HEMS を使えば電力情報を見ることはできるが，ユーザが自ら主体的にアクセスして閲覧しなければならない．その時ユーザが何らかの刺激を受けて節電しようと意欲がわいたとしても，そのための行動が煩わしいものだと意欲が低下し実行しそこねてしまう可能性がある．また，健康を意識する情報を目にした際に，思い立った行動をすぐに実行できないとその意識は低下してしまう．例えば，週 1 回 30 分のランニングをすることが健康に良いという情報に刺激を受け，週末の予定に書きこもうとカレンダーを取り出そうとしても時間がかかったり面倒であったりするとその意識は損なわれてしまう．あるいは，洗濯をしていることを忘れてしまい，

終わっていることを知らせる音にも気づかず意識されないまま放置されてしまう情報もある．

このように手間がかかったり忘れてしまったりするといった問題は，高齢化社会を迎えた日本 [20]では高齢者の増加に伴い顕在化する．また共働きや子育てをしている働き世代にとっても，家庭内で忙しくしているときにあれこれと気にしていることは心理的に負担が大きく，知らなかったあるいは気づかなかった役立つ情報（アンコンシャス情報）を知るすべがない．

以上のように，家庭内における様々な世代において，有益なアンコンシャス情報があっても知る機会を喪失しているという問題があった．

## 1.3 研究の目的と方針

本研究は，有益なアンコンシャス情報をユーザが逃すことなく見ることができるアンコンシャス情報表示技術の研究開発を目的とする．この目的を達成するためには，ユーザの様々な状況に応じて適切なアンコンシャス情報を生成することと，そのアンコンシャス情報をユーザが逃すことなく表示することの２つの課題がある．

１つ目の課題であるアンコンシャス情報の生成については，クラウド上ですでに提供されている WebAPI [21]を利用して取得する情報と，家庭内にある家電やセンサの情報とを統合して取り扱う．そしてユーザの状況あるいは家電やセンサなどの周囲の状況に応じて，ユーザに気付きを与えるアンコンシャス情報を生成する．このようなユーザに気付きを与えるアンコンシャス情報の生成処理を「気付きエンジン」と呼び機能を実装する．

次に２つ目の課題であるアンコンシャス情報を表示する方法について，本研究では Web に着目する．ユーザが閲覧している Web にアンコンシャス情報を重畳して表示することにより，アンコンシャス情報の存在を認識し，その内容に容易にアクセスできるようにする．ユーザが新たな情報を得る機会は Web による閲覧時が多いことや，HTML5 [22]が W3C により勧告[23]されたことを受けて現在提供されているアプリが Web に移行すると考えられる [24]こと，そして TV も HybridCast [25]などのサービスにより Web と容易に連携可能になり身近な表示デバイスの Web 化が進行していること [26]を考慮し，Web を使うことを着想した．さらに，Web の閲覧を極力邪魔しないように，閲覧中の Web にアンコンシャス情報を重畳して表示す

るアイデアに至った．このように既存の Web にアンコンシャス情報を新たに追加し表示させる処理を「気付き表示エンジン」と呼び機能を実装する．

## 1.4 本論文の構成

以下，第２章では，アンコンシャス情報表示技術＜設計指針＞について，実現するサービスの具体例を交えて，実現の方針とアプローチを示した．従来技術では任意の Web ページに組み込めなかったアンコンシャス情報を，HTTP 通信と DOM，CSS の各技術を拡張した家電情報重畳技術を研究開発することで実現し，さらにアンコンシャス情報の通信方式を WebSocket 化することでリアルタイムな表示を可能とした．これらの技術の実現にあたり，利用中の Web サービスを阻害しないという本研究独自の要件を挙げ，以下の章にて実現技術を解説しながら可能性を示した．

第３章では，アンコンシャス情報表示技術＜Ⅰ．基本形・操作＞について実装検証した．まずシステムの基本形となるアーテキテクチャとして，Web ページ内の興味あるキーワードに気付かせ即座に家電を操作実行できる家電操作タグを重畳表示する家電情報重畳技術を確立し，プロトタイプを実現した．気付きエンジンでは，ユーザが興味のあるキーワードをインターネットから取得し，画面に表示されている領域内のキーワードにだけ家電操作タグを表示するようにキーワード抽出の機能を設計した．この技術を実装したプロトタイプを構築し，評価実験により時間的有効性を示した．

第４章では，アンコンシャス情報表示技術＜Ⅱ．複数の家電設定＞について家電情報重畳技術を拡張する手法を開発し，実装検証した．利用する Web サービスごとに周囲の家電状態を登録・照合するマルチデバイス照合技術仕様を明らかにし，次回利用時に意識しなくても状態を再現する技術を確立した．複数の家電設定を再現するアンコンシャス情報の生成技術を気付きエンジンに追加実装し，第３章の気付き表示エンジンを拡張したプロトタイプを構築した．複数家電へのアクセス時間の観点から最適化検討し，一度に取得する状態数と所要時間の関係を実機の家電８種を用いた計測実験により導き，利用中の Web サービスに影響しない仕様を示した．

第５章では，アンコンシャス情報表示技術＜Ⅲ．家電状態の変化通知＞について実装検証した．家電の状態変化に応じたアンコンシャス情報を表示することで，ユーザは家電の状態変化やなすべき行動を意識し続ける必要がなくなる．そこで，気付きエンジンを拡張し，状態変化にリアルタイムに気付きを与える Web 通信の WebSocket 化を図った．洗濯乾燥機の脱水運転完了時のプロトタイプを構築し，アンケートの結果，天気予報と乾燥運転の電気代が外干しを促す節電情報として役立つアンコンシャス情報であることを確認した．さらに Web サービスの長時間利用時も家電を意識しなくてすむ Web 要素通知技術を気付きエンジンに実装し，実現可能性を示した．

第６章では，結論として本研究で得られた成果を要約し，事業性の観点から技術的に普及可能であり，幅広い分野に応用することで様々な場面における活動を支援できることを示した．

# 第2章　設計指針

本章では，本研究の設計指針を明確にするために，まず本研究が目指すアンコンシャス情報表示技術がもたらす便利で豊かな生活について具体例を挙げてイメージを示す．そして，従来技術で達成できていない取り組むべき課題を明らかにし，解決のためのアプローチを示す．

## 2.1 サービスイメージとアンコンシャス情報

本節では，便利で豊かな生活を実現するアンコンシャス情報の具体的なイメージを３つ示す．

**(S1) Web の興味ある内容に気付かせ即座に操作可能とするアンコンシャス情報の例**

インターネットでお気に入りのタレントのブログ（Web ページ）にアクセスすると，そこには TV 番組の出演情報が掲載されていた．文字が多くてすぐには気付かなかったが，「番組予約」のボタンがアンコンシャス情報として表示されたのでその出演番組に気付き，アンコンシャス情報のボタンを押すだけで HDD レコーダの番組予約を行うことができた．

このように，ブログにアクセスするときユーザは番組予約することは意識していなかったが，アンコンシャス情報が先回りしてユーザに気づかせ簡単に番組予約を実行する機能を提供する．

**(S2) 利用する Web サービスに応じて複数家電状態を設定するアンコンシャス情報の例**

Web で動画を視聴するときは，外からの光が入らないようにカーテンを閉め，シーリングライトを消して部屋を暗くし，音声や字幕のモードを変える，といったようにいつも決まった状態に周囲の家電を設定している．プロジェクタとスクリーン，5.1ch スピーカまで準備する場合もある．Web の作業に集中したいときは，TV の電源を OFF にし，洗濯機やエアコン，ロボット掃除機を静音モードにするといった環境設定を行うが，個別に実行する作業は手間がかかり，すぐに集中できる環境を作ることが難しい場合もある．このように，いくつもある周

囲のデバイスの設定を１つ１つ操作するのは大変である．アンコンシャス情報は Web を利用するそれぞれの場面で複数の家電を設定するワンタッチ実行ボタンを提供する．これを押すだけで，利用する Web に対して設定すべき周囲の家電の状態設定に気付き，簡単に実行することができる．

**(S3) 家電の状態変化を関連情報と合わせて通知するアンコンシャス情報の例**

洗濯乾燥機で脱水運転が終わって乾燥運転に移行する前に，ユーザが見ていたスマートフォンに，脱水運転完了とともに，今日これからの天気が晴れであることや乾燥運転の電気代も合わせてアンコンシャス情報として表示する．

このようなアンコンシャス情報により，ユーザは洗濯運転が完了したことに気付かされるのと同時に，天気が良いので外干しできること，そして電気代を節約できることを知る．天気予報や電気代はその場で調べれば分かることだが，その手間を省き気付かせることで外干しが促進され，結果として節電につなげることができる．

以上に示したサービスイメージを実現するために，それぞれで述べたアンコンシャス情報を生成し，ユーザに提示する．図 2-1 は上記に示した各サービスイメージにおいて，ユーザが意識している情報，意識していなかったアンコンシャス情報，そしてアンコンシャス情報をもとに実行可能な対処行動を簡略図で示したものである．

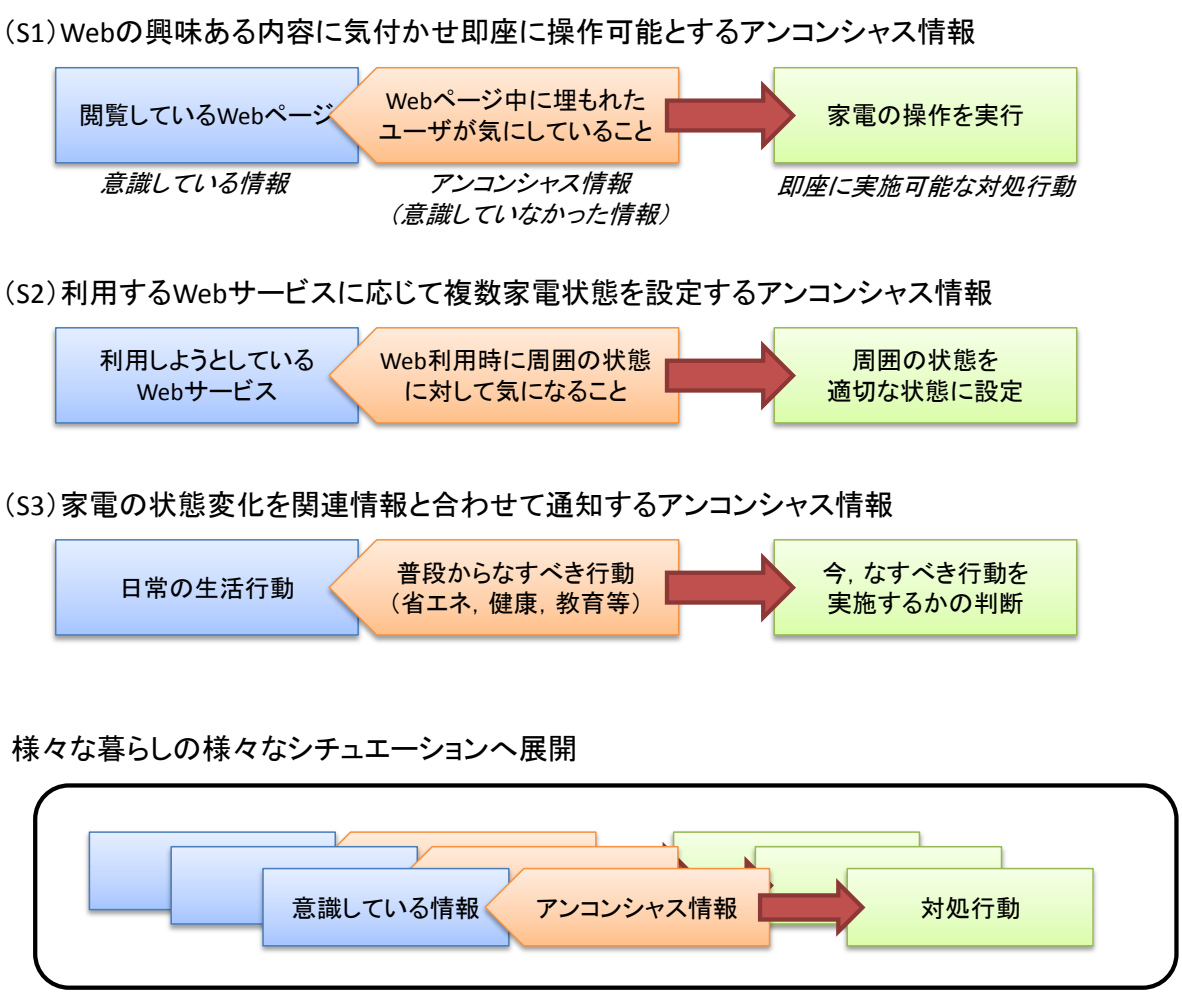



図 2-1　サービスイメージにおけるアンコンシャス情報

## 2.2 基本アーキテクチャ

2.1 節に挙げた例は個々に役立つサービスや機能であるが，アンコンシャス情報を表示する共通の基本アーキテクチャとして設計・実装することにより，様々なシチュエーションに展開することが可能となる．

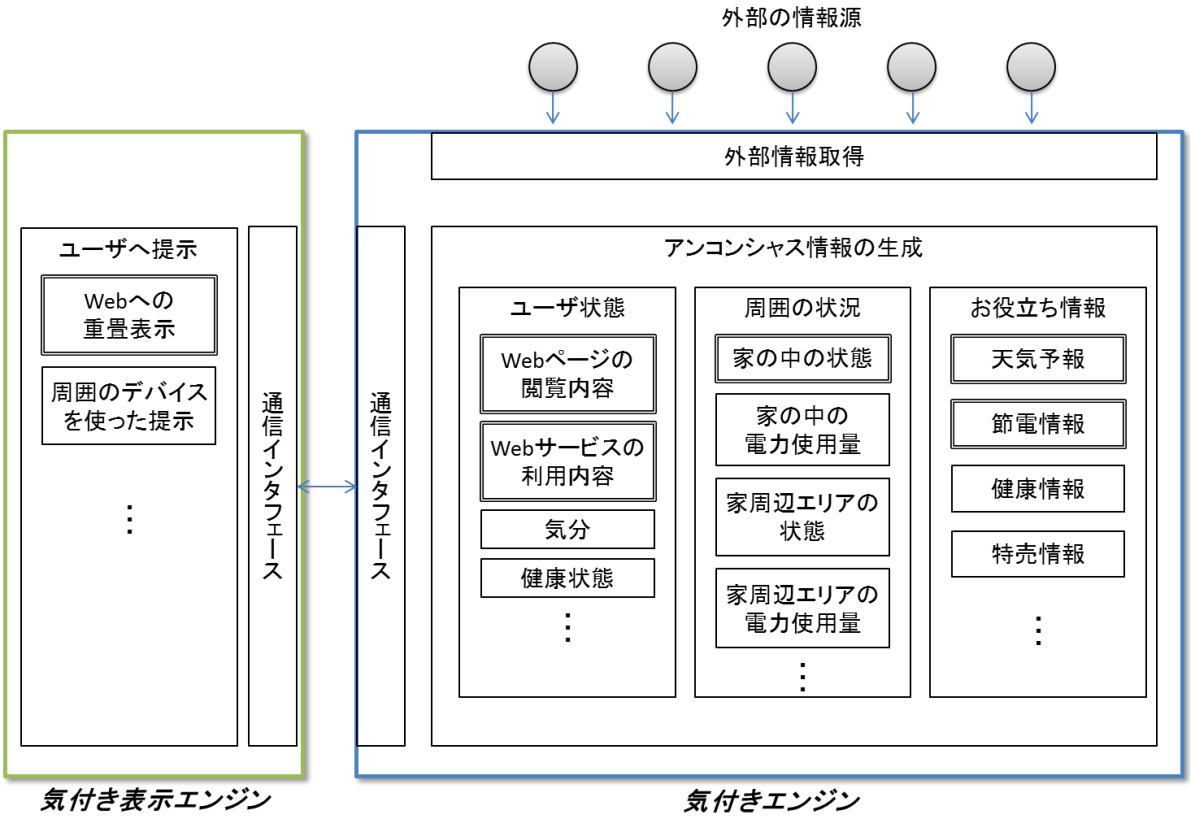



図 2-2　アンコンシャス情報表示の基本アーキテクチャ

図 2-2 は，様々な状況に対応し，有益なアンコンシャス情報を取り込み，ユーザに表示するためのアーキテクチャである．図のとおり，「気付きエンジン」と「気付き表示エンジン」の２つの要素から構成する．気付きエンジンは，ユーザの状態や周囲の状況に関する情報をもとに，適切な情報源から表示すべきアンコンシャス情報を取得し，気付き表示エンジンに提供する通信インタフェースを備える．一方の気付き表示エンジンは，ユーザが気付き，即座に対処行動を実行できるように表示する機能を備える．

図 2-2 に示すアーキテクチャにより，様々な情報源の情報をもとに状況を判断し，様々な役立つアンコンシャス情報を生成することを可能とする．表示についても通信インタフェースを通じてネットワークにつながる適切な表示デバイスを使って表示できる仕組みとした．他のデバイスと連携して表示する方法としては，HTML 文書を分割して複数のデバイスで表示する著者らの研究 [27]がある．なお本論文では，図中の二重線枠の箇所について実装し，それぞれ評価を行った．

## 2.3 設計要件と実現方法

前節で述べた基本アーキテクチャに基づき，システム実現するための設計要件として以下の３項目を設定した.

**設計要件①：状況に応じて適切な情報を組み合わせてアンコンシャス情報を生成すること**

**設計要件②：ユーザがアンコンシャス情報に気付き対処行動を簡単に実行できること**

**設計要件③：ユーザが利用する／している元のサービスを邪魔しないこと**

設計要件①において，アンコンシャス情報に含めるべき情報はユーザあるいは状況によって異なる．そのため様々なユーザや状況に対応できるように気付きエンジンにより様々な情報を取捨選択できるようにする．設計要件①を実現するために利用可能な情報源が WebAPI として多数提供されている．例えば，天気予報，テレビ番組の詳細情報，家電の機能を提供するものなどが存在する．また文献 [28]の研究では，各家電の機能を Web サービスとして利用できるシステムを開発している．このシステムは，家電機器に対する操作だけでなく，宅内に設置された環境センサへのアクセスも WebAPI として提供している．このような WebAPI を通じて様々な情報を取得することができる．設計要件①を実現するためには，これらの情報の中から，ユーザがまだ意識しておらず，気付きエンジンは役に立つ情報をアンコンシャス情報として抽出しなければならない．そのため，アクセス先の Web に関する情報，周囲の環境に関する情報，ユーザの身体的あるいは精神的な状態に関する情報などを用いて判断する必要がある．例えば，シャツを着るだけで心拍数を取得できるセンサ [29] [30]が開発されており，将来的にユーザの状態推定への活用が期待される．本研究の気付きエンジンでは，アクセスした Web 情報と周囲の環境情報の２種類の情報を用いて判断させた.

設計要件②については，ユーザがアンコンシャス情報を見て気付きを得た時に，なすべき行動がすぐに実行できないと忘れたり時間が経ってしまったりして気付きが意味をなさなくなることに対する要件である．なすべき行動を意識したら，家電を操作したり自ら行動したりし

てすぐに実行できるようにする．確実にユーザが知覚できる表示を行うためには，ウェアラブルデバイスや環境センサを用いてユーザの居る位置や姿勢を把握し知覚可能な表示デバイスを選択・表示する方法や，文献［31］[32]のように頭部装着側ディスプレイ（HMD: Head Mounted Display）を用いてユーザの視覚を奪う方法がある．ウェアラブルデバイスを用いた研究は盛んであるが，位置と姿勢を特定できたとしてもユーザが見ている方向に表示可能なデバイスがあるとは限らない．また音などのモダリティを使った表示も可能であるが，聞いて操作を指示するといったインタラクションが煩雑となり適切ではない．また，HMD を日常生活で装着するのは現実的ではないが，メガネ型の表示装置であれば適用できる可能性がある．ただし，表示できる情報量や操作指示の方法の面で課題が残る．本研究ではユーザの何らかの情報獲得の場面に着目した．緊急性の高い情報は，警報アラームや TV への字幕など，様々な手段があるが，本研究ではそのような場面ではなく，ユーザが獲得した情報に付加価値を与えるものを想定した．ユーザが情報獲得する手段として現在の主流は Web であり，そのときユーザは Web を注視しているはずであるから，その Web の中に表示するのが望ましいと考えた．

設計要件③はアンコンシャス情報が多用され，ユーザが意識して行っている作業を阻害すると提案方法が使われなくなる恐れがあるため要件として設定した．作業の阻害要因としては，時間的な要因と視覚的な要因を考えた．具体的には，Web へのアンコンシャス情報の表示処理において，Web 内に表示する場所とその処理時間が本来の Web アクセスに影響しない範囲に収めることを目標とする．視覚的に影響するアンコンシャス情報の表示場所については，元の Web サービスの表示内容やレイアウトを変えることなく，アンコンシャス情報を重畳する数やタイミングを制御する．

以上の要件を満たすためには，気付きエンジンと気付き表示エンジンをネットワーク上に適切に配置し実装する必要がある．以降ではこれら２つのエンジンを実装するための実現構成について説明する．

## 2.4 実現構成

気付きエンジンと気付き表示エンジンの配置についてそれぞれ分けて説明する．

### 2.4.1 気付きエンジンの実現構成

気付きエンジンは，天気予報などインターネット上の WebAPI だけでなく家庭内の家電も扱うため，家電にアクセスできる場所に配置されなければならない．そのため，家庭内 LAN に配置するのが一般的な解決手段である．この場合新たな装置を家庭内 LAN に設置しなければならず，ユーザの手間とコストがかかりサービス導入の障壁となる可能性がある．すでに HEMS サービスを利用している場合，専用の装置が家庭内 LAN 上に配置されているため，この装置に追加実装できる仕組みを設ける方法がある．あるいは，インターネットサービスで用いるホームゲートウェイ（HGW）と呼ばれる，家庭内とインターネットを結ぶルータに必要最低限の機能を配置し，残りの機能をクラウドに配置するというやり方で実現することもできる．HGW に特別新たなハードウェアを装備することなく，サービス提供者側からの操作で HGW 内部に機能を追加できるため有効な手段である．ただし，クラウドから各家庭の家電情報にアクセスできる可能性があるため十分なセキュリティ対策が求められる．

### 2.4.2 気付き表示エンジンの実現構成

気付き表示エンジンを既存の Web ページに適用する方法としては３つに大別できる．１つ目は Web を表示するブラウザ内にその機能を備える方法，２つ目は Web アクセスの流通過程で追加する方法，そして３つ目は Web サーバ側で追加する方法である．これら３つの表示エンジンの適用方法について順に説明する．

#### （適用方法１）Web ブラウザへのアドオン方式

気付き表示エンジンを Web ブラウザに機能拡張としてアドオンする方法がある．ユーザが利用するブラウザは概ね１つか２つであると想定されるため，それが機能追加できるブラウザであれば一度の機能追加の作業を実施するだけで適用できるというメリットがある．しかし，利用するブラウザによって機能を追加する手段が異なる，あるいは機能追加できない場合があるため，汎用的に適用できる手段とは言えない．

**（適用方法２）HTTP プロキシ方式**

２つ目の方法はブラウザとサーバの間のネットワーク上に存在するルータや HTTP プロキシなどの装置により追加を行う方法である．具体的には，ブラウザからの要求に応答した Web ページの内容をこのプロキシにおいて展開し，気付きエンジンのモジュールを追加しブラウザに送信する．気付きエンジンで述べた HGW にさらに気付き表示エンジンの追加機能を備えれば実現できる．実装評価した事例としては文献 [33]などがある．ただし，Web アクセスの流通過程で送受信されるデータに処理を施すため，事前にユーザやサーバを運用する情報提供者等への承諾を得る必要がある．

**（適用方法３）サーバでの事前組込み**

３つ目の方法は情報提供者側のサーバで処理する方法である．ユーザの環境に応じたアンコンシャスな情報を提供するためにはユーザの宅内環境の情報をサーバ側にアップロードしなければならない．多数のユーザを対象とした場合，この方法はサーバやネットワークに甚大な負担をかける可能性がある．またユーザのプライバシーに関わる情報も含まれるため，ユーザにその承諾を得たり，厳重にデータを管理したりするためのコストの面からサービス提供者の負担が大きくなる．

以上３つの適用方法について整理した内容を表 2-1 に示す．いずれのパターンでも技術的には実現可能であるが，サービス導入の観点から差がある．例えば，通信事業者がサービスを提供する場合は，HGW 上に気付きエンジンと気付き表示エンジンの必要最低限の機能を配置し，それ以外の機能をクラウド側で処理するパターンが，ユーザ負担が少なく望ましい．家庭内 LAN に HEMS の装置を設置している事業者であれば，その装置に提案方式の機能を追加し，ブラウザの HTTP プロキシとしてこの装置の IP アドレスを設定すれば実現できる．また，家庭内に設置する装置を安価に構成できるのであれば，導入コストを下げられるため，小規模な事業主体でも実現可能である．このように，本論文では社会実装することに重点をおき，サービスの実現性や導入のしやすさを考慮して研究を進めた．

表 2-1　気付き表示エンジンの組み込み方式の比較

| | （適用方法１）<br>Web ブラウザアドオン | （適用方法２）<br>HTTP プロキシ方式 | （適用方法３）<br>サーバ事前組込み |
|---|---|---|---|
| HTTPS(SSL)への対応 | ○可能 | ×原則不可　（SSL 通信の中身を可視化するプロキシも存在する[1)]） | ○可能 |
| 端末やブラウザの自由度 | △ブラウザや端末が限られる | ○任意のブラウザに対応 | ○任意のブラウザに対応 |
| ユーザ側の利便性 | △インストールをする手間が発生する（Google Chrome の場合はGoogleアカウントが必要となる） | △環境構築に手間がかかる | ○ユーザ側に手間は発生しない |
| 導入にかかるコスト | △ブラウザごとにアドオンが必要．ブラウザの仕様変更に伴うメンテが必要 | △ユーザに HTTP プロキシを設置・導入してもらう作業が必要 | △気付きエンジンのスクリプトを開発時に組み込む作業が発生 |
| 運用上の注意 | △アドオン利用時に，ブラウザのユーザへの事前承認が必要 | △ユーザや情報提供者への事前承認が必要． | △家電情報の扱いにおいてセキュリティ対策が必要 |

## 2.5 アンコンシャス情報の気付き表示方式

アンコンシャス情報を表示する気付き表示方式の概念図を図 2-3 に示す．この気付き表示方式により 2.3 節で挙げた設計要件①と設計要件②を満たす．ここでは分かりやすさのため，サービスイメージ(S1)を例に挙げ説明する．

1) http://www.swatbrains.co.jp/csp.html

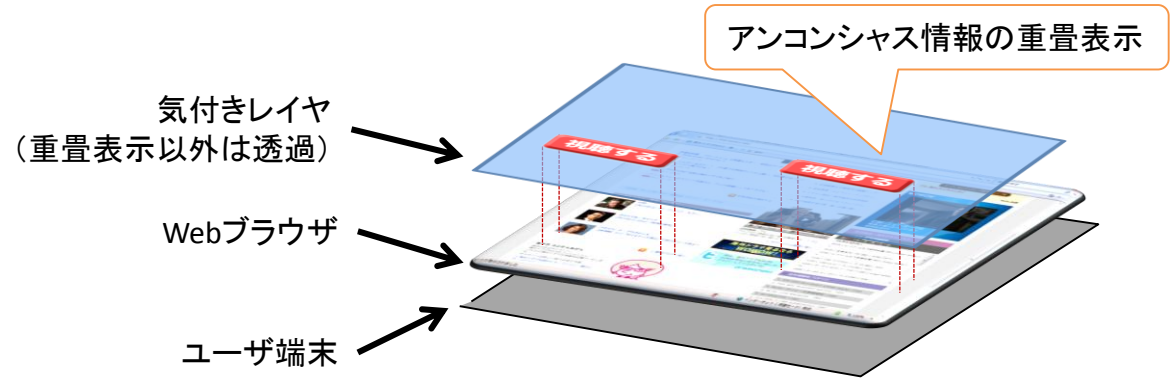



図 2-3　気付き表示方式の概念

この方式は，タブレット端末に表示されている Web ブラウザの上に「気付きレイヤ」と呼ぶ仮想的な表示のレイヤを重畳させ，この気付きレイヤに気付きエンジンから得たアンコンシャス情報を表示することが特徴である．アンコンシャス情報として番組予約を実行する「家電操作タグ」をこの気付きレイヤに表示する．気付きレイヤは，気付きエンジンによって提供された家電操作タグの配置や表示方法を制御し，閲覧している Web ページの情報に応じてユーザに気付きを与える．そして，ユーザがその情報を見て通常リモコンで行うと予測される操作を，家電操作タグのタッチ操作により実行できるように気付きエンジンがアンコンシャス情報を提供する．気付きレイヤを用いることで，画面を切り替えることなく家電を操作することが可能で，同じ Web ページを閲覧していても，ユーザ毎にカスタマイズされた操作への気付きの表示が可能となる．

この方式であれば，設計要件③に挙げた，ユーザが閲覧する Web の表示に影響しないという視覚的側面の条件を満たすことができる．重畳して下の Web が見えない場合は，アンコンシャス情報の ON/OFF を切り替えることで対応できる．

一方，設計要件③の時間的な側面については，気付き表示方式に関わる処理の間が，ユーザにとって不快感を抱かせないようにしなければならない．これについてはプロトタイプを用いて，ユーザが不快感を抱く時間の指標を基に比較することにより検証する．

## 2.6 関連技術

本節では，以上に述べてきた技術に類似する研究やサービスについて，

- 家電を連携制御する技術
- Web ページ情報を付加する技術
- コンテキストアウェアネス

の 3 つの観点から従来研究や関連サービスを挙げ，本研究との違いを明らかにする．

### 2.6.1 家電を連携制御する技術

● **非ネットワーク接続家電の制御**

ネットワークにつながっていない家電を操作する方法として，何らかの機器を利用する場合と，何ら特別な機器を利用しない場合に分けられる．機器を利用する場合は，専用リモコンを利用する場合と，スマートフォンなど汎用の端末を利用する場合に分けられる．汎用の端末を用いる場合は，端末から直接家電を操作する場合と，中継装置を用いて家電を操作する場合がある．

専用リモコンを利用する場合，例えば，家電毎にリモコンがあると，数が増えて散らかったり，リモコンが紛失しやすかったりするため，ユニバーサルリモコンを使うことで複数のリモコン機能を 1 つにまとめることができる．しかし，家電によって異なる操作を，モードを切り替えて 1 つのボタンで操作させるため，ボタンの形状や名称が一致していない場合があり，使い勝手が悪い．

汎用の端末を利用して家電を直接操作する場合，その多くは，家電機器の操作を行う独自のアプリケーション [16] [17]や独自のネットワークサービスを各メーカが提供している [34] [35] [36]．物理的には 1 つの端末内で複数の家電を操作できるようになるが，使いたい機器に応じてアプリケーションを切り替える必要があり煩わしい．また，全ての家電メーカから全ての機種に対応するアプリケーションが提供されている訳ではないため，既に設置してある任意の家電を使えない．

汎用の端末から中継装置を利用して家電を操作する場合，身の回りの多くの家電で使われている赤外線方式によるリモコンに対応させるものがある．例えば，文献 [37] [38]では，WebAPI を備えた IrRC（Infrared Remote Control）装置を使って，Web からのアセシビリティを向上

しようとする研究が行われている．また文献 [39]では，赤外線信号を発するデバイスを指先につけ, 家電を指さしながら指のジェスチャにより機器を指定･操作する提案がなされている．このような赤外線を中継する装置を利用すれば，Web から任意のリモコン操作が可能となり有効である．しかしながら，家電操作を目的としたアプリケーションや専用の Web ページを利用しなければならないとすると，現在使っている画面を切り替える必要があり，使い勝手が良くない．

特別な機器を利用しない場合として，手を使わずに音声を使った方法がある．例えば， 文献 [40]では音声認識を用いて操作コマンドを実行する研究が，文献 [41]では自由な発声内容を理解し，家電操作を行う研究が報告されている．これらの研究によれば，家電の状態やユーザの意図を理解し操作することができるが, 操作のためにユーザが能動的に発声しなければならない．

以上に述べた技術では，家電を操作するために画面を切り替えたり，ユーザが操作のために能動的な行動をおこさなければならなかったりするという課題があった．

● ネットワーク接続家電の制御

ネットワークにつながる家電には, 1.1 項の背景で述べたように, DLNA や ECHONET Lite などの規格が制定され普及してきている．海外でも，米国の SEP [42]や欧州の KNX [43]といった規格により家電のネットワーク化が進んでいる．昨今では電力使用量の関心が高まり，家庭にある周囲の情報を取得する代表的なシステム HEMS [4] [44]が注目されている．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 などの Web ブラウザを使って見えるようにしたシステムである．電力使用量など有用な情報を得ることができるが, ユーザが主体的にシステムにアクセスしなければならない．

文献 [45]では，DLNA, ZigBee, Bluetooth 等に対応した機器を Web から制御するアーキテクチャの提案を行っている．特に無線通信規格の ZigBee は低コスト・低消費電力であり，文献 [46] [47]をはじめ広く利用されている．文献 [48] [49]では，電力線(PLC)を使って HEMS

を活用する方法が提案されている．文献 [50]では，Web から ECHONET Lite 家電を扱うための開発環境（SDK）を提供し，ゲームを交えたリモコンアプリを容易に開発できるとしている．一方，東芝は「ホームクラウドサービス」と呼ぶコンセプトデモを 2013 年の CES(Consumer Electronics Show)で紹介している [51]．これは電気の使用量がある閾値を超えないように，他の家電の使用状況を見ながら連携して使用状態を自律的に制御するものである．また文献 [38]では，各機器が自己の機能をサービスとしてネットワークに公開し，他の機器と互いに連携する仕組みを提案している．文献 [52]では，予め指定した時刻に家電を制御するシステムを提案し，制御のリアルタイム性の検証がなされている．また著者文献 [53]では，音声やジェスチャを用いて家電を操作するシステムを提案している．これらのように家電をネットワーク経由で利用し，自律的に制御する技術開発は行われているが，天気予報やユーザの状態などが考慮されておらず，ユーザに省エネや健康など必要に応じて気付きを与えるものではなかった．さらに筆者文献 [54]では，家庭内の家電の状態を確認し，外出時にエアコンがついていると判断されたときには通知が届くシステムを提案している．この方法であればユーザからアクセスすることなく受動的に家電の状態を把握することができる．無駄な電力消費を防ぐ意味では有効であり，無意識だった切り忘れという情報を得るという観点では本論文と同じ考え方である．本論文では，家電の状態や操作を行うアンコンシャス情報を表示することを基本とし，それに加えて役に立つ情報を付加できる仕組みを考案し実現する．

### 2.6.2 Web への情報付与技術

元々ある何らかの情報に新たな情報を付加する既存研究について説明する．カメラで捉えた実世界の映像情報に景観情報を付加する AR 技術に関する研究は多数存在する（例えば文献 [55]）．一方で Web に情報を付加する取組みについては，例えば文献 [56]では，interFORest というプロジェクトの中に設置した Web サイトに付加情報を表示している．この付加情報は当該 Web サイトにアクセスしたユーザ間の交流を目的としており，アクセス数の状況やプロジェクトの活動地域の情報などを付加情報として表示している．また，「iKnow! ポップアップ辞書 [57]」では Google Chrome にアドオンを追加することにより，Web 上の英単語にマウ

スを乗せると自動的に和訳や発音を聞ける音声再生ボタンなどを重畳して表示するサービスを提供している．これらで提供される付加情報は役に立つものであるが， ユーザのアクセス数の状況や和訳など表示される内容が固定化されている．すなわち，アクセスした Web に対して付加的に表示される情報に柔軟性がなく，ユーザの好みや周囲の状況変化に応じて表示内容を変えることができなかった．

これに対し本研究では，これらの問題点を解消するため，様々な状況変化に応じてユーザが意識しなかったアンコンシャス情報を提供できる仕組みを実現する．

### 2.6.3 コンテキストアウェアネス

提案するアンコンシャス情報を作成するためには，センサなどを使ってユーザを取り巻く状況を分析し，その状況に応じて情報を表示しなければならない．これはコンテキストアウェアネスと呼ばれる技術と類似する．

文献 [58]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捉え，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．文献[59]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．また文献 [28]では，家電やセンサなどの情報が Web サービスとして提供される世界を想定し，既存の Web サービスと連携させるためのフレームワークが提案されている．さらにモノのインターネット化と呼ばれる IoT の分野においては，身の回りの家具などにセンサを取り付けネットワークでデータを取得できるようにし，これらの動きを解析した結果に基づくアクションを行うといった研究もなされている．例えば，文献 [13]では，ドアや引き出しにセンサを取り付け，その動作の解析結果をもとにモノの動作を判断し，それに応じた音声メッセージを流す，といった研究が報告されている．

また文献 [60]では，日常生活の中で触れる機会のある鏡に付加情報を表示する研究が報告されている．この研究では，鏡を利用中のユーザの意識は“鏡“に絞られているが，本研究では多様な目的で利用する Web を対象としており，それに対するアンコンシャス情報も柔軟に対応するといった拡張性の高さの面において異なっている．

上記にあげたいずれの研究も，状況に応じて何らかのアクションを実行する点においては本研究と類似であるが，ユーザが意識しなかったアンコンシャス情報について言及されておらず，またその表示手段の実現や気付き後のアクションに移す手段についても実現されていない．したがって，本研究で提案するアンコンシャス情報の気付き表示方式は独創性の高い研究であると言える．

# 第3章　アンコンシャス情報表示技術
# ＜Ⅰ．基本形・操作＞

本章では，気付きエンジンと気付き表示エンジンを実現する基本システムとして，2.1 節で述べたサービスイメージ(S1)が実現可能であり，アンコンシャス情報の表示が有効であることを示す．すなわち，Web ページ内の情報に気付きを与え，家電操作の対処行動を簡易に実現する番組予約のプロトタイプ A を実現する．プロトタイプ A は，気付きエンジンによりアクセスした Web の内容を検査し，その内容に見合った番組予約情報をアンコンシャス情報として気付き表示エンジンが表示するといった動作を行う．さらに，プロトタイプ A を使った評価を行い，従来の Web サービスでの番組録画予約システムよりも提案するアンコンシャス情報を用いた方が短時間に実行できることを示す．

具体的には，Web ページの閲覧内容に応じた「家電操作タグ」を Web ページに重畳させることで，ユーザにさりげないレコメンドと簡易な家電操作を可能とする家電情報重畳技術を実現する．構成として，一般家庭に急速に普及してきているタブレット端末と，非ネットワーク接続の家電を操作するリモコン信号プロキシ [61] [62]（以降，「RC プロキシ」と呼ぶ）を利用する．タブレット端末は，ノート PC などと比較して可搬性がよく，また操作が直感的で分かりやすいというメリットがある．そのため，昨今では幅広い年齢層に浸透してきており，今後の普及が期待されている．このタブレット端末のブラウザにおいて，閲覧している情報に対し誘発される家電操作を推定し，その操作実行が可能な家電操作タグをその Web ページに重畳して表示する．具体的には，閲覧している Web ページにユーザが興味のある TV 番組名があると，TV での視聴を促す家電操作タグを重畳して表示し，部屋にある TV のチャンネルをその番組に合わせる，といった動作を実現する．さらに，このプロトタイプ A を用いた実験を行い，従来の意識的な家電操作と比較して，提案方式がユーザに負担をかけないという観点で

優れていることを示す.

なお，本章の内容は著者論文 [63]に該当する．当該論文では，任意の Web ページ内の気になる箇所に家電操作タグを重畳表示することにより気付きを与え，この家電操作タグから即座に操作できる環境を「アンビエント」，気付きレイヤを「タグレイヤ」と称し，家電情報重畳技術の研究成果を報告している.

## 3.1 プロトタイプ A の機能構成

図 3-1 に示す機能構成によりプロトタイプ A を，気付きエンジン(N)，気付き表示エンジン(D)，家電操作処理部(C)の処理部により構成し，家電情報重畳技術の動作検証と評価を行う．プロトタイプ A では気付きエンジン(N)と気付き表示エンジン(D)のそれぞれの処理部を 2.4.2 項で説明した Web ブラウザへのアドオン方式（適用方法１）により実装し，２つの処理部の間で処理の指示や DB へのアクセスを可能とした．家電操作処理部(C)はホームサーバとして見立て，赤外線リモコンに対応するレガシーデバイスに対応させる．操作対象としている家電機器に確実にリモコン信号が届くように，タブレット端末からのリモコン操作の指示を中継させる構成とした．また，リモコン信号のデータについては，柔軟なデータの追加拡張が可能なようにインターネット上に共有のリモコン信号管理サーバを設置し，そこから各家庭に設置されている家電のリモコン信号データを取得し，リモコン信号 DB（データベース）に格納する.

ユーザが興味を持っている TV 番組は上述のとおりアドオン内で処理するが，リモコン操作をするために放送チャンネルや放送時間に関する番組情報が必要である．この番組情報はインターネット上ですでに提供されているサービス [64]を利用して取得する．このサービスはユーザが興味のある番組のジャンルやタレント名を登録しておくと，1 週間以内に放送される番組名と放送時間のリストを API により提供している．このような API を利用することによりサービスの実現性と汎用性を高めると同時に，システム構築の効率性を高めた．全体の動作としては,

- Web ブラウザが表示した Web ページの中からユーザが興味を持っている TV 番組名を気付きエンジンが検索し，その番組情報を取得する

- 気付き表示エンジンがその番組情報を埋め込んだ家電操作タグの数や配置を制御して Web ページにアンコンシャス情報として気付きレイヤに重畳表示する
- ホームサーバは家電操作タグのタッチ操作により受信した番組情報をもとにリモコン信号を選択し赤外線信号を送出する

という流れとなる．各処理部における詳細な処理内容について次節で述べる．

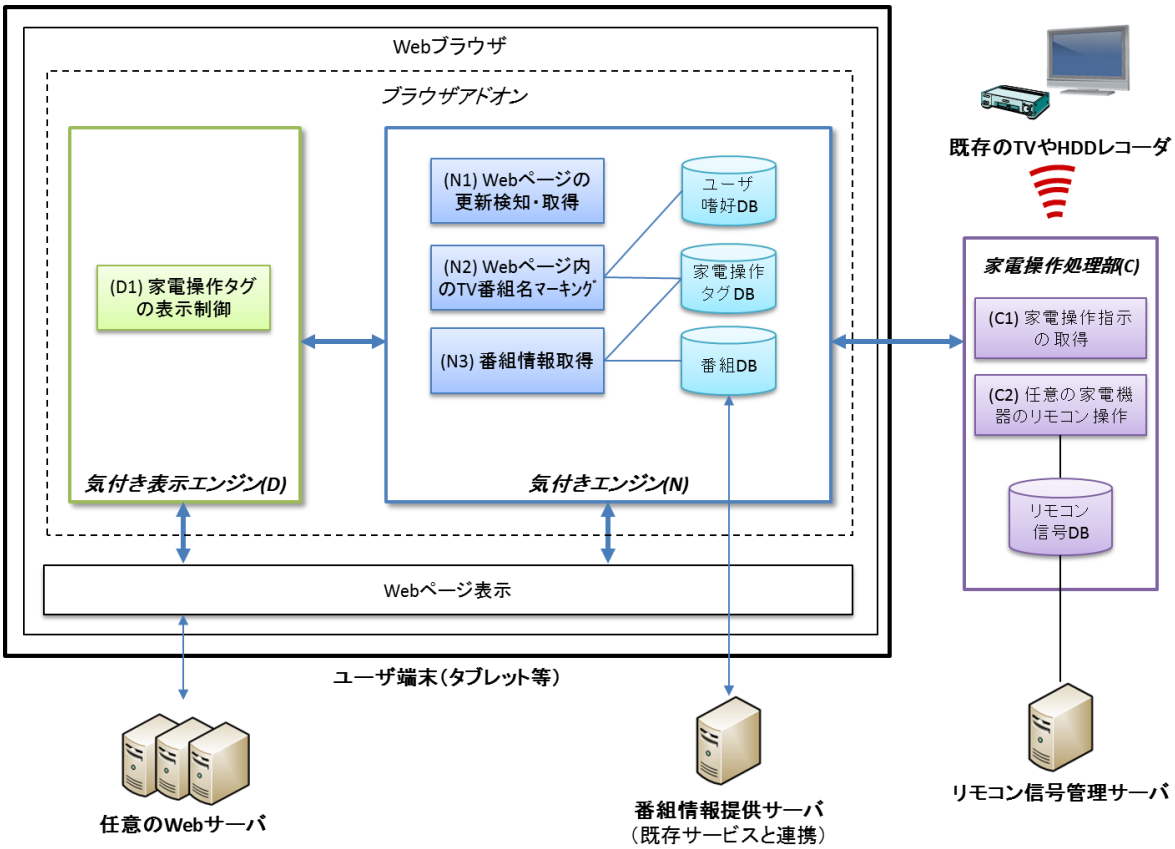



図 3-1　プロトタイプ A の機能構成

## 3.2 プロトタイプ A のシステム設計

家電情報重畳技術の実装に当たっての前提条件とポイントとなる処理の実現方法について述べる．まずシステム構築の前提として，ユーザ嗜好 DB と番組 DB は事前にデータが格納されているものとする．すなわち，ユーザ嗜好 DB にはユーザが興味を持っている TV 番組名とその興味度を事前にユーザ毎に登録してもらう．番組 DB は，外部にある番組情報提供サーバから提供される番組表を基に，番組情報の各要素（TV 番組名，放送局のチャンネル番号，番

組開始日時，番組終了日時）をバッチ処理により抽出し格納しておく．また，家電操作処理部では，操作対象とする TV や HDD レコーダのメーカおよび機種をユーザに選択してもらい，リモコン信号管理サーバから該当するリモコン信号をリモコン信号 DB に登録しておく．以下では，各処理部の実現方法を述べる．

### (1) 気付きエンジン

Web ブラウザで読み込んだ Web ページをブラウザのアドオンで取得する（図 3-1 の N1）．その Web ページの中からユーザ嗜好 DB に登録されている TV 番組名（テキスト）をキーワードマッチングにより検索する．その TV 番組名の位置に家電操作タグを重畳できるように，固有の ID を付けたダミーの HTML タグを挿入しマーキングする（図 3-1 の N2）．一方，家電操作に必要なチャンネル番号等の番組情報を，その TV 番組名を検索キーとして番組 DB から取得する．そして，上述の固有の ID と合わせて番組情報を家電操作タグ DB に記録する（図 3-1 の N3）．

### (2) 気付き表示エンジン

気付き表示エンジンは CSS（Cascading Style Sheets） [65]と DOM（Document Object Model） [66]を使って家電操作タグの表示位置，数，順序に関わる表示を制御する（図 3-1 の D1）．表示する位置は，操作レコメンド処理部で挿入したダミーの HTML タグを DOM により参照し，TV 番組名に被さらないように予め定めた方向にずらして重畳させる．

次に，家電操作タグを表示する TV 番組名とその順番を決定する．まず家電操作タグ DB に登録されている TV 番組名が家電操作タグの表示候補となる． Web ページの全体が Web ブラウザに表示されていない場合は，表示されている領域の中の TV 番組名に表示候補を絞り込む．そして，表示候補となった TV 番組名の内，ユーザの興味度の高いものから順に，予め指定した同時表示数分の TV 番組名を選択し，それに対し家電操作タグを表示する．ただし，興味度が同じ場合はユーザ嗜好 DB 登録の順とした．

### (3) 家電操作処理部

ユーザのタッチ操作によりタブレット端末から送信された番組情報を取得する（図 3-1 の C1）と，その中の番組開始/終了日時から現在放送中か否かを判断し，放送中であれば TV で

の視聴を，未放送であれば HDD レコーダでの録画予約をするようにリモコン信号を設定し送信する（図 3-1 の C2）.

## 3.3 プロトタイプ A のシステム構成

プロトタイプ A では，Web ブラウザで見ている Web ページに事前に登録した TV 番組名があると下方向にずらして家電操作タグを重畳表示し，それをタッチ操作するだけで TV のチャンネル操作あるいは HDD レコーダの予約を実行する，という動作を実現する.

プロトタイプ A のシステム構成図を図 3-2 に示す．タブレット端末には ONKYO 製タブレット PC TW317（CPU: Intel Atom N450，ディスプレイ: マルチタッチ 11.6 型ワイド液晶，OS: Windows7 Home Premium，重量: 1.0kg）を利用し，ブラウザとして FireFox 3.6.13 [67] を選択し，この上で動作するアドオンを開発しインストールした．任意の家電機器の操作には後述の RC プロキシを設置した．タブレット端末と RC プロキシはルータ内蔵の無線 LAN（802.11b/g）により接続され，B フレッツ [68]によりインターネットと接続する．また，操作対象機器として，Panasonic 製 TV TH-42PZ800 と東芝製 HDD レコーダ RD-BR600 を設置した.

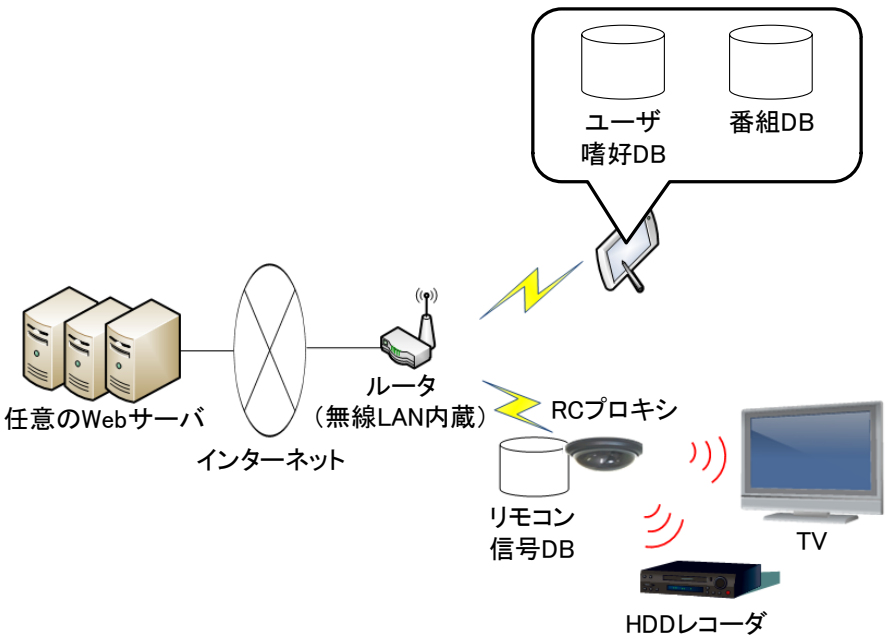



図 3-2　プロトタイプ A のシステム構成

```
{"items":[
        {"name":"NTT ニュース 21",          ・・・①TV 番組名
        "priority":10},                     ・・・①興味度
        {"name":"グルメマップ",             ・・・②TV 番組名
        "priority":5},                      ・・・②興味度
 ・・・
]}
```

図 3-3　ユーザ嗜好 DB の一例

```
{"items":[
    {"name":"NTT ニュース 21",          ・・・①TV 番組名
    "starttime":"2011-02-28 21:00",     ・・・①番組開始日時
    "endtime":"2011-02-28 21:59",       ・・・①番組終了日時
    "channel":"011"},                   ・・・①放送局のチャンネル番号
    {"name":"グルメマップ",             ・・・②TV 番組名
    "starttime":"2011-02-28 22:00",     ・・・②番組開始日時
    "endtime":"2011-02-28 22:24",       ・・・②番組終了日時
    "channel":"021"},                   ・・・②放送局のチャンネル番号
  ・・・
]}
```

図 3-4　番組 DB の一例

ユーザ嗜好 DB と番組 DB については, 予めタブレット端末に配置する. ユーザ嗜好 DB は, 興味のある TV 番組名とその興味度を登録する（図 3-3）. 興味度は初期値として 0 から 10 までの 11 段階で，それぞれの TV 番組名の興味の度合いを数値で設定する．一方，番組 DB には，TV 番組名とその番組開始日時，番組終了日時，放送局のチャンネル番号を登録する（図 3-4）．以降では，家電操作タグと RC プロキシ，番組興味度の更新について詳細を述べる.

● **家電操作タグ**

家電操作タグは図 3-5 のようなボタンアイコンを用いる. そして, それぞれの家電操作タグ

には, 図 3-6 に示す HTTP の GET メソッドによりチャンネル番号や番組開始日時等の情報を RC プロキシに送る URL をリンク情報として設定する.


図 3-5　家電操作タグ

http://192.168.1.16/rcproxy/sendir.php?channel=011&name=*URL エンコードされた番組名* &starttime=2011-02-28_21:00&endtime=2011-02-28_21:59

図 3-6　家電操作タグに設定するリンク情報（例）

● **RC プロキシ**

任意の家電機器のリモコン操作には，著者らが開発した RC プロキシを採用する．著者らが実施したフィールド実験でこの RC プロキシを使用した実績があり，日常利用において信頼性の高いリモコン操作を実現できていることが示されている [69]. 本節ではこの RC プロキシについて，気付き表示方式を実現するために必要な機能に絞って説明する.

プロトタイプ A では，図 3-7 に示すシーリングライト内蔵タイプの RC プロキシを採用した．赤外線には指向性がある [70]ため，部屋にある複数の家電機器に確実に伝達できるように見通しの良い天井に設置することを考え, 既存のシーリングライトに機能を内蔵したタイプを用いた．このタイプは，赤外線信号を同時に 8 方向に送出するモジュールを備えており，部屋の中にある家電を 1 台の RC プロキシで制御できるように工夫した．またこの RC プロキシでは, 番組の放送時間に関する情報を取得し, 現在放送中か否かを判断できるように実装した. これにより，現在放送中の番組であれば TV のチャンネルを操作，そうでなければ HDD レコーダに予約するといった機能を実現した．具体的には，予め家電機器のメーカと機種をユーザに選択させ, それを基に放送局のチャンネル番号とリモコン信号の組をリモコン信号管理サーバから取得しリモコン信号 DB に登録する機能と，タブレット端末からの番組情報を取得しその内容に応じて TV か HDD レコーダかを判断してリモコン信号を送出する機能を実現した.

図 3-7　RC プロキシの外観・設置図

- **家電操作タグの操作履歴を用いた番組興味度の更新**

家電操作タグをタッチ操作された TV 番組は，ユーザが意図して視聴や録画の操作を行おうとしており，他の TV 番組に比べて興味の度合いが高い．そこで，家電操作タグがタッチ操作されるたびに，その TV 番組名の興味度を＋１するようにユーザ嗜好 DB を更新する．これにより，タッチ操作される回数が多くなると興味の度合いが次第に高くなり，結果としてタッチ操作回数が多く興味の度合いの高い家電操作タグが優先的に表示され，ユーザの使い勝手を向上させた．

## 3.4 プロトタイプ A の動作確認

上記に述べたプロトタイプ A を使って実際の Web ページにアクセスし，気付き表示エンジンによる重畳表示を実行した．

図 3-8 はある TV 番組表，図 3-9 はあるニュース記事の Web ページを閲覧中にプロトタイプ A の家電操作タグ表示を実行した際のブラウザの表示画面である．登録した TV 番組名の下に図 3-5 の家電操作タグが表示されている．また家電操作タグの同時表示数を 5 つに設定しており，それぞれ 5 つまでの表示になっている．家電操作タグは，瞬間的に表示するのではなく，上から降ってくるような動きを入れているため視覚的に把握しやすくした．これらの家電操作タグを使って，実際に TV のチャンネル切り替えの操作が可能であることを確認した．

家電操作タグが TV 番組名の真上に被さらないように配慮して下に配置したが，TV 番組名とやや離れてしまったものもあり，配置する位置に関しては調整が必要であった．また図 3-9 では，同じ TV 番組名にそれぞれ家電操作タグが表示されている．いずれも操作の内容は同じであり，同じ家電操作タグを複数表示する必要性について検討し，例えば最初の家電操作タグを 1 つだけ表示する，といった検討課題を抽出した．

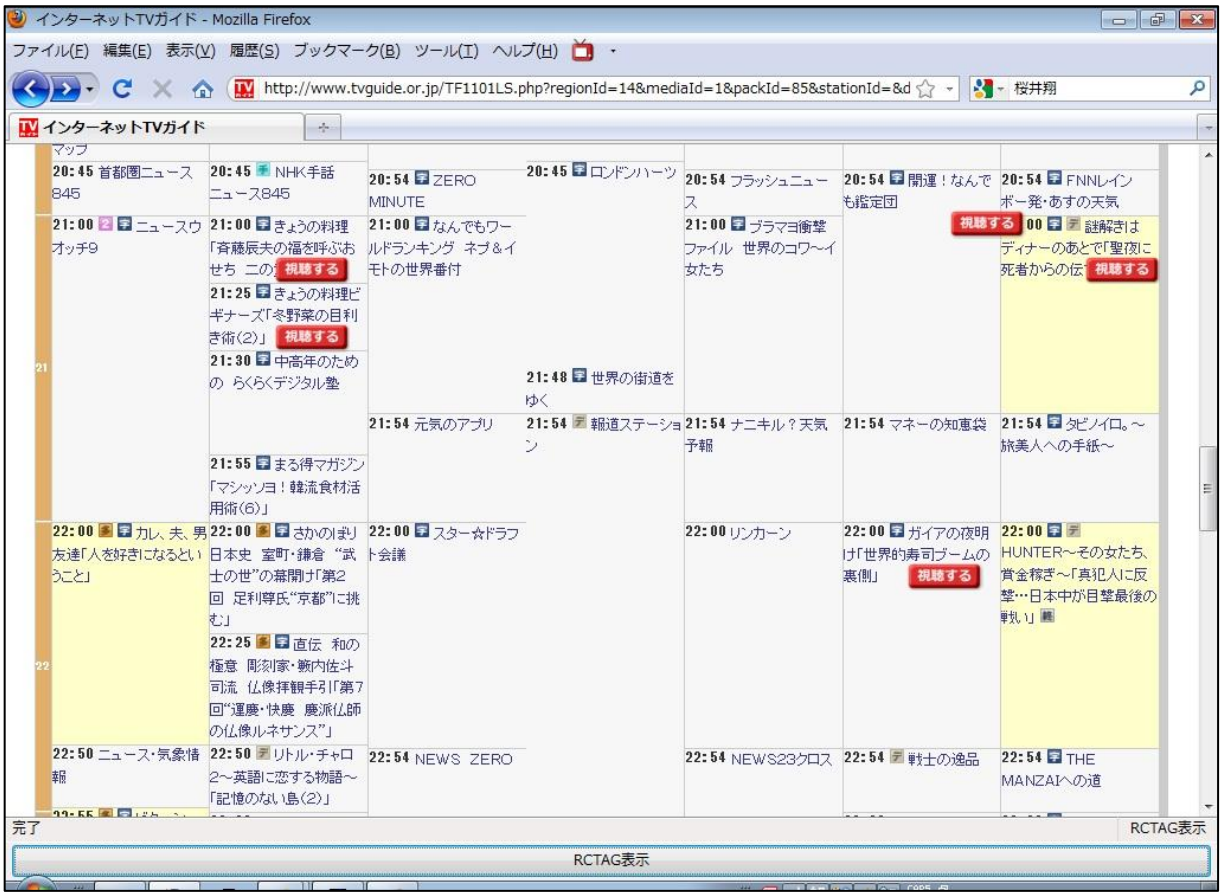



図 3-8　TV 番組表でのプロトタイプ A の実行例


図 3-9　ニュース記事でのプロトタイプ A の実行例

## 3.5 プロトタイプ A を使った評価

前節において，構築したプロトタイプ A の動作確認を行い，気付きエンジンと気付き表示エンジンが実現可能であることを示した．次に本節では，本論文で提案しているアンコンシャス情報の表示が有効であることを示す．具体的には，プロトタイプ A を用いて操作時間に関する評価実験を実施し，家電操作タグによる録画予約が従来の操作と比べて，時間短縮の効果があることを確認する．

### 3.5.1 実施概要

従来方式の Web ページで TV 番組の録画予約を行う操作と，提案する家電操作タグ方式による操作の時間を計測し，その時間を比較評価する．実験環境は上記で述べたプロトタイプ A を使用する．被験者は 10 名（PC を普段使っている 20～30 代の男女）とした．実験の中で予約してもらう TV 番組は，被験者全員同一とし，実際に存在する 5 つ｛CSI，みんなの体操，

ひるおび，3 分クッキング，徹子の部屋〕を設定した．

またシステムの事前準備として，家電操作タグが振られるようにこれらの TV 番組 5 件と，これらとは別の 95 件の計 100 件の TV 番組名を番組名ファイルに登録しておいた．実験でアクセスする Web ページは，従来方法として機器専用の番組予約サイト，提案方式として上記 5 つの TV 番組名がそれぞれ Web ページに含まれる 5 つの Web サイトの計 6 サイト（図 3-10）とし，予めブラウザのブックマークに登録しておいた．

◆従来方法：パナソニック DIGA 番組予約サイト（「番組予約サイト」と略す）

- https://dimora.jp/

※ このページから 5 つの番組を検索

◆提案方法：番組表，ブログ，ニュースなどの実験当時実在する以下の 5 サイト

1. http://ja.wikipedia.org/wiki/CSI:%E7%A7%91%E5%AD%A6%E6%8D%9C%E6%9F%BB%E7%8F%AD
2. http://tv.goo.ne.jp/index.html
3. http://tv.moshimore.net/program/title/minnanotaisou/
4. http://www.youtube.com/watch?v=r8-B0gPuPl0
5. http://news.livedoor.com/article/detail/5283498/

図 3-10　実験に利用した番組予約のための Web サイト

### 3.5.2 タスク

予約する TV 番組名を 1 つずつ指定し，以下の 2 通りの方法で予約を行ってもらった．

- **従来方式のタスク**

被験者は，予約する TV 番組名を指示されたら，ブックマークから番組予約サイトにアクセスし，番組名のキーワードを入力し，検索された結果の中から指定された TV 番組の予約実行のボタンを押して予約を完了する．計測する時間は，ブックマークの番組予約サイトを選択した時点から，検索結果の予約実行のボタンを押すまでとした．

- **提案方式のタスク**

被験者は，予約する TV 番組名を指示されたら，ブックマークから番組名に対応する Web

サイトにアクセスし，そのページの中にある家電操作タグを表示し，ボタンを押して予約を完了する．計測する時間は，ブックマークの Web サイトを選択した時点から，家電操作タグのボタンを押すまでとした．

### 3.5.3 結果

予約した番組名毎に，予約に要した時間を比較したグラフを図 3-11 に示す．提案方式の方が，3.4～8.7 倍速く予約を完了することができた．また，提案方式では分散が小さく比較的一定であるのに対し，従来方式では分散が大きくなっている．

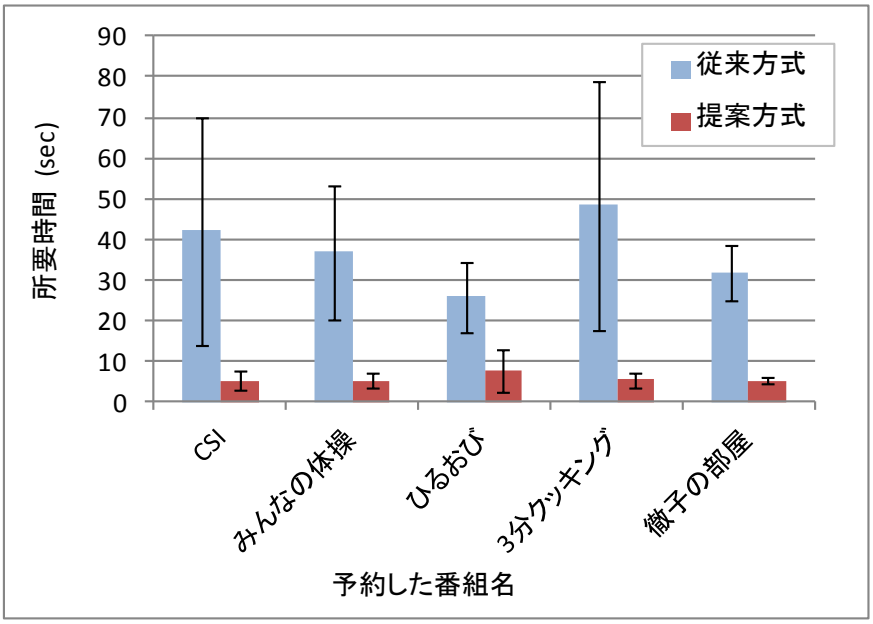



図 3-11　プロトタイプ A を用いた実験結果

### 3.5.4 考察

従来方式で分散が大きくなった原因は，番組予約サイトで番組名を入力するのに時間を要したためと考えられる．「CSI」以外は番組名の長さに関連が見受けられる．一方「CSI」はアルファベット入力に切り替える操作に時間を要したと考えられる．

提案方式では，100 件の番組名が登録された番組名ファイルを取得し，Web ページ内の検索，タグの付与といった一連の処理を行っているが，従来方式よりも速く安定して予約を完了することができた．

## 3.6 本章のまとめ

本章では，任意の Web ページを閲覧中に気になった番組を見つけ，その状況からすぐに TV 視聴や録画予約を行うことが可能なアンコンシャス情報の実現例を示し，家電情報重畳技術として実装と評価を行った．アンコンシャス情報を生成する気付きエンジンと Web 上に表示する気付き表示エンジンを Web ブラウザのアドオンにより実装し，これらが実現可能であることを確認した．さらに従来方式と時間短縮効果を比較する評価実験を実施し，従来方式よりも速く操作が可能でありアンコンシャス情報が有効であることを確認した．

# 第4章　アンコンシャス情報表示技術
# ＜Ⅱ．複数の家電設定＞

多様なサービスが増加すると，家電やセンサ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事態が頻繁に発生する．2.1 項のサービスイメージ(S2)に示したように，例えば，父親が動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複数の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知れない．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．また，従来研究 [71]においても，２～６つの家電操作の手順を１つの操作で連携するシステムが提案されており，ユーザ評価により有効かつ高い満足度が検証されている．このような状況を鑑み，本章では，サービスを利用する際に周囲のデバイスが期待している状態にあるかを意識しなくても適切に制御するアンコンシャス情報を提案する．なお以降では，ネットワークにつながる家電やセンサ，モノを含めて「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．

解決方法としては，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．このような対応をサービス提供者側で制御することは難しい．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，状況変化に伴うサービス提示 [58]や，サービス実行の条件としてデバイス状態を利

用する研究 [59]などが行われていた．また，著者らの文献 [72]では，Web で利用する映像や音声のメディアに応じて，LAN 上の TV やスピーカなどを自動的に選択する方法を提案している．しかし，任意の Web サービスと任意の複数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，各ユーザがサービス毎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，現在の状態が異なっている場合は再現するように制御するマルチデバイス照合技術を実現しアンコンシャス情報として提供する．対象とするサービスとしては，HTML5 [22]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．本章では，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する．

以下，マルチデバイス照合技術の要件をユーザアンケートから導き，第 3 章で実現した家電情報重畳技術を拡張することによりマルチデバイス環境に対応するプロトタイプ B を構築し，実現可能性を示す．さらにサービスを提供する上で課題となる各デバイスの管理・制御に関わるアクセス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元にサービスとしての実現性を考察する．

なお，本章の内容は著者論文 [73]に該当し，第 3 章と同様，当該論文でも気付きレイヤを「タグレイヤ」と称し成果を報告している．

## 4.1 マルチデバイス照合技術を実現するプロトタイプ B

本節では，マルチデバイス照合技術の具体例としてサービスイメージ(S2)に挙げた複数のデバイスを同時に扱うプロトタイプ B を作成し，提案手法の処理により本来の Web アクセスに影響しないという時間的要件の検証を行う．そこで，任意の Web サービスとその時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を確認するプロトタイプ B を実現する．この中では，ユーザが利用している Web サービスとその時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．まずは，サービス提供する際に欠かせない想定されるデバイス数とその状態数をユーザアンケートにより導き，実装要件と性能条件を得る．

## 4.2 想定するデバイス数調査のためのアンケート

- **目的**

サービス提供におけるマルチデバイス照合技術の実装要件を得るため，現実的なシーンにおいて設定したいデバイスおよびその状態数を調査した．

- **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）を対象とした．

- **回答方法**

目的とする以下の 2 つの設問のほか，「サービスを利用する時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」，などの補足の設問（表 4-1）を用意し，さらに回答後にインタビューを行った．

表 4-1　ユーザアンケートの代表的な設問

| | |
|---|---|
| 設問 1 | 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． |
| 設問 2 | 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挙げてもらう．<br>状況をイメージしやすいように，具体的な家電の例（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（電源状態，運転残り時間，運転モードなど）を表示し，合計 26 の状態から選択してもらう．<br>その他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． |

● アンケート結果

マルチデバイス照合技術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 4-2 は，各回答者が登録したいと挙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 つ以下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言える．

表 4-2　ユーザアンケートによる登録したいデバイス状態数

| 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 |

## 4.3 プロトタイプ B のシステム構成

マルチデバイス照合技術を実現するプロトタイプ B の実装構成を図 4-1 に示す．本構成では，第 3 章で述べた家電情報重畳技術を元に，2.4.2 項で説明した HTTP プロキシ方式（適用方法 2）を新たに導入して気付き表示エンジンを閲覧する Web ページに組み込む．気付きエンジンはホームサーバと呼ぶ同一 LAN 上にサーバを設置する構成としたが，簡易なコンピュータにより導入実現できることをプロトタイプ B の実現により示す. この気付きエンジンには，Web との対話 I/F, 家庭内デバイス群の状態確認, 設定するデバイス状態の登録・管理と照合,

デバイスへの問い合わせの各機能を備える．

一方の気付き表示エンジンでは，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デバイス状態登録処理，登録している状態を照合するデバイス状態照合処理の機能を備えた気付きエンジンの実行モジュールを実装し，上述のとおり HTTP プロキシ方式により Web ページに組み込む．サービス・デバイス状態連携処理はユーザが 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デバイス状態登録処理によりデバイス状態を登録し，登録済であればデバイス状態照合処理により登録されているデバイス状態を取得する．デバイス状態登録処理とデバイス状態照合処理の処理においては，ユーザに登録する状態などを表示し入力を求める画面が必要となるが，気付きレイヤを使って HTML5 の CSS による柔軟な表示を実現した．

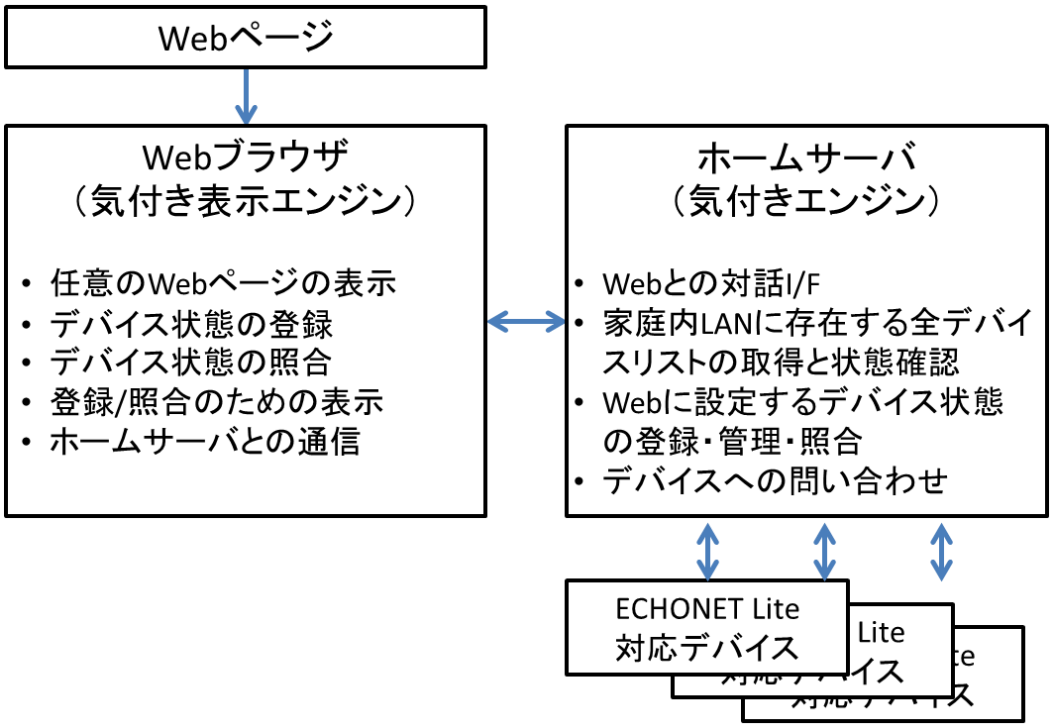



図 4-1　マルチデバイス対応プロトタイプ B の実装構成

気付き表示エンジンには，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 つの機能を備える．こ

れら 2 つの機能を JavaScript で実装し，HTTP プロキシ方式により閲覧中の Web ページに追加する．そして，Web ページの更新毎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，気付き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 4-2 を用いてそれぞれ説明する．

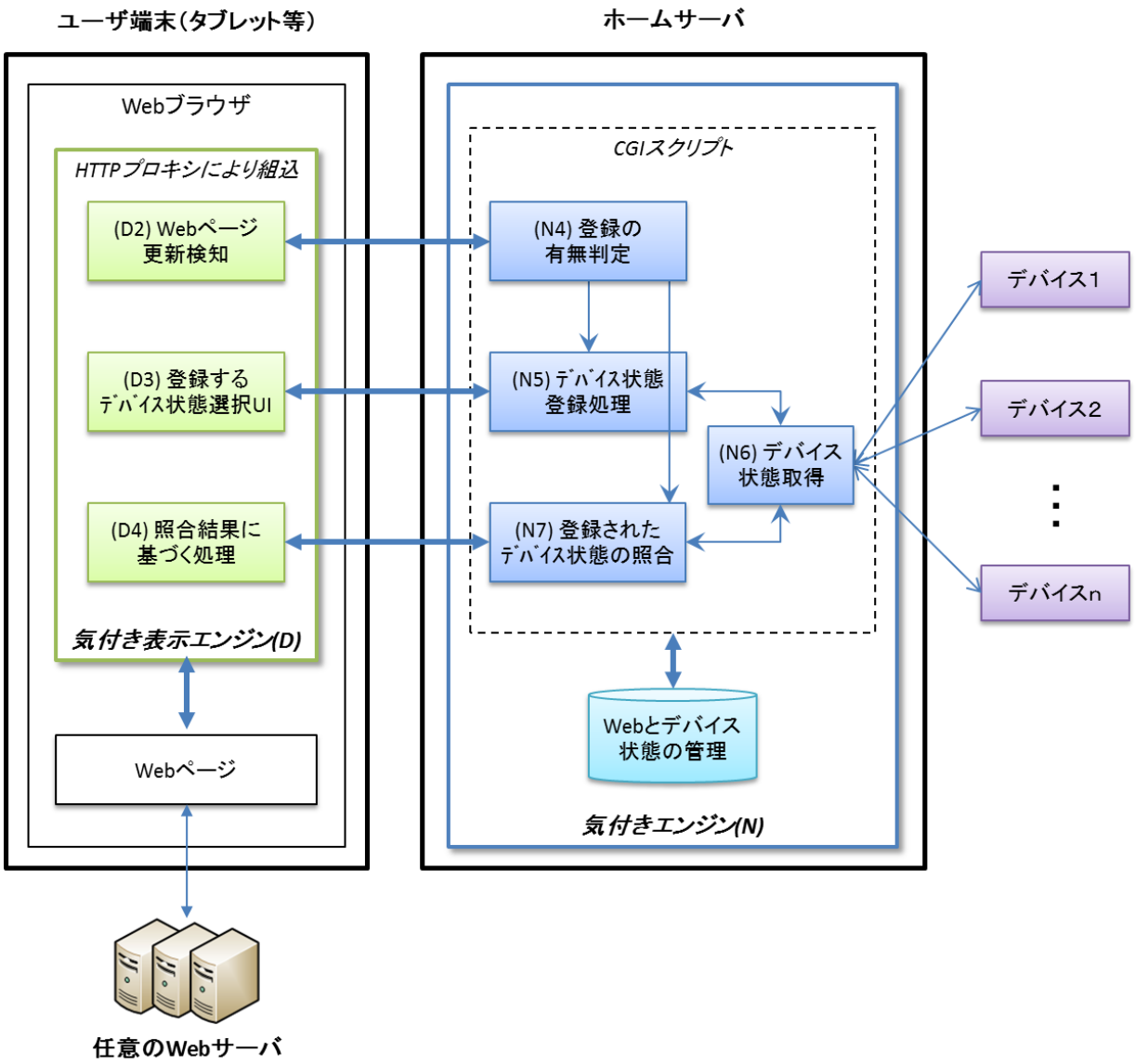



図 4-2　プロトタイプ B の機能構成

● **登録有無の判定と気付き表示エンジンの画面**

Web ブラウザ上で Web ページの更新が発生する（図 4-2 の D2）とその URL が気付きエンジンに通知される（図 4-2 の N4）．その URL がホームサーバに登録されていなければ，現在

のデバイスの状態を登録する画面を，登録されていたら現在のデバイス状態と登録されているデバイス状態を照合して結果を気付き表示エンジンに送信する．

具体的には，図 4-2 の N5 の処理によりデバイス状態の照合の画面を応答メッセージとして返送し，登録するデバイス状態登録 UI を表示する（図 4-2 の D3）．登録されていれば，登録されたデバイス状態の照合（図 4-2 の N7）を実行し，デバイス状態取得（図 4-2 の N6）により現在のデイバス状態を取得し，その結果を，気付き表示エンジンに送信し，照合結果に基づく処理（図 4-2 の D4）により Web ページ上に重畳表示する．

- **デバイス状態の登録**

気付きエンジンが取得した各デバイスの現在の状態（図 4-2 の N6）を取得し，登録画面を作成して CGI（Common Gateway Interface） [74]により応答返却（図 4-2 の N5）する．その中からユーザが選択したデバイスの選択設定した状態と，その時の Web の URL を気付きエンジンに送信し，Web とデバイス状態の管理 DB に登録する．

- **デバイス状態の照合**

登録の有無判定（図 4-2 の N4）で受け付けた Web の URL をキーに登録されているデバイス状態を Web とデバイス状態の管理 DB から取得し，それぞれのデバイスに対し状態を確認する（図 4-2 の N6）．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，全てが OK であるかを判定した結果を含めて CGI により応答返却する（図 4-2 の N7）．この結果を照合結果に基づく処理（図 4-2 の D4）が受信し，判定結果を元に，サービス利用の許可等の制御を行う．

## 4.4 プロトタイプ B の実装要件

マルチデバイス照合技術を実現するプロトタイプ B の実装要件 RB を示す．

〈実装要件 RB1〉Web ブラウザから LAN 上の全てのデバイスの状態を確認できること

〈実装要件 RB2〉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

〈実装要件 RB3〉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〈実装要件 RB1〉の「Web ブラウザから LAN 上に接続する全てのデバイスの状態を確認できること」について説明する．ユーザの端末がつながっている LAN 上の全ての 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，文献［45]や文献［75］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〈実装要件 RB2〉の「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」について説明する．ユーザは本来サービスを享受する目的で Web を閲覧しており，その本来目的としている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．

〈実装要件 RB3〉の「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」は，提案方式の処理に要する時間が長くなると，ユーザが不快感を抱き，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た．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，〈実装要件 RB1〉と〈実装要件 RB2〉については実装面でのシステム要件，〈実装要件 RB3〉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．

## 4.5 プロトタイプ B の性能要件

2.3 節の設計要件③である「ユーザが利用する元のサービスを邪魔しないこと」で示したように，提案方式に関わる処理の間，ユーザにとって不快感を抱かせないようにしなければならない．

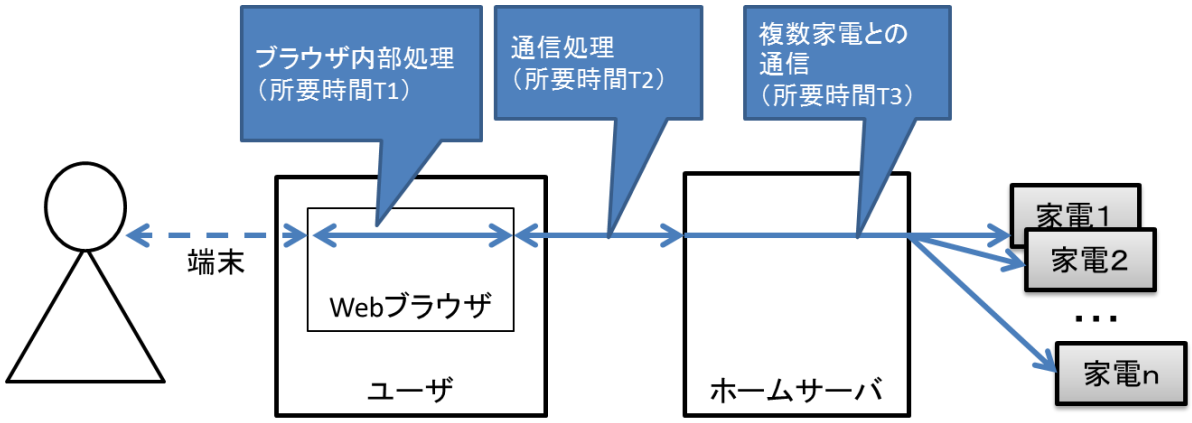



図 4-3　Web から複数の家電状態を確認するまでの処理時間

マルチデバイス照合技術に関わる主な処理は図 4-3 に示した構成にあるとおり，以下の３つの注目すべき処理がある．

〈所要時間 T1〉　提案方式を実行するため Web ブラウザ内部の処理

〈所要時間 T2〉　Web ブラウザとホームサーバ間の双方向の通信

〈所要時間 T3〉　複数の家電との通信応答を含むホームサーバ内部処理

ここでは〈所要時間 T3〉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，〈所要時間 T1〉の Web ブラウザ内部の処理は，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〈所要時間 T2〉の Web ブラウザとホームサーバ間の双方向の通信は，有線 LAN だけでなく IEEE802.11g/n/ac などにより無線 LAN の高速化が図られていること [76]などから，提案方式独自の処理にあたる〈所要時間 T3〉の複数の家電との通信応答を含むホームサーバ内部処理に着目する．

〈所要時間 T3〉の処理時間に大きく影響を与えるのは，問い合わせするデバイスの数と状

態数であると考えた．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おらず，高い処理性能を期待できないからである．そこで，実際の ECHONET Lite 対応デバイスを使って，デバイスへの問い合わ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．

## 4.6 プロトタイプ B を使った評価

本節では，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 B の動作により確認する．さらに，実機のデバイスへの問い合わせ時間を調査と，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 B について説明し，プロトタイプ B 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．

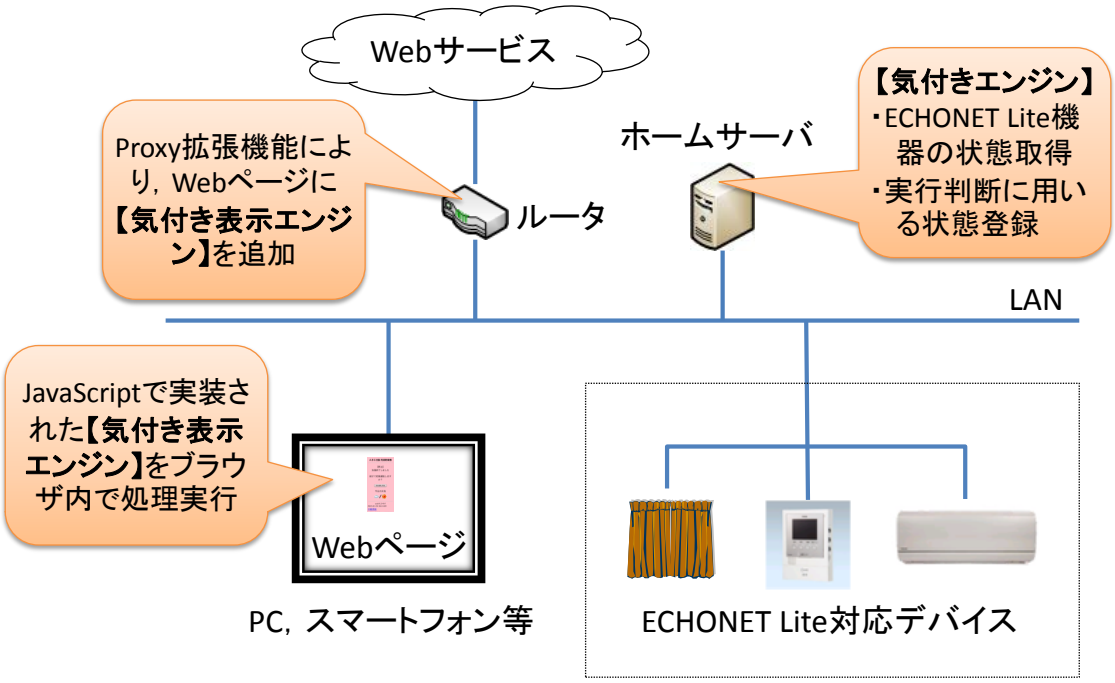



図 4-4　プロトタイプ B のシステム構成

図 4-4 は，構築したプロトタイプ B のシステム構成図である．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予め登録されている状態から動作を開始する．そ

の理由は，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抱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 [77]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．また文献[78] [79]など，多数の活用事例が報告されていて実績もある．ホームサーバおよびユーザ端末となる PC の仕様は表 4-3 の通りであり，表 4-4 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 4-5 は実際に用いた ECHONET Lite デバイスの一部の外観である．

表 4-3　プロトタイプ B のシステム仕様

| | | |
|---|---|---|
| ホームサーバ | | |
| | ハードウェア | Raspberry Pi TypeB |
| | OS | Linux rasberrypi 3.10.25+ #622<br>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 4-4　プロトタイプ B で使用する ECHONET Lite デバイス

| 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 |
| 畜電池 | ENG-B6630A1 |
| シーリングライト | LEDH-LT1 |

図 4-5　プロトタイプ B で利用したデバイス（一部）


図 4-6　プロトタイプ B の動作画面（照合中）

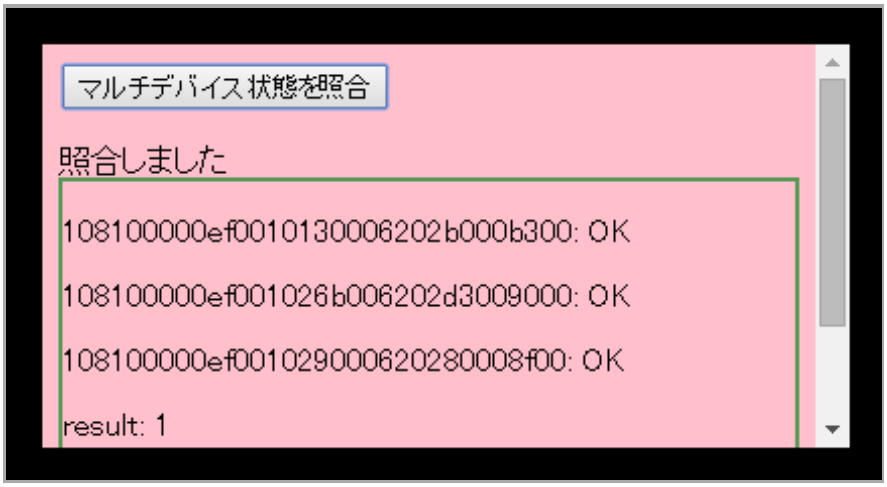



図 4-7　プロトタイプ B の動作画面（結果表示の拡大）

### 4.6.1 動作検証

図 4-6 は気付き表示方式により追加されたモジュールにより，一般の Web ページにアンコンシャス情報が重畳され，アンコンシャス情報にある照合実行ボタンの押下後，気付きエンジンに問い合わせている画面である. このとき, 気付きエンジンでは今回用いた 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かを照合判定している．図 4-7 は気付きエンジン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，気付きエンジンから受信した状態確認の結果を図 4-8 に示す.

```
{"ref":"http://*******.html",
"108100000ef0010130006202b000b300":"OK",
"108100000ef001026b006202d3009000":"OK",
"108100000ef00102900062028000 8f00":"OK",
"result":"1"}
```

図 4-8　ホームサーバから受信した結果データ

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 [80]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

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 状態を問い合わせしている.

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 [81]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．これらが全て"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，全て"OK"で"result"は"1"となっており，その結果を気付き表示エンジンにより判断して，図 4-7 のとおり照合された旨のメッセージを表示している．ここでは動作確認用にこのような簡易な表示をしているが，実際のサービスでは気付き表示エンジンを使って表示や処理の実行を制御し，動画の視聴開始など元のサービスの実行を制御する．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 B の実行例では動画再生のボタン（動画領域中央）の上に照合のアンコンシャス情報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，マルチデバイス照合技術の実現性を確認した．

### 4.6.2 性能評価

本項では，デバイスごとの状態問い合わせ時間について，実際のデバイスを用いて調査する．

- **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 B の気付きエンジンの 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．

- **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 4-5 と図 4-9，図 4-10 に示す．

表 4-5　ECHONET Lite デバイスの応答時間[3)]

| 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 |

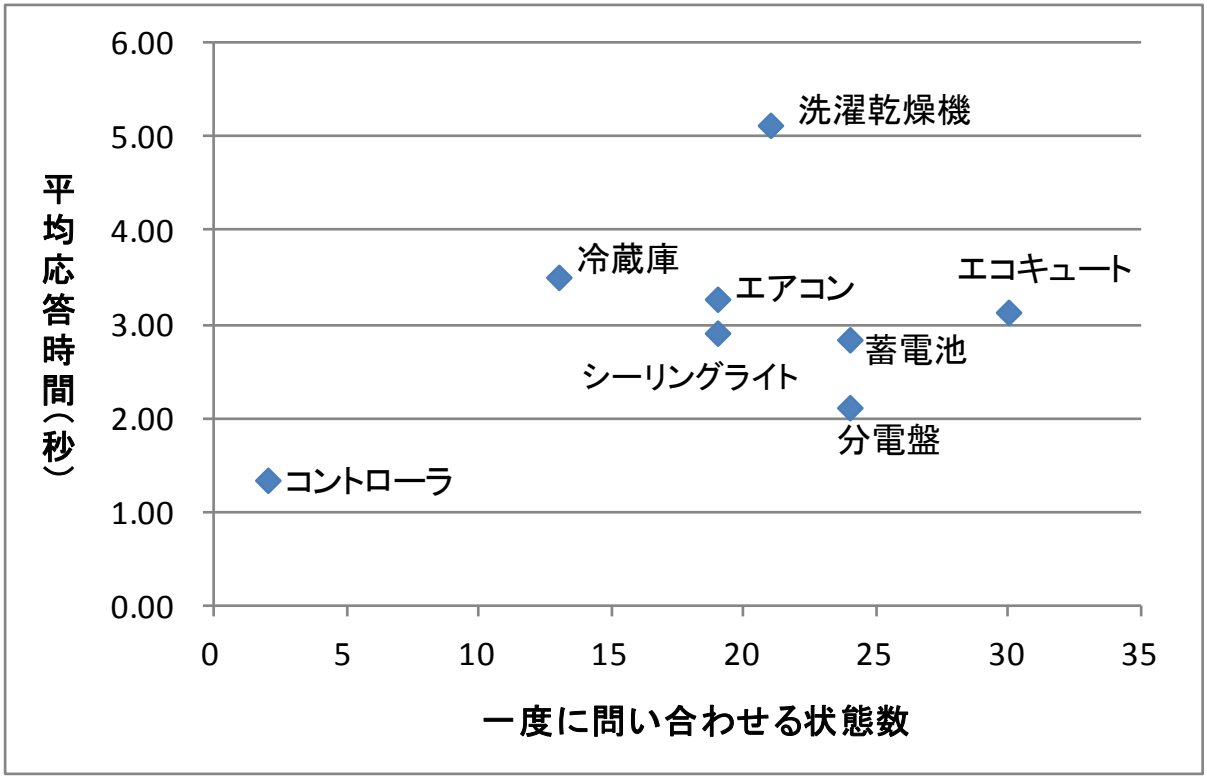



図 4-9　問い合わせ状態数と平均応答時間

3) 1 つのデバイスタイプに付き，あるメーカの 1 機種を用いて測定したものであり，他のメーカの応答時間とは異なる場合がある．

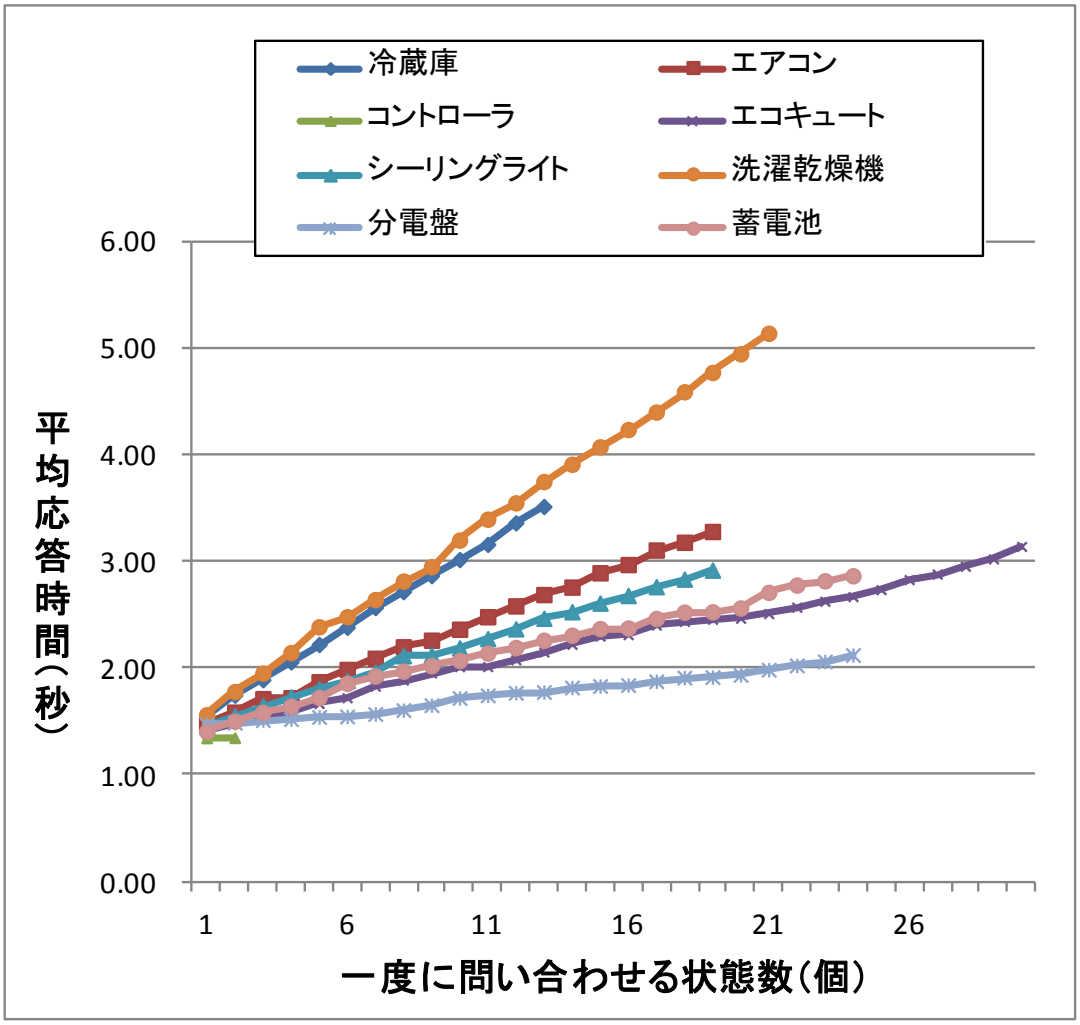



図 4-10　問い合わせる状態数と応答時間

表 4-6　状態数と応答速度の算出

| 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 |

デバイス毎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毎の傾きと状態数 1 つの場合の応答時間を算出した（表 4-6）．この値をもとに，デバイス状態数から概ねの応答時間を予測することができる．

### 4.6.3 考察

● **デバイス問い合わせの所要時間**

4.2 節の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 つずつ問い合わせて取得した場合の時間を算出する．表 4-6 で示した 1 状態取得あたりの平均応答時間の値（Y 切片： 1.402sec，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+0.082×1)×6＝8.904sec 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 秒の時間を要する．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．これらを含めた時間が Web ページの更新時間だと考えると，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 [82]」「ユーザはウェブページを 10 秒から 20 秒で離れてしまうことが多い [83]」という基準を満たせる．一方で「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 [84] [85]」という別の指標もあるため，さらなる時間の短縮が求められる．ここでは想定した 5 つのサービスでのアンケート結果をもとに 6 つの状態取得の時間を導いたが，それ以外のサービスでは 6 つより多い状態取得が必要となり，その時間が長くなる可能性がある．以上の考察から，

対処１：デバイスの状態取得時間の短縮化

対処２：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を検討する．

まず対処１の「デバイスの状態取得時間の短縮化」については，ECHONET Lite プロトコルによるデバイスへの問い合わ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．

次に対処２の「ユーザがサービスから離脱しないための画面更新」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，例えば，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．このような有益な情報を組み合わせてサーバからプッシュ方式で表示する方法については第５章で述べるが，画面が更新されるまでの間，アンコンシャス情報を表示することでサービスからの離脱を防ぐことができる．

以上のことから，提案方式におけるデバイスへの問い合わせは，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，デバイスへの問い合わせ処理の並列化と，バックグラウンド処理により問い合わせを行い，その間に有効な情報をユーザに表示する方法を挙げた．これらの方法は実現可能性が高く，設計指針③で示した「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」の時間的側面において達成可能であると言える．

● **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，その上でユーザは所有するデバイスを想起し，好みの状態を登録する．この時，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を登録できない．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言えない．

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．

## 4.7 本章のまとめ

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，サービスを利用可能とするマルチデバイス照合技術を提案した．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，その時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，元のサービスの画面を遷移しないこと，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実装要件とした．この実装要件に従って，アーキテクチャを示し，HTTP プロキシ方式による気付き表示方式を採用したプロトタイプ B を実装し実現性を確認した．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い合わせ所要時間を測定し，ユーザ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，デバイスへの問い合わせ処理の並列化と，バックグラウンド処理中におけるユーザへの情報表示を実施することにより，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ことが見通せた．以上のことから，2.3 節にあげた設計要件③である，本来の Web 閲覧の時間的な影響を与えずに実現できることを確認した．

# 第5章　アンコンシャス情報表示技術 ＜Ⅲ．家電状態の変化通知＞

これまでユーザ端末で閲覧していた Web ページの内容に基づいて周囲の家電などのマルチデバイスを設定・制御する方法について述べてきた．実際には，家電の運転終了など家電の状態変化についてユーザが知りたい情報もある．そのような周囲の状況変化に加えて，その時にユーザが意識していなかった知るべきアンコンシャス情報がある．例えば，2.1 節のサービスイメージ(S3)で示したように，洗濯乾燥機の脱水が終わり乾燥運転を実行する場面において，乾燥運転にかかる電気代や天気予報の晴れであるというアンコンシャス情報を表示することにより，外干しが促され自然と省エネ行動を実践することができる．本章ではこのようなプロトタイプ C を構築し，ユーザ評価から，電気代や天気情報といったアンコンシャス情報が有効であること確認する．

さらに，家電の状態変化が気になって Web サービスに集中できなくなることを鑑み，Web サービスを利用する前あるいは利用した後に運転が完了するように気付きエンジンにより制御し，アンコンシャス情報を使ってユーザを導く方法を示す．具体的には，動画視聴サービスの利用時に洗濯が完了するという場面を想定したプロトタイプ D を実現し，提案方法の処理時間を計測した結果，サービスとして実現可能であることを確認する．

なお，本章 5.5 節までの内容は著者英文論文 [86]に該当し，気付きレイヤを”Concierge Layer”と称し研究成果を発表している．また本章 5.6 節は著者研究会発表文献 [87]に該当し，気付きレイヤを「タグレイヤ」と称し発表している．

## 5.1 アンコンシャス情報による新たな気付き

本章では気付き表示レイヤを用いて，任意のタイミングでアンコンシャス情報を表示し，ユ

ーザに気付かせることを可能とする．先に述べたように，洗濯乾燥機の脱水終了をトリガに，乾燥運転の実行ボタンと天気予報をアンコンシャス情報として表示する．天気予報の情報は，天気が良いと分かると外に干すことで乾燥運転にかかる電気代を節約することができ，また悪天候ならそのまま乾燥運転をアンコンシャス情報に表示される実行ボタンから操作できるため，節電あるいは簡単な遠隔操作というメリットをユーザに提供できる．あるいは，来客がドアホンを鳴らしたときに前回訪問時の日付や用件を合わせて通知したり，掃除機のごみがいっぱいになったことを知らせる際にごみパックの安売り情報を合わせて通知したりするといった様々な利用方法がある．このように，家電の状態変化に応じて，役立つ情報に柔軟に結び付けることにより様々な用途への展開を実現する．つまり家電の状態変化に応じたアンコンシャス情報を表示することにより，

- 気にしていなくても家電の状態変化に気付ける
- 推奨される家電の設定や操作を自然と簡単に実行できる
- 家電とは異なる情報と結びつけて，新たな役立つ行動を実践できる

といったメリットを提供できる．

## 5.2 プロトタイプ C の実装要件

上記に述べたように，家電状態が変化したタイミングでアンコンシャス情報をユーザ端末に表示できること，そしてこのようなアンコンシャス情報の有効性ついて検証する．以下に，本プロトタイプを実現するための実装要件を示す．

〈実装要件 RC1〉ホームサーバの気付きエンジンは，LAN 上のあらゆる家電の状態取得と操作実行が可能であること

〈実装要件 RC2〉気付きエンジンは，家電の状態と操作のボタン，節電操作に関連する情報をアンコンシャス情報として生成すること

〈実装要件 RC3〉気付き表示エンジンは，適切なタイミングでアンコンシャス情報を表示し，ユーザに節電のための行動を促すこと

まず〈実装要件 RC1〉は，出来る限り多くの家電に対応することにより，多様な家電の状態と役立つ情報との組み合わせを可能とする．さらに，複数の家電状態を統合的に扱うことも可能となるため，複数の家電を連携させた節電効果も期待できる．

次に〈実装要件 RC2〉に関しては，家電の状態を把握し，家電の前まで行かなくとも操作できる利便性を提供するだけでなく，節電の促しや行動を補完する情報を与える．この節電関連情報は多岐にわたるため柔軟に対応できるように設計する．例えば，家電の種類により電力特性や可能な運転操作の仕様に差異があり，また他の家電との併用可否，電力会社の電気料金，さらには天気予報のような家電とは直接関係のない情報も節電行動につながるため，これらの情報を組み合わせて利用できるようにする．

そして〈実装要件 RC3〉では，このような節電行動を継続して使ってもらえるように，2.3 節で述べた設計要件③に従い，元の Web サービスを阻害しないようにアンコンシャス情報を表示し，家電状態変化となすべき行動をユーザに気付かせる．

## 5.3 プロトタイプ C のシステム設計

上記の要件に従い，システムの設計を行った．図 5-1 はプロトタイプ C の構成図である．プロトタイプ C では，

〈設計要件 DC1〉ホームサーバの気付きエンジンにより家電の状態と操作の一元管理を行うこと

〈設計要件 DC2〉Web を利用して表示のタイミングや内容の柔軟性を担保すること

〈設計要件 DC3〉HTTP プロキシ方式を採用し，既存の Web ブラウザ上にアンコンシャス情報を表示すること

を実現した．

〈設計要件 DC1〉のホームサーバでは，LAN 内の家電を常時一元的に状態管理し，状態取得と操作指示を行うアンコンシャス情報を CGI により提供する．これにより，ユーザ端末毎に状態管理する方法と比べて，ユーザ端末における処理負荷，および状態通知に関わる各家電とネットワークの負荷を軽減するとともに，状態把握と操作を 1 箇所に集約し，CGI で提供す

ることにより，汎用的な Web ブラウザからのアクセスを可能とした．

また〈設計要件 DC2〉は，家庭内の家電状態や天気予報などの外部の情報をリアルタイムに取得し，CGI で提供するアンコンシャス情報に柔軟に組み合わせて表示させる．

一方，〈設計要件 DC3〉に述べたように，2.4.2 項で述べた HTTP プロキシ方式を適用する．この HTTP プロキシは第 4 章で述べたプロトタイプ B と同様に，通常の HTTP プロキシの処理に加えて，アンコンシャス情報を重畳表示するように JavaScript でプログラムされた気付き表示エンジンを Web ページに埋め込む処理を行う．これにより，ユーザが任意の Web ページを閲覧中でもアンコンシャス情報を表示させることができる．

プロトタイプ C の気付き表示エンジンでは以下の 2 つの処理を行う．

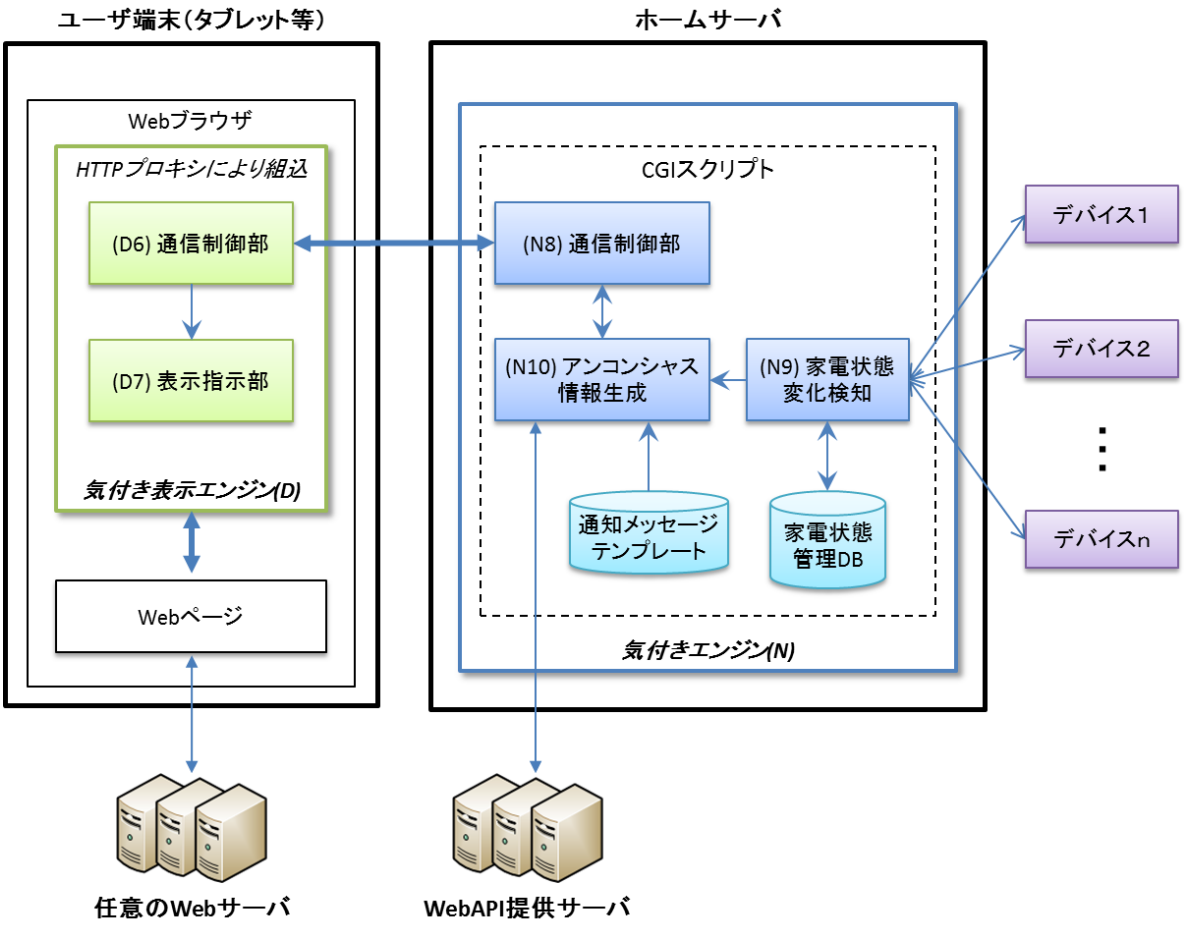



図 5-1　プロトタイプ C の機能構成

- **アンコンシャス情報の内容と表示指示を気付きエンジンから受信**

HTML5 で規定されている WebSocket [88]を用いて，ホームサーバの気付きエンジン（図 5-1 の N8）と，ユーザ端末の Web ブラウザで表示されている Web ページに組み込まれた気付き表示エンジン（図 5-1 の D6）との間の通信を確立する．従来手法として Web ブラウザから定期的にホームサーバへ問い合わせを行うポーリング [89]という手法もあるが，WebSocket を用いることで通信ロスがなくリアルタイムな指示伝達が可能となる．

- **閲覧している Web ページに受信したアンコンシャス情報を重畳して表示**

家電状態変化通知（図 5-1 の N9）がデバイスからの変化通知を取得すると，アンコンシャス情報を生成する（図 5-1 の N10）．家電の状態変化に対応する通知メッセージテンプレートを取得し，天気予報などインターネットで提供されている WebAPI 提供サーバから取得し，通知メッセージテンプレートに代入することによりアンコンシャス情報を Web 要素で生成する．この Web 要素で構成されたアンコンシャス情報を，上記で確立した通信制御部間の WebSocket を経て気付き表示エンジンに送信する（Web 要素通知技術）．この Web 要素通知技術を実現することにより，リアルタイムかつ柔軟な Web での情報表示が可能となった．アンコンシャス情報を受信した気付き表示エンジンは表示指示部（図 5-1 の D7）に表示を指示し，表示中の Web ページ上に重畳表示する．

## 5.4 プロトタイプ C による実現可能性の確認

実装したプロトタイプ C の動作確認により，システムの実現性を確認した．まず既存の Web ページに気付きエンジンの指示によりアンコンシャス情報が重畳表示されることを確認した（図 5-2）．アンコンシャス情報には，現在の洗濯乾燥機の状態と乾燥運転実行の操作ボタン，そして節電関連情報として天気予報を表示した．これにより Web 要素通知技術の動作を確認できた．さらに，乾燥運転実行のボタン操作により洗濯乾燥機を制御できることを確認した．


図 5-2　プロトタイプ C のシステム画面

## 5.5 プロトタイプ C の有効性評価

アンコンシャス情報の有効性を確認するため，アンコンシャス情報として節電に関する情報が操作実行のボタンと一緒に表示されると，操作実行ボタンだけのときと比べて節電行動を取りやすいことをユーザ評価により確認した．アンコンシャス情報の有効性について，プロトタイプ C で作成した画面を用いた被験者実験により節電に対する意識や行動の変化を検証した．

評価方法としては，プロトタイプ C で作成した 4 種類のアンコンシャス情報（表 5-1）により節電への意識や行動の変化を検証した．節電行動を行うかどうかを被験者に回答してもらい，スコア（外干しする：1，分からない：0，外干ししない：－1）を集計する．被験者は 6 名（普段家電を利用している 20～30 代の男女）とした．

表 5-1　実験でアンコンシャス情報として組み合わせる表示パターン

| | | 天気情報 | |
|---|---|---|---|
| | | 非表示 | 表示 |
| 電気代 | 非表示 | 表示パターン P（図 5-3） | 表示パターン Q（図 5-4） |
| | 表示 | 表示パターン R（図 5-5） | 表示パターン S（図 5-6） |


図 5-3　プロトタイプ C のアンコンシャス情報（表示パターン P）


図 5-4　プロトタイプ C のアンコンシャス情報（表示パターン Q）


図 5-5　プロトタイプ C のアンコンシャス情報（表示パターン R）


図 5-6　プロトタイプ C のアンコンシャス情報（表示パターン S）

● 結果

表 5-2 はユーザ評価の結果である．天気予報と電気料金を両方表示する表示パターン S の場合が最も高評価であった．また，天気予報だけ表示する表示パターン Q と電気料金を表示する表示パターン R, そして天気予報も電気料金も表示しない表示パターン P はこの順番で高い評価ではあったが，これらの間にはほとんど差はなかった．被験者ごとに見ると，天気予報を出す方を好む被験者と料金を出す方を好む被験者とに分かれていたが，どの被験者も天気予報と電気料金を両方表示する表示パターン S は邪魔にはならず表示されることに問題ないとの回答であった．

表 5-2　アンコンシャス情報表示パターンの評価結果

| | パターン P | パターン Q | パターン R | パターン S |
|---|---|---|---|---|
| 平均 | -0.50 | -0.17 | -0.33 | 0.33 |

● 考察

プロトタイプ C のアンコンシャス情報により, 少なくとも節電に関わる情報を表示することの有効性を確認した．しかし，結果にはばらつきがあり個人差があることも確認できた．よって，アンコンシャス情報に節電操作を表示する際は，ユーザを認識し，そのユーザに適した節電情報を表示することがより有効であると考えられる．実際の日常生活においてユーザは様々な状況にあり，アンコンシャス情報を受け取る際に，元の Web サービスを阻害する可能性がある．次節では，このような場面を考慮した対処方法について述べる．

## 5.6 周囲に対する無意識化を支援するアンコンシャス情報

プロトタイプ C の評価結果から, 家電の状態変化に応じてユーザにアンコンシャス情報を通知することの有効を確認できたが，ユーザが利用している Web サービスについての考慮が十分でなかった．例えば，Web サービスで長編動画を視聴している時などは途中で中断したくない場合や，脱水後の洗濯物や調理した料理をそのまま放置しておきたくもない場合もある．

そこで, 本節では, このように周囲の環境に対して気にせず無意識でいられることを支援し，サービスに集中できるアンコンシャス情報を提供することにより生活をさらに便利で豊かに

する．プロトタイプ C で示した家電の状態変化に応じた通知機能に，第 4 章の Web サービス利用における家電の状態確認機能を組み合わせ，Web サービスを利用している最中に家電の状態が変化する場合は，サービス利用時間と家電の運転時間を調停する手段をアンコンシャス情報として表示する仕組みを実現する．

### 5.6.1 無意識化を支援するアンコンシャス情報に必要な処理

家電の運転終了時間を考慮した従来研究として，オーブンの調理時間と同じ時間長の映像をインターネット上で探し，オーブンの前のディスプレイで再生することにより，オーブンの調理時間を退屈させない研究がある [90]．ユーザが家電の運転終了まで待てる場合はよいが，長時間に渡る場合や家電の前にずっと居ることが負担となる場合には適用することが難しい．また文献 [91]では，センサ反応後の家電動作イベントを設定するフレームワークの提案がなされているが，Web サービスの利用状況については考慮されていなかった．そこで，これから利用する Web サービスの時間長と家電の運転終了までの時間を比べ，運転終了までの時間をアンコンシャス情報として表示する．ユーザはこれを見て，Web サービスの利用を待つ，あるいは Web サービスの利用終了時に合わせて家電の運転が終了するように制御することで，Web サービスに集中できるようにする．このような方法を実現するにあたり，以下の処理を実現する．

〈処理 D1〉 利用する Web サービス情報と時間長の取得

〈処理 D2〉 家電状態の把握と変化の予測

〈処理 D3〉 Web サービスの利用時間と家電の状態変化の時間調停

〈処理 D1〉はユーザが利用しようとしている Web サービスが他の割込み作業が発生して中断しても良いかどうか，中断できないとしたらそのサービスの終了時間を判断する．〈処理 D2〉は〈処理 D1〉で得た Web サービスの利用時間長の間に，ユーザが対応すべき家電の有無を判断する．対応すべきか否かには，変化する状態の内容に基づく．例えば，洗濯機乾燥機の洗濯から脱水に移行した変化は対応すべき変化ではないが，脱水完了は対応すべき変化である，といった状態変化の対応付けを事前に設定する．〈処理 D3〉は〈処理 D2〉の判断の結果，対応

すべき状態変化があった場合は，アンコンシャス情報としてユーザに通知する．以下に具体的な実現方法を述べる．

- **Web サービス情報の取得**

Web サービスの中断を許容できるかについてはその内容に依存するが，Web ページに含まれる情報から類推できる．例えば，Twitter や Facebook，YouTube などのように著名なサービスであれば，その URL を手掛かりに判断することもできる．また，<meta>に記載されたタイトル情報も非常に有効な情報源であり，さらにチャットであれば WebSocket，映像視聴であれば HTML の<video>タグ [92]といったように，Web ページの中で使用されているタグや JavaScript の関数などからも判断することができる．

サービスを終了する時間については，厳密に推定することは難しいが，平均的な所要時間や過去の所要時間を参考にすることができる．また動画や音声の視聴サイトであれば，当該コンテンツの時間長を取得することにより推定することが可能である．

また，先行文献 [93]では，Web ページの閲覧滞在時間とユーザの目的についてモデル化や履歴分析による研究がなされており，これらのモデルや分析手法も利用できる．

本節は Web サービス利用と家電運転の時間調停する実現可能性を確認することが目的であるため，長時間での利用頻度が多いと想定される動画視聴を Web サービスとして利用している場面を想定し進めた．

- **家電状態の把握と変化の予測**

家電の状態変化とその時間を予測する方法としては，ECHONET Lite を使ってオーブンレンジの調理時間といった終了までの所要時間を直接取得したり，洗濯乾燥機のすすぎから脱水，乾燥といった既知の工程における現在の状態を取得し，その工程の所要時間を算出したりすることができる．あるいは，文献 [94]や文献 [95]のように行動履歴から予測することも可能である．このように様々な状態変化の予測手法を組合せることは有効であり，このような組合せができるようにシステムを構築する．

- **調停の方法**

上記に述べた方法により，Web サービスの利用時間と家電の状態変化の時間を取得し，そ

れらが時間軸上で重なるか否かを判断する．重なる場合は設定された調停ルールに基づき調停する．調停ルールは，Web サービスの利用における緊急性や中断の受容性，また家電は状態変化後の人的作業に要求される迅速性や放置した場合の経済的あるいは物的なダメージへの影響度合い，さらには作業完了までの時間を比較することにより判断し，変更される側に対し要求を行う．

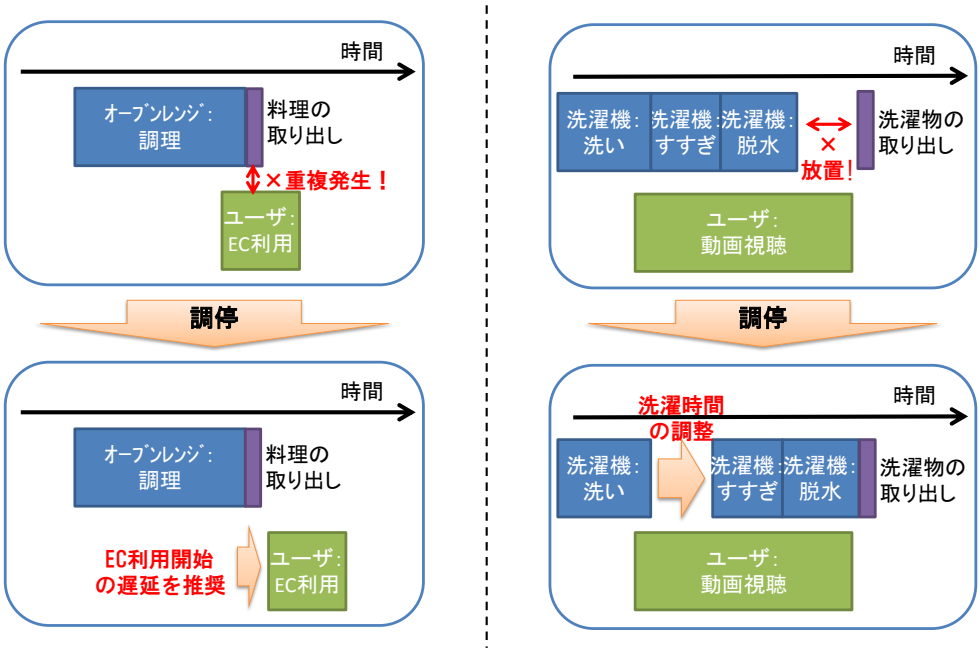



図 5-7　インターネット利用と家電運転の時間調停例

図 5-7 は具体例としてオーブンレンジと洗濯機の調停方法を示したものである．図 5-7 の左側はユーザが EC 利用中にオーブンレンジの調理が完了し料理の取り出さなければならない場面である．この時，調理の変更は不可であり，調理完了までの時間が短いことから，EC 利用の開始を遅らせるという提案をアンコンシャス情報として提示する．一方，図 5-7 の右側は動画視聴中に洗濯機の脱水が完了し洗濯物を取り出すまで放置されてしまうという場面である．脱水完了までに時間があり，また洗濯の運転時間は変更可能と判断し，動画視聴終了後に脱水が完了するように洗濯機の運転時間を変更する．そして，動画視聴の終了と同時に洗濯機の脱水運転が完了すると，状態変化を通知するアンコンシャス情報を表示する．

### 5.6.2 プロトタイプ D のシステム構成

上記に述べた方法を実現するシステムを設計する．図 5-8 は，そのシステム構成を示したものである．PC やスマートフォンは無線 LAN アクセスポイントを使って家庭内の LAN と接続し，ルータを経由して Web サービスを利用する．

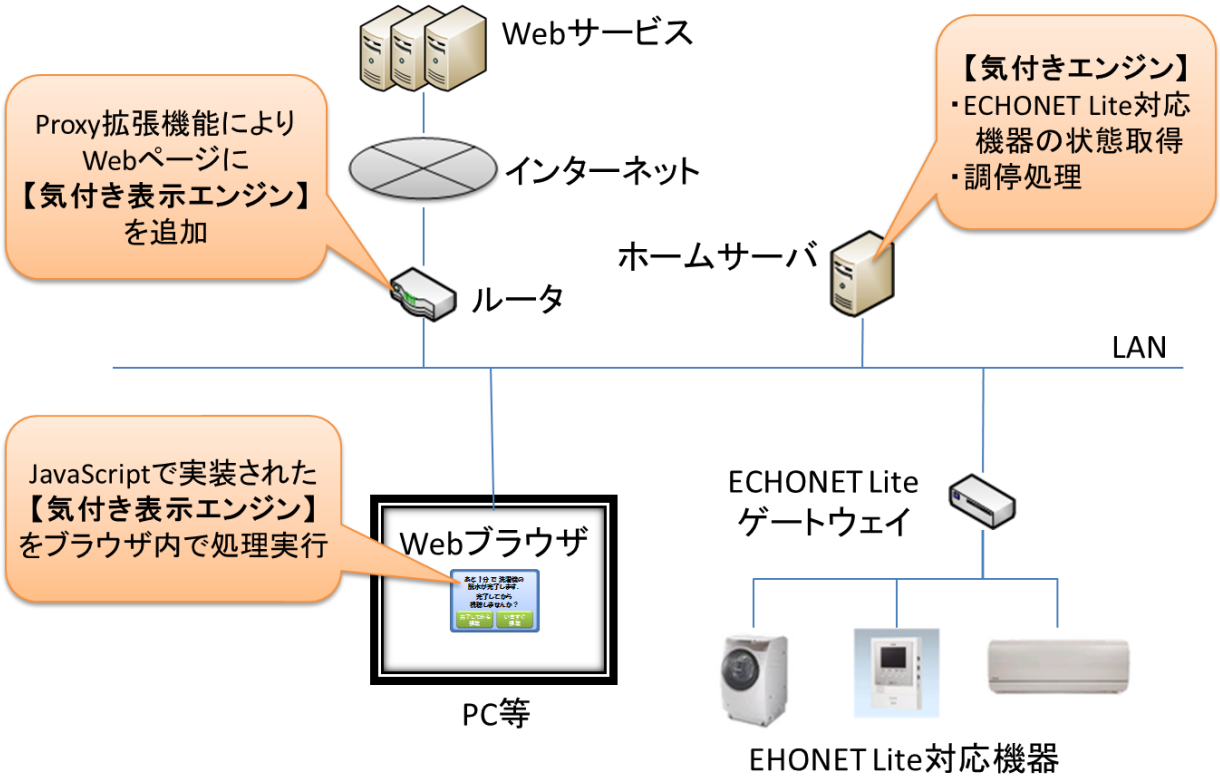



図 5-8　プロトタイプ D のシステム構成

LAN に気付きエンジンを搭載したホームサーバを設置し，ECHONET Lite を用いて各家電の状態取得と制御を行う．気付きエンジンは，

- ECHONET Lite による家電とのやりとりを一元的に管理制御することにより家電へのアクセス負荷を低減する
- PC からのアクセス先を集中することにより家電とのレスポンスのリアルタイム性を確保する
- 家庭内の家電を網羅的に把握することにより総合的な状況判断を可能とする

といった役割を担う．また気付きエンジンでは，先に述べたアルゴリズムに従って，ユーザの Web サービスの利用状況と比較して時間調停を行う．

一方，ユーザの手元にある PC の Web ブラウザには，Web サービスの利用状況を把握し，家電の対応作業を通知する機能を追加する．具体的には，Web のアクセス先 URL により，ユーザのサービス状況を推測する．そのため，ユーザにはこれらの解析処理が実行されることを承諾した上で利用してもらう手順とする．

### 5.6.3 プロトタイプ D のユースケース

本項では提案モデルとアーキテクチャを具体的なユースケースに適用し，実現性を検証する．

- **映像視聴と洗濯機のユースケース**

長編の映像を Web で視聴中に洗濯機の脱水が完了する場面を想定する．従来は，最後まで映像を視聴し終わってから，あるいは視聴を一時停止するなどして洗濯物を取り出していた．しかし提案方法では，映像が終了するまで洗濯物が放置されたり，映像視聴を中断されたりすることないように，脱水が完了することを映像視聴前に通知し，脱水後の対応作業が終わってから視聴できるようにメッセージで通知する．

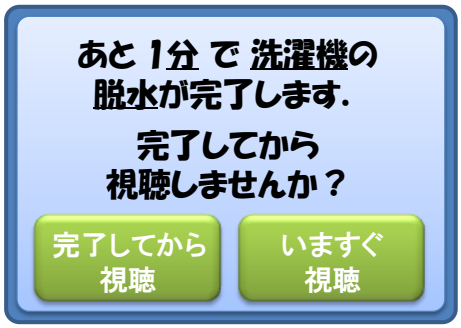



図 5-9　家電運転完了を促すアンコンシャス情報（例）

具体的には，洗濯機の脱水完了前に Web の映像を視聴しようとすると，図 5-9 のようなダイアログを表示し，脱水が完了してから視聴することを促す．これを見たユーザは，脱水後に洗濯物を取り出して（干すなどして）から視聴を開始することで視聴を中断されることを回避できる．

- **プロトタイプ D の実装**

図 5-10 のモジュール構成によりプロトタイプ D を実装した．気付きエンジンを搭載するホームサーバには，EPSON 社製 Endeavor MR3500 に Linux OS (CentOS 5.8 Final)を載せ，

Web サーバ（apache 2.2.3），データベース（MySQL 5.0.95），Web-CGI（ruby 1.9.3p194）を実行させた．また，ユーザ端末の PC には，Panasonic 社製 Let's Note CF-W8（Windows 7 Professional） の Google Chrome 27.0.1453.47 beta-m を使用し，Tampermonkey v2.12.3124.188 [96]を使って２つの機能拡張（映像検出モジュール，気付き表示エンジン）を追加した．Web ブラウザとホームサーバの気付きエンジンとの間はプロトタイプ C と同様に WebSocket [88]により常に双方向での通信を可能とした．

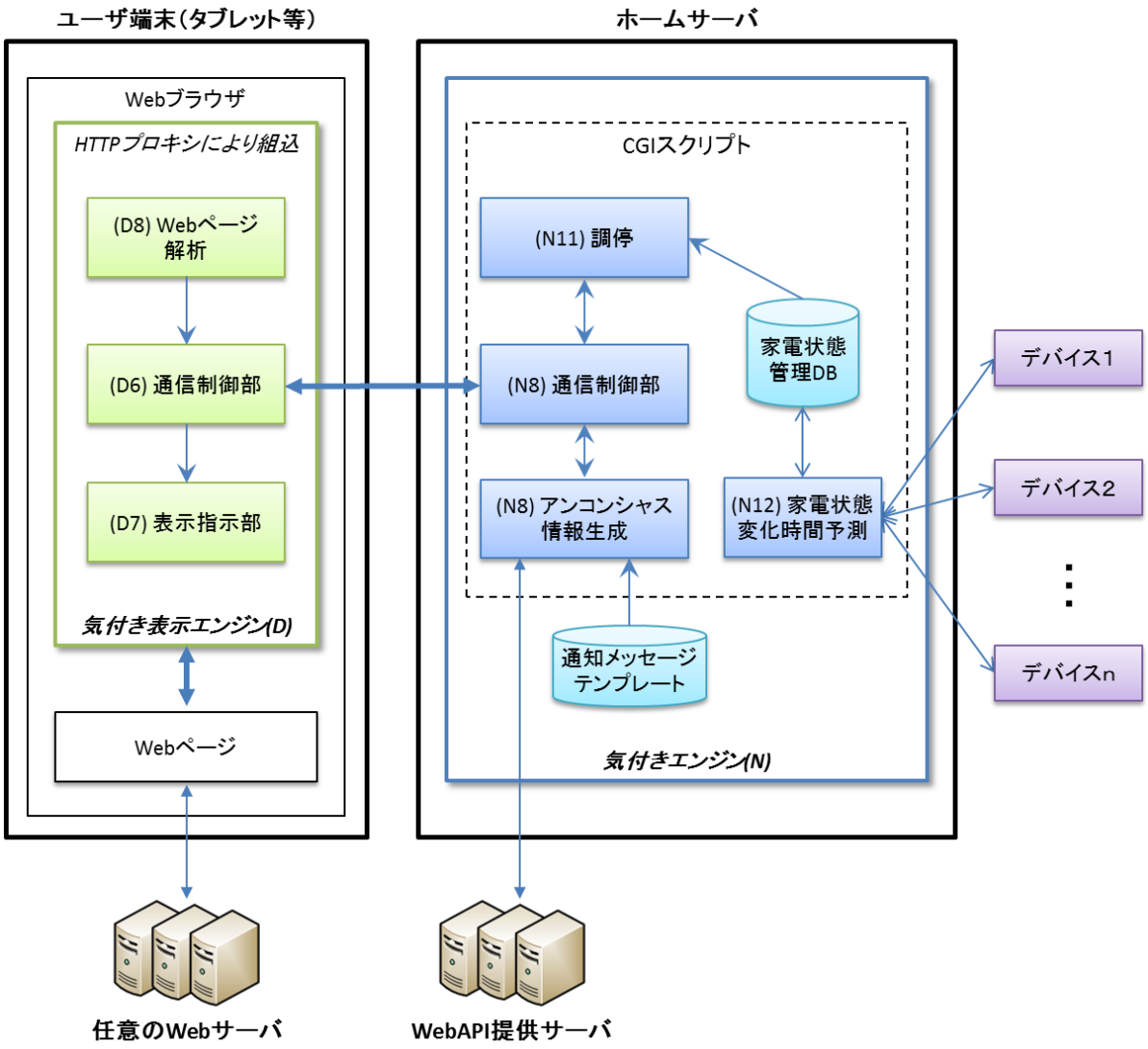



図 5-10　プロトタイプ D のモジュール構成

Web ページ解析（図 5-10 の D6）ではユーザが Web サービスを利用しようとしているかを判断する．ユースケースにおいては，ユーザが映像視聴しようとしているかを判断する．その

ため，ユーザが閲覧している Web ページ内に映像があるかを解析し，ある場合はその所要時間として動画時間長を取得し，ホームサーバへ送信する．

気付きエンジンでは，家電状態の変化時間を予測する（図 5-10 の N12）．各家電と定期的に ECHONET Lite で通信し，家電の状態を監視する．プロトタイプ D では取得した状態が家電状態管理 DB 上に常に保存されているものとして運用する．

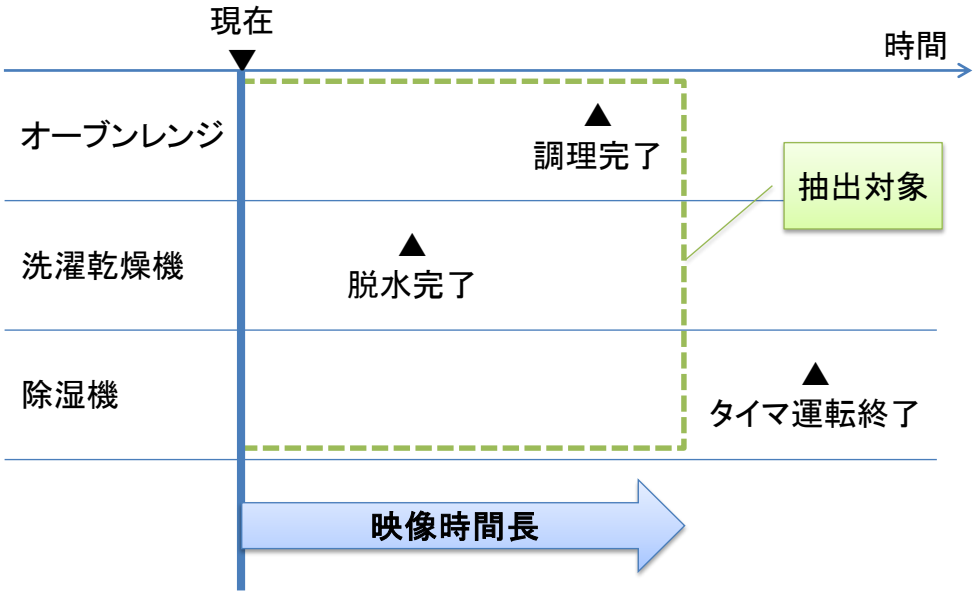



図 5-11　家電状態変化の抽出イメージ

気付きエンジンの調停部（図 5-10 の N11）は Web ブラウザの Web ページ解析部から映像時間長を取得すると，家電状態管理 DB の中から，図 5-11 のイメージのように映像時間長に含まれる家電の状態変化の有無を抽出する．このとき複数の状態変化がある場合も考えられる．この場合，

〈抽出方法１〉　現在に近いものを１つだけ抽出（図中の「脱水完了」のみ）

〈抽出方法２〉家電や状態に優先度を設定し高いものを１つだけ抽出．例えば，オーブンレンジの方が洗濯乾燥機よりも優先度が高い場合は「調理完了」を抽出

〈抽出方法３〉　抽出対象にある全てを抽出

などの対応方法が考えられる．ただし，〈抽出方法３〉の場合，抽出対象にある全ての状態変化を抽出するため表示内容が多くなって煩雑にならないよう１つずつ順番に出すなどの工夫が必要である．プロトタイプ D では〈抽出方法１〉の現在に近いものを１つだけ抽出する方

法を採用し，状態変化が抽出された場合はユーザへの待機を促すメッセージを表示させることとした．なお，家電の運転時間を制御する方法については，具体的な家電の運転工程を調査した後，本手法に適用する．

上記の調停処理の結果，抽出された家電状態の変化をユーザが閲覧中の Web ブラウザに通知する．プロトタイプ C で実装したように，任意の家電の様々な状態変化に対応するメッセージテンプレートを用いてアンコンシャス情報の通知文を生成する．

気付きエンジンのアンコンシャス情報生成（図 5-10 の N8）は，テンプレートからメッセージを生成し，プロトタイプ C と同様に，WebSocket により接続された通信経路（図 5-10 の D6 と N8 の間）を利用して Web 要素通知技術によりアンコンシャス情報を気付き表示エンジンに送信し，ユーザが閲覧している Web 上に表示する（図 5-10 の D7）．

● **プロトタイプ D の動作**

図 5-12 と図 5-13 はプロトタイプ D において YouTube の映像にアクセスした画面である．アクセスすると画面中央にアンコンシャス情報を重畳表示し，その中にはテンプレートで生成された「04 月 16 日 21:00 に Washer1 の WashAndDryStatus が Dry になります．続行しますか？」というメッセージが表示された．Washer1 と WashAndDryStatus，Dry といった文字列は ECHONET Lite で定義された値で，家電状態管理 DB に保持されたものである．この画面から表示された「OK」ボタンを押すとアンコンシャス情報を表示したまま状態が変わるまで待機し，「閉じる」ボタンを押すとアンコンシャス情報を閉じて，背面に隠れていた元々アクセスした Web ページが表示されることを確認した（図 5-14）．


図 5-12　家電調停プロトタイプ D のアンコンシャス情報（画面全体）

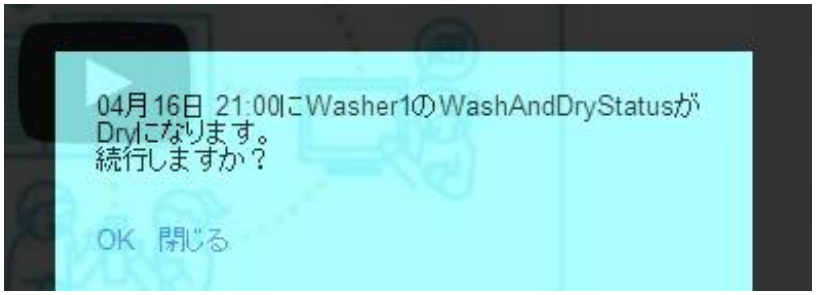



図 5-13　家電調停プロトタイプ D のアンコンシャス情報（拡大）


図 5-14　ユーザがアクセスした動画サイト

### 5.6.4 プロトタイプ D のシステム評価

● **概要**

元々ユーザが視聴しようとしていた Web サービスに影響を与えないという設計要件③に基づき, 調停までに要する処理時間を測定し, 時間的な側面から影響を与えないことを確認する. そこで，プロトタイプ D を用いて，YouTube と他 3 つの動画サイトにアクセスし，提案方式により家電状態の変化を確認してアンコンシャス情報を適用する場合と適用しない場合のぞれぞれの時間を計測した. タスクを実行する際は, 1 分以内に Washer1 の WashAndDryStatus が Dry になる状態を家電状態管理 DB に保持し，1 分以上の映像再生を選択することにより, 確実に家電ダイアログが表示される状況にしてから実施した.

● **結果と考察**

各映像サイトにアクセスした時点から,

従来方式： 映像が再生されるまでの時間

提案方式： アンコンシャス情報が表示されるまでの時間

をそれぞれ 5 回計測し，平均した結果を表 5-3 に示す.

表 5-3 映像再生あるいは情報表示までの所要時間

| 映像配信サイト | 映像再生あるいはアンコンシャス情報表示を開始するまでの平均所要時間（秒） | | |
|---|---|---|---|
| | 従来方式 | 提案方式 | 提案方式により増加した所要時間 |
| Youtube [4] | 5.2 | 6.4 | 1.2 |
| Dailymotion [5] | 7.0 | 12.6 | 5.6 |
| FC2 [6] | 4.8 | 9.6 | 4.8 |
| Yahoo!video [7] | 4.0 | 13.0 | 9.0 |

4) http://www.youtube.com/watch?v=ASO_zypdnsQ
5) http://www.dailymotion.com/video/xcsvd7_bleach-super-good-feeling_music
6) http://video.fc2.com/content/20130417NAU23QFm&t_nnormal
7) http://screen.yahoo.com/episode-14-box-110000011.html

その結果, 提案方式を適用すると適用しなかった場合よりも 1 秒から 9 秒程度長くなることが分かった．これは映像有無の検出と映像の時間長を取得するために，映像視聴を一旦開始しなければ取得できない映像配信の仕様が原因となっていた．しかし，4.6.3 項で考察したように，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 Web ページの更新時間では「10 秒までならユーザの注意力は続く [82]」という基準には収まっている．このことから，提案方式によりサービスを提供することは可能であり，元のサービスに影響を与えない範囲内であると判断できる．また，4.6.3 項で考察した対処方法と同様に，進捗状況を表示しながらバックグラウンドで映像時間長を取得する方法も適用できる．ただし，サービス提供する際は，提供される映像配信の仕様が変わる場合があるため，対応状況を確認しながら進める必要がある．

## 5.7 本章のまとめ

本章では，節電のための家電操作をユーザが手軽に実行可能とするアンコンシャス情報のプロトタイプ C を作成した．このアンコンシャス情報は，家電の状態変化に応じて適切なタイミングでユーザの閲覧画面に重畳して通知される．アンコンシャス情報には節電のための操作を実行するボタンや節電を促す情報を柔軟に組み合わせ，Web 要素として構成されたアンコンシャス情報を，WebSocket を使ってリアルタイムにブラウザに通知し表示する Web 要素通知技術を確立した．節電に関する情報を組み合わせたアンコンシャス情報について被験者評価を実施し，その有効性を確認した．

さらに Web サービスを利用している最中に家電の状態が変化する場合に，サービス利用時間と家電の運転時間を調停するメッセージをアンコンシャス情報として表示する仕組みを実現した．これにより Web サービスを利用している最中に，洗濯乾燥機の運転完了などユーザの対処行動が発生しないように家電を制御したり，あるいは Web サービスの利用開始を一時待機するように促すアンコンシャス情報を表示したりすることにより，Web サービスの利用を中断しない調停方法について述べた．そして，映像視聴における洗濯乾燥機の状態変化のユースケースを示し，そのプロトタイプ D を構築し，サービスとして実現可能であることを確認した．

また，著者らの文献 [27]では，Web ブラウザを搭載した TV と，スマートフォンやタブレットなどのユーザ端末とが連携し，TV では映像をユーザ端末では詳細情報や映像操作のリモコンなどを表示制御するシステムを提案している．このシステムにおいては，アンコンシャス情報をユーザ端末に表示することで，TV の視聴を阻害せずにユーザに気付かせることができる．なお，本論文では複数のユーザを想定しなかったが，複数のユーザ端末をそれぞれ識別・管理し，利用ユーザに応じたアンコンシャス情報を個別に表示することもできる．

以上のことから，家電の状態変化に応じてアンコンシャス情報を拡張できることを示した．

# 第6章　結論

本研究では，インターネットと家電を融合し，人々の暮らしを豊かにする世界の実現を目指し，ユーザが意識していなかったが知らされると役立つアンコンシャス情報を適切な場面で享受できるという，これまでになかった新たな情報獲得を体験する仕組みを提案した．これによってユーザは何も意識しなくても，省エネのために推奨されているが実際に実行すべきタイミングでは気付きにくい行動を簡単に実現することが可能となる．この提案方式を実現するために，アンコンシャス情報を収集する気付きエンジンと，それをユーザに分かるように表示する気付き表示エンジンを開発した．

第１章では，本研究を取り組むに至った背景について述べた．

第２章では，アンコンシャス情報を生成し表示するための設計指針を示した．実現するサービスの具体例を挙げ，その方針とアプローチを示し，関連する従来技術との違いから提案する技術の独自性を明らかにし，気付きエンジンと気付き表示エンジンの設計要件を示した．

第３章では，アンコンシャス情報を表示する基本システムとして，閲覧 Web ページに興味のある番組名があると，録画予約が可能なボタンをアンコンシャス情報として重畳表示する家電情報重畳技術を確立し，プロトタイプ A により実現した．Web を使った番組予約と比較した結果，提案方法は短時間で予約を完了できることを示し，有効であることを確認した．

第４章では周囲の家電に対する意識を排除するために，ある Web サービス利用時に周囲の家電に対し常に意識して操作設定している行動に着目し，意識しなければならない複数の家電設定をアンコンシャス情報として提供し適切な状態を手軽に再現できるといった価値を実現した．そのため，家電情報重畳技術を拡張し，Web と複数の家電状態とを連携させる機能を気付きエンジンに実装した．複数の家電にアクセスする所要時間が長くなると利用中の Web サービスに影響を与える可能性があるため検証実験を行った．検証実験に先立ち，想定される家電の状態設定数をユーザアンケートにより抽出した．そして，その状態設定数を取得する時

間を計測し，Web サービスの指標値を参考にサービスの実現可能性を確認した．

第５章では，家電の状態変化に応じてアンコンシャス情報を表示し，家電の状態やその関連知識に対して意識することからユーザを解放する価値を提供した．具体的には，洗濯乾燥機の脱水運転完了を状態変化として捉え，この洗濯機の状態と乾燥運転を続けて実行するボタンに加えて，外干しを促す天気予報や乾燥運転の電気代をアンコンシャス情報として生成する機能を気付きエンジンに実装した．そして，生成されたアンコンシャス情報を気付きエンジンが指示したタイミングで閲覧中の Web 上に重畳表示する機能を Web 要素通知技術として実装した．さらにユーザアンケートを実施し，意識していなかった天気予報や電気代が外干しを促す節電行動に役立つ情報であることを確認した．加えて，家電の運転完了が気になって Web サービスに集中できなくなることを鑑み，Web サービスを利用する前あるいは利用した後に運転が完了するように気付きエンジンが制御し，アンコンシャス情報によりユーザを導く方法を実現した．動画視聴サービスの利用時に脱水運転が完了するという場面を想定したプロトタイプを実現し，提案方法の処理時間を計測した結果，サービスとして実現可能であることを確認した．

● **実現性の高いサービス**

このような便利で役立つ技術が社会に貢献するためには，実際にサービスとして提供できなければならない．提案方式を実現するためには，既存 Web への気付き表示エンジン機能の追加と，気付きエンジンを実装するホームサーバ等の設置が家庭内に必要である．

前者の対応方法としては 3 つの手段がある．それは，Web ブラウザにブラウザ提供者が事前にあるいはユーザが手動でインストールする手段，HTTP の通信途中で挿入する手段，そして Web を配信する提供側で事前に組み込む手段である．それぞれ一長一短であるが，通信事業者等が提供するルータにバンドルと呼ばれるソフトウェアを配布・インストールすることにより，既存 Web への機能追加は技術的に実現することが可能である．例えば，文献 [97]によれば，共有なプラットフォームである OSGi 規格 [98] [99] [100]に基づき，バンドルを配布する仕組みがサービス提供されている．

また，後者のホームサーバの設置についても，例えばニュースリリース [101]で報道発表さ

れているように，宅内に既に設置されているハードウェアに ECHONET Lite などの家電と通信する機能を追加することは技術的に可能となっている．このように，本研究で必要となる機能はソフトウェアで実現できるため，既存に設置されたハードウェアを活用してサービスを提供し，広くユーザに利用してもらうことが可能である．またプロトタイプで実装したように安価なコンピュータ機器でも実現できることを確認しており，個人でも実現し利用することは難しくない．

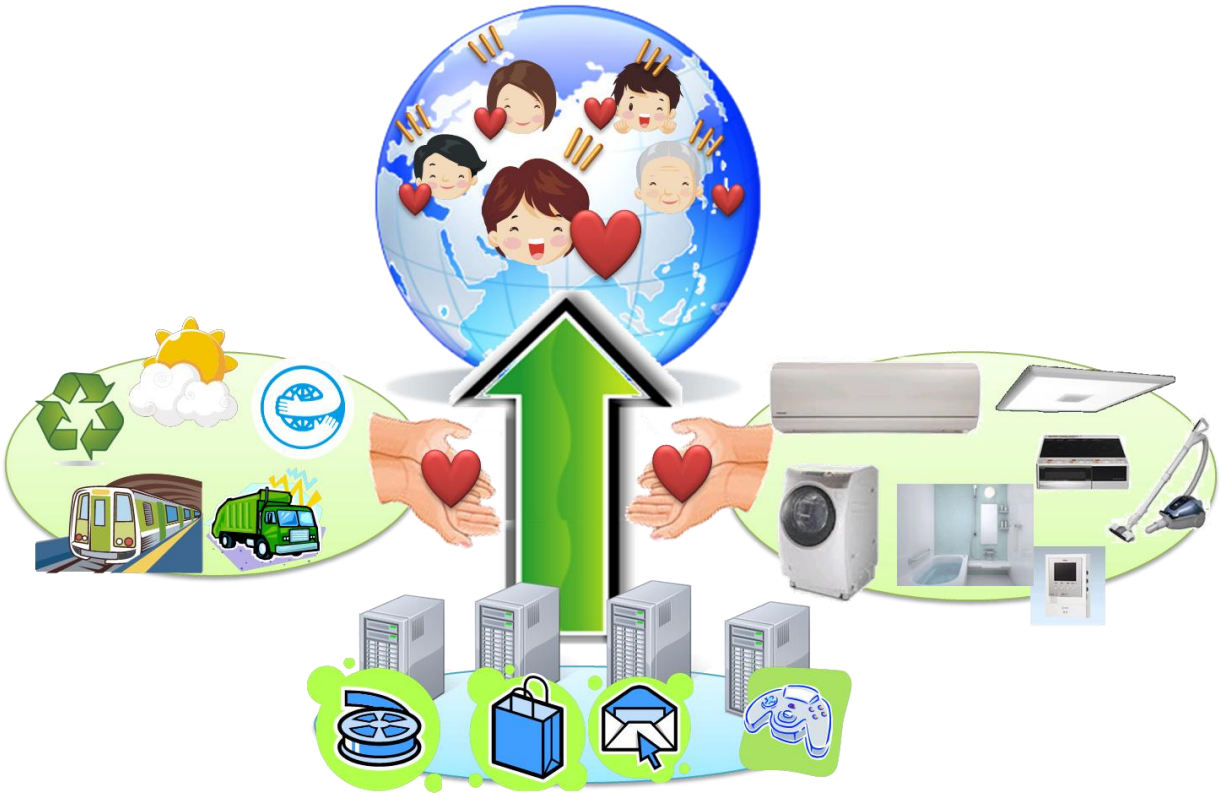
- **社会への貢献**

本研究で実現した気付きエンジンによりアンコンシャス情報をサービス提供することで，人々の生活を豊かにすることができる．家庭内には様々な家電があり，さらに IoT と呼ばれる家具やモノなども含めて今後ますますネットワークにつながる世界が予測されるが，身の回りにあるこのような多種多様なモノが存在する環境に対して，ユーザが意識しなくても役立つ情報が自然と得られ，恩恵を得ることができる．特に昨今のエネルギー問題に対しては大変有効であり，電力利用を考慮した節電運転やユーザの節電行動をアンコンシャス情報として提供することで，意識しなくとも自然と導かれ実施することができる．そしてそれは日本だけでなく，グローバルに活用することができ，全世界の人々が豊かな生活を過ごすことに貢献できる．

アンコンシャス情報は電力だけでなく，日常生活における様々な分野で活用できる．近年，65 歳以上の人口が国民の約 4 分の 1 という高齢化社会を迎えている日本 [20]においては，高齢者の振り込め詐欺被害の抑止や薬の服用忘れ防止など，アンコンシャス情報の表示により日常生活でのサポートにも貢献する．また急増する介護の現場で要支援の方々に使ってもらい，脳活性化やモノ忘れ防止の支援ツールとして活用することを強く推奨していきたい．

さらに，アンコンシャス情報の表示方法についても，本論文では Web 上に重畳表示する方式を示したが，目の不自由な方には大きな文字や色合いを変えて表現するといった，使うユーザに応じて対応できるアンコンシャス情報のパーソナライズについても考案（特許登録）している．

このようにスマートフォンなど Web 技術が組み込まれた端末と本技術，さらには音声による出力機能などを有機的に組み合わせて提供することでさらに発展し，未来のあらゆる生活シーンにおいて便利で豊かな生活の実現に貢献できる．本研究のアンコンシャス情報により与えられる気付きにより，人と家電，人と環境，そして人と人の間でより良く連携できる関係を築くことに貢献すると信じる．



# 謝辞

本研究を進めるにあたり，多くの方々に大変御世話になりました．ここに深く感謝の意を表します．

研究活動全般にわたり格別なる御指導と御高配を賜りました 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 一色正男 教授に甚大なる謝意を表します．社会人ドクタとして企業での活動を行いながらも３年間で博士論文をまとめることができたのは，一色先生が温かくも力強く励まし，博士としての見識を指導してくださったおかげです．著者の研究の基本要素となっている Web と家電の両方の分野に精通しグローバルで活躍されている一色先生に御指導頂けたことは，著者にとって大変幸運でした．一色先生から御指導頂いた，楽しく世の中に役に立つという研究姿勢を大切にし，研究者として世の中に役に立つよう今後も努めていく所存です．

神奈川工科大学博士課程への進学の機会と多大なる御指導を頂きました 長崎大学大学院工学研究科電気・情報科学部門 小林透 教授に心より感謝申し上げます．小林先生は，前職の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で上司として著者を指導してくださった後も気にかけてくださり，何度となく相談にのっていただきました．博士課程への進学は著者にとって人生の大きな分岐点でした．その機会を与えてくださり，そして著者をいつも励ましてくださった御恩は一生忘れません．本当にありがとうございました．

博士論文について貴重なご教示を賜りました 神奈川工科大学 武尾英哉 教授，奥村万規子 教授，黄啓新 教授，松本一教 教授，電気通信大学 大須賀昭彦 教授 に心より感謝申し上げます．先生方の御助言により，主張点がより明確になり，論文の完成度を高めることができました．本当にありがとうございました．

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 杉村博 助教に感謝いたします．杉村先生には，論文の書き方や ECHONET Lite のプログラムなど，細かな相談にも分かりやすく御指導を頂きました．また雑談にも御付き合いいただき，凝り固まった頭をリフレッシュすることができました．ありがとうございました．

一色研究室の皆様には３年間にわたり多方面で御支援を頂きました．特に，神奈川工科大学一色研究室卒業生の甲斐純平 様，栗田拓実 様，中尾圭佑 様，阿部聡明 様，池田雅人 様，稲田貴仁 様にはアンケート調査や分析など著者の研究を直接手伝って頂きました．また，博士課程の宇佐美真 様，村上隆史 様，修士課程の横山悠平 様，中島義人 様とは，研究議論で有益なアイデアを頂いたり，研究室での事務作業を助けて頂いたりしました．いつも明るく楽しい研究室で皆様方と出会えたことは大変貴重な財産です．一色研究室の博士第１号となれたことを大変光栄に思うと同時に，皆様方に深く感謝いたします．

神奈川工科大学スマートハウス研究センターの関家一雄 様，笹川雄司 様，町田誠志 様には，実機の ECHONET Lite 家電を使った実験で大変お世話になりました．データ収集においては細やかな気配りを頂きまして大変ありがとうございました．

株式会社富士通研究所 森信一郎 様には，学会活動を通じて著者の研究の方向性に貴重な御指導を賜りました．論文とは何かを諭し御教示いただいたおかげで，止まっていた論文の執筆活動を進めることができました．深く感謝いたします．

著者の勤務先である 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の茨木久 前所長，川添雄彦 現所長，如澤裕尚 企画部長をはじめ，所属するプロジェクトの上司・諸先輩方には，博士号取得の進捗状況を心配して頂き，応援や支援を賜りました．ここに感謝の意を表します．

最後になりましたが，著者の博士課程進学を遠くから見守り続けてくれた両親に深く感謝いたします．そして，いつもそばで優しく応援してくれる妻 眞由美に深く深く感謝します．

本研究の成果がお世話になった皆様のご期待に添えるか疑問は残りますが，ここに重ねて厚く謝意を表し，謝辞とさせていただきます．

# 参考文献


[1] Digital Living Network Alliance, "DLNA for Industry," http://www.dlna.org/, 2014.

[2] Apple, "Apple - AirPlay - iOS デバイスにあるコンテンツを Apple TV で再生。," http://www.apple.com/jp/airplay/, 2014.

[3] エコーネットコンソーシアム, "ECHONET CONSORTIUM," http://www.echonet.gr.jp/index.htm, 2014.

[4] HEMS(ECHONETLite)認証センター, "HEMS(ECHONETLite)認証センター," http://sh-center.org/, 2014.

[5] 山崎毅文, "HEMS 分野におけるホームネットワークでの通信プロトコル標準化動向について," *日本 ITU 協会 ITU ジャーナル,* vol. 43, no. 11, pp. 36-38, 2013.

[6] 杉村博，関家一雄，渡部智樹，一色正男, "Web サービスによる HEMS 機器相互接続テスト環境の開発," *電気学会論文誌 C（電子・情報・システム部門誌）研究開発レター,* vol. 133, no. 4, pp. 818-819, 2013.

[7] スマートライフジャパン, "スマートライフジャパン," http://smart-life-japan.jp/, 2014.

[8] Atzori Luigi，Antonio Iera and Giacomo Morabito, "The internet of things: A survey," *Computer Networks 54.15,* vol. 54, no. 15, pp. 2787-2805, 2010.

[9] 横谷哲也, "IoT/M2M を取り巻く環境と標準化(クリティカル応用のための通信技術)," *日本信頼性学会誌 ： 信頼性,* vol. 34, no. 8, pp. 540-545, 2012.

[10] 一色正男, 笹川雄司, "HEMS のための WoT(IoT) (特集 ネットがあらゆるモノとつながることで社会・生活が変わる! モノのインターネット)," *OHM,* vol. 101, no. 3, pp. 12-14, 2014.

[11] Simon Duquennoy, Gilles Grimaud, Jean-Jacques Vandewalle, "The Web of Things: Interconnecting Devices with High Usability and Performance," *Embedded Software and Systems, 2009. ICESS '09. International Conference on,* pp. 323-330, 2009.

[12] 象印マホービン株式会社, "みまもりほっとラインー親の元気がポットでわかるー," http://www.mimamori.net/, 2014.

[13] 宮田章裕，有賀玲子，宮下広夢，佐藤隆，井原雅行，小林透, "貼りつけるだけで家具をインテリジェント化するデバイス," *情報処理学会インタラクション 2013 インタラクティブセッション,* 2013.

[14] 総務省, "スマートフォン等の急速な普及と端末市場の変化," *H24 年度版情報通信白書,* 2013.

[15] 片岡泰之，渡部智樹，田中清，東野豪, "モバイルアプリのファセット検索を実現するインデクシング手法," *情報処理学会研究報告. UBI, [ユビキタスコンピューティングシステム],* vol. 2013, no. 5, pp. 1-6, 2012.

[16] 東芝, "レグザ Apps コネクト," http://www.toshiba.co.jp/regza/apps/index_j.htm, 2014.

[17] シャープ, "COCOROBO SQUARE," シャープ, http://www.sharp.co.jp/cocorobo/manual/square.html, 2014.

[18] Panasonic, "ルームエアコン「X シリーズ」を発売 | プレスリリース," http://panasonic.co.jp/corp/news/official.data/data.dir/2013/09/jn130918-2/jn130918-2.html, 2014.

[19] 国土交通省, "第 2 節　震災後の国民意識の変化," *国土交通白書 2012, 第 2 部,* 2012.

[20] 内閣府, "（1）高齢化率が 24.1%に上昇," *平成 25 年版　高齢社会白書（全体版）*, 2014.

[21] 藤井章博, "広がる Web API の活用 ―マッシュアップの幅広い可能性―," *科学技術動向,* no. 106, pp. 9-18, 2010.

[22] W3C, "W3C HTML Working Group," http://www.w3.org/TR/html5/, 2014.

[23] W3C, "HTML5 勧告–オープン・ウェブ・プラットフォームの重要なマイルストーンを達成," http://www.w3.org/2014/10/html5-rec.html.ja, 2014.

[24] 石井晋司，渡部智樹，井原雅行，小林透, "次世代のコンテンツ流通にかかわる W3C における標準化動向," *電気通信協会 NTT 技術ジャーナル,* vol. 25, no. 1, pp. 56-59, 2013.

[25] Hisayuki Ohmata, Masaru Takechi, Shigeaki Mitsuya, Kazuhiro Otsuki, "Hybridcast: A new media experience by integration of broadcasting and broadband," *ITU Kaleidoscope: Building Sustainable Communities (K-2013), 2013 Proceedings of ,* pp. 1-8, 2013.

[26] 小松健作, “HTML5 で理解する次世代 Web(第 6 回)標準化は?普及見通しは? HTML5 の今後を見通す,” *日経コンピュータ,* 第 778, pp. 126-129, 2011.

[27] Maiko Imoto, Hiroki Watanabe, Tomoki Watanabe, and Shinji Miyahara, "Flexible Assignment of Components of Web Pages for Multiple Device Environments," in *AVI 2014 International Working Conference on Advanced Visual Interfaces*, ACM, 2014.

[28] 高塚広貴，佐伯幸郎，まつ本真佑，中村匡秀, "異種分散 Web サービスに基づくコンテキストアウェアサービスの管理プラットフォームの実装," *電子情報通信学会技術報告 LOIS 研究会,* vol. 113, no. 327, pp. 71-76, 2013.

[29] 小野一善，小田直規，高河原和彦, "業界の垣根を超えて結実したウェアラブルセンサ -hitoe 技術," *電気通信協会ＮＴＴ技術ジャーナル,* vol. 26, no. 5, pp. 42-44, 2014.

[30] 塚田信吾，河西奈保子，川野龍介, "着るだけで心電図を測るウェアラブル電極インナー (特集新分野事業の開拓に貢献する先端デバイス・材料技術)," *電気通信協会 NTT 技術ジャーナル,* vol. 26, no. 2, pp. 15-18, 2014.

[31] 清川清, "バーチャルリアリテイにおける視覚提示技術(<特集>最新バーチャルリアリティ)," *知能と情報 : 日本知能情報ファジィ学会誌 : journal of Japan Society for Fuzzy Theory and Intelligent Informatics,* vol. 19, no. 4, pp. 318-325, 2007.

[32] 中尾誠幸，寺田努，塚本昌彦, "ウェアラブルコンピューティング環境における周辺環境を考慮

した装着型ディスプレイへの情報提示手法," *情報処理学会研究報告. HCI, ヒューマンコンピュータインタラクション研究会報告,* vol. 2012, no. 9, pp. 1-8, 2012.

[33] 関口克己，北口雅哉，鶴巻宏治, "透過型 Proxy 方式の実装と評価," *電子情報通信学会技術研究報告. RCS, 無線通信システム,* vol. 101, no. 684, pp. 99-105, 2002.

[34] 大盛善啓，樋口靖和，菊池匡晃, "家電機器への IP リモコン機能組込みを容易にするソフトウェアプラットフォーム," *東芝レビュー,* vol. 67, no. 6, pp. 24-27, 2012.

[35] ソニー, "CHAN-TORU 機能詳細 | ソニーのヒト・コト |," ソニー, http://www.sony.jp/hitokoto/weblabo/chantoru/details.html, 2014.

[36] Panasonic, "ディーガのネットサービス「Dimora」＆「Memora」," http://panasonic.jp/diga/dimora_memora/index.html, 2014.

[37] Masahide Nakamura, Akihiro Tanaka, Hiroshi Igaki, Haruaki Tamada and Ken-ichi Matsumoto, "Adapting Legacy Home Appliances to Home Network Systems Using Web Services," ICWS'06, 2006.

[38] 井垣宏，中村匡秀，玉田春昭，松本健一, "サービス指向アーキテクチャを用いたネットワーク家電連携サービスの開発," *情報処理学会論文誌,* vol. 46, no. 2, pp. 314-326, 2005.

[39] Koji Tsukada, Michiaki Yasunuma, "Ubi-Finger: Gesture Input Device for Mobile Use," *Proceedings of APCHI 2002,* vol. 1, pp. 388-400, 2002.

[40] 榊原弘記，中村匡秀，井垣宏，松本健一, "ホームネットワークシステムにおける家電状態を利用した音声操作インタフェースの改善," *電子情報通信学会 2 種研究会 サイバーワールド(CW)第 9 回研究会,* pp. 13-18, 2008.

[41] 内田尚和，常盤大樹，高木朗，麻生英樹，森彰，橋本政朋，伊東幸宏，小林一郎，中島秀之，八名和夫, "意味の位置づけを可能にする意味表現を用いた情報家電操作のための対話的インタフェース," *人工知能学会全国大会論文集,* vol. 5, pp. 20-20, 2005.

[42] U.S. Department of Energy, "State Energy Program," http://www1.eere.energy.gov/wip/sep.html, 2014.

[43] KNX Association, "KNX Association," http://www.knx.org/knx-en/index.php, 2014.

[44] Sundramoorthy, V., Qi Liu, Cooper, G., Linge, N. and Cooper, J., "DEHEMS: A user-driven domestic energy monitoring system," *Internet of Things (IOT), 2010,* vol. 29, pp. 1-8, 2010.

[45] 松村剛志，安川健太，村上慎吾，村上慎吾，ヨハンイェルム，小田稔周, "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. IN, 情報ネットワーク,* vol. 110, no. 449, pp. 181-186, 2011.

[46] Ha, Young-Guk, "Dynamic Integration of Zigbee Home Networks into Home Gateways Using OSGi Service Registry," *IEEE Transactions on Consumer Electronics,* vol. 55, no. 2, pp. 470-476, 2009.

[47] P. Kinney, "ZigBee technology: wireless control that simply works," Proc. of Communications Design Conference, 2003.

[48] M. Inoue, T. Higuma, Y. Ito, N. Kushiro, and H. Kubota, "Network Architecture for Home Energy Management System," *IEEE Transactions on Consumer Electronics,* vol. 49, no. 3, pp. 606-613, 2003.

[49] Seong Ho Ju, et al, "An Efficient Home Energy Management System based on Automatic Meter Reading," *Power Line Communications and Its Applications (ISPLC), 2011 IEEE International Symposium on,* pp. 479-484, 2011.

[50] Shigeru Owada, Fumiaki Tokuhisa, "Kadecot: HTML5-based Visual Novels Development System for Smart Homes," *Consumer Electronics (GCCE), 2012 IEEE 1st Global Conference on,* pp. 17-19, 2012.

[51] 東芝, "TOSHIBA CES2013 特設サイト次世代のデジタルライフを実現!," http://www.ad-toshiba.jp/exhibition/ces2013/, 2014.

[52] 福田将之，井垣宏，中村匡秀, "ホームネットワークシステムにおけるリアルタイムな家電制御サービスの実現," *電子情報通信学会技術研究報告，情報ネットワーク研究会,* vol. 108, no. 136, pp. 41-46, 2008.

[53] Hiroshi SUGIMURA, Kazuki UTSUMI, Masayuki KANEKO, Kazuki ARIMA, Masahiro Sakamoto, Yuki Imaizumi, Tomoki Watanabe and Masao Isshiki, "Development of Immersive Display System of Web," in *SS-STV-1: Smart TV (1), GCCE, IEEE*, 2014.

[54] 阿部聡明，池田雅人，渡部智樹，杉村博，一色正男, "家電の状態に応じたユーザーへの通知システム," 情報処理学会 第 76 回全国大会, 5Y-1, 2014.

[55] 岸下直弘，間下以大，清川清，竹村治雄, "広視野シースルーHMD による情報提示のための VR 環境を用いた周辺視野の影響の調査," *電子情報通信学会技術研究報告. MVE,* マルチメディア・*仮想環境基礎,* vol. 112, no. 106, pp. 85-90, 2012.

[56] 筧康明, "interFORest プロジェクト：Discover Shiretoko キャンペーンにおける Web と実世界をつなぐ試み," *電子情報通信学会技術研究報告. MVE,* マルチメディア・*仮想環境基礎,* vol. 109, no. 215, pp. 41-46, 2009.

[57] セレゴ・ジャパン株式会社, "iKnow! ポップアップ辞書," http://iknow.jp/, 2014.

[58] 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. 2004, no. 112, pp. 21-27, 2004.

[59] 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クストア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),* vol. 91, no. 1, pp. 22-34, 2008.

[60] 藤波香織，カウサルファヒム，中島達夫, "鏡を拡張したコンテクストアウェア情報表示装置," *情報処理学会論文誌,* vol. 49, no. 6, pp. 1972-1983, 2008.

[61] 渡部智樹，小林稔，阿部匡伸, "ユーザの操作を記録し活用するライフログリモコン," *NTT 技術ジャーナル,* vol. 22, no. 7, pp. 16-19, 2010.

[62] 渡部智樹，青木良輔，井原雅行，小林稔，阿部匡伸, "「リモコン信号プロクシ」を用いた機器

操作ログ収集システム," *電子情報通信学会技術研究報告. LOIS, ライフインテリジェンスとオフィス情報システム,* vol. 110, no. 141, pp. 23-28, 2010.

[63] 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.

[64] So-net Corporation, "テレビ番組表【G ガイド.テレビ王国】," http://tv.so-net.ne.jp/, 2014.

[65] W3C, "Cascading Style Sheets," http://www.w3.org/Style/CSS/, 2014.

[66] W3C, "W3C Document Object Model," http://www.w3.org/DOM/, 2014.

[67] Mozilla Japan, "次世代ブラウザ Firefox とメールソフト Thunderbird," http://www.mozilla.jp/, 2014.

[68] NTT 東日本, "B フレッツ | サービス・料金一覧 | フレッツ光," https://flets.com/bflets/, 2014.

[69] 青木良輔，渡部智樹，小林透, "アンビエントな家電操作実現に向けた家電機器状態ログ収集システムの提案," *情報処理学会 CDS 研究会トランザクション,* vol. 2, no. 2, pp. 63-72, 2012.

[70] 上坂洋紀，秋田純一，北川章夫，美馬義亮, "スポコン: 狙う動作によって機器選択を行うリモートコントローラ," *インタラクション 2011 論文集,* pp. 729-732, 2011.

[71] 湯浅直弘，伊原誠人，中村匡秀，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のユーザビリティ評価," *電子情報通信学会技術研究報告. IN, 情報ネットワーク,* vol. 106, no. 578, pp. 399-404, 2007.

[72] Yasuyuki Kataoka, Tomoki Watanabe, Kiyoshi Tanaka and Tomohiro Yamada, "Consumer Device Recommendation Method for Web-Based Distributed Browsing," *Consumer Electronics (ICCE), 2014 IEEE International Conference on,* pp. 87-88, 2014.

[73] 渡部智樹，高嶋洋一，杉村博，一色正男, "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," *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. 5, no. 1, pp. 38-46, 2015.

[74] IETF, "RFC 3875 - The Common Gateway Interface (CGI) Version 1.1," http://tools.ietf.org/html/rfc3875, 2014.

[75] 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," *研究報告コンシューマ・デバイス&システム（CDS）,* Vols. 2012-CDS-3, no. 22, pp. 19-28, 2012.

[76] 市川武男，石原浩一，村上友規, "ワイヤレスホームネットワークを実現する高速無線 LAN (特集 ネットワーク連携で利便性を高めるワイヤレスアクセス技術)," *電気通信協会 NTT 技術ジャーナル,* vol. 25, no. 1, pp. 18-22, 2013.

[77] RASPBERRY PI FOUNDATION, "Model B | Raspberry Pi," http://www.raspberrypi.org/product/model-b/, 2014.

[78] Brock, J. Dean, Rebecca F. Bruce, and Marietta E. Cameron., "Changing the world with a Raspberry Pi," *Journal of Computing Sciences in Colleges,* vol. 29, no. 2, pp. 151-153, 2013.

[79] Trapp Brian, "Raspberry Pi: The Perfect Home Server," *Linux Journal,* vol. 229, no. May 2013, 2013.

[80] IETF, "RFC 4627 - The application/json Media Type for JavaScript Object Notation (JSON)," http://tools.ietf.org/html/rfc4627, 2014.

[81] エコーネットコンソーシアム, "ECHONET Lite 規格書 Ver1.01（日本語版） 第 2 部 ECHONET Lite 通信ミドルウェア仕様," エコーネットコンソーシアム, 2 章 2-4～2-8, 2012.

[82] IID, Inc., "Web サイトの応答時間," http://www.usability.gr.jp/alertbox/20100621_response-times.html, 2014.

[83] Nielsen Norman Group, "How Long Do Users Stay on Web Pages?," http://www.nngroup.com/articles/how-long-do-users-stay-on-web-pages/, 2014.

[84] CROSS-STYLE, "CROSS-STYLEーウェブデザイン講座「８秒ルールは今は昔！？進化した「３秒絶対ルール」とは？」," http://www.cross-style.com/webdesign/006.html, 2014.

[85] 永井学, "特集 システム戦略 Web サイト高速化のツボ--「3 秒ルール」に対応して商機を掴め," in *日経ネットビジネス 2002 年 12 月号*, 日経 BP 社, pp. 99-113 , 2002

[86] Tomoki Watanabe，Rika Mochizuki，Toru Kobayashi and Masao Isshiki, "HACCS: Home Appliance Control Concierge System: Extending Functions on Web Service," *Computer Software and Applications Conference (COMPSAC), 2013 IEEE 37th Annual,* pp. 208-213, 2013.

[87] 渡部智樹，望月理香，小林透，杉村博，一色正男, "視聴映像の時間長を考慮した家電制御システム," *情報処理学会研究会報告 CDS 研究会,* vol. 7, no. 4, pp. 1-8, 2013.

[88] W3C, "The WebSocket API," http://www.w3.org/TR/2011/WD-websockets-20110929/, 2014.

[89] 増尾剛，中村二朗，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," *電子情報通信学会技術研究報告. NS, ネットワークシステム,* vol. 112, no. 350, pp. 1-6, 2012.

[90] Keita Watanabe, Shota Matsuda, Michiaki Yasumura, Masahiko Inami, and Takeo Igarashi, "CastOven: a microwave oven with just-in-time video clips," *ACM Ubicomp '10,* pp. 385-386, 2010.

[91] 丸尾彰宏，松尾周平，まつ本真佑，中村匡秀, "サービス指向ホームネットワークにおける複数センサとタイミング制約を用いた高度コンテキストの抽出," *電子情報通信学会技術研究報告. IN, 情報ネットワーク,* vol. 110, no. 449, pp. 187-192, 2011.

[92] Wikipedia, "HTML/Elements/video - W3C Wiki," http://www.w3.org/wiki/HTML/Elements/video, 2014.

[93] 戸田航史，中道上，島和之，大平雅雄，阪井誠，松本健一, "Web ページ閲覧者の視線に基づいた情報探索モデルの提案," *情報処理学会研究報告. HI, ヒューマンインタフェース研究会報告,* vol. 2005, no. 52, pp. 35-42, 2005.

[94] 川原圭博，司化，猪鹿倉知広，登内敏夫，森川博之，青山友紀, "行動履歴と制約条件を考慮した情報家電制御機構," *情報処学会研究報告ユビキタスコンピューティングシステム研究会(UBI),* vol. 10, no. 6, pp. 55-60, 2006.

[95] 山田祐輔，加藤丈和，松山隆司, "スマートタップネットワークを用いた家電の電力消費パターン解析に基づく人物行動推定," *電子情報通信学会技術研究報告 USN,* vol. 111, no. 134, pp. 25-30, 2011.

[96] Google, "Chrome Web Store – Tampermonkey," https://chrome.google.com/webstore/detail/tampermonkey/dhdgffkkebhmkfjojejmpbldmpobfkfo?hl=ja, 2014.

[97] NTT, "フレッツ・ジョイントサービスを支えるホーム ICT," *電気通信協会 NTT 技術ジャーナル,* vol. 24, no. 2, pp. 13-15, 2012.

[98] OSGi Technology, "OSGi Technology," http://www.osgi.org/About/Technology, 2014.

[99] NTT, "OSGi サービス・アグリゲーション・プラットフォーム（OSAP）," http://www.ntt.co.jp/RD/OFIS/active/2007pdf/pdf/h_pf01.pdf, 2014.

[100] 丹康雄, "Chapter 5 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(制御・通信プロトコル,<特集>エネルギーの情報化～IT による電力マネジメント～)," *情報処理,* vol. 51, no. 8, pp. 959-965, 2010.

[101] NTT 西日本, "MEMS アグリゲーター向け「光 BOX＋」対応機器の拡充について," http://www.ntt-west.co.jp/news/1410/141003a.html, 2014.

# 図目次

# 表目次

# 付録

(ア) HTTP プロキシ方式によるスクリプト（気付き表示エンジン）挿入のソース（抜粋）

```
#!/usr/bin/env ruby
# Isshiki.Lab@KAIT Tomoki Watanabe
# HTTP Proxy for Injection my Script
#ブラウザ判定
def DecisionBrowser(req)
   agent = req.header['user-agent'].to_s
   type = ""
   if agent.include?("Safari")
   if agent.include?("iPad")#ipad
      type = "ipad"
      puts ">>>>>ipad"
   end
   if agent.include?("Android")#Android
      type = "Android"
      puts ">>>>>Android"
   end
   if agent.include?("Windows")#Windows
      type = "Windows"
      puts ">>>>>Windows"
   end
   end
   return type
end

#スクリプト挿入
def AddScript(res_body, type, enco)
   new_res_body = ""
   res_body.lines do |line|
   doc = Hpricot(line)
   hrefs = (doc/:head)
   hrefs.each do |head|
   if type == "ipad"
      puts "Add script for ipad"
```

```
      if line.include?("<head>") == true || line.include?("<HEAD>") == true
      line = line.gsub(/<head>|<HEAD>/,"<head><script type=¥"text/javascript¥"
src=¥"https://www.google.com/jsapi¥"></script><script type=¥"text/javascript¥" src=¥"http://"+ $ipaddress +
"/noticeLayerApp.js¥"></script>¥n")
      elsif line.include?("<head") == true || line.include?("<HEAD") == true
        line += "<script type=¥"text/javascript¥" src=¥"https://www.google.com/jsapi¥"></script><script
type=¥"text/javascript¥" src=¥"http://"+ $ipaddress + "/noticeLayerApp.js¥"></script>¥n"
      end
   end
   if type == "Android"
      puts "Add script for Android"
      if line.include?("<head") == true || line.include?("<HEAD") == true
      line = line.gsub(/<head>|<HEAD>/,"<head><script type=¥"text/javascript¥"
src=¥"https://www.google.com/jsapi¥"></script><script type=¥"text/javascript¥" src=¥"http://"+ $ipaddress +
"/noticeLayerApp.js¥"></script>¥n")
      elsif line.include?("<head") == true || line.include?("<HEAD") == true
        line += "<script type=¥"text/javascript¥" src=¥"https://www.google.com/jsapi¥"></script><script
type=¥"text/javascript¥" src=¥"http://"+ $ipaddress + "/noticeLayerApp.js¥"></script>¥n"
      end
   end
   if type == "Windows"
      puts "Add script for Windows"
      if line.include?("<head>") == true || line.include?("<HEAD>") == true
        line = line.gsub(/<head>|<HEAD>/,"<head><script type=¥"text/javascript¥"
src=¥"https://www.google.com/jsapi¥"></script><script type=¥"text/javascript¥" src=¥"http://"+ $ipaddress +
"/noticeLayerApp.js¥"></script>¥n")
      elsif line.include?("<head") == true || line.include?("<HEAD") == true
        line += "<script type=¥"text/javascript¥" src=¥"https://www.google.com/jsapi¥"></script><script
type=¥"text/javascript¥" src=¥"http://"+ $ipaddress + "/noticeLayerApp.js¥"></script>¥n"
      end
   end
    end
   new_res_body += line
   end
   return new_res_body, flag
end
```

(イ) 気付き表示エンジンを実現する JavaScript ソース（抜粋）

```
///気付きレイヤ表示用スクリプト
// Isshiki.Lab@KAIT Tomoki Watanabe
// huwahuwa notification

// 設定項目
var KeywordServerIP = "192.168.0.8"; //キーワード配信サーバ IP アドレス
var KeywordServerPort = "15488"; //ポート番号（Websocket 用）
var KeywordShow = 5; //一度にタグを表示するキーワード個数
var ClickCount_toSave = 5; //X回タグがクリックされたら操作ログを送信する
var timeflag = []; //番組開始時間が現在か未来か

//ページ読み込み時 Webscoket でキーワードリストを取得
window.addEventListener("load", function () {
  //再度優先度の上位から表示
  on_off = 1;
  //再度ダミータグを付与
  tag_set_flag = false;
  try {
    //Websocket を利用したキーワードリスト取得
    ws = new WebSocket("ws://" + KeywordServerIP + ":" + KeywordServerPort); //接続
    //データ受信時
    ws.onmessage = function (evt) {
      _json = evt.data;
    };
    //接続完了時
    ws.onopen = function () {
      ws.send("loaded");
    };
    //エラー
    ws.onerror = function (ex) {
      window.alert(ex.data);
    }
  }
  catch (e) {
    //window.alert(e.message);
  }
},
```

```
false);

//タッチイベントを取得　Android,iPad 用
document.addEventListener("touchstart", function () {
  // マルチタッチ（４本）
  if (event.touches.length == 4) {
    main(); //タグの表示
  }
},
false);

/* タグ表示準備 */
function main()
{
  //キーワードリストの有無をチェック
  if (_json == "") {
  //websocket 通信ができなければ http 通信でキーワードリストを取得
    var req = new XMLHttpRequest();
    req.open("GET", "http://" + KeywordServerIP + "/getjson.php", false);
    req.send(null);
    _json = req.responseText;
    if (_json == "") {
      window.alert("【タグ表示エラー】¥n キーワードリストが未取得です。")
      return;
    }
  }
  //ページ内の全文字列を取得
  var doc = document;
  var result = doc.evaluate(XPATH, doc.body, null, PathResult.ORDERED_NODE_SNAPSHOT_TYPE,
null);
  for (var i = 0; i < result.snapshotLength; ++i)
  {
    var node = result.snapshotItem(i); //ページ内文字列
    var text = node.nodeValue.replace(/^¥s+|¥s+$|¥r|¥n/g, "");
    if (text.length == 0) continue;
    //キーワードリストとページ内文字列のマッチング処理
      var nodeStr = stringChecker(node.nodeValue);
      var matchedArr = node.nodeValue.match(keyword);
```

```
        var links = [];
        //各ページ内文字列にキーワードが含まれているかチェック
        if (matchedArr != null) {
          for (var y = 0; y < matchedArr.length; y++) {
            var str = matchedArr[y];
            var newNode = node.splitText(nodeStr.indexOf(str));
            node = newNode.splitText(str.length);
          for (var y = 0; undefined != name[y]; y++) {
            //マッチング（文字列の完全一致）
            if (str == name[y]) {
              var listcounter = y;
              //ダミータグ付与（RC）
              var aElem = doc.createElement("RC");
              aElem.setAttribute("style", "text-decoration:none;");
              aElem.setAttribute("title", str);
              aElem.setAttribute("target", "_blank");
              var textNode1 = doc.createTextNode("");
              var textNode2 = doc.createTextNode("");
              aElem.appendChild(textNode1);
              aElem.appendChild(newNode.cloneNode(true));
              aElem.appendChild(textNode2);
            }
          }
          node.parentNode.replaceChild(aElem, newNode);
          links.push({ target: node.parentNode, aElem: aElem, newNode: newNode });
        }
      }
    }
    tag_set_flag = !tag_set_flag;
    //タグの表示処理(1)へ
    RCtags(json);
}

/*  タグ表示処理(1)    */
function RCtags(json){
    var doc = document;
    var tagContainer;
    //タグ用表示用のタグを作成（TAGS）
```

```
    tagContainer = doc.getElementById('tag_container');
    if (tagContainer) {
        doc.body.removeChild(tagContainer);
    }
    tagContainer = doc.createElement('TAGS');
    tagContainer.id = "tag_container";
    doc.body.appendChild(tagContainer);
    var tagTemplate = doc.createElement('SPAN');
    tagTemplate.style.cssText = 'z-index:500;position:absolute;display:inline;';
    tagTemplate.setAttribute('name', 'tag');
    //ダミータグから各値を取得
    var res = doc.evaluate('//RC', doc, null, XPathResult.ORDERED_NODE_SNAPSHOT_TYPE, null);
    //位置調整
    var area = new Array(4);
    if (window.scrollY<100) {
        area[0] = window.pageXOffset ;
        area[1] = window.pageYOffset ;
        area[2] = area[0] + window.innerWidth ;
        area[3] = window.pageYOffset + window.innerHeight*1;
        assignTags(tagContainer, tagTemplate, res, area, doc, json);//タグ表示処理(2)へ
    }    else    {
        area[0] = window.pageXOffset ;
        area[1] = window.pageYOffset ;
        area[2] = area[0] + window.innerWidth ;
        area[3] = window.pageYOffset + window.innerHeight*1;
        assignTags(tagContainer, tagTemplate, res, area, doc, json);//タグ表示処理(2)へ
    }
    //アニメーション処理
    $("#tag_container IMG", document).animate({ width: "88", height: "33" }, 300).animate({ width: "79",
height: "29" }, 150);
    //タグのクリックイベント設定
    var element = document.getElementById('tag_container');
    element.addEventListener("click", function (event) {
        click(json);
    }, false);
}
```

```
/*タグ表示処理(2) */
function assignTags(tagContainer, tagTemplate, res, area, doc, json) {
   //優先度順にソートされたキーワードを取得
   var prioritylist = priorityTags(json);
   //ページ内でマッチした数だけ処理する
   for (var i = 0; i < res.snapshotLength; i++) {
   //文字列が非表示であれば処理しない
      var node = res.snapshotItem(i);
      var cs = doc.defaultView.getComputedStyle(node, null);
      if (cs.getPropertyValue("visibility") == "hidden")
         continue;
      //位置情報取得
      getAbsolutePositions(node, 1);
      //表示オフ
      if (node.absTop < area[1] || node.absTop >= area[3]
            || node.absLeft > area[2] || node.absLeft < area[0])
         continue;
      //表示オフ
      if (on_off == 0) {
         continue;
      }
      tagElem = tagTemplate.cloneNode(false);
      tagContainer.appendChild(tagElem);
      //一回の表示で(KeywordShow)個のキーワードまで表示
      var z = (Number(on_off) * KeywordShow);
      for (x = (z - KeywordShow); (z - KeywordShow) <= x && x < z; x++) {
         if (prioritylist[x] != undefined) {
            var y = prioritylist[x];
            if (node.textContent == json.items[y].name) {
               //リンク文字列(操作コマンド)生成
               s_time = json.items[y].starttime.split(" ");
               e_time = json.items[y].endtime.split(" ");
               //JSON 項目"url"が undefined であれば通常通りのリンク文字列を生成
               if (json.items[y].url == undefined || json.items[y].url == "") {
                  //リンク文字列
                  url += "http://" + KeywordServerIP + "/204.php?mame=" + json.items[y].name + "&channel="
+ json.items[y].channel + "&starttime=" + s_time[0] + " " + s_time[1] + "&endtime=" + e_time[0] + " " +
e_time[1] + "&priority=" + json.items[y].priority;
```

```
                    //URL エンコード
                    url = encodeURI(url);
                } else {
                    url = json.items[y].url;
                }
                //タグ表示
                tagElem.innerHTML = getTagHTMLByDataURI(url, timeflag[y]);
                //タグの位置調節
                tagElem.style.top = node.absTop + 'px';
                tagElem.style.left = node.absLeft + 'px';
                tagElem.style.opacity = 1; //new
                tagElem.refElem = node;
            }
        } else {
            oflag = false;
        }
    }
  }
  //"表示なし"の制御
  if (oflag == false) {
    on_off = 0;
    oflag = true;
  } else {
    on_off++;
    oflag = true;
  }
}
```

(ウ) 気付きエンジンの通信インタフェースを実現する Ruby ソース（抜粋）

```
#!/usr/bin/env ruby
# Isshiki.Lab@KAIT Tomoki Watanabe
# WebSocket Interface
require 'em-websocket'
require 'rubygems'

#キーワードリスト
keywordlist = "bangumi.json"

#デーモン化
Process.daemon(nochdir=true) if ARGV[0] == "-D"
connections = Array.new
EventMachine::WebSocket.start(:host => "0.0.0.0", :port => 51234) do |ws|
  ws.onopen {
    puts Time.now.to_s + " conected."
  }
  ws.onmessage {|data|
  if(data == "loaded")#(ブラウザのページロード時)
    #キーワードリスト読み込み
    jsondata = open(keywordlist,"r:utf-8")
    keywords = jsondata.read
    jsondata.close
    #キーワードリストの送信
    ws.send keywords
    puts Time.now.to_s + " send keywordlist to browser."
  else#(ブラウザの 5 回クリック時)
    #キーワードリスト:操作ログの保存
    file = File.open(keywordlist, "w")
    file.print data
    file.close
    puts Time.now.to_s + " saved keywordlist."
    #キーワードリスト読み込み
    jsondata = open(keywordlist,"r:utf-8")
    keywords = jsondata.read
    jsondata.close
    #キーワードリストの送信
    ws.send keywords
```

```
    puts Time.now.to_s + " send keywordlist to browser."

    end
  }
  ws.onclose {
    puts Time.now.to_s + " closed."
  }
end
```

(エ) アンコンシャス情報出力を指示する気付きエンジン Ryby ソース（抜粋）

```
#coding:utf-8
# Isshiki.Lab@KAIT Tomoki Watanabe
#  アンコンシャス情報出力を指示する WebSocket サーバ
require 'json'
require 'em-websocket'

Process.daemon(nochdir==true) if ARGV[0] == "-D"

# WebSocket のコネクション
connections = Hash.new

EM::run do
  EM::PeriodicTimer.new(2) do
  Dir::glob(NOTICELAYER_DIR + "/*.tag").each { |name|
    puts "TAG file found - " + name
    next if FileTest.directory?(name)
    uuid = File.basename(name, ".tag")
    if connections.has_key?(uuid) then
    f = open(name)
    content = f.read
    f.close
    connections.fetch(uuid).send(content)
    puts "TAG sent - " + content
    end
    File.delete name
  }
  Dir::glob(NOTICELAYER_DIR + "/*.json").each { |name|
```

```
      next if /¥d+_((¥w|-)+)¥.json/ !~ name
      next if FileTest.directory?(name)
      uuid = $1
      puts "JSON file found - " + name
      begin
      if connections.has_key?(uuid) then
        f = open(name)
        json = f.read
        f.close
        connections.fetch(uuid).send(json)
      end
      rescue
      puts "Illegal JSON file - " + name
      end
      File.delete name
    }
    end
    EM::WebSocket.start(:host => "0.0.0.0", :port => WS_PORT) do |ws|
    ws.onopen { |handshake|
      ipaddr = ws.get_peername[2,6].unpack("nC4")[1,4].join(".")
      puts "WS: onopen " + ipaddr
    }
    ws.onmessage { |msg|
      puts "WS: onmessage " + msg
      connections.store(msg, ws)
    }
    ws.onclose {
      # 保持しているコネクション情報を削除する
      if connections.key(ws) != nil then
      puts "WS: onclose " + connections.key(ws)
      connections.delete(connections.key(ws))
      end
    }
    end
end
```

# 論文目録

１．国内論文（査読付き）

(1) 渡部智樹，高嶋洋一，杉村博，一色正男，“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”, 情報処理学会論文誌　コンシューマ・デバイス＆システム, Vol.5, No.1, pp.38-46, 2015

(2) 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, Vol.2, No.2, pp.73-80, 2012

ほか２件

２．国際会議論文（査読付き）

(1) Tomoki Watanabe, Rika Mochizuki, Toru Kobayashi and Masao Isshiki, “HACCS：Home Appliance Control Concierge System”, 37th Annual Computer Software and Applications Conference(COMPSAC), 2013 IEEE 37th Annual, Kyoto, Japan, pp.208-213, 2013

ほか８件

３．特許出願

(1) 出願 2014 年，渡部智樹，山田智広，“認証装置および認証装置の動作方法”, (特願 2014-086225)

(2) 出願 2011 年，成立 2014 年，渡部智樹，小林透，“コンテンツ表示方法、コンテンツ表示装置及びそのプログラム”(特願 2011-217966，登録 5568537)

(3) 出願 2010 年，成立 2014 年，渡部智樹，青木良輔，小林稔，“遠隔操作システム、方法及びプログラム”(特願 2010-270476，登録 5460564)

(4) 出願 2010 年，成立 2013 年，渡部智樹，小林稔，阿部匡伸，“リモートコントロールシステム、リモートコントロール方法及びプログラム”(特願 2010-035091，登録 5292337)

(5) 出願 2009 年，成立 2013 年，渡部智樹，阿部匡伸，小林稔，前田篤彦，“視聴データ取得方法、視聴ログ取得装置、及び視聴システム”(特願 2009-134062，登録 5193953)

ほか１６７件（内 筆頭 50 件，登録 60 件）

４．国内学会発表

(1) 渡部智樹，高嶋洋一，杉村博，一色正男，”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”，情報処理学会研究報告コンシューマ・デバイス＆システム（CDS），2014-CDS-10，No.10，pp.1-7, 2014

(2) 阿部聡明，池田雅人，渡部智樹，杉村博，一色正男，”家電の状態に応じたユーザーへの通知システム”，第 76 回情報処理学会全国大会, 5Y-1, 2014

(3) 渡部智樹，望月理香，小林透，杉村博，一色正男，”視聴映像の時間長を考慮した家電制御システム”，情報処理学会研究報告コンシューマ・デバイス＆システム（CDS），2013-CDS-7, No.4, pp.1-8, 2013

(4) 青木良輔，渡部智樹，小林透，小林稔，”アンビエントな家電操作を実現する家電制御システムの提案”，2012-CDS-3，No.23，pp.1-8, 2012

(5) 渡部智樹，青木良輔，小林稔，阿部匡伸，”リモコン操作時間間隔に着目した TV 番組選択に関する一考察”，情報科学技術フォーラム講演論文集，2010-08-20，Vol.9，No.4，pp.513-514, 2010

(6) 渡部智樹，青木良輔，井原雅行，小林稔，阿部匡伸，”「リモコン信号プロクシ」を用いた機器操作ログ収集システム”，電子情報通信学会技術研究報告 LOIS，ライフインテリジェンスとオフィス情報システム, Vol.110，No.141, pp.23-28, 2010

ほか５２件

５．記事掲載

(1) 杉村博，関家一雄，渡部智樹，一色正男，”Web サービスによる HEMS 機器相互接続テスト環境の開発”，電気学会論文誌Ｃ（電子・情報・システム部門誌）, Vol.133, No.4, pp.818-819, 2013

(2) 石井晋司，渡部智樹，井原雅行，小林透，”次世代のコンテンツ流通にかかわる W3C における標準化動向(特集 次世代 Web プラットフォーム)”，NTT 技術ジャーナル，Vol.25，No.1，pp.56-59, 2013

(3) Shinji Ishii, Tomoki Watanabe, Masayuki Ihara, and Toru Kobayashi, “Standardization Trends in W3C Relating to Next-generation Content Distribution Services”, NTT Technical Review, Vol.11,No.4, 2013

(4) Tomoki Watanabe, Youichi Takashima, Minoru Kobayashi, and Masanobu Abe, ”Lifelog Remote Control for Collecting Operation Logs Needed for Lifelog - based Services”, NTT Technical Review, Vol.9, No.1, 2011

(5) 渡部智樹，小林稔，阿部匡伸，”ユーザの操作を記録し活用するライフログリモコン”，NTT 技術ジャーナル，Vol.49，No.7，pp.16-19, 2010

６．受賞

(1) 望月理香，渡部智樹，小林透，2013 年度 電子情報通信学会ＬＯＩＳ研究賞，電子情報通信学会, 2014

(2) 望月理香，渡部智樹，小林透，2012 年度 電子情報通信学会ＬＯＩＳ研究賞，電子情報通信学会, 2013