

ホームネットワークにおけるデバイス間連携に基づく 新サービス構築手法の検討

秋田 浩也¹ 川上 智史¹ 佐藤 健哉¹

概要: 近年ネットワークに接続できる家電が発売されている。これらの家電はネットワークに接続することで、お互いの情報を共有し、互いに制御しあうといったことが可能となっている。現在、日本ではECHONET Lite や DLNA(Digital Living Network Alliance) といった規格が混在しており、単体での利用しかできないのが現状である。また、近年はIoT(Internet of Things) の普及により、組み込みデバイスの発売も行われ始めている。ECHONET Lite と呼ぶ規格や、組み込みのセンサーの情報を単一のインターフェイスで操作するためには、それぞれの差異を吸収するミドルウェアが必要となる。本稿では、規格の違うデバイス同士の連携を可能にするミドルウェアを提案する。デバイスの連携とは、デバイスが持つ機能の連携である。また、機能の抽象度にも注目した。異なる規格では、提供する機能の抽象度が異なるため、機能同士を連携させるためには、適切な処理が求められる。また、実環境での利用時のユースケースを上げ、実環境での優位性について考察した。最後に、関連する研究や技術との比較を行い、本提案システムの有効性を検証した。

A Consideration for the Method of Constructing New Services by Cooperation between Devices in Home Network

HIROYA AKITA¹ SATOSHI KAWAKAMI¹ KENYA SATO¹

1. はじめに

近年、ネットワークに接続できる情報家電の普及が進んでいる。情報家電は、自身の情報を共有するだけでなく、他のデバイスからの制御も可能となっている。例えば、外出先から家にあるエアコンの電源を操作するといったことが可能である。

現在 IoT(Internet of Things) の流れを受けて、様々な情報家電が発売されている。それに伴って、日本における情報家電の通信規格の標準化も進められている。代表的な通信規格として、DLNA(Digital Living Network Alliance)[1] や ECHONET Lite[2] といった規格があげられる。DLNA は主にデジタル家電に採用されている規格であり、映像や音声を共有するためのガイドラインである。ECHONET Lite は、主に白物家電に採用されている規格である。また、これらの家電だけではなく、組み込みデバイスといったものも徐々に普及している。例えば、ウェアラブルデバ

イスに搭載されている脈拍計等のセンサーを使って健康管理をするといったことが可能である。ウェアラブルデバイスと、家にある体重計等の情報を統合することで、より正確な健康管理が可能となる。

しかし、現状では、組み込みデバイスや複数の規格に対応したミドルウェアやインターフェイスが存在しない。ECHONET Lite や DLNA といった規格の違いを意識せずに制御できる iHAC システム [3] の研究は行われているが、これは制御のみとなっており、デバイス間の連携は考慮されていない。また、情報家電は情報を共有することが可能であるため、デバイス間の連携も注目されている。例えば、あるデバイスのセンサー情報を利用して別のデバイスが動作するといったことである。web サービスの連携を実現する先行技術として、IFTTT[4] があげられる。IFTTT は機能の抽象化レベルを最大まで下げることで、ユーザーが自身でルールを作成することができる。しかし、機能の抽象化レベルが低いと、多くのルールを作成しなければならないという欠点がある。

¹ 同志社大学大学院理工学研究科

本稿では上記の問題を解決するために、組み込みデバイスや複数の規格に対応した汎用インターフェイスを提案するとともに、汎用インターフェイス作成時に問題となる機能の処理についても検討を行う。提供された機能を組み合わせることで、複数の処理を提供可能である。以下、2章で関連研究、関連技術について、3章で提案システムについて述べ、4章で提案システムのユースケースについて記述し、5章で既存技術・研究との比較を行う。最後に、6章でまとめを記述する。

2. 既存研究・既存技術

2.1 iHAC(intuitive Home Appliance Control)

iHAC システムは通信規格の違いを意識することなく、ARを用いて、デバイスの制御が可能となるシステムである。ユーザーはカメラでデバイスを映すことで、ディスプレイ上にデバイスを操作する画面が表示される。よって、ユーザーは通信規格の違うデバイスごとにアプリケーションを切り替えるといった動作をする必要がない。

図1にiHACシステムの概要を示す。汎用インターフェイスがECHONET Lite デバイス、DLNA デバイスに対してそれぞれのプロトコルを用いて命令を送る。汎用インターフェイスがプロトコルの差異を隠ぺいすることで、ユーザーはプロトコルの違いを意識せずにデバイスの制御ができる。このシステムの課題として、デバイスの制御のみしか取り扱っていないということが挙げられる。情報家電は、情報も共有することができるので、プロトコルが異なるデバイス同士の連携も可能となる。連携とは、例えばあるデバイスの情報を基にして、別のデバイスが動作する等のことである。

2.2 IFTTT

デバイスやwebサービスの連携を目的としたサービスと

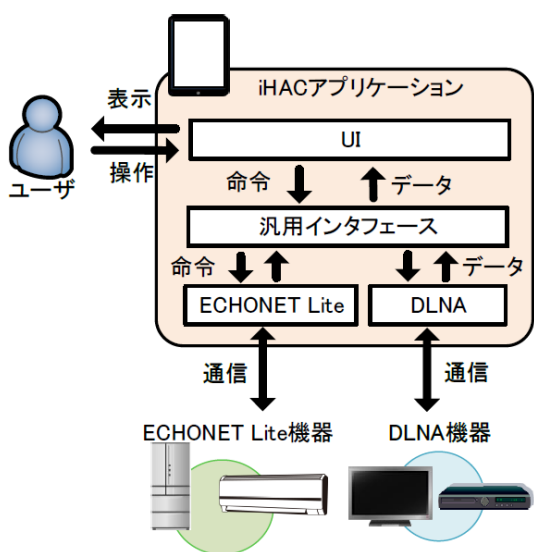


図1 iHACシステム構成

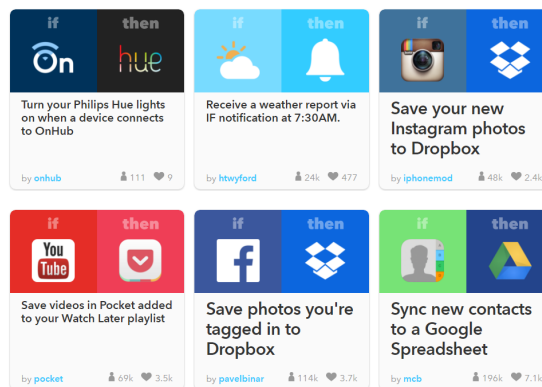


図2 IFTTTによるレシピ

して、IFTTTがある。このサービスは、if this then that というシンプルなコンセプトに基づいて「レシピ」と呼ばれるルールを作成することができる。レシピは「きっかけ」と「行動」から構成されている。図2にレシピの具体例を示す。例えば、「Instagramで投稿した写真が自動的にdropboxに保存される」というレシピを作成する。ここでは、「Instagramで投稿」が「きっかけ」であり、「dropboxに保存」が「行動」である。このサービスは機能の抽象度を下げることで、デバイスの連携を可能にしている。機能の抽象度とは、その機能がどの程度細かく分類されているのかということである。例えば、エアコンの温度変更という機能に対して、エアコンの冷房時の温度を一度下げる。と言う機能は抽象度が低い。抽象度を下げると言うことは、より詳細にルールを作成できるというメリットと引き換えに、多くのルールを作成しなければならないという問題も起こる。

ホームネットワークにおける、デバイス連携を考えたときに、できるだけ処理を減らすことを目指す。これは、老若男女が使用できることを目的としているためである。本稿では、機能の抽象度を上げることによって、処理の削減を実現する。

3. 提案システム

3.1 概要

規格の違いを内包した、汎用インターフェイスを提案する。汎用インターフェイスとは、ユーザーが操作する部分であり、ユーザーインターフェイスに相当する。本研究の目的を以下にまとめる。

(1) 規格の違いを吸収

ECHONET LiteやDLNAといった通信規格の違いをミドルウェアで吸収することで、単一のインターフェイスで複数の規格のデバイスの制御ができる

(2) 組み込みデバイスに対応

ウェアラブルデバイスの普及に伴って、多種多様なデ

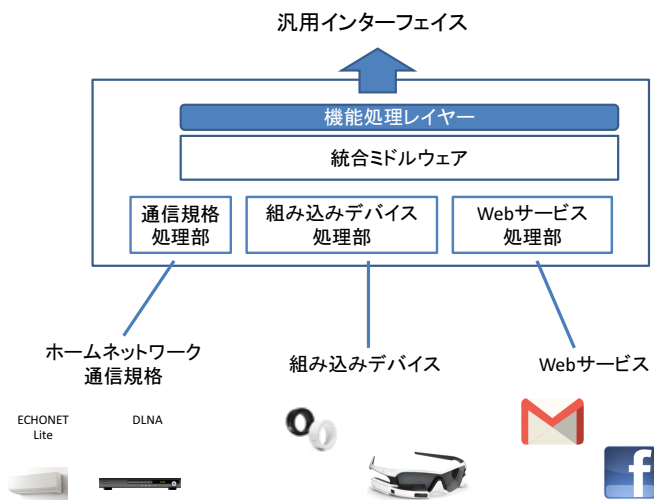


図 3 システム概要

デバイスが発売されている。これらのデバイスには複数のセンサーが搭載されており、これらの情報をホームネットワークに組み込むことができる

(3) Web サービスとの連携

近年は多くの web サービスが存在する。Gmail や facebook といった web サービスをホームネットワークに組み込むことができる

(4) 簡単な操作

IFTTT とは違い、機能の抽象度を上げることで、より少ない操作で新しい機能の作成ができる

上記の 4 点を満たした、汎用インターフェイスの開発を目指す。家の中だけではなく、web サービスとも連携することで、複数の機能を提供することができる。

3.2 システム構成

概要を図 3 に示す。まず、通信規格処理部について説明する。この部分は、ECHONET Lite や DLNA といったデバイスの制御を担当する。制御とはデバイスの探索や、デバイスの操作である。

次に、組み込みデバイス処理部の説明をする。組み込みデバイスとは、ウェアラブルデバイス等のことである。これらのデバイスは、専用の API を用いることで制御する。制御とは、センサー情報の共有やデータの送信などである。

次に web サービス処理部について説明する。web サービスとは、gmail や facebook といったサービスである。例えば、特定の人からのメールに対して、照明の色を変えろといったことが可能となる。

次に統合ミドルウェアについて説明する。統合ミドルウェアの主な役割は、通信規格の差異を吸収することである。例えば、ECHONET Lite の場合は、デバイス探索を行うと、デバイスのオブジェクトコードとして、数字が返ってくる。その数字がどのデバイスを意味しているのか

は、他のプロトコルはわからない。このような問題に対して、ミドルウェアが仲介することによって、プロトコル同士の差異をなくし、ユーザーはプロトコルを意識することなく、システムの利用ができる。

最後に機能処理レイヤーについて説明する。まず機能とは何かを定義する。機能とは、デバイスに対する何らかの処理である。例えば、エアコンの場合は、電源をつける、切るといった機能があり、もしカメラを搭載しているデバイスの場合は、映像情報を取得するといった機能がある。機能の定義は、プロトコルによって異なる。例えば、ECHONET Lite の場合を考える。家庭用エアコンクラスは、除湿モード時相対湿度設定値取得、節電動作設定、ユーザーリモコン温度設定値取得、換気モード設定、... といったように多くの機能がある。これらの機能を使って詳細な家電の操作が可能であるが、一方でユーザーにとっては複雑になりすぎるといった欠点がある。よって、本稿では、機能の抽象度を上げることでこの問題を解決する。ECHONET Lite の機能をそのままユーザーに表示するのではなく、このレイヤーを通過することで抽象度を上げる。例えば、エアコンクラスには以下の機能が規定されている。

- 冷房モード時温度設定値
- 暖房モード時温度設定値
- 除湿モード時温度設定値

上記の 3 つはすべて温度を設定するという機能に集約される。機能の抽象度を上げることによって、ユーザーはどのモードかを意識して機能を選択するのではなく、単純に設定温度を変更することができる。このようにして、それぞれのプロトコルが提供する機能をユーザーレベルの抽象度まで上げる操作を機能処理レイヤーは行う。

3.3 動作イメージ

動作イメージとして、ユーザーインターフェイスについて、システムの動作について説明する。

3.3.1 ユーザーインターフェイス部

システムのユーザーインターフェイスについて図 4 に示す。汎用インターフェイスに対して、Drag&Drop することでルールを作成する。基本的なルール作成はすべて Drag&Drop 操作で行う。Drag&Drop を採用した理由として、パーソナルコンピュータ (パソコン) の学習の初期で習うため、多くの人が容易に理解できる操作だからである。具体例として、ドアの開閉に対して、電気 on とエアコンの電源 on を依存関係として持たせる。これによって、玄関のドアが開閉されると自動的に証明の電気がつき、エアコンの電源が入る、というルールを作成することができる。

3.3.2 システムの動作

システムの動作の例として、デバイス検索シーケンスを挙げる。その様子を図 5 に示す。

まずアプリケーションが起動する時に、デバイスの検索

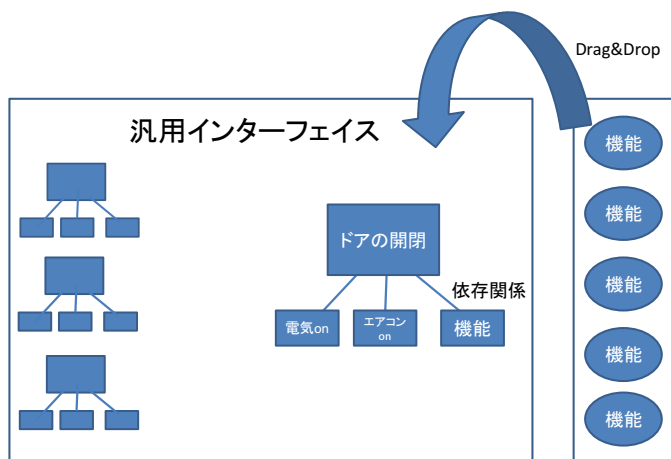


図 4 提案システムのユーザーインターフェイス

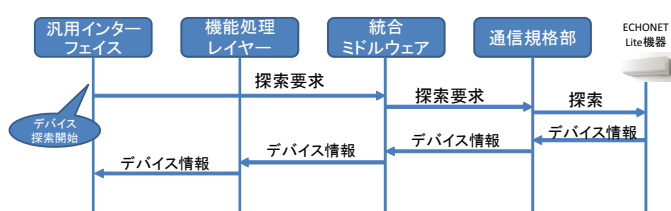


図 5 提案システムのデバイス検索シーケンス

を行う。汎用インターフェイスはまず、統合ミドルウェアにデバイス探索命令を送る。統合ミドルウェアはホームネットワーク全体に対して、デバイスの検索を行う。通信規格によってデバイス検索方法は異なるが、それぞれの通信規格にあった方法でデバイスの検索を行う。

今回はECHONET Lite デバイスを例に挙げて説明する。ECHONET Lite デバイスを検索すると、ホームネットワークに接続されたすべてのデバイスが、自身の情報をクラスコードとして送信する。そのクラスコードを元に、そのデバイスがどのようなデバイスであるのかを判断する。例えば、エアコンクラスのクラスコードは0x30と規定されているので、その数字が送信される。通信処理部はその数字を共通の指標に変換する。今回は、デバイスがエアコンであるので、エアコンと言う文字と関連付けた情報として、統合ミドルウェアに渡す。統合ミドルウェアは、機能処理レイヤーにデータを渡す。その時のデータは、JSONのような共通のデータ形式となっている。

最後に機能処理レイヤーで機能の抽象度を調整する。今回は、エアコンクラスが持つ機能を調整する。通信規格では、詳細な機能が提供されているが、そのまま表示すると、ユーザーの利用は困難になる。よって、ある程度機能の抽象度を落として、ユーザーに提供する。通信規格レベルでは、複数提供されている機能も、利用時を考慮してひとつに的おメルコとができる場合は、機能処理レイヤーで抽象度の処理を行う。

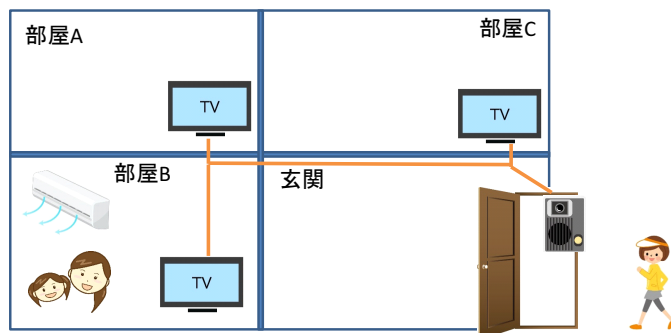


図 6 提案システムのユースケース

4. ユースケース

4.1 具体例 1

ユースケースとして、家電と映像との連携を挙げる。多くの白物家電には様々な機能が搭載されている。以前は分厚い紙の説明書が付属していたが、現在ではデータとして提供されているデバイスも存在する。しかし、ユーザーはすべての機能を把握することや、分厚い説明書をすべて読むことは困難である。

本提案システムを利用することで、ホームネットワークに接続されている家電を把握することができる。よって、ホームネットワーク内の家電の機能の紹介動画や使い方の動画を表示することが可能である。また、もしデバイスに不具合が出た場合、不具合の解決策をよりスムーズにインターネットで検索することができる。わざわざデバイスの前まで行って、型番を調べる必要はない。また、インターネットで説明書を探す必要もない。

4.2 具体例 2

もうひとつのユースケースとして、インターホンが押された場合のデバイス連携を挙げる。その様子を図6に示す。インターホンが押された場合、もし家に人がいない場合は、音声を用いて、人がいないことを来訪者に告げる。人がいるかどうかは、照明や家電が動作しているのかで判断できる。また、エアコン等の人感センサーを搭載している家電があった場合、人がどの部屋にいるのかを把握することができる。人がいる部屋に表示可能なデバイス、例えば、テレビ等がある場合は、そのテレビにインターホンの映像を表示するといったことも可能である。現在のデバイス連携はルールをユーザーが作成する形が主流である。しかし、ミドルウェアが自動的に判断し制御することで、ユーザーの負担は軽減する。

5. 既存研究・技術の比較

既存研究、既存技術との比較を表1に示す。本提案システムでは、デバイス間の連携が可能である。連携とは、あ

るデバイスの情報を利用して、別のデバイスが動作するということである。IFTTT は、web サービス同士の連携は可能であるが、実環境のデバイスについては十分考慮されていない。また、iHAC に関しては、デバイスの操作に関してのみ考慮しており、別のデバイスの情報を利用しての連携ができない。

web サービスの利用に関しては、iHAC は想定していない。最後に、抽象度の操作について述べる。本提案システムでは、抽象度の概念を導入している。基本的には抽象度が高い状態で、ユーザーに提供するが、詳細な設定も可能である。

表 1 既存研究・技術との比較

	提案システム	iHAC	IFTTT
デバイス同士の連携	○	△	△
web サービスの利用	○	×	○
デバイスの抽象度の操作	○	○	×

6. まとめ

本稿では、近年の情報家電の普及、IoT 技術の普及や web サービスの充実を背景に、今後求められるホームネットワークサービスの一形態を提案した。現在の日本は通信プロトコルが乱立しているという現状がある。他の通信プロトコル側では、別の通信プロトコルのデバイスを見つけることができないため、連携は不可能である。本稿では、ミドルウェアを導入することで、通信プロトコルの差異を吸収し、さらに web サービスや組み込みデバイスとの連携が可能となる。

また、機能の抽象度にも注目した。現在のデバイスには多くの機能が存在するがゆえに、ユーザーはすべて把握していないという現状がある。機能の抽象度を処理することで、よりユーザーに適したサービスを提供できると考える。プロトコルによって、提供する機能の抽象度が異なるため、機能処理レイヤーで適切に処理することが求められる。

今後は、提案したシステムが実際の空間で適切に動作するのかの検証を行っていく。

謝辞

本研究の一部は JSPS 科研費 (JP26540038) の助成を受けたものである。

参考文献

- [1] DLNA. 入手先 (<http://www.dlna.org/>) (参照 2016-5-01).
- [2] ECHONET CONSORTIUM. 入手先 (<https://echonet.jp/>) (参照 2016-5-01).
- [3] 梅山莉奈, 増田剛志, 鈴木秀和: 規格の違いを意識しない直感的家電制御システムの提案, 研究報告コンシューマ・デバイス&システム (CDS), Vol. 2015-CDS-14, No. 9, p p. 1-8 (2015)
- [4] IFTTT. 入手先 (<https://ifttt.com/>) (参照 2016-5-01).