

# 通信プロトコルの違いを意識しない直感的家電制御システムの実装

梅山 莉奈<sup>†1</sup> 増田 剛志<sup>†2</sup> 鈴木 秀和<sup>†1</sup>  
<sup>†1</sup> 名城大学大学院理工学研究科 <sup>†2</sup> 名城大学理工学部

## 1 はじめに

近年、ネットワークを通じて操作可能な情報家電が普及しつつある。しかし、情報家電には複数の通信規格が存在し、宅内には DLNA (Digital Living Network Alliance) [1] や ECHONET Lite[2] などが混在している。そのため、ユーザは機器の規格に合わせてアプリケーションや操作方法を変更する必要がある。

著者らは、規格の違いを意識することなく直感的に機器を制御することができる iHAC (intuitive Home Appliance Control) システムを提案している [3]。文献 [3] では、iPad で動作する iHAC アプリケーションにおいて DLNA, ECHONET Lite 機器の探索 API の実装を行っており、両規格の機器の探索結果を表示できることを確認している。

本稿では、DLNA, ECHONET Lite 制御用の API を実装し、動作検証を行った結果について述べる。

## 2 iHAC システム

### 2.1 概要

iHAC システムは、規格の違いを意識することなく、直感的に機器を制御することを目的としたシステムである。本稿における“直感的”という用語が意味する内容を4つに分ける。

1つ目は、規格の違いを意識しない操作であり、アプリケーション内に規格の違いを吸収するインタフェースを組み込むことにより実現する。2つ目は、位置情報の付加であり、型番や IP アドレスなどの機器から取得できる情報だけでなく、部屋名などの位置情報を関連付けて表示を行う。3つ目は、カスタマイズ可能な UI による表示であり、ユーザインタフェース (UI) の表示に HTML5 と CSS (Cascading Style Sheets) を用いることにより、自由なカスタマイズを可能とする。4つ目は、AR (Augmented Realty) を用いた表示であり、機器の情報や操作画面をカメラ映像上に重畳表示をする。

### 2.2 システムの構成

図1に iHAC システムの構成を示す。iHAC システムは、UI 部、汎用インタフェース部、各規格の通信処理部

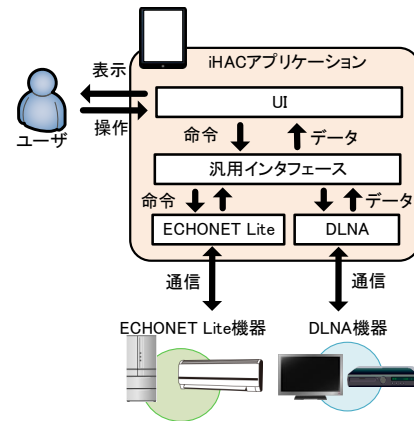


図1 提案システムの構成

で構成され、アプリケーションとして操作端末にインストールする。

UI 部はユーザに対して機器の情報や操作メニューの表示を行う。ユーザの操作に従って汎用インタフェース部の API をコールする。表示方法として HTML5 と CSS で表示を行う HTML モードと、AR で表示を行う AR モードがある。なお、位置情報や AR オブジェクトの表示位置は、あらかじめユーザが登録を行う。

汎用インタフェース部は、通信規格の違いを吸収する部分である。機器の探索や操作に関わる API を定義し、UI 部がその API をコールすることで、各規格の通信処理部で定義されている API をコールする。通信処理部からの応答に位置情報の付加や、フォーマットの統一を行って UI 部に情報を渡す。また、規格ごとにデータベースを持ち、機器名、機器を一意的に識別する識別子、位置情報、TV やエアコンなどのデバイスタイプを保持する。

各規格の通信処理部は汎用インタフェース部からのコールに従って、実際に機器の探索や命令などを行う。図1では ECHONET Lite 通信プロトコルおよび DLNA 通信モジュールが通信処理部に該当し、当該規格の家電機器と通信を行う。

## 3 実装

すでに実装済みである iPad で動作する iHAC アプリケーションに、DLNA, ECHONET Lite 制御用の API の実装を行った。なお、UI 部には HTML モードのみを実装した。

汎用インタフェース部には、すでに実装済みの探索 API、登録 API に加え、サービスリスト取得 API、サー

Implementation of Intuitive Home Appliance Control System without Considering Differences in Protocols

Rina Umeyama<sup>†1</sup>, Takeshi Masuda<sup>†2</sup> and Hidekazu Suzuki<sup>†1</sup>

<sup>†1</sup> Graduate School of Science and Technology, Meijo University

<sup>†2</sup> Faculty of Science and Technology, Meijo University

ビス実行 API を実装した。探索 API は、ネットワークに接続されている機器の探索を行い、発見した機器に位置情報を付加する。登録 API は、指定された機器をデータベースへ登録する。サービスリスト取得 API は、指定された機器で実行可能な電源 ON/OFF やコンテンツの再生などのサービスの種類を取得する。サービス実行 API は、指定したサービスを実行する。各規格の通信処理部には、各規格のライブラリを用いてすでに実装済みの探索 API に加え、サービスリスト取得 API、サービス実行 API の実装を行った。DLNA ライブラリには Platinum<sup>1</sup>、ECHONET Lite ライブラリには KAIT-4S-EZ<sup>2</sup> を利用した。

汎用インタフェース部の各 API は、UI 部からコールされ、それに従って各規格の通信処理部の API をコールする。サービスリスト取得 API の場合は、UI 部からコールされると、指定された規格の通信処理部のサービスリスト取得 API をコールする。各規格の通信処理部は、それに従ってネットワーク上にサービスリスト要求メッセージを送信する。応答結果として取得したサービスリストのデータは、JSON 形式で汎用インタフェース部から UI 部へ渡す。

データベースは、すでに実装済みの登録機器データベースと位置情報データベースを用いた。登録機器データベースには、機器名、識別子、位置 ID、デバイスタイプの登録を行う。位置情報データベースには、位置 ID と部屋名の登録を行う。

#### 4 動作検証

実装したアプリケーションの動作検証を行った。ローカルネットワークに操作端末である iPad と図 2 に示す操作対象である 5 台の機器を接続した。ECHONET Lite 機器には、ECHONET Lite 機器のエミュレータである MoekadenRoom<sup>6</sup> を利用して検証を行った。また、機器の位置情報はあらかじめ図 2 に示す通りに登録を行った。

図 3 に機器の探索結果を示す。データベースへ登録済みの機器 5 台が UI に表示されており、DLNA 機器と ECHONET Lite 機器の双方が表示されている。図 3 の探索結果から操作する機器を選択するとサービスリストが表示され、サービスを選択するとサービスが実行されることを両規格の機器で確認した。図 3 から“家庭用エアコン”を選択をし、サービスリストを取得した画面を図 4 に示す。“家庭用エアコン”で操作可能な動作状態や運転モード設定などのリストとその値を取得でき、またそれらの値を変更できることを確認した。DLNA 機器の場合は、サーバやレコーダ内に保存されているコン

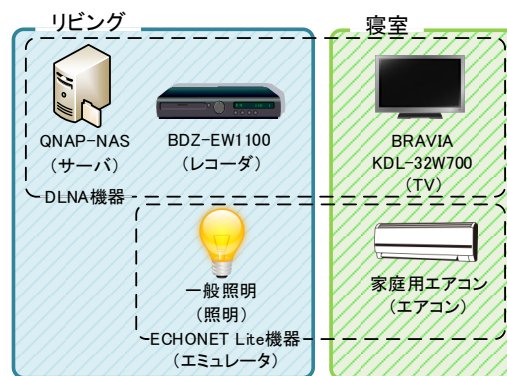


図 2 動作検証の環境

🏠 リビング	
📁	QNAP-NAS
📺	BDZ-EW1100
⚙️	家庭用エアコン
🛏️ 寝室	
📺	BRAVIA KDL-32W700B
💡	一般照明

図 3 機器リストの動作検証

動作状態	ON
運転モード設定	暖房
温度設定値	20°C

図 4 家庭用エアコンの制御検証例

テンツを閲覧でき、それを TV で再生できることを確認した。

#### 5 まとめ

本稿では、iHAC アプリケーションに DLNA、ECHONET Lite 制御用の API の実装を行った。動作検証により、両規格の機器を規格の違いを意識することなく制御できることを確認した。今後は、AR を用いた表示の実装を行う予定である。

#### 謝辞

本研究は JSPS 科研費 15K15987 の助成を受けたものである。

#### 参考文献

- [1] DLNA. <http://jp.dlna.org/>
- [2] ECHONET CONSORTIUM. <http://www.echonnet.gr.jp/>
- [3] 梅山莉奈ほか: 研究報告コンシューマ・デバイス&システム (CDS), 2015-CDS-14, No.14, pp.1-8 (2015).

<sup>1</sup> <http://www.platinosoft.com/platinum>

<sup>2</sup> <https://smarthouse-center.org/sdk/download/form/18>