

多様な機器を赤外線で制御可能な Web サービスの構築

春原雅志^{†1} 河口信夫^{†2}

近年、いつでもどこでも様々な情報やサービスを享受できるユビキタス環境が実現されつつある。一方、日常環境には赤外線リモコンやシリアル通信を用いて外部からの機器制御が可能であるが、機器連携機能を持たない「非情報家電」が多数存在している。本研究では非情報家電の制御情報を効率的に登録し、プログラムのコーディングなどを行わずにネットワークへの接続性や他のシステムと連携可能な環境を Web サービスを通じて構築する手法を提案する。また、機器の情報を登録や制御のための赤外線送受信機を試作し、様々な機器の情報が容易に登録でき、連携制御ができることを確認した。本手法により複雑な手順を踏まずに非情報家電の制御を Web サービス化し、ユビキタス環境へ組込むことが可能となる。

Web Service for Infrared Remote Controllable Appliances

MASASHI SUNOHARA^{†1} and NOBUO KAWAGUCHI^{†2}

Recent advancement of ubiquitous computing environment enables deployment of anywhere information service. However, we still have a lot of “non-information appliances” which has no integration functions. These devices can be controlled through infra-red remote controller. In this study, we propose a system which can easily configure and control these appliances through a Web-service. We implement a infra-red interface to control infra-red appliances and confirm remote control through the Web-service.

1. はじめに

近年、ネットワーク環境の整備や情報端末の小型化により、いつでもどこでも誰でも情報やサービスを受け取ることができるユビキタス環境¹⁾が実現されつつある。ユビキタス環境では多数のセンサや機器を相互に連携させることにより、我々の快適な生活をさりげなく支援することが期待される。機器同士の連携を実現するために UPnP²⁶⁾ や DLNA²⁷⁾, Bonjour²⁸⁾ といった仕組みが利用され始めている。これらの仕組みを用いると、機器をネットワークに接続するだけで IP アドレスの設定や機器が提供するサービスの公開を自動的に行うことができ、ネットワークに接続して利用することが想定される情報家電を中心に広まりつつある。しかしながら我々が生活している環境にはネットワークへの接続性や機器連携機能を持たない「非情報家電」が多数存在している。非情報家電の多くは赤外線リモコンやシリアル通信などを用いて外部からの機器制御が可能であるが、他の機器と連携動作させるためには、機器自体にそのような機能を組込むか、赤外線やシリ

アル通信を制御を行うシステムを構築する必要がある。しかし、機器自体に組込むのは容易ではなく、またそのようなシステムを構築するためには、機器ごとに赤外線通信やシリアル通信を行うためのプログラムを実装しなければならない。また様々な機能を持つ機器は制御を行うたびに内部状態が変化し、単純な制御指示では意図しない動作をさせてしまう可能性がある。例えば、機器の電源を ON にするために ON/OFF 制御を行った場合、既に機器の電源が ON の状態の場合は電源が OFF になってしまう場合がある。そのため非情報家電を制御するシステムは、一方的に制御に対して指示を行うのではなく、機器の現在の状態や状態遷移を考慮して指示を行う必要がある。

本稿では非情報家電の内部状態を含んだ制御情報を効率的に制御システムに登録し、プログラムのコーディングなどを行わずにネットワークへの接続性や他のシステムと連携可能な環境を構築する手法を提案する。提案手法ではまず機器情報の登録を簡略化するために、様々な機器が持つ内部の状態をモデル化し状態遷移の仕方についてパターン分けを行う。そして、各パターンごとに状態遷移を含んだ機能テンプレートを作成する。この機能テンプレートに対して、機器を制御するために必要な赤外線信号やシリアルコマンドな

^{†1} 名古屋大学大学院情報科学研究科
Graduate School of Information Science, Nagoya University

^{†2} 名古屋大学大学院工学研究科
Graduate School of Engineering, Nagoya University

どの対応付けを行う。このような手順を繰り返すことにより、様々な機器の情報を効率的に制御システムに登録することが可能となる。そして制御システムに登録した機器の制御を Web サービス化することにより、ネットワーク経由での機器制御や他のシステムとの連携が容易になる。さらに、Web サービスを用いて登録した機器情報を共有することにより、登録作業の手間を軽減することが可能となる。

上記の手法に基づき、機能テンプレートを用いた機器情報の登録と Web サービスの提供を行うシステムを Web サーバ上に実装した。機器情報の登録は、Web ブラウザ上から利用可能な Web アプリケーションとして実装し、機器情報を管理する Web サーバ上に XML ファイルとして蓄積した。また、登録された機器を Web アプリケーションや他のシステムから利用するための Web サービス API を実装した。実装したシステムを用いて赤外線リモコンによる制御が可能な機器の情報を登録するために赤外線送受信機を作成し、様々な機器の情報を登録した。そして、実際に API を用いて機器情報の取得や制御を行う Web アプリケーションを作成し、携帯端末からの非情報家電の制御や連携制御が可能になることを確認した。本手法を用いることにより複雑な手順を踏まずに非情報家電の制御を Web サービス化し、ユビキタス環境へ組込むことが可能となる。

2. ユビキタス環境における機器制御

ユビキタス環境には様々な機器が動作しているが、これらは一般に電源や音量やチャンネルといった「状態」を持つ。本稿では「機器の制御」を「機器を現在の状態から別の状態に遷移させるために指示を行うこと」ととらえる。以下では、現時点で利用可能な機器制御を行う手段を列挙する。

- 供給電力を用いた制御: 機器に供給する電力を ON または OFF にすることにより機器の制御を行う。また、供給電力を調整することによりモータの回転数や照明の明るさを制御することが可能となる。
- 機器のボタンなどを用いた制御: 機器本体に備え付けられたボタンやスイッチを用いて制御を行う。遠隔制御をするためには物理的な操作を行う機構が必要となる。
- 赤外線リモコンを用いた制御: 機器の赤外線リモコンを用いた制御を行う。複数の機器のリモコンの信号を記録し、一つのリモコンに統合することができる「学習リモコン」と呼ばれる製品も見られ、外部から制御も容易である。

- シリアル通信を用いた制御: 機器に対してシリアル通信経由で制御信号を送ることにより制御を行う。専用のコントローラなどと組み合わせて利用される場合が多く見られる。なお、シリアル通信を TCP/IP 通信に変換する XPort²⁹⁾ や、Bluetooth に変換する Parani10³⁰⁾ 等を用いることにより、遠隔からの制御を実現することができる。
- Web ブラウザを用いた制御: ネットワークに接続された機器が Web サーバの機能を持ち、ユーザは Web ブラウザを用いて機器の設定や制御を行う。WebCodget System¹⁵⁾ を用いることにより、シリアル通信機器の制御を Web ブラウザから行うことが可能になる。
- 無線通信を用いた制御: Bluetooth などの無線通信を用いて機器の制御を行う。通信が可能な範囲において機器制御が可能な点は、赤外線リモコンによる制御と類似しているが、赤外線のように指向性が無いため、機器が設置されている場所を考慮しなくても良いという利点がある。

3. 赤外線を用いた機器制御

本稿では、前節で述べた制御法の中から、特に赤外線リモコンを用いた制御について取り上げる。これは (1) 赤外線を用いて制御可能な機器が多種多様あり、(2) 機器から離れた場所からの操作が可能、(3) 機器を制御するための信号を赤外線リモコンで登録可能、であるからである。

赤外線リモコンを用いた機器制御の例を表 1 に示す。この表より、赤外線リモコンによる機器制御についていくつかの特徴があることが分かる。

電源の制御: 1 つめの特徴として、各機器に共通な制御として電源制御 (ON/OFF) がある。1 つのボタンによって機器の電源 ON/OFF を行うものも、電源を ON/OFF ボタンを別々に付けられているものもある。

機能のグループ化: 2 つめの特徴として、機能のグループ化がある。制御する機器の機能が増えると赤外線リモコンのボタンも多くなるが、各機能を状態としてとらえると図 1 のように状態ごとにグループ化が行える。例えば、ビデオレコーダには「再生」「停止」「早送り」「巻戻し」などのボタンがあるが、これらはビデオテープの再生状態を制御するためのグループとみなすことができる。このグループに属しているボタンを操作することにより、「ビデオテープを再生し、映像の出力を行う」、「映像の出力を停止し待機状態にする」、「ビデオテープを早送りする」、「ビデオテープを巻戻す」などの状態を遷移させ、プレイヤーの制御を行える。

表 1 赤外線リモコンによる機器制御

機器名	制御内容
照明	電源制御 (ON/OFF), 明るさ (明/暗)
扇風機	電源制御 (ON/OFF), 風量調整 (強/弱), 機能切替 (首振り/リズム), タイマ
エアコン	電源制御 (ON/OFF), モード切替 (自動/冷房/暖房/ドライ), 温度 (上/下), 風量, 機能切替 (快眠冷房/健康冷房) タイマ,
テレビ	電源制御 (ON/OFF), 音量 (上/下/消音), チャンネル (1ch ~ 12ch/BS5 ~ BS11/次/戻), 入力切替 (テレビ/ビデオ入力 1/ビデオ入力 2), 機能切替 (画面表示, 明るさ, 省電力モード, サラウンド, 二重音声), 設定 (メニュー/カーソル移動/決定)
ビデオレコーダ	電源制御 (ON/OFF), チャンネル (1ch ~ 12ch/次/戻), 再生機能 (再生/停止/一時停止/早戻/早送/録画), 再生速度切替 (標準/3倍/5倍), 入力切替 (外部入力 1, 外部入力 2), 設定 (メニュー/カーソル移動/決定/取消)
プロジェクタ	電源制御 (ON/OFF), 音量 (上/下/消音), 入力切替 (ビデオ入力.Sビデオ入力, RGB入力 1, RGB入力 2), 機能切替 (アスペクト比, 省電力モード), 明度調整 (明/暗), フォーカス調整 (遠/近), 設定 (メニュー/カーソル移動/決定/キャンセル)
ステレオアンプ	電源制御 (ON/OFF), 音量 (上/下/消音), モード切替 (ムービー/ミュージック/スポーツ/ゲーム), 入力切替 (入力 1~4), 機能切替 (ナイトリスニング/多重音声/テスト)

る。同様に、ビデオレコーダやプロジェクタには外部からの入力切替を行うボタンが見られるが、これらに属するグループのボタン操作により、複数の外部入力端子の中から「入力端子 X からの入力を行う」という状態に遷移させる制御を行える。このようなグループ化を行うと、赤外線リモコンでの操作は複数の機能グループの集合であると考えられる。

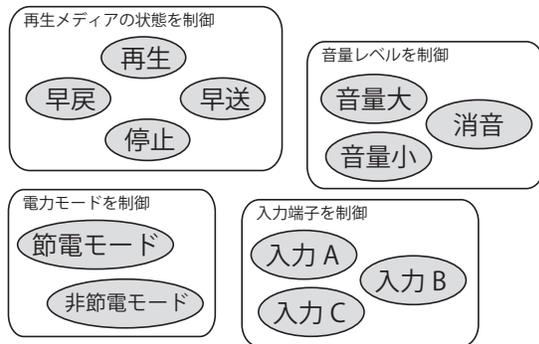


図 1 機能のグループ化

状態遷移の指示法: 3つめの特徴は、赤外線リモコンのボタンを押したときの状態遷移の仕方には2つのパターンに分類することができる点である。1つは、ある状態 X から目的とする状態へ直接遷移させること

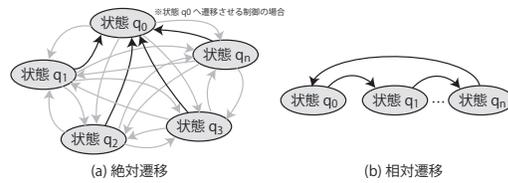


図 2 状態遷移の指示法

ができるボタンであり、もう1つはある状態 X から別の状態 Y へ相対的に遷移させることができるボタンである。前者は機器の状態とリモコンのボタンが直接対応付けされているため、ユーザは目的とする状態に対応するボタンを押すだけで、機器をその状態に遷移させることができる。例えば、表 1 にあげた機器ではビデオレコーダの「再生」や「停止」ボタンが直接遷移可能なパターンに該当する。一方、後者はボタンを押すことにより状態を遷移させることができるが、直接目的の状態にすることはできないため、目的の状態に達するまで複数の操作が必要となる。例えば、表 1 にあげた機器ではエアコンの「温度」ボタンやテレビの「音声」ボタンがこのパターンに該当する。本研究では前者のような遷移をさせることを「絶対遷移」、後者のような遷移をさせることを「相対遷移」と呼ぶことにする。

表 1 から分かるように、テレビの赤外線リモコンの「チャンネル」ボタンのように絶対遷移と相対遷移の両方を備えているものも見られる。

状態に依存するボタン: リモコンのボタンの多くは絶対遷移か相対遷移させるためであるが、中には機器の状態に応じて機能が変化するボタンがある。例えば、地上デジタル放送を受信可能なテレビでは [青, 赤, 緑, 黄] のボタンがあり、これらはデータ放送の受信時に画面に表示されている選択肢を選ぶ機能がある。また、上下左右に移動可能なカーソルキーを備えたりリモコンは、画面上に表示されるメニューを操作するために用いられる。このようなボタンによる機器状態の遷移は、メーカーや機器によって遷移方法が多様であるため、モデルとして標準化するのは容易ではない。

3.1 システムによる機器の制御

本節では前節に述べた機器制御の特徴を用いて、システムによる機器制御を行う際に必要となる要素について述べる。ユーザが赤外線リモコンなどを用いて機器を操作する際には、現在の機器の状態を視覚や聴覚によって確認した上で、目的の動作をさせるために適切な操作を行う。そのため、システムがユーザに変わって機器に対して制御を行う場合は、次のことを認

識している必要がある。

- 制御対象とする機器と制御の内容
- 制御対象とする機器の現在の状態
- 機器を目的の状態にするための制御法

まず、システムが機器に対して制御を行う際には、どの機器に対してどのような制御を行うかということを決める必要がある。これらの決定は、ユーザからシステムに対して指示する場合だけでなく、システムがセンサ情報などを元に機器を適切な状態に保つために指示する場合も考えられる。システムは、機器が複数存在する環境においてどのような機器がどこに存在しているかを把握している必要がある。

続いて、システムは制御対象とする機器の現在の状態を把握している必要がある。システムは現在の状態から目的とする状態にするためには機器に対してどのような指示をする必要があるのかを判断し、適切な制御を行う。そのためには、制御を行う機器にはどのような状態があり、どのような遷移が可能であるかを把握している必要がある。機器の状態遷移は図2に示すように、ある状態から別の状態への状態遷移が多数存在する。システムを用いて機器制御を行う場合は、これらの状態と状態を遷移させるための情報をシステムに登録することが必要となる。しかし、機器の機能の増加とともに増えていく機器の状態の定義と遷移について一つ一つ登録をさせるのは、ユーザにとって手間がかかる。

以下では機器制御の特徴を用いることで状態の定義と遷移の登録を効率的に行う手法について述べる。

3.1.1 状態の定義

状態の定義では、制御を行う機器がどのような状態を持ち、それぞれの状態においてどのような動作を行うかを定義するものである。それぞれの機器が持つ機能は、複数の機器で共通に見られるものと機器が独自に備えている機能がある。複数の機器に共通に見られるものについては、予め想定される機能の状態をまとめたテンプレートを用意し、必要に応じてテンプレートに修正を加えて適用することによりユーザが一つの状態を定義する手間を軽減することが可能となる。

3.1.2 絶対遷移の登録

絶対遷移の登録では、目的とする状態に絶対遷移させることができる操作をシステムに登録するものである。図2のように、機能ごとにグループ分けした状態において、各状態を遷移させる制御は多数存在する。しかし絶対遷移では、状態間に状態遷移を行うことができる関連があれば、いかなる状態においても目的とする状態に絶対遷移することが可能である。つまり、

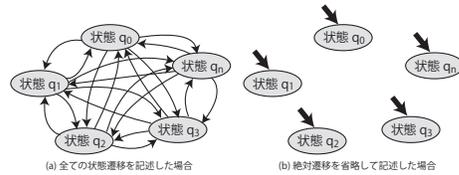


図3 絶対遷移の登録

目的とする状態にするための操作を登録することにより、各状態からの遷移を定義することが可能となる。ここでは図3に示すように、絶対遷移による状態の遷移を太矢印で表すことにする。

3.1.3 相対遷移の登録

相対遷移の登録では、2で述べたように、状態間を遷移させるために必要な操作を登録するものである。表1に示した制御内容では、音量調整やチャンネルの変更、温度調整などがこれに該当する。相対遷移の例として、赤外線リモコンなどに見られる「+」や「-」のボタン操作によって「次の状態に遷移」または「前の状態に遷移」させることがあげられる。この例では「次の状態」は音量調整では出力する音量を大きくしたり、温度調整では温度を上げることに該当し、「前の状態」は音量を小さくしたり温度を下げることに対応する。この対応は機器や機能によってどのように対応付けするか異なるが、状態遷移の仕方を3つのパターンに分類することができる。ここではそれぞれのパターンについて説明するために以下のように定める。

- 状態数は2以上とする
- 各状態に $q_0, q_1, q_2, \dots, q_n$ とラベルを付ける
- 「次の状態に遷移」とはある状態 q_x から状態 q_{x+1} へ遷移することである
- 「前の状態に遷移」とはある状態 q_x から状態 q_{x-1} へ遷移することである

Pattern 1: Sequential

この遷移は図4に示すように、一定方向へ順に遷移を行い、目的の状態まで遷移するパターンである。このパターンでは、ある状態 q_x から状態 q_{x+1} へ遷移を行うが、状態 q_n の時は次の状態に遷移すると状態 q_0 へ遷移する。なお状態数が2の時は、電源がONまたはOFFの状態などの切替スイッチとして利用することもできる。

Pattern 2: Bidirectional-Nonloop

この遷移は図4に示すように、双方向へ順に遷移を行い、目的の状態まで遷移するパターンである。このパターンでは、ある状態 q_x から状態 q_{x+1} または q_{x-1} へ遷移を行うが、状態 q_0 の時は前の状態に遷移を行っても状態 q_0 のままであり、状態 q_n の時は次の状態に

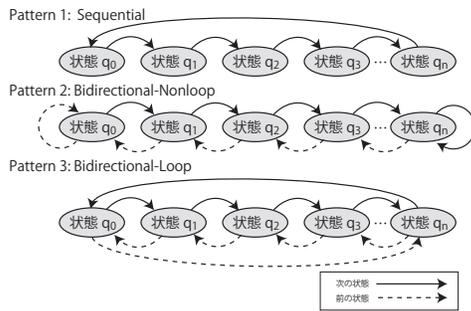


図 4 相対遷移の登録

遷移を行っても状態 q_n のままである。

Pattern 3: Bidirectional-Loop

この遷移は図 4 に示すように、双方向へ順に遷移を行い、目的の状態まで遷移するパターンである。このパターンでは、ある状態 q_x から状態 q_{x+1} または q_{x-1} へ遷移を行うが、状態 q_0 の時は前の状態に遷移すると状態 q_n へ遷移し、状態 q_n の時は次の状態に遷移すると状態 q_0 へ遷移する。なおこのパターンは状態数が 2 の時はパターン 1 とみなすこともできる。

相対遷移の登録では該当するパターンを元に、「次の状態」に遷移する操作と「前の状態」に遷移する操作を登録することにより効率的に登録することができる。

3.2 システムへの機器情報の登録

これまで述べてきた登録の手法を用いて、システムへ機器情報の登録を行う。まず、効率的な登録のために予め複数の機器に共通に見られる機能について状態のグループ化を行い、それぞれのグループにおいて絶対遷移と相対遷移の登録が想定されるかどうか分類を行う。ここでは機能ごとに分類したものを機能テンプレートと呼ぶことにする。表 2 にグループを分類した例を示す。表中の は絶対遷移または相対遷移が想定されることを示し、×は想定されないことを示す。そして は機器によって想定される場合と想定されない場合があることを示す。相対遷移については 4 節で述べたどのパターンに該当するか記している。ユーザはシステムに登録をさせる際に、表 2 に示された機能テンプレートとして利用することにより、機器状態や状態遷移についての定義をする手間を省くことができる。なお、機能テンプレートの状態と遷移は多くの機器で想定されるものになっているが、機器に合わせて適宜調整することにより様々な機器に適応可能になる。

実際に機能テンプレートを用いてシステムに登録する際の手順を以下に示す。

- (1) 登録する機能に最も近い機能テンプレートを選択する

表 2 機能テンプレートの例

機能	状態	絶対遷移	相対遷移
切替	OFF, ON	×	(Sequential)
温度	機器で調整可能な温度	×	(Bidirectional-Nonloop)
音量	機器で出力可能な音量	×	(Bidirectional-Nonloop)
入力切替	機器で入力可能な外部入力		(Sequential)
チャンネル	機器で調整可能なチャンネル		(Bidirectional-Loop)
モード	機器で設定可能なモード		(Sequential)
プレイヤー	再生, 停止, 早戻り, 早送り, 一時停止		×
コントローラ	メニュー, キャンセル, カーソル (上下左右)		×
その他	機器特有の機能		

- (2) 選択した機能テンプレートにおいて絶対遷移が想定されていれば登録する
- (3) 現在の状態を登録する
- (4) 選択した機能テンプレートにおいて相対遷移が想定されていれば登録する
- (5) 現在の状態を登録する
- (6) 必要に応じて機能テンプレートの内容を設定する
- (7) 制御する機器に必要な機能の機能が揃うまで 1.~6. を繰り返す

このような手順で登録することにより、機器の状態遷移を含めた情報を効率的にシステムに登録することができる。

4. Web サービスを用いた機器制御

4.1 Web アプリケーション

Web サーバ上のコンテンツの閲覧のために利用されてきた Web ブラウザ上で、様々なアプリケーションを Web サービスとして提供する Web アプリケーションが広がりつつある。これらの Web アプリケーションはソフトウェアの機能をネットワークを通じて利用できるようにしたものである。メールの送受信や画像処理など単一の機能に特化した Web アプリケーションだけでなく、Web ブラウザ内にデスクトップ環境を構築し、その上で様々なソフトウェアを実行する Web デスクトップと呼ばれる Web アプリケーションも開発されている。Web アプリケーションの利用には次のようなメリットがあると考えられる。

- 利用者はソフトウェアのインストール作業をせずにサービスを利用可能

- ネットワークへに接続することができれば、場所に依らずに利用可能
- Web ブラウザ内で動作するため様々な OS 上での動作が可能
- サービス提供者はサーバ上でコンテンツや情報の一括管理が可能
- ネットワークを介して情報の共有や公開が可能



図 5 Web ブラウザ上でのアプリケーション利用

4.1.1 Web サービスを支える技術

このような様々な Web サービスが広がるきっかけの一つとして、Web ブラウザ上での様々な操作インタフェースが実現可能になってきたことがあげられる。例えば、これまでの Web サービスでは、Web サーバへリクエストを送信し、サーバから受け取った情報を Web ページとして画面に表示していたため、ユーザが画面上のリンクやボタンのクリック操作を行うたびに画面遷移が発生していた。しかし、JavaScript と XML の要素を組合せた Ajax²³⁾ などの技術を用いて画面の一部を動的に書き換える技術などが提案され、Web ブラウザ上で、通常のアプリケーションと同じような操作性でアプリケーションやサービスを利用することが可能となった。最近では Ajax を用いた Web アプリケーションの開発を容易にするための様々な Javascript ライブラリなども提供されている。また、他にも Flash や Silverlight といったような Web ブラウザにプラグインを追加することにより音声や動画などのマルチメディアコンテンツの再生や、双方向性を持った Web アプリケーションを提供可能な技術も普及しつつある。また、Ruby on Rails 等の Web アプリケーションを構築フレームワークを用いることにより、効率的な Web アプリケーションの開発が可能となる。

4.1.2 Web サービスの利用環境

Web サービスやそれらを支える技術が発展する一方で、それらを利用するための環境の発展も見られる。インターネットに接続可能な環境は屋内だけでなく、屋外や公共施設でも利用可能な場所が増えている。また、携帯電話にもフルブラウザを搭載したものが増え、最近では無線 LAN と Web ブラウザを搭載した音楽プ

レイヤも発売された。今後さらにネットワーク基盤の発展や、ユーザが利用する端末の小型化、性能の向上により、あらゆる場所において Web サービスが利用可能になると期待される。

4.1.3 Web サービスを用いた情報の取得

Web サービスの発展と共に、Web サービスを用いた情報の配信や共有が行われるようになってきた。そのような情報を利用する際には、送り手と受け手の双方が理解可能な共通の情報表現のルールが必要となる。そこで汎用的なデータ記述言語として XML (Extensible Markup Language) が利用されている。XML はデータ構造をタグ付けしたテキストファイルによって記述されるため、様々なプラットフォーム間でデータの交換が可能である。XML 内のタグは自由に定義することができ、XML を利用することによりサービス間のメッセージ交換などを容易に実現することが可能となる。

4.2 機器制御と Web サービス

前節で述べたような特徴を持つ Web サービスを、Web アプリケーションのようなソフトウェアを利用するためだけでなく、様々な機器を制御可能な Web サービスとして構築する。機器制御を Web サービスとして提供する際には、ユーザによる Web サービスの利用だけでなく、他のシステムからの Web サービスの利用にも対応する必要がある。システムによる制御は第 3 章で述べたように、制御対象とする機器の情報を認識させる必要がある。本研究ではネットワーク上に機器情報を管理するサーバを設置し、そのサーバ上に機器情報の登録を行う。

4.2.1 Web アプリケーションを用いたシステムへの機器情報の登録

機器情報の登録はできるだけ簡単な手順により、手間をかけずに行えることが望ましい。そこで本研究ではシステムへ機器情報を登録するためのソフトウェアを Web アプリケーションとしてユーザに提供する。機器情報の登録を Web アプリケーション上で行う事により、ユーザは専用のソフトのインストールや設定作業を行わずに、普段利用している Web ブラウザを用いて機器情報の登録が可能となる。Web ブラウザ上から登録された機器情報をサーバ上で一括管理することにより、複数機器の制御を Web サービスとして提供することが可能となる。また、登録した機器情報をネットワークを用いて共有し、別のユーザが再利用することにより機器登録の手間を減らすことができる。

4.2.2 Web サービス API を用いた機器制御

システム上に登録した機器情報を利用するために

Web サービス API を利用する。Web サービス API は、機器情報を登録したサーバに対して 80 番ポートを用いた HTTP 通信によって情報を得るためのリクエストを行うことにより登録されている機器の一覧や機器が持つ状態の一覧を取得することが可能となる。本研究では機器制御を行うために次のような Web サービス API を提供する。

- システム上にどのような機器が登録されているかを取得する API コマンド
- それぞれの機器に対して可能な制御を取得する API コマンド
- システムに登録されている機器情報ファイルを取得するための API コマンド
- 機器の現状態を取得する API コマンド
- システム内に保持している現状態を外部から変更するための API コマンド
- 機器登録情報を変更するための API コマンド
- 機器を制御するための API コマンド

これらの API を用いて Web アプリケーションを作成することにより、様々な機器のネットワーク経由での利用が可能となる。また、他のシステム上からこれら HTTP 通信によってこれらの API を利用し、機器制御や連携制御を行うことが可能となる。さらに、Web サービスを用いて登録された機器情報の共有を行うことにより、システムへの機器情報の登録の手間を省くことが可能となる。

4.2.3 セキュリティ対策

Web サービス API を用いるとネットワーク経由で非情報家電の制御が可能になる反面、悪意のあるユーザにより機器制御が行われたりする可能性がある。Web サービスを提供するサーバに実装されている認証機能 (Apache の BASIC 認証) などを用いて API の利用に制限をかけることが可能となる。また、利用可能な機器や機能を詳細にコントロール可能な仕組みを実装することにより、よりセキュアな機器利用が可能になると考えられる。

5. システムの実装

実装したシステムの構成を 6 に示す。次の節よりそれぞれの役割や機能の詳細を述べる。

5.1 赤外線送受信機

赤外線送受信機は、シリアル通信を用いて赤外線の送受信を行う機能を持つ。赤外線送受信機の仕様を 3 に示す。このような仕様に基づく赤外線送受信装置を PIC マイコン³¹⁾ を用いて実装した¹⁷⁾¹⁸⁾¹⁹⁾。実装した赤外線送受信機の回路図を図 7 に、装置の写真を図 8 に

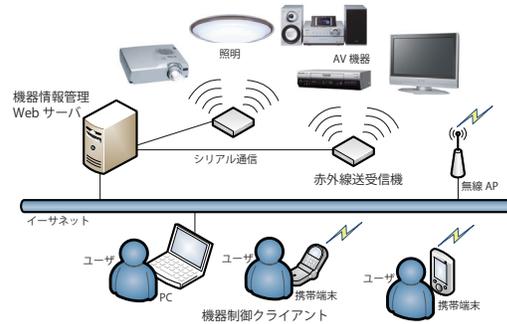


図 6 システム構成

表 3 赤外線送受信装置の仕様

電源	9V 乾電池または AC 電源
赤外線受信機能	赤外線リモコンから送信された赤外線を受信し、シリアルポートでデータを出力する
赤外線送信機能	シリアルポートよりデータを入力し、赤外線の送信を行う。

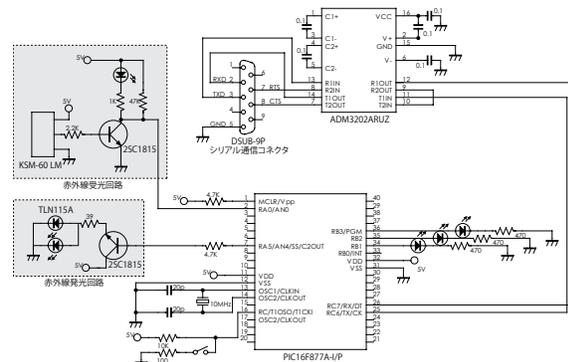


図 7 赤外線送受信装置回路図



図 8 赤外線送受信装置

示す。

5.2 赤外線受信機能

赤外線の受信部には赤外線リモコン受信モジュール KSM-60 LM³³⁾ を利用した。このモジュールは赤外線リモコンから送信された赤外線信号のパルス信号を受信し、内部でフィルター処理をし 37.9kHz の復調をし

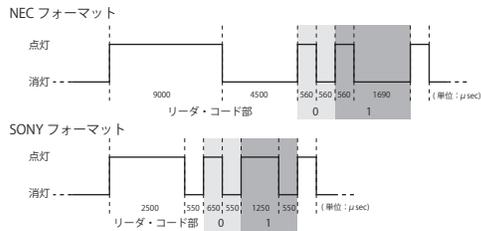


図 9 PPM 方式を用いた赤外線信号

た後、TTL レベルのデジタル信号として出力するものである。家電製品に付属のリモコンの多くは、信号の 0 と 1 を表すために PPM(Pulse Position Modulation) という変調方式を採用している。PPM 方式では赤外線の点灯時間と消灯時間の長さの組み合わせによって 0 と 1 の信号を表現する方式である。ただし、太陽光や蛍光灯などの光による誤動作を避けるため、点灯状態の時は 37.9kHz の周波数で点灯と消灯を繰り返すことによって表現している。さらに、PPM 方式による表現方法には、NEC フォーマットと呼ばれる赤外線の消灯時間の長さで 0 と 1 を表現する方式と、SONY フォーマットと呼ばれる赤外線の点灯時間の長さで 0 と 1 を表現する 2 つの方式があり、多くの機器を制御するためには、それぞれの方式に対応する必要がある。そこで本システムの実装では、赤外線モジュールから出力されると 0 と 1 の信号の経過時間をシリアル通信を用いてシステム上に記録することにより、フォーマット形式に依存せずにシステムに記録することを可能とした。装置が受信モードの時は、常に赤外線を監視し、赤外線を受信すると PIC の USART 機能を用いてシリアルポートへ赤外線データの出力を行う。赤外線送信部には赤外線 LED TLN115A³⁴⁾ を利用した。赤外線の送受信とシリアル通信の制御には PIC マイコン PIC 16F877A-I/P を利用した。

5.3 機器情報管理サーバ部

機器情報管理サーバは登録された機器情報の管理や状態を考慮した機器制御などを行う。機器情報と登録する Web アプリケーションの提供や、機器制御 API の処理を行い、本システムにおいて中心的な役割を持つ部分である。機器情報管理サーバの環境は、Web サーバに Apache-2.2.6 を、スクリプト言語に PHP-5.2.4 を用いた。これらの環境は XAMPP³⁵⁾ と呼ばれる、Apache、MySQL、PHP などの一括インストールが可能なパッケージを利用して構築した。

5.3.1 機器情報管理モジュール

機器情報管理モジュールは機器情報管理サーバと外部との通信を Web サーバを通して行うためのモジュール

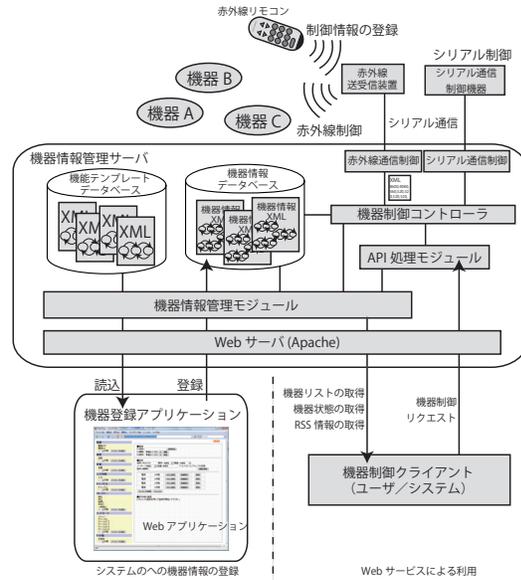


図 10 機器情報管理サーバ

である。機器情報管理サーバ上のデータベースや各モジュールと連携し次のような役割を担う。

- 機器情報の登録の際にデータベースより機能テンプレートを呼び出す
- 機器情報登録 Web アプリケーションを提供する
- 機器情報を機器情報データベースに登録する
- API 処理モジュールと連携し、機器情報の提供や制御を行う WebAPI サービスを提供する
- 機器制御コントローラに対して機器制御の指示を出し、機器制御を行う

5.3.2 機器情報データベース

機器情報データベースには、システムに登録された機器情報を XML ファイルとして保存している。この XML ファイルには機器を識別するための固有の ID が付けられ、また、機器が設置されている場所や型番なども含めることができる。機器情報と登録する際には、機能テンプレートデータベースから必要なテンプレートを選択し、機器を制御するための情報を対応付けすることにより登録を行う。登録された情報は機器ごとの XML ファイルとして機器情報データベースに登録する。機器情報データベース内に蓄積し、Web サービスとしての公開や、機器制御 API を用いた機器制御を行う。XML による機器情報データベースの記述例として図 11 にテレビの機器情報を登録した例を示し、表 4 に XML の要素の詳細を示す。なお、XML ファイルの全ての記述を付録 A に示す。

```

<?xml version="1.0" encoding="utf-8"?>
<vre>
  <file_name>20080125_231140.xml</file_name>
  <label>テレビ</label>
  <place>自宅</place>
  <category>ディスプレイ</category>
  <manufacture>SONY</manufacture>
  <model>RM-J248</model>
  <description>リビングに設置してあるテレビ</description>
  <url>http://192.168.207.130:8080/frc/deviceparts.php?file=20080125_231140.xml</url>
  <date>Fri, 25 Jan 2008 23:11:40 +0900</date>

  <parts id="20080125_231218">
    <function>power</function>
    <label>電源</label>
    <number>0</number>
    <type>0</type>
    <status>
      <state id="0">
        ...機能テンプレートに記述されている内容...
      </state>
      ...状態数の数だけ<state></state>を繰り返し...
    </status>
  </parts>

  <parts id="20080125_231241">
    <function>input</function>
    <label>入力切替</label>
    <number>7</number>
    <type>1</type>
    <status>
      <state id="0">
        ...入力切替の状態数だけ<state></state>を繰り返し...
      </state>
    </status>
  </parts>

  <parts id="20080125_231313">
    ...チャンネル切替の機能テンプレートに記述されている内容...
  </parts>

  ...読み込んだテンプレートの数だけ<parts></parts>を繰り返し...
</vre>

```

図 11 機器情報の XML 記述 (テレビ)

表 4 機器情報の XML 属性

XML 要素	内容
file_name	機器情報 XML ファイル名を記述
label	テンプレートファイルの名前を記述
place	機器の設置場所を記述
category	機器の種類を記述
manufacture	機器のメーカーを記述
model	機器の型番を記述
description	機器の情報を記述
url	機器情報を取得するための URL を記述
date	機器が登録された年月日時間を記述
parts	機能テンプレートから読み込んだ制御情報の XML を記述

5.3.3 機能テンプレートデータベース

図 10 中の「機能テンプレートデータベース」はテンプレートを用いて機器情報の登録をするために実装した。機能テンプレートは様々な機器が共通に持っている機能の状態遷移をテンプレートとして用意し、各状態に機器制御を対応づけることにより機器登録の手間を軽減する。本実装では 3.2 節で述べた、電源、温度、音量、入力切替、チャンネル、モード、プレイヤー、コントローラのテンプレートを作成した。テンプレートに存在しない機能については、自由にカスタマイズ可能なテンプレートが利用可能である。本実装では機器の状態遷移を XML の記述により管理することにし、機能ごとに XML ファイルを作成した。XML によるテンプレート記述の例として図 12 に入力切替のテンプレートを示し、表 5 に XML の要素の詳細を示

```

<?xml version="1.0" encoding="utf-8"?>
<template>
  <function>input</function>
  <label>入力切替</label>
  <number>7</number>
  <type>1</type>
  <status>
    <state id="">
      <label>入力</label>
      <action>input</action>
      <active></active>
      <rcv_signal></rcv_signal>
      <next_signal></next_signal>
      <next_id></next_id>
      <back_signal></back_signal>
      <back_id></back_id>
    </state>
  </status>
</template>

```

図 12 入力切替のテンプレートの XML 記述

表 5 機能テンプレートの XML 属性

XML 要素	内容
function	テンプレートファイルの機能を記述
label	テンプレートファイルの名前を記述
number	テンプレートファイルの表示順を記述
type	相対遷移の Pattern 番号を記述
state	状態ごとに固有の id を付け記述
label	状態の名前を記述
action	状態における機器の動作を記述
active	現在の状態に対して "true" を記述
rcv_signal	絶対遷移するための制御情報を記述
next_signal	次の状態に相対遷移するための制御情報を記述
next_id	次の状態の相対遷移先を state id 属性で記述
back_signal	前の状態に相対遷移するための制御情報を記述
back_id	前の状態の相対遷移先を state id 属性で記述

す。機器の状態は state 属性 (図中のグレー網掛け部分) により表現する。

5.3.4 API 処理

機器情報データベース内に登録されている機器情報の取得や機器制御を外部から行うために機器情報管理サーバに API を実装した。機器情報管理サーバに対して HTTP 通信を行う事により本システム以外のシステムや機器との通信が可能となる。API を利用する際には http://ServerURL/api.php に対して POST または GET メソッドによりリクエストを送る。ここで ServerURL とは機器情報管理サーバの IP アドレスまたはホスト名である。本システムに実装した API の一部を表 6 に示す。

5.3.5 機器制御コントローラ

機器制御コントローラは機器制御管理モジュールや

表 6 API 一覧

API 説明
<ul style="list-style-type: none"> ・機器管理サーバに登録されている機器の一覧を XML 形式で取得 . ・ file_name で指定した機器が持つ機能の状態一覧を取得 . ・ file_name で指定した機器に対し、機器制御を実行 . ・ file_name で指定した機器の function で指定した機能の現状態を取得 . ・ file_name で指定した機器の function で指定した機能の現状態を設定 . ・ file_name で指定した機器の XML ファイルの属性を設定 . ・ file_name で指定した機器の function で指定した機能の XML ファイルの属性を設定 . ・ model で指定した機器の型番により機器情報ファイルを検索し、XML ファイルを取得 .

制御 API からの指令を受け、機器情報データベースに登録されている内容を元に赤外線やシリアル通信などを行うプログラムに対して制御指示を送る .

5.3.6 赤外線・シリアル通信制御

赤外線・シリアル通信制御は、赤外線送受信装置やシリアル通信制御機器と通信を行い、機器情報の取得や機器制御を行う . 本システムではシリアル通信を用いて制御するプログラムを Java を用いて実装した . このプログラムは、機器情報管理サーバに接続された機器の監視と、機器制御コントローラからの指示を監視している . 機器情報コントローラは、XML データを用いてデータの入出力を行うように実装した . そのため、例えば本システムで実装した赤外線送受信装置をデータ形式の異なる別の赤外線送受信装置に置き換える場合でも、その装置に対応した赤外線通信制御プログラムから赤外線データを XML 形式として出力することにより対応可能である .

5.4 状態を用いた機器制御

本研究で実装したシステムでは、機器が持つ状態を含んだ機器制御情報を登録可能である . 機器制御を行う際には制御対象とする機器をどのような状態に遷移させるかという指示を表 6 に示した API を用いて HTTP リクエストすることにより制御する . API 処理で受け取ったリクエストは機器情報モジュールによって機器状態を考慮した最適な制御情報の探索が行われ、機器制御コントローラに送られる . 最適な制御情報を探索するために、機器情報が記述されている XML ファイル内の (active) 要素を利用する . ここでは例として絶対遷移可能な場合の制御と相対遷移のみ可能な場合の探索アルゴリズムについて述べる .

5.4.1 絶対遷移可能な場合の例

ここでは絶対遷移可能な場合の例として「機器制御クライアントが自宅の部屋にあるテレビを 5 チャンネルにする」という制御を扱う . ただし、このテレビの機器情報は予め機器情報データベースに登録されており、目的とするチャンネルに直接切り替える制御が可能であるとする .

アルゴリズム

- (1) 機器情報管理サーバに対して、登録されている機器の一覧を取得するリクエストを送る
- (2) 取得した機器の一覧を元に、制御対象とする機器（自宅の部屋にあるテレビ）に対してどのような制御が可能であるかを取得するリクエストを送る .
- (3) 取得した制御一覧よりテレビを 5 チャンネルにするための API コマンドを取得し、API 処理モジュールに対してリクエストを送る .
- (4) リクエストを受け取った API 処理モジュールは機器情報管理モジュールに操作対象とする機器と目的とする状態の情報を送る .
- (5) 機器情報管理モジュールは機器情報データベースを参照し、制御対象とするテレビの現在の状態を確認する .
- (6) 現在の状態が既に 5 チャンネルの状態になっていれば、機器制御コントローラに対して指示を行わない .
- (7) 現在の状態が 5 チャンネル以外である場合は、5 チャンネルの状態に絶対遷移させるための制御情報を取得し、機器制御コントローラに指示を行うと共に、データベースの機器状態を変更する .

このように、絶対遷移をさせるための情報が機器情報データベースに記録されていれば、現在の状態と比較するだけで制御が可能である .

5.4.2 相対遷移のみ可能な場合の例

続いて相対遷移のみ可能な場合の例として「機器情報クライアントが自宅の部屋にあるテレビの音量レベルを 15 にする」という制御を扱う . ただし、前節の例と同様に機器情報管理サーバに機器情報の登録がされているとする .

アルゴリズム

- (1) 機器情報管理サーバに対して、登録されている機器の一覧を取得するリクエストを送る .
- (2) 取得した機器の一覧を元に、制御対象とする機器（自宅の部屋にあるテレビ）に対してどのような制御が可能であるかを取得するリクエスト

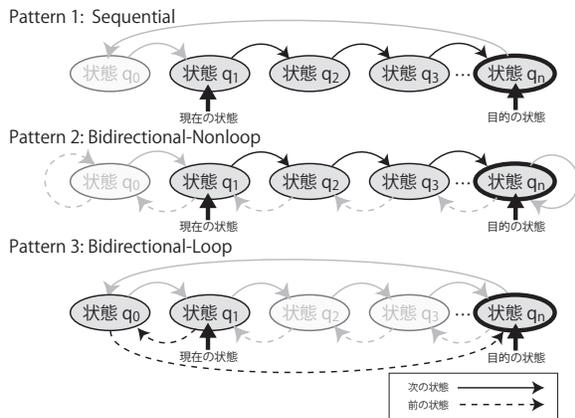


図 13 相対遷移の経路探索

- を送る。
- (3) 取得した制御一覧よりテレビの音量レベルを 15 にするための API コマンドを取得し、API 処理モジュールに対してリクエストを送る。
 - (4) リクエストを受け取った API 処理モジュールは機器情報管理モジュールに操作対象とする機器と目的とする状態の情報を送る。
 - (5) 機器情報管理モジュールは機器情報データベースを参照し、制御対象とするテレビの現在の状態を確認する。
 - (6) 現在の状態が既に音量レベル 15 になっていれば、機器制御コントローラに対して指示を行わない。
 - (7) 現在の状態が音量レベル 15 以外である場合は、その状態にするための指示を機器制御コントローラに対して送る必要があるが、絶対遷移をするための制御情報が登録されていないため、相対遷移によって目的の状態（音量レベル 15）にするための操作情報を探索する。
 - (8) 相対遷移による遷移が可能な場合は、機器情報 XML ファイル内の $\langle \text{next.id} \rangle$ 要素と $\langle \text{back.id} \rangle$ 要素を用いて図 13 のように幅優先探索を行う。
 - (9) 探索結果より相対遷移によって音量レベルを 15 にするための指示を機器制御コントローラに対して行い、データベースの機器状態を変更する。

6. 評価

本システムを用いて、実際に機器情報の登録と機器制御 API を利用した機器制御を行い、非情報家電のネットワークからの利用が可能であることを確認する。まず、実装したシステムを用いて様々な機器の制御情報の登録を行い、実装システムからの機器制御が可



図 14 Web アプリケーションによる機器情報の登録

能であるか確認する。続いて、機器制御を行うための API を用いた Web アプリケーションを作成し、実際にシステムに対して機器制御を行い、状態を考慮した機器制御が可能か確認する。

6.1 システムへの機器情報の登録

本研究で提案している手法では、登録者の手間を軽減するために予め用意された機能テンプレートを組合せ、それに対して制御情報を登録する。図 14 は機器情報の登録を行う Web アプリケーションである。登録者は Web ブラウザを用いて機器管理サーバにアクセスし、Web アプリケーション上から機器情報の登録を行う。そこで、実装した Web アプリケーションを用いて実際に我々の身の回りにある機器の登録を行い、効率的に機器登録が可能であることを確認した。

6.2 被験者による機器登録実験

システムの機器情報を手軽に登録可能であることを確認するために被験者による実験を行った。

6.2.1 実験手順

実験では図 15 のように赤外線送受信機を接続した機器情報管理サーバをネットワーク上に設置し、機器登録の操作を行うクライアント端末は無線 LAN によってネットワークに接続されたノート PC を利用した。実験は以下の手順により行った。

- (1) 機器登録アプリケーションの使い方や登録作業の一連の流れを説明（約 5 分）。
- (2) 具体的な登録作業をテレビの機器情報の登録により例示（約 5 分）。
- (3) 被験者に登録してもらう機器の説明とリモコン操作の説明（約 2 分）



図 15 被験者による機器登録実験

表 7 実験で登録作業を行った機器及び機能

機器 (メーカー/型番)	機能 (状態【現状態】)	遷移
プロジェクタ (EPSON/EMP-TW20)	電源 (ON【OFF】)	絶対
	入力切替 (InputA【InputB】, S-Video, Video)	相対
	音量 (音量 0 ~ 【音量 10】 ~ 音量 23)	相対
	コントロール (Menu【Esc】, , , , Enter)	絶対
テレビ (EIZO/SC26XD2)	電源 (ON【OFF】)	絶対
	外部入力 (【DVD】, ビデオ, HDMI, 携帯)	絶対
	放送切替 (【地上 A】, 地上 D, BS, CS)	絶対
	チャンネル (1 ~ 【5】 ~ 12)	絶対/相対
	音量 (音量 0 ~ 【音量 23】 ~ 音量 50)	相対

- (4) 以下の手順により機器情報の登録を行ってもらい、登録時の観察と時間を計測
- (a) 登録する機器の名称や型番、設置場所などの情報を登録
 - (b) 機能テンプレートから必要なものを選択し、名称などを変更
 - (c) 赤外線リモコンを用いて赤外線情報を登録

実際に登録を行ってもらった機器及び機器の機能を表 7 に示す。

6.2.2 結果

3人の被験者に対して実験を行い、登録作業を開始してから必要なテンプレートを選択するまでの時間 (Time1) と赤外線リモコンの操作を行い赤外線登録するまでの時間 (Time2) を計測した。結果を表 8 に示す。プロジェクタの登録に比べ、テレビの登録に Time1

表 8 被験者実験の結果

被験者	機器	Time1	Time2	合計
被験者 A	プロジェクタ	1 分 55 秒	1 分 03 秒	2 分 58 秒
	テレビ	2 分 46 秒	2 分 28 秒	5 分 14 秒
被験者 B	プロジェクタ	2 分 55 秒	1 分 30 秒	4 分 25 秒
	テレビ	3 分 31 秒	2 分 20 秒	5 分 51 秒
被験者 C	プロジェクタ	2 分 12 秒	1 分 17 秒	3 分 29 秒
	テレビ	3 分 19 秒	2 分 42 秒	6 分 01 秒

の時間が多くかかっているのは、外部入力や放送切替などをテンプレートから登録する際に、ラベル名の変更する状態が多かったためである。また、Time2の時間を比較するとプロジェクタに比べテレビの登録に時間がかかっているが、これはチャンネルを登録する際にチャンネル 1 ~ チャンネル 12 まで順に赤外線を登録する必要があったためである。今回の実験では機器一台を登録するのに長くて 6 分程度かかっているが、提案手法を用いてテンプレートの選択から赤外線の登録まで順に行うことで、状態遷移を含んだ機器情報の登録を容易に実現できることが確認できた。

6.3 様々な機器情報の登録

続いて、実装したシステムが様々な機器に対して対応可能であるか確認するために、機能やメーカーが異なる機器の登録を行った。登録を行った機器の一覧を表 10 に示す。登録する機能に完全に一致するテンプレートが無い場合には、状態遷移のパターンが類似するテンプレートを適用し登録を行った (表中の*印)。表中の「赤外線登録」の項目で「×」が付いているものは登録後に実装した赤外線送受信装置を用いて制御が行えたことを意味し「×」は制御が行えなかった事を意味する。制御が行えなかった原因を調べるために、オシロスコープを用いて信号の波形を確認した。その結果、表中で×が付いている「照明」と「エアコン」は NEC フォーマットや SONY フォーマットの赤外線とは大きく異なり、信号の長さが非常に長いものであった。実装した赤外線送受信装置は赤外線フォーマットの種類に依存しないように信号の HI と LOW の時間で記録しているが、PIC 上に読み込むために用意した配列のサイズを大きく上回る長さであったため、受信した赤外線信号と同じ信号を送信することができなかった。

7. 機器状態を利用した制御

本システムが機器の状態を考慮した機器制御が可能であるか確認をした。

7.1 状態を考慮した制御

状態を考慮した制御の確認を行うための機器として

表 9 テレビの電源制御

現在の機器状態	目的の状態	次の機器状態	システムの動作
電源 OFF	テレビ電源 OFF	電源 OFF	赤外線出力無し
	テレビ電源 ON	電源 ON	赤外線出力
電源 ON	テレビ電源 OFF	電源 OFF	赤外線出力
	テレビ電源 ON	電源 ON	赤外線出力無し

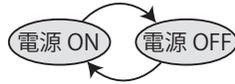


図 16 電源の状態遷移

SONY のテレビ (RM-J248) の機器情報を用いた . このテレビのリモコンには電源を ON/OFF するためのボタンがあり , 電源が ON の状態と OFF の状態を切り替えるための赤外線信号は同一のものである . そのため , 電源の状態を考慮せずに赤外線を送信すると誤った状態に遷移させてしまう . 本システムを用いて機器制御を行った結果を表 9 に示す . 状態を考慮した機器制御を行うことにより , システムに対して目的とする状態を指示するだけで適切な機器制御が可能であることを確認した .

7.2 相対遷移を行う機能の制御

相対遷移を行う機能の制御の確認を行うための機器として SONY のプロジェクタ (RM-PJM17) を用いた . このプロジェクタには複数の入力端子があり , 入力切替ボタンを用いてどの入力端子からの信号を入力するか選択する . それぞれの入力端子からの入力を状態としてとらえると図 17 のような状態遷移となる . 赤外線リモコンから入力切替の制御を行う場合は , 目的の状態になるまで信号を繰り返し送る必要があるが , 本システムでは機器の状態を保持しているため目的とする状態を直接指示することが可能である .

ここでは , 目的とする状態を API を用いてシステムに指示し適切な制御が行われるか確認した . 機器が「入力 A」の状態の時に , 「ビデオ」の状態にするための指示を API を用いて送ったところ , 入力切替を行うための赤外線信号が 3 回連続して送信され , 機器の状態が「ビデオ」に変更されることを確認した . このように , 機能の状態遷移が絶対遷移か相対遷移かに関わらず目的の状態を指示するだけで制御が可能となる .

8. API を用いた機器制御

本研究で実装した機器制御 API は , サーバ上に登録されている機器の詳細な情報を HTTP 通信を用いて XML 形式で取得することができる . XML ファイ

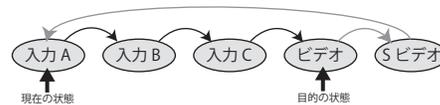


図 17 入力切替の状態遷移



図 18 RSS による機器情報の公開と制御

ル内では , それぞれの機器が持つ機能やどのような状態遷移をするか構造的に記述されているため , 他のシステムや Web アプリケーションとの連携が期待できる . そして , 取得した情報を元に HTTP 通信で機器管理サーバに対して機器制御のリクエストを行い , サーバ上に登録された機器の制御を行う . その際 , 機器を制御するために必要な情報や機器の状態はシステムが把握しているため , 制御対象の機器と目的の状態を指示するだけで良い . 実際に表 6 で定義した API を用いていくつかの Web アプリケーションを作成した .

8.1 RSS による機器情報の公開と制御

API を用いると , 登録されている機器の情報が容易に取得できる . そこで , 取得した機器情報の XML データを RSS 形式に変換し , RSS リーダを用いて機器の一覧を取得可能にした . RSS 形式で提供されるデータは機器情報の追加 , 削除により動的に変化する . 機器の一覧から辿ることができる機能の一覧には機器を制御するための URL が含まれており RSS リーダ上から機器制御が可能である .

8.2 携帯端末からの機器制御を可能にする Web アプリケーション

API によって取得した XML データを整形するためのスタイルシートを作成することで , 様々な端末上での機器制御が可能となる . ここでは Apple 社製の iPod touch に搭載されている Web ブラウザ「Safari」に対応したスタイルシートを作成した . iPod touch は無線 LAN デバイスを内蔵し , Web ブラウザやフォトビューアなどのアプリケーションを利用可能な携帯音楽プレイヤーである . 画面をタッチすることで様々な操作が可



図 19 携帯端末からの機器制御を可能にする Web アプリケーション

能なマルチタッチインターフェイスを搭載している。今回実装した Web アプリケーションでは API によって Web サーバ上の機器情報を取得し、機器制御用のインタフェースをユーザに提供している。状態数の少ない機能はボタンによる表示、温度調整や音量調整のような状態数が多い機能はセレクトボックスによる表示になるようにしている。

8.3 機器情報の共有

上記のアプリケーションを利用する際には、予め機器管理サーバに機器の情報を登録した。本研究で実装したシステムでは機器の型番によりサーバ上の機器情報を検索可能な API を実装している。そのため機器情報サーバに登録された機器情報を共有し、API を用いて検索することで、別のサーバに登録されている機器の情報を利用することが可能となる。

9. ま と め

本稿ではネットワークへの接続性や機器連携機能を持たない非情報家電を、プログラミングの実装などの手間をかけずにシステムに登録し、Web サービスとして利用可能な環境を構築する手法を提案した。本手法では機器が持つ様々な機能を状態としてとらえ、モデル化した状態遷移のパターンごとにテンプレートを作成した。機器に登録する際には、必要なテンプレートを組合せ、機器を制御するための赤外線信号やシリアルコマンドを対応付けすることにより効率的に機器情報を登録することが可能となった。また、本手法を Web アプリケーションとして実装し、ユーザによる Web ブラウザからの登録を可能にした。登録された機器情報はネットワーク上のサーバ上に XML 形式の機器情報データベースとして保存し、API を用いた機器情報の取得や機器制御が可能な Web サービスとして利用できる。機器情報データベース内では制御対象とする機器の状態を保持しているため、API を用いて機

表 10 システムへの機器登録

機器 (メーカー/型番)	機能 (利用テンプレート)	状態遷移	IR 登録
照明 (NEC/9LKZ121GS)	電源 (切替)	相対	×
	電源 (切替)	相対	
	モード切替 (*入力切替)	相対	
	温度 (温度)	相対	
	風量 (*温度)	相対	
エアコン (DAIKIN/AN25BDS-W)	風量 (*温度)	相対	×
	機能切替 (切替)	相対	
	タイマ (*入力切替)	相対	
	電源 (切替)	相対	
	風量調整 (*入力切替)	相対	
扇風機 (SHARP/PJ-K3HR-H)	機能切替 (切替)	相対	
	タイマ (*入力切替)	相対	
	電源 (切替)	相対	
	音量 (音量)	相対	
	チャンネル (チャンネル)	絶対/相対	
テレビ (SONY/RM-J248)	入力切替 (入力切替)	相対	
	機能切替 (切替)	相対	
	設定 (コントローラ)	絶対	
	電源 (切替)	相対	
	音量 (音量)	相対	
テレビ (EIZO/SC26XD2)	チャンネル (チャンネル)	絶対/相対	
	外部入力 (入力切替)	絶対	
	放送切替 (入力切替)	絶対	
	設定 (コントローラ)	絶対	
	DVD 開閉 (切替)	相対	
ビデオレコーダ (TOSHIBA/RM-F100)	DVD 再生機能 (プレイヤー)	絶対	
	電源 (切替)	相対	
	チャンネル (チャンネル)	絶対/相対	
	再生機能 (プレイヤー)	絶対	
	再生速度切替 (*入力切替)	相対	
プロジェクト (EPSON/EMP-TW20)	入力切替 (入力切替)	相対	
	機能切替 (切替)	相対	
	明度調整 (切替)	相対	
	フォーカス調整 (*音量)	相対	
	設定 (コントローラ)	絶対	
ステレオアンプ (YAMAHA/TSS-15)	電源 (切替)	相対	
	音量 (音量)	相対	
	モード切替 (入力切替)	絶対	
	入力切替 (入力切替)	相対	
	機能切替 (切替)	相対	

器制御を行う際には目的の状態を指示するだけで適切な機器制御を行うことが可能となった。

さらに、実際に赤外線制御可能な機器を登録するために、シリアル通信で赤外線の送受信を行う赤外線送受信機を作成し、様々な機器に対する登録や動作確認を行った。また、Web サービスを用いた機器制御を行うために API を利用して機器情報の取得や制御を行うための Web アプリケーションを作成し、機器の状態を考慮した適切な機器制御が可能であることを確認した。

9.1 今後の課題

本研究では登録された機器の状態を XML ファイルとしてサーバ上に保持した。赤外線制御する機器のように、機器の現状態を取得するのが困難な機器はシステム上で保持している状態と制御対象となる機器の状態を合わせておく必要がある。この保持のためには、状態遷移が発生する制御を行うと同時に機器情報を更新することが必要になる。一方、赤外線リモコンや機器自体のスイッチなど、外部制御によって機器操作が行われると、システム上での機器状態と実際の機器の状態が異なってしまう、適切な制御ができなくなってしまう。より正確な状態を取得するための方法として次のようなことが考えられる。

- 機器の状態を取得する仕組みを実装する

- 外部から機器制御を行う際の信号を取得する機器の状態を取得する仕組みを実装するとは、例えば家電の中には電源が ON/OFF の状態であることを示す LED を搭載しているものが見られるが、この LED の状態を光センサやカメラなどによってとらえ、機器の状態を取得する方法である。しかし、この方法では電源状態のようなごく限られた状態しか取得できない可能性がある。外部から機器制御を行う際の信号を取得するとは、例えば赤外線リモコンで機器制御を行った際に送信される赤外線をシステムでも取得し、機器の状態を取得する方法である。外部の制御を確実にとらえることにより、より正確な状態を取得することが可能になると考えられるが、機器の赤外線受光部に近い場所に赤外線受信機を設置する必要が生じる。

これらの方法を複合的に組合せることにより、正確な機器状態を取得することが期待できる。

謝辞 本研究の一部は、総務省「戦略的情報通信研究開発推進制度 (SCOPE)」の支援を受けて行われた。

参考文献

- 1) Mark Weiser, "The computer for Twenty-First Century", Scientific American, pp.94-100(1991).
- 2) Brad Johanson, Armando Fox, and Terry Winograd, The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms, IEEE Pervasive Computing 1:2, pp.67-75(2002).
- 3) 森川博之, 南正輝, 青山友紀: STONE: 環境適応型ネットワークサービスアーキテクチャ, 信学技報, IN2001-12(2001).
- 4) Okoshi, T., Wakayama, S., Sugita, Y., Iwamoto, T., Nakazawa, J., Nagata, T., Furusaka, D., Iwai, M., Kusumoto, A., Harashima, N., Yura, J., Nishio, N., Tobe, Y., Ikeda, Y. and Tokuda, H.: Smart Space Laboratory Project: Toward the Next Generation Computing Environment, IEEE Third Workshop on Networked Appliances (IWNA 2001), (2001).
- 5) 小川耕司, 小暮悟, 小西達裕, 伊東幸宏, "異機種を统一的に利用可能な家電統合インタフェースの構築", 情報学ワークショップ 2007 (WiNF2007), pp. 99-102 (2007.9).
- 6) Jeffrey Nichols, Brad A. Myers, Michael Higgins, Joseph Hughes, Thomas K. Harris, Roni Rosenfeld, Mathilde Pignol, "Generating Remote Control Interfaces for Complex Appliances", Proc. of the 15th Annual ACM Symposium on User Interface Software and Technology 2002, pp.161-170
- 7) Hidenori Yamamoto, Takaaki Sekiguchi, Hiromitsu Kato, Shigetoshi Sameshima, Masaya Kadota, Jun'ichi Yura, Kazunori Takashio, Hideyuki Tokuda, "Ubiquitous Information Platform for Interoperation among Different Middleware", マルチメディア, 分散, 協調とモバイル (DICOMO2005) シンポジウム, pp.673-676,2005
- 8) Atsushi Tanaka, Tomoyuki Yashiro, "Routing Method Considering Node Density in Wireless Sensor Networks", マルチメディア, 分散, 協調とモバイル (DICOMO2006) シンポジウム, pp.181-184,2006
- 9) Ryota Kawamoto, Takumi Emori, Seiichiro Hara, "Performance Evaluation of Routing Protocol for Sensor Networks (ZigBee) concerning Connectivity to DLNA/UPnP-ZigBee Gateway", マルチメディア, 分散, 協調とモバイル (DICOMO2006) シンポジウム, pp.81-84,2006
- 10) 石井智康, "LegacyDevice を中心としたホームネットワークに関する研究", 国立大学法人北陸先端科学技術大学院大学情報科学研究科修士論文, 2003.3.
- 11) 松崎寛之, "レガシーデバイスを用いたホームネットワーク構築における機器の状態取得及び管理に関する研究", 国立大学法人北陸先端科学技術大学院大学情報科学研究科修士論文, 2005.3.
- 12) 植田健太, 小坂隆浩, 佐藤健哉, "Google Maps を用いた統合型家電機器操作システムの構築", 社団法人情報処理学会, Vol.2007, No.16(20070301), pp. 399-404
- 13) 吉田竜二, 井上亮文, 平石絢子, 重野寛, 松下温, "Web ベースのモバイルホームネットワーク連携方式の提案", マルチメディア, 分散, 協調とモバイル (DICOMO2001) シンポジウム, pp.411-416,2001
- 14) 河口信夫, 稲垣康善, "cogma:動的ネットワーク環境における組み込み機器間の連携用ミドルウェア", 情報処理学会コンピュータシステム・シンポジウム, pp.1-8, 2001.
- 15) 河口信夫, 春原雅志, "WebCodget: Web サーバに移動して動作する組み込み機器向け移動ソフトウェア", 情報処理学会ユビキタスコンピューティング研究会,2005-UBI-009 pp.41-44 (2005).
- 16) Masashi Sunohara, Yohei Iwasaki, Nobuo Kawaguchi, "Mobile Software System for Simple Device Coordination using Web Server", 2006 International Symposium on Ubiquitous Computing System The Proceedings of Poster Session, pp.139-150, Oct.11 2006.
- 17) 神崎 康宏, "作りながら学ぶ PIC マイコン入門 豊富な各機能を動かして確認しながら理解する", CQ 出版, 2005.4.
- 18) 鈴木 美朗志, "PIC&C 言語でつくる赤外線リモコン わかるマイコン電子工作 リモコン実用回路の製作と改造ロボット工作", 電波新聞社, 2007.8.
- 19) 後閑 哲也, "電子工作のための PIC16F 活用ガイドブック", 技術評論社, 2004.9.
- 20) 三宅基治, 吉川貴, 中土昌治, "モバイル/ホームネットワーク連携技術とホームゲートウェイ装置の試作", NTT DoCoMo テクニカル・ジャーナル

Vol. 14 No.4, 2007

- 21) 西田雄也, 福島志斗, 藤原圭, 原元司, "merisa - 電子メールによる汎用遠隔操作システム - ", 情報処理学会第 62 回全国大会, 特別トラック 5, pp.37-40,2001
 - 22) Grundig AG, Hitachi Ltd., Matsushita Electric Industrial Co., Ltd., Philips Consumer Electronics, Sharp Corporation, Sony Corporation, Thomson Multimedia, Toshiba Corporation, "HAVi", "<http://www.havi.org/>"
 - 23) Jesse James Garrett, "Ajax: A New Approach to Web Applications", "<http://www.adaptivepath.com/ideas/essays/archives/000385.php>"
 - 24) ECHONET CONSORTIUM, "ECHONET", "<http://www.echonet.gr.jp/>"
 - 25) Sun Microsystems, Inc., "Jini", "<http://www.sun.com/software/jini/index.xml>"
 - 26) UPnP(TM) Forum, "UPnP", "<http://www.upnp.org/>"
 - 27) Digital Living Network Alliance, "DLNA", "<http://www.dlna.org/>"
 - 28) Apple Inc., "Bonjour", "<http://www.apple.com/macosx/features/bonjour/>"
 - 29) LANTRONIX Inc., "XPort / WiPort", "<http://www.lantronix.com/>"
 - 30) InterSolutionMarketing Inc., "Parani10", "<http://www.intersolutionmarketing.com/>"
 - 31) Microchip Technology Inc., "PIC", "<http://www.microchip.com/>"
 - 32) Microchip Technology Japan Inc., "PICC Lite", "<http://microchip.htsoft.com/products/compilers/PICClite.php>"
 - 33) KODENSHI Inc., "KSM-60 LM", "<http://www.kodenshi.co.kr/download/DB/4-1/KSM-60xLM2E.PDF>"
 - 34) TOSHIBA Inc., "TLN115A", "[http://www.semicon.toshiba.co.jp/docs/datasheet/ja/Opto/TLN115A\(F\)_ja_datasheet_071001.pdf](http://www.semicon.toshiba.co.jp/docs/datasheet/ja/Opto/TLN115A(F)_ja_datasheet_071001.pdf)"
 - 35) apache friends, "XAMPP", "<http://www.apachefriends.org/en/index.html>"
-