

動的サービスバインディング機構を用いたマルチベンダホームネットワークシステムの一実現手法

福岡 佑介[†] 西岡 隆司[†] 中村 匡秀^{††} 井垣 宏^{††} 松本 健一[†]

[†] 奈良先端科学技術大学院大学情報科学研究科 〒630-0192 奈良県生駒市高山町 8916-5

^{††} 神戸大学大学院工学研究科 〒657-8501 兵庫県神戸市灘区六甲台町 1-1

E-mail: †{yuusuke-f,takashi-ni,matumoto}@is.naist.jp, ††{masa-n,igaki}@cs.kobe-u.ac.jp

あらまし 本稿では、複数のベンダのネットワーク家電を収容可能なホームネットワークシステム (HNS) を構築する手法を提案する。まず、ネットワーク家電のベンダに依存しない標準家電サービスを構築し、HNS アプリケーションに向けて公開する。次に、各標準家電サービスとネットワーク家電を動的に結び付ける機構を構築する。これにより、様々なベンダのネットワーク家電に対応できる HNS アプリケーションが開発できるとともに、HNS や HNS アプリケーションを再構築することなく、HNS に収容されているネットワーク家電を置き換えることができる。また、本稿では提案手法に基づいた HNS フレームワークを実装するとともに、そのフレームワークを用いて複数ベンダのネットワーク家電からなる HNS を構築し、提案手法の有用性を確認した。

キーワード ホームネットワークシステム, ネットワーク家電, マルチベンダシステム, ホームネットワークサービス, 動的バインディング

Implementing Multi-Vendor Home Network System Using Dynamic Service Binding Mechanism

Yusuke FUKUOKA[†], Takashi NISHIOKA[†], Masahide NAKAMURA^{††}, Hiroshi IGAKI^{††}, and
Ken-ichi MATSUMOTO[†]

[†] Graduate School of Information Science, Nara Institute of Science and Technology Takayama-cho 8916-5,
Ikoma-shi, Nara, 630-0192 Japan

^{††} Graduate School of Engineering, Kobe University Rokkodai-cho 1-1, Nada-ku, Kobe, 657-8501 Japan
E-mail: †{yuusuke-f,takashi-ni,matumoto}@is.naist.jp, ††{masa-n,igaki}@cs.kobe-u.ac.jp

Abstract This paper presents a method that constructs the home network system (HNS) with multi-vendor networked home appliances. The proposed method first prepares vendor-neutral appliance services, and exhibits the services to the HNS applications. We then prepare a mechanism, which dynamically binds each vendor-neutral service and concrete appliances during runtime. Thus, the HNS service provider can build a common HNS application for various combinations of multi-vendor appliances. Moreover, it is unnecessary to rebuild the vendor-neutral services as well as the HNS applications even when any appliance is replaced with another. We have implemented the proposed method, and deployed the system in an actual HNS. The experimental evaluation showed that the implementation worked well with sufficiently small overhead for the practical settings.

Key words home network system, networked appliances, multi-vendor system, home network services, dynamic binding

1. はじめに

家電やセンサを家庭内のネットワークに接続することで、家電単体では実現できない、より便利なサービスをユーザに提供

するホームネットワークシステム (HNS) の研究開発が進められており、既に商品化されたものも存在する [12] [13] [14]。一般的に、HNS に接続される家電はネットワーク家電と呼ばれ、ネットワークインタフェースやプロセッサ、ストレージを持ち、

さらに外部のソフトウェアから家電を制御するための API (家電 API と呼ぶ) を備えている。HNS サービスは、これらの家電 API を決められた手順で呼び出すソフトウェア (HNS アプリケーションと呼ぶ) によって実現される。

しかしながら、現在提供されている HNS アプリケーションは、単一ベンダ製のネットワーク家電にしか対応していないものがほとんどである。その原因として、同種のネットワーク家電であっても、そのベンダによって家電 API の仕様が異なることが挙げられる。そのため、HNS サービスを各家庭に提供する事業者 (HNS サービス提供者と呼ぶ) は様々なベンダのネットワーク家電に対応した HNS アプリケーションを開発することができず、またユーザも、好みのベンダのネットワーク家電を任意に選択し、HNS に収容させることができない。

この問題を解決するために、本稿では、複数のベンダのネットワーク家電を収容可能な HNS (マルチベンダ HNS と呼ぶ) を構築する手法を提案する。

まず、家電の種類ごとに、任意のベンダのネットワーク家電を制御可能な API を持つ標準家電サービスを準備し、HNS アプリケーションに向けて公開する。次に、これらの標準家電サービスと、HNS に収容するネットワーク家電との結び付けを動的に行う仕組みを構築する。これにより、HNS に任意のベンダのネットワーク家電を収容できるとともに、ネットワーク家電の置き換えが発生した場合でも、標準家電サービスとネットワーク家電との結び付け定義を変更するだけで対応することができる。

また、本稿では提案手法に基づいた HNS フレームワークを用いてマルチベンダ HNS を構築し、HNS や HNS アプリケーションを再構築することなく、HNS に収容されているネットワーク家電を置き換えられることを確認した。

2. 準備

2.1 ホームネットワークシステム (HNS)

ホームネットワークシステム (HNS) は、複数のネットワーク家電 (ネットワークに接続する機能を持った家電) と、ホームサーバで構成されており、それらは家庭内 LAN に接続されている。図 1 は HNS を表した図である。各ネットワーク家電には制御用の API (家電 API) が備えられており、外部のソフトウェアは家電 API を通してそのネットワーク家電を制御することができる。ホームサーバは、ユーザに提供する HNS サービスを管理するアプリケーションサーバ、ならびに外部ネットワークへのゲートウェイの役割を持つ。各 HNS サービスは、あらかじめ決められた手順で各ネットワーク家電の API を呼び出すソフトウェア (HNS アプリケーション) によって実現される。

HNS サービスの一例として、ここではシアターサービスを挙げる。シアターサービスとは、リビングの照明・カーテン・テレビを連携動作させて、ユーザが映画館のような環境でテレビ番組を視聴できるようにする HNS サービスである。ユーザがシアターサービスを呼び出すと、照明を暗くする、カーテンを閉じる、テレビの電源を入れて最適な音量に設定する、とい

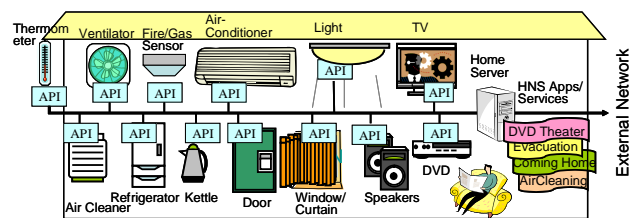


図 1 ホームネットワークシステム (HNS)

た一連の操作が自動的に行われる。

なお、家電が持つ API は、いくつかのレイヤに分類することができる。例えば、ネットワークレイヤの API は、アドレス設定、メッセージや信号の形式、通信プロトコルなど、機器がネットワークを介して通信するための処理を扱う。ネットワークレイヤの標準化を図る規格として、DLNA [1], UPnP [10], ECHONET [2], X10 [8], HomePlug [4] などが提唱されている。

一方、アプリケーションレイヤの API は、低レイヤのモデルウェアやネットワークプロトコルを隠蔽して、ネットワーク家電の論理的な機能を容易に呼び出せるようにしている。例えば、カーテンが持つアプリケーションレイヤの API としては、open() や close() などが考えられる。通常、これらの API は、そのネットワーク家電のベンダによって開発され、HNS アプリケーションに向けて公開される。このようなアプリケーションレイヤの API を構築するためのフレームワークとして、OSGi フレームワーク [6] が挙げられる。また、ネットワーク家電のアプリケーションレイヤの API を、Web サービス [11] を用いて公開する研究も存在する [3]。

本稿では、ネットワークレイヤのような低レイヤの家電 API ではなく、アプリケーションレイヤの家電 API について言及する。

2.2 HNS の問題点

現在商品化されている HNS アプリケーションは、単一ベンダ製のネットワーク家電にのみ対応しているものがほとんどである [12] [13] [14]。その要因として、家電 API の仕様がネットワーク家電のベンダによって異なることが挙げられる。

図 2 は、2.1 で述べたシアターサービスを実現する HNS アプリケーションの実装を表したものである。(a) の TheaterService_for_A は A 社製ネットワーク家電用の HNS アプリケーション、(b) の TheaterService_for_B は B 社製ネットワーク家電用の HNS アプリケーションである。また、それぞれの内部に描かれている図形は、各ベンダの家電 API の仕様に合わせた呼び出し処理を表している。TheaterService_for_A は、照明・カーテン・テレビの家電 API を順に呼び出してシアターサービスを実現する。一方、TheaterService_for_B も、同じく照明・カーテン・テレビの家電 API を順に呼び出して TheaterService_for_A と同様にシアターサービスを実現するが、それぞれのネットワーク家電が持つ家電 API の仕様 (図 2 における図形) が、A 社と B 社で異なるため、TheaterService_for_B の実装は TheaterService_for_A の実装と全く異なっている。従って、例えば (a) において A 社製の

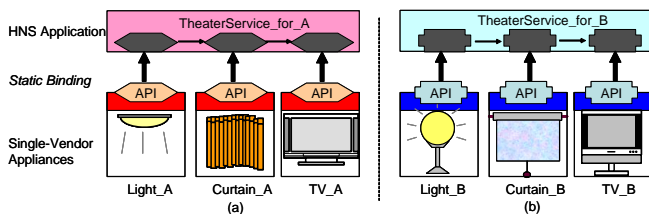


図 2 従来の HNS におけるネットワーク家電と HNS アプリケーションの関係

照明を B 社製の照明に置き換えることはできず、各 HNS アプリケーションは単一ベンダ製のネットワーク家電にしか対応できない。

現在の HNS が持つこのような制約は、HNS の普及促進を妨げるものである。HNS アプリケーションが対応可能なネットワーク家電のベンダが限定されているため、ユーザは好みのベンダのネットワーク家電を HNS に収容させることができない。また、家電 API の仕様はそのネットワーク家電のベンダが決定するため、それらの API を呼び出す HNS アプリケーションも同一のベンダによって提供されることが多く、サードパーティの HNS サービス提供者の参入が困難である。

様々なベンダのネットワーク家電を収容できるマルチベンダ HNS を構築できるようにすることが、HNS の次なる目標である。

2.3 家電 API 仕様の標準化

マルチベンダ HNS を構築するためのシンプルな手法として、家電 API の仕様を標準化し、全てのベンダの間で統一することが考えられる。しかしながら、家電 API の仕様を厳格に決めてしまうと、結果的に各ネットワーク家電ベンダは他社と同じような製品を生産しなければならず、他社との差別化を図ることができなくなる。さらに、全てのベンダが納得できるような形で家電 API を標準化することは非常に困難であり、この手法は現実的とはいえない。

2.4 本稿における仮定

以降、本稿では以下の 3 つの仮定に基づいて議論を行う。

仮定 A1: 各家電 API は、そのネットワーク家電のベンダが準備する。

仮定 A2: 各家電 API の仕様は、そのネットワーク家電のベンダが決定する。

仮定 A3: 各家電 API の仕様は、ユーザや、サードパーティの HNS サービス提供者に公開されている。

3. 提案手法

3.1 要求定義

本稿では、以下の 3 つの要求を満たす、マルチベンダ HNS を構築するためのフレームワークを提案する。

要求 R1: ベンダを問わず、任意のネットワーク家電を HNS に収容できること。

要求 R2: HNS、ならびに HNS アプリケーションを再構築することなく、HNS に収容されているネットワーク家電を置き換えられること。

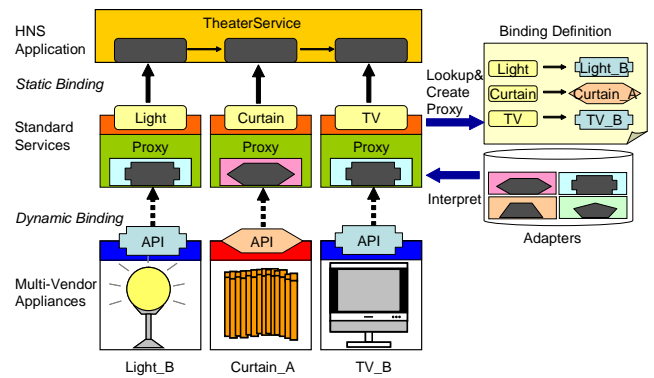


図 3 提案手法のアーキテクチャ図

要求 R3: ネットワーク家電のベンダに限らず、サードパーティの事業者でもユーザに HNS サービスを提供できること。

一般的に、家庭内には様々なベンダの家電が設置され、また家電のベンダの組み合わせは家庭ごとに異なる。要求 R1 を満たすことにより、HNS はあらゆるベンダの組み合わせに対応できるようになるため、様々な家庭にこの HNS を導入できるようになる。

ユーザがネットワーク家電を買い換えた際、HNS に収容されているネットワーク家電を置き換える必要がある。また、将来的には音楽プレーヤーや小型テレビなど、携帯可能な家電を帰宅時に HNS に接続し、外出時に HNS から切断するという利用形態も予想される。HNS が要求 R2 を満たすことにより、このような収容されているネットワーク家電の変化に柔軟に対応できるようになる。

さらに、HNS フレームワークが要求 R3 を満たすことにより、HNS サービス事業に参入する事業者が増加し、これまでに無かった新しいサービスや、より低コストで高品質なサービスがユーザに提供されることが期待できる。

3.2 提案手法の概要

従来の HNS では、HNS アプリケーションが、ベンダによって仕様の異なる家電 API を直接呼び出しているため、2.2 で述べた問題が発生している。この問題を解決するために、HNS サービス提供者がネットワーク家電のベンダに依存しないインタフェースを準備し、それらと HNS に収容するネットワーク家電を結びつけることによってマルチベンダ HNS を構築する手法を提案する。

図 3 は、提案手法のアーキテクチャを、2.1 のシアターサービスを例として模式図で表したものである。提案手法は、2 つの大きなステップから構成される。

まず、家電の種類ごとに、ネットワーク家電のベンダに依存しない標準家電サービスを準備し、これらを HNS アプリケーションに向けて公開する。HNS アプリケーションは、標準家電サービスを通してネットワーク家電を制御することができる。なお、標準家電サービスのインタフェース定義ならびに構築は、HNS サービス提供者が行う。

次に、標準家電サービスとネットワーク家電との結び付け定義を記述したバインディング定義と、標準家電サービスの各

表 1 HNS 構成要素を準備する主体

	従来手法	提案手法
家電 API	家電ベンダ	家電ベンダ
HNS アプリケーション	家電ベンダ	HNS サービス提供者
標準家電サービス, アダプタ	—	HNS サービス提供者
バインディング定義	—	ユーザ

API が呼び出された際に行うべき家電 API 呼び出し処理を記述した各ネットワーク家電用のアダプタ, さらにこれらを用いて標準家電サービスとネットワーク家電との結び付けを動的に行う, 動的サービスバインディング機構を構築する. 動的サービスバインディング機構は, 標準家電サービスの各 API が呼び出された際にバインディング定義を参照し, そこで結び付け先として指定されたネットワーク家電用のアダプタに記述された処理を解釈し, 家電 API 呼び出し処理を実行する.

表 1 は, 提案手法における HNS の各構成要素を準備する主体について示したものである. 仮定 A1 ならびに仮定 A2 の通り, 各家電 API の仕様はそのネットワーク家電のベンダが決定し, 実装する. これは従来手法と同様である. しかし, HNS アプリケーションは, ネットワーク家電のベンダではなく, サードパーティを含めた HNS サービス提供者が準備する. また, 各標準家電サービス, ならびに各ベンダ製品用のアダプタは HNS サービス提供者が準備し, バインディング定義はユーザが準備することを想定している.

以下に, 提案手法を構成する 2 つのステップについて詳しく述べる.

3.3 標準家電サービスインタフェースの構築

家電 API の仕様はそのベンダによって異なっており, このことが HNS アプリケーションとネットワーク家電が静的に結合する原因となっている. そこで, ベンダを問わず, 標準的な API でネットワーク家電を制御可能な標準家電サービスを家電の種類ごとに構築し, これらを HNS アプリケーションに向けて公開する.

ところが, 家電 API の仕様だけでなく, ネットワーク家電に搭載されている機能もそのベンダによって異なっている. そのため, 各標準家電サービスのインタフェースを定義する前に, 標準家電サービスが対象とするネットワーク家電の機能 (標準的な機能と呼ぶ) を決定する必要がある. なお, この決定は標準家電サービスを準備する HNS サービス提供者の裁量で行うことができるが, どのベンダの製品にも搭載されており, かつ日常的に使用する機能を標準的な機能と定義することが望ましい.

次に, 標準的な機能と呼び出すための API を持った各標準家電サービスのインタフェースを定義する. 図 4 は, 標準テレビサービスのインタフェースの定義例を擬似コードで示したものである. なお, 各標準家電サービスの API 仕様についても, HNS サービス提供者の裁量で決定することができる.

3.4 標準家電サービスとネットワーク家電の動的結び付け

HNS アプリケーションが標準家電サービスを通してネットワーク家電を制御できるようにするためには, 3.3 で定義した標準家電サービスの各 API について家電 API を呼び出す処理

```
TVInterface { // 標準テレビサービスのインタフェース
// 各 API は成功時に 0 を, 失敗時にその他の値を返す.
int on(); // 電源入
int off(); // 電源切
int changeVolume(int vol); // 音量調節 (0:最小, 100:最大)
int mute(); // 消音
int unmute(); // 消音解除
int changeChannel(int ch); // チャンネル変更
// 入力切替 (0:テレビ, 1:HDD レコーダー, ...)
int changeInput(int input);
...
}
```

図 4 標準テレビサービス API の定義例

```
BindingDefinition {
// 標準照明サービスと B 社製照明を結び付ける.
LightService -> Light_B;
// 標準テレビサービスと B 社製テレビを結び付ける.
TVService -> TV_B;
...
}
```

図 5 バインディング定義の例

```
adapter for TV_B { // B 社製テレビ用アダプタ
if TVService.on() is called {
// テレビの電源を入れるために必要な家電 API 呼び出し処理
ret = tv_B.power(true);
return ret;
}
...
}
```

図 6 アダプタの記述例

を実装し, 標準家電サービスとネットワーク家電を結び付ける必要がある. ところが, 標準家電サービスとネットワーク家電を静的に結び付けてしまうと, ネットワーク家電の置き換えが発生する度に標準家電サービスを再構築しなければならなくなり, HNS フレームワークが要求 R2 を満たすことができなくなる. そこで, 標準家電サービスとネットワーク家電との結び付け定義を外部化し, 標準家電サービス実行時に, その定義に従って標準家電サービスとネットワーク家電を動的に結び付ける機構 (動的サービスバインディング機構) を構築する.

まず, 各標準家電サービスとネットワーク家電とのバインディング定義と, 標準家電サービスの各 API が呼び出された際に行うべき家電 API 呼び出し処理を記述した, 各ベンダのネットワーク家電用のアダプタを準備する. 図 5 はバインディング定義の例である. バインディング定義では, 各標準家電サービスについて, 結び付けるネットワーク家電を指定する. また, 図 6 はアダプタの記述例である. 仮定 A3 により, 各家電 API の仕様は公開されているため, HNS サービス提供者は各ベンダのネットワーク家電の API 仕様に合わせたアダプタを準備し, ユーザに提供することができる.

次に, 各標準家電サービスについて, 標準家電サービスとネットワーク家電との結び付けを動的に行うプロキシオブジェクトを構築する. このプロキシオブジェクトは, 標準家電サービスの API が呼び出された際にバインディング定義を参照し, そこで指定されたネットワーク家電用のアダプタを解釈して, 家電 API の呼び出し処理を実行する.

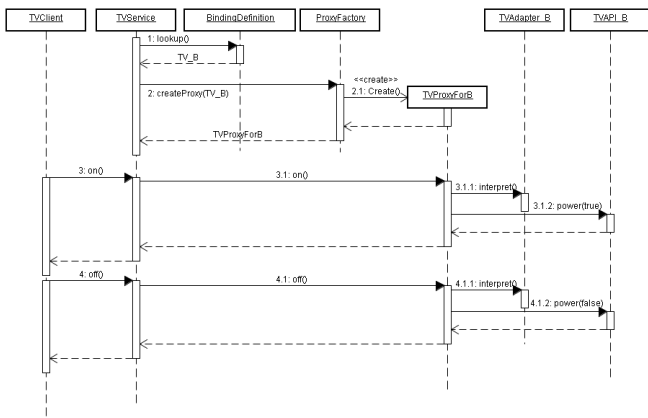


図 7 提案フレームワークのシーケンス図

3.5 提案手法における処理の流れ

HNS アプリケーションが標準家電サービスを用いて B 社製のテレビを制御する際のシーケンス図を図 7 に示す。

まず、標準テレビサービス TVService はバインディング定義 BindingDefinition を参照し、そこで結び付け先として指定されている B 社製テレビ用のプロキシオブジェクト TVProxyForB を ProxyFactory に生成させる。TVProxyForB は、TVService と同様に 3.3 で定義したインタフェースを実装しており、各 API が呼び出された際に B 社製テレビの家電 API を呼び出す処理を行う。

HNS アプリケーション TVClient が標準家電サービスの on() API を呼び出すと、標準家電サービスは TVProxyForB の on() メソッドを呼び出す。TVProxyForB は B 社製テレビ用のアダプタ TVAdapter_B に記述された、電源を入れるために必要な家電 API 呼び出し処理を解釈し、実行する。その結果、B 社製テレビの電源が入る。

4. 実装

提案手法に基づいた HNS フレームワークである Verbena を、Java を用いて実装した。

4.1 標準家電サービスインタフェースの定義

まず、家電の種類ごとに、現在市場に出回っている各ベンダの製品に搭載されている機能の調査を行った。なお、ネットワーク家電の機種がまだ少ないという現状を踏まえ、この調査は従来の家電を対象として行った。

次に、この調査結果から各標準家電サービスが対象とする標準的な機能を抽出し、それらの機能を呼び出すための API を持った各標準家電サービスのインタフェースを定義した。

4.2 動的サービスバインディング機構の構築

3.4 で述べた、各標準家電サービスとネットワーク家電を実行時に結び付ける機構を実装した。

Verbena では、バインディング定義が結び付け先のネットワーク家電用のアダプタを兼ねる仕様とした。そのため、標準家電サービスの各 API が呼び出された際に行うべき処理を、直接バインド定義に JavaScript で記述する。各標準家電サービスクラスは、実行時に各ネットワーク家電用のプロキシオブジェ

クトを生成し、各メソッドが呼び出された際に、そのプロキシオブジェクトの同一名称のメソッドを呼び出す。各ネットワーク家電用のプロキシオブジェクトは、メソッドが呼び出された際にバインディング定義に JavaScript で記述された処理を解釈し、実行する。なお、各ネットワーク家電用プロキシの生成には動的プロキシクラス [9] を、またバインディング定義に記述された JavaScript の解釈・実行には、Java で記述された JavaScript の実装である Rhino [5] を用いた。

さらに、各標準家電サービスと HNS アプリケーションとの相互運用性を確保するために、各標準家電サービスクラスを、Apache AXIS を用いて Web サービス [11] として公開した。

5. 適用実験

実装した Verbena を、実際に各種家電に適用してマルチベンダ HNS を構築する実験を行った。

5.1 実験環境

本実験では、以下の家電を用いて HNS を構築した。

- プラズマディスプレイ: 日本電気 PX-50XM2
- DVD プレーヤー: パイオニア HTZ-535DV
- 照明: 松下電工 HHFZ5310, キシマ KFF-1708, 仮想 UPnP 照明

- カーテン: ナビオ パワートラック
- 空気清浄機: 日立 EP-V12
- サーキュレータ: 森田電工 MCF-257NR

また、家電の他に以下の機器も使用した。

- リモコン: スギヤマエレクトロン クロッサム 2+USB, 玄人志向 KURO-RS
- ホームサーバ: PC/AT 互換機 (Core Solo U1400, メモリ 1.5 GB, HDD 40 GB, Windows XP Professional)

本実験で使用した家電は、仮想 UPnP 照明を除き、ネットワーク接続機能を持たない従来型の家電である。そこで、PC に接続可能な 2 種類のリモコンと、それぞれに付属の制御用ライブラリを用いてこれらの家電 API を実装し、ネットワーク家電と見なして実験を行った。また、Java 用の UPnP コンポーネントである Cyberlink for Java [7] を用いて、PC 上で動作する仮想 UPnP 照明を実装し、同様に Verbena の適用を試みた。

次に、各家電の API 仕様に合わせてバインディング定義を作成し、上記の各家電を HNS に収容させた。

5.2 HNS アプリケーションの実装

各標準家電サービスを用いて、以下の 3 つのサービスを実現する HNS アプリケーションを実装した。

DVD シアターサービス: テレビの入力切り替えと音量調節を行い、カーテンを閉じ、照明を消灯して、DVD を再生する。

空気清浄サービス: 空気清浄機とともに扇風機を動作させ、室内の空気を効率良く浄化する。

おでかけサービス: カーテンを閉じ、全ての家電の電源を切る。

また、実装した HNS アプリケーションを実行し、HNS アプリケーションが標準家電サービスを通して各家電を制御できることを確認した。

表 2 HNS アプリケーションの応答時間

	おでかけ	DVD シアター	空気清浄
標準家電サービス API 数	5	5	2
応答時間: 静的結び付け [ms]	9,709	17,369	6,659
応答時間: 動的結び付け [ms]	9,916	17,522	6,709
応答時間差 [ms]	207	153	50

5.3 家電置き換えへの対応

バイディング定義を変更することで、HNS に収容されている家電の置き換えに対応できることを確認する実験を行った。

本実験では、松下電器製の照明からキシマ製の照明への置き換えを想定し、バイディング定義を変更した上で、5.2 で実装した各 HNS アプリケーションを実行した。その結果、バイディング定義を変更するだけで、HNS や HNS アプリケーションを再構築することなく、キシマ製の照明への置き換えに対応することができた。また、同様にバイディング定義を変更することで、仮想 UPnP 照明への置き換えに対応することも確認した。

6. 考 察

6.1 要求の満足度

Verbena は、ネットワーク家電のベンダに依存しない標準家電サービスと、HNS に収容するネットワーク家電を実行時に結び付けることによって、HNS アプリケーションが標準家電サービスを通してネットワーク家電を制御できるようにしている。そのため、任意のベンダ製のネットワーク家電を HNS に収容することができる。従って、Verbena は要求 R1 を満たす。

また、Verbena では HNS に収容されているネットワーク家電の置き換えが発生した場合でもバイディング定義を変更するだけで対応でき、その家庭の HNS を再構築する必要はない。加えて、ネットワーク家電の置き換えが発生しても標準家電サービスの API 仕様は一切変更されないため、既にユーザが使用している HNS アプリケーションを再構築する必要もない。従って、Verbena は要求 R2 を満たす。

さらに、各標準家電サービスの API 仕様は、標準家電サービスを準備する HNS サービス提供者の裁量で決定することができる。そのため、ネットワーク家電のベンダのみならず、サードパーティの HNS サービス提供者も、HNS アプリケーションを構築し、ユーザに各種 HNS サービスを提供することができる。従って、Verbena は要求 R3 を満たす。

6.2 性 能

我々は、標準家電サービスとネットワーク家電の結び付けを動的に行う際のオーバーヘッドを明確にするために、Verbena の他に標準家電サービスとネットワーク家電を静的に結び付けた HNS を準備し、2 つの環境下で、5.2 で実装した HNS アプリケーションの応答時間を測定した。表 2 はその結果である。最上段は HNS アプリケーションが呼び出す各標準家電サービスの API の数である。また、応答時間、ならびに応答時間差の単位は全てミリ秒である。

表 2 を見ると、標準家電サービスとネットワーク家電の結び付けを動的に行った場合の応答時間と、静的に結び付けた場合の応答時間との差、すなわち標準家電サービスとネットワーク家電の結び付けを動的に行う際のオーバーヘッドは、最大のおでかけサービス実行時でも 200 ミリ秒強に留まっている。一般的に、HNS アプリケーションに高い実行性能を求められることは少ないため、このオーバーヘッドは大きな問題とはならないと考えられる。

7. おわりに

本稿では、大半の HNS アプリケーションが単一ベンダ製のネットワーク家電にしか対応していないという、現在の HNS の問題点を指摘した。そして、この問題を解決するために、提案するマルチベンダ HNS の構築手法が満たすべき 3 つの要求を挙げた。これらの要求を満たすために、ネットワーク家電のベンダに依存しない標準家電サービスを構築し、標準家電サービスとネットワーク家電の結び付けを動的に行う手法を提案した。また、提案手法に基づいた HNS フレームワークである Verbena を実装するとともに、Verbena と複数のベンダの家電を用いてマルチベンダ HNS を構築し、提案手法の実用性の検証を行った。

今後の研究課題として、標準家電サービスが対象としていない機能を HNS アプリケーションから呼び出すための仕組みの提案などが挙げられる。

謝 辞

この研究は、科学技術研究費(若手研究 B 18700062, 基盤研究 B 17300007), および日本学術振興会日仏交流促進事業(SAKURA プログラム)の助成を受けて行われている。

文 献

- [1] Digital Living Network Alliance, <http://www.dlna.org/>
- [2] ECHONET Consortium, <http://www.echonet.gr.jp/>
- [3] H. Igaki, M. Nakamura, and K. Matsumoto, "A service-oriented framework for networked appliances to achieve appliance interoperability and evolution in home network system" (short paper), *Proc. of Int'l Workshop on Principles of Software Evolution (IWPSE 2005)*, pp.61-64, Lisbon, Portugal, Sep. 2005.
- [4] HomePlug Powerline Alliance, <http://www.homeplug.org/>
- [5] Mozilla Foundation, "Rhino," <http://developer.mozilla.org/ja/docs/Rhino>
- [6] OSGi Alliance, <http://www.osgi.org/>
- [7] S. Konno, "CyberLink for Java," <http://www.cybergarage.org/net/upnp/java/>
- [8] Smart Home Systems, Inc., "The X-10 Story," <http://www.smarthomeusa.com/info/x10story/>
- [9] Sun Microsystems, Inc., "ダイナミックプロキシクラス," <http://java.sun.com/j2se/1.5.0/ja/docs/ja/guide/reflection/proxy.html>
- [10] UPnP Forum, <http://www.upnp.org/>
- [11] W3C Web Services Activity, <http://www.w3.org/2002/ws/>
- [12] 株式会社東芝, "東芝ネットワーク家電 フェミニティ," <http://www3.toshiba.co.jp/femininity/>
- [13] 松下電工株式会社, "ライフィニティ," <http://biz.national.jp/Ebox/kahs/>
- [14] 三菱電機株式会社, "三菱ルームエアコン みまもりさーバー," http://www.mitsubishielectric.co.jp/home/kirigamine/it_b.html