

2007 年度  
卒業論文

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

神戸大学工学部情報知能工学科

福田 将之

指導教員 中村匡秀 准教授  
井垣宏 助手

---

2008 年 2 月 26 日



# 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

福田 将之

## 要旨

本論文では，HNS を利用した新たなサービスとしてリアルタイム家電制御サービスを考え，そのシステムを作成するための手法を提案する．リアルタイム家電制御サービスとは，元来制御工学の分野で使用されているリアルタイム制御の概念を HNS に適用し，HNS の家電機器を指定された時刻に動作させるサービスである．提案手法として，まず始めに，既存の HNS をリアルタイム制御するためのリアルタイム実行基盤の開発を行う．この実行基盤では，家電操作とその実行時刻のペアを「タスク」として管理する．そのためのキーアイデアとして，マルチスレッドを用いたタスクの管理法を紹介する．次に，エンドユーザがタスク登録を容易に行えるように HNS リアルタイムサービス作成 GUI を提案する．本論文では，提案手法に基づき，我々のグループで開発している実際の HNS においてリアルタイム家電制御サービスを実現するシステムの実装を行い，その有用性の確認を行った．

キーワード ホームネットワークシステム，生活リズム，リアルタイム制御，  
Web サービス，AIR，マルチスレッド

# 目次

第 1 章	はじめに	1
第 2 章	準備	4
2.1	研究に関連する既存技術 . . . . .	4
2.1.1	ホームネットワークシステム (HNS) . . . . .	4
2.1.2	WEB サービス . . . . .	4
2.1.3	AIR アプリケーション . . . . .	6
2.1.4	リアルタイム制御 . . . . .	6
2.1.5	マルチスレッド . . . . .	6
2.2	HNS における家電機器のリアルタイム制御 . . . . .	7
2.3	先行研究：サービス指向アーキテクチャに基づく従来家電を 対象としたホームネットワーク適応手段の設計と評価 . . . . .	7
第 3 章	提案手法	9
3.1	要求 . . . . .	9
3.2	システム概要 . . . . .	10
3.3	HNS リアルタイム実行基盤 . . . . .	11
3.3.1	サービス定義 . . . . .	11
3.3.2	マルチスレッド機構を用いたタスク監視と API 実行 . . . . .	13
3.3.3	Web サービスによる実行基盤の機能公開 . . . . .	13
3.3.4	HNS リアルタイムサービス作成 GUI . . . . .	15
第 4 章	実装	16
4.1	使用した HNS . . . . .	16

4.2	HNS リアルタイム実行基盤の実装 . . . . .	16
4.3	HNS リアルタイムサービス作成 GUI . . . . .	17
4.3.1	機能 . . . . .	17
4.3.2	操作 . . . . .	18
4.4	HNS リアルタイムサービス実行 GUI . . . . .	20
4.4.1	機能 . . . . .	20
4.4.2	操作 . . . . .	21
<b>第 5 章 評価</b>		<b>23</b>
5.1	評価実験 . . . . .	23
5.2	HNS リアルタイム家電制御サービスの例 . . . . .	24
5.2.1	おはようサービス . . . . .	24
5.2.2	おやすみサービス . . . . .	24
5.2.3	お出かけサービス . . . . .	25
5.2.4	空き巣防止サービス . . . . .	25
5.2.5	快適帰宅サービス . . . . .	25
5.2.6	照明連動サービス . . . . .	25
<b>第 6 章 まとめ</b>		<b>26</b>
謝辞		<b>27</b>
参考文献		<b>28</b>

## 目次

2.1	ホームネットワークシステム (HNS) . . . . .	5
2.2	NAIST-HNS . . . . .	8
3.1	提案システム . . . . .	10
3.2	サービス定義の例 . . . . .	12
4.1	HNS リアルタイムサービス作成 GUI . . . . .	17
4.2	日付指定方式 . . . . .	19
4.3	時間指定方式 . . . . .	20
4.4	HNS リアルタイムサービス実行 GUI . . . . .	21
4.5	ファイル読み込み完了 . . . . .	22

# 表目次

1.1	ある会社員 (独身) の一日 . . . . .	2
3.1	サービス定義の形式 . . . . .	12
3.2	図 3.2 の概要 . . . . .	12
3.3	リアルタイム家電制御サービス . . . . .	14
3.4	TimerBean クラスのフィールド . . . . .	14
5.1	実験で作成したタスク . . . . .	23

## 第 1 章

# はじめに

宅内の家電機器やセンサーをネットワークに接続し、より便利な快適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)[1][2] の研究・開発が盛んである。HNS では、TV や DVD プレーヤー、エアコン、扇風機、カーテン、照明といった家電（ネットワーク家電と呼ばれる）や、温度計・照度計といったセンサを、宅内外から遠隔制御・監視したり、連携制御することで、付加価値の高いサービスを実現している。

HNS を利用したアプリケーション・サービスには、携帯電話等を用いて宅外から自宅の家電機器を操作できる遠隔制御サービス [3] や、機器の状態を監視して通知する機器状態監視・見守りサービス [4]、複数の家電を連携制御して便利な付加価値サービスを実現する家電連携サービス [5] などが存在する。

本研究では、HNS の新たなアプリケーションとして、リアルタイムな家電制御サービスを実現することを考える。元来リアルタイム制御とは、システムが行う各処理に期限（デッドライン）を設定し、各処理を期限内に終わるように制御を行う方式の総称である。自動車や航空機、医療機器、産業用ロボット、プラント制御などに代表的に用いられている [7][8]。我々のねらいは、リアルタイム制御の概念を HNS 内の家電制御に適用して、より便利で付加価値の高いサービス・アプリケーションを実現することにある。

家庭内のユーザの日常生活は、一般に生活リズムとよばれる周期的な行動パターンに基づいて行われている。また、その行動パターンに基づいて、様々な家電も利用されている。例として、以下にあるユーザの平日の典型的な行動パターンを表 1.1 に示す。

Table 1.1. ある会社員 (独身) の一日

---

6:00	起床．目覚まし時計を止める．
6:10	カーテンを開ける．テレビのニュースをつける．
6:15	洗面所にて洗面，歯磨きをする．
6:20	コーヒーを沸かし，トースターでパンを焼く．その間服を着替える．
6:40	朝食を食べる．
6:55	テレビを消す．電気がすべて消えているか確認．カーテンを閉める．
7:00	戸締り．出勤．
	会社
19:00	帰宅．電気をつける．
19:10	洗面所で手洗い，うがい．服を着替える．
19:20	テレビをつける．電子レンジをつけて弁当を温める．
19:30	夕食．
20:00	風呂に入る．
	:

---

6:10 にカーテンを開け，テレビの電源をいれて，6:20 にコーヒーメーカーの電源をいれ，トースターを余熱する，等の操作を登録，HNS からリアルタイム制御することで，ユーザが操作する手間を省ける．ただし，こうした生活リズムはユーザごとに異なるため，サービス実現においては，ユーザが好きな時刻に，好きな家電操作を手軽に登録できることが望ましい．

本研究の目的は，家庭内の HNS ユーザが，自らの生活リズムに合わせたこうしたユーザの行動パターンが事前にわかっており，おおよその行動時刻が決まっているのであれば，家電を操作する時刻をあらかじめ HNS に登録しておき，その時刻に家電をリアルタイム制御すれば，そのユーザの生活を効率的にサポートできる．上記の例では，6:10 にテレビをつけ，カーテンを開ける，6:20 リアルタイム家電制御サービスを，手軽に実現するための枠組みを提案することである．提案法では，まず既存の HNS をリアルタイム制御するための HNS リアルタイム実行基盤の開発を行う．実行基盤は，家電操作とその実行時刻のペアを「タスク」として管理する．タスクが登録されると，実行基盤

は現在時刻からの差分を計算し実行時刻まで待機する．指定時刻に達すると，実行基盤は HNS に対して家電 API の実行を要求し，目的の家電操作が実行される．複数のタスクを並行して監視するため，実行基盤はスレッド技術に基づいて実現される．

さらに提案手法では，エンドユーザが容易にタスク登録できるようにするための HNS リアルタイムサービス作成 GUI を提案する．ユーザは家電のアイコンをマウスで選択し好みの時刻を選択することで，自分の生活リズムに合わせたリアルタイム家電制御サービスを容易に定義できる．GUI で定義されたサービスは，実行基盤にタスクとして登録され，指定された時刻にタスクが実行されることになる．

提案手法に基づき，我々のグループで開発している実際の HNS において，リアルタイム家電制御サービスを実現するシステムの実装を行った．実行基盤は Java Web サービスを，また GUI は AIR を用いて実装した．また，このシステムを用いていくつかのリアルタイム家電制御サービスを開発し，その有効性を確認した．

本論文の以降の構成は次のとおりである．まず 2 章では，必要な概念の定義や先行研究，現状の課題について述べる．3 章では，要求定義を行った後提案手法を述べる．4 章では提案手法に基づく実装，ケーススタディについて述べる．5 章では，提案法の評価を行う．最後に 6 章で，今後の課題とともにまとめを述べる．

## 第 2 章

# 準備

### 2.1 研究に関連する既存技術

#### 2.1.1 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、宅内にある複数のネットワークに接続された家電、ネットワーク家電から構成される。図 2.1 は HNS を表した図である。ネットワーク家電は、ユーザや外部エージェントがネットワーク越しに制御できるように、制御 API を備えている。この API 呼び出しを実行するため、ネットワーク家電はプロセッサ及びストレージを持つことが一般的である。また、HNS の代表的な機能として、連携サービス [5] が挙げられる。連携サービスとは、複数のネットワーク家電を同時、または逐次制御することで、ユーザの目的を最小限の操作で実現するサービスである。

例えば、ユーザが DVD を視聴したい、という目的を持っているとする。この目的を達成する際、HNS が無い場合では TV、DVD プレイヤー、音響家電機器、照明といった個々の家電機器を操作する必要がある。しかし、HNS を用いて連携サービスを実行することで、これらの操作を DVD シアターサービスとして、簡単に実現することができる。

#### 2.1.2 WEB サービス

WEB サービスとは、ソフトウェアの機能を WWW 関連の技術を使い、ネットワークを通じて利用できるようにしたものであり、インターネット上に分散するアプリケーションの提供する機能（サービス）を、クライアントのアプリケーションにプログラミング可能なインターフェイスとして提供してい

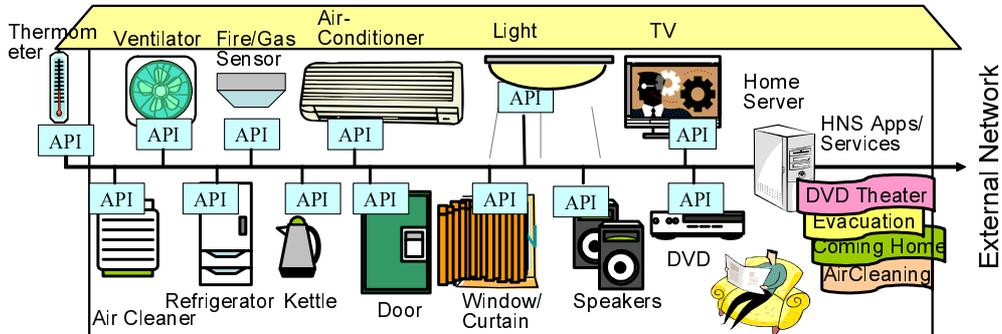


Fig. 2.1. ホームネットワークシステム (HNS)

る．WEB サービスを支える WWW 関連技術の一つとして，WSDL がある．WSDL とは，WEB サービスのインターフェイスとバインディング情報の定義を記述する XML 形式の言語である．この WSDL を参照することによって，サービスの利用方法が取得できる．サービスの利用は，クライアントアプリケーションから XML-RPC および SOAP といったリモートプロシージャコール (RPC) で行われる．アプリケーションを作成するための開発環境も，RPC に対応してさえいれば，他は何の制限も無く，また RPC の一つである SOAP は，多くのプログラミング言語で専用の API を導入することで利用できる．ここで，Web サービスに対してより容易にアクセスする方法として REST (*Representational State Transfer*) が存在する．REST では URL を指定するだけで Web サービスの呼び出しを行えるため，複雑なデータをやりとりしないサービスに用いられる．

### 2.1.3 AIR アプリケーション

AIR(Adobe Integrated Runtime) とは,Adobe Systems 社が提供しているデスクトップアプリケーションの実行環境のことで,Flash の技術をベースとしている [9] . Flash との違いは,Web ブラウザに束縛されず,ウィンドウの生成と操作,ファイルの読み書き,ドラッグ&ドロップといったローカルマシンの機能を利用できる点である . OS に依存しないデスクトップアプリケーションの実行環境としては AIR ランタイムの他に,Microsoft 社の「Common Language Runtime(CLR)」や,Sun Microsystems 社の「Java Virtual Machine(JVM)」がある . しかし,CLR は Windows 以外の対応が鈍く,JVM は実行環境のインストールがユーザにとって敷居が高いという欠点がある . それに対し,AIR ランタイムは初めから Windows,Mac に対応しており (Linux にも対応予定), 実行環境のインストールも,Web ページのクリック一つで AIR ランタイムと AIR アプリケーションを同時にインストールできるため容易である [10] .

### 2.1.4 リアルタイム制御

外部からの要求に対して,それぞれの要求時間内に要求仕様の仕事を終了させることが要求される場合に使用される方式のことを指す [11] . 元来ロボット産業,宇宙船,航空機,医療機器などのシビアな要求がなされる制御工学の分野で用いられている . 今回の場合,家電機器を対象としているため,ある家電のある動作をこの時間に行って欲しい,という事を要求仕様とする .

### 2.1.5 マルチスレッド

1 つのアプリケーションソフトがスレッドと呼ばれる処理単位を複数生成し,平行して複数の処理を行うこと . CPU の処理時間を非常に短い単位に分割し,複数のスレッドに順番に割り当てることによって,複数の処理を同時に行っているようにみせている [12] .

## 2.2 HNS における家電機器のリアルタイム制御

現在，様々な家電機器が開発され世に出回っている．これら家電機器の動作は利用者の生活リズムと深く関連性がある．例えば，朝起きればカーテンを開けて，顔を洗い，服を着替えて，ご飯の支度をする．ここで使用されている家電機器はカーテン，炊飯器，トースター，コーヒーメーカーなどである．会社/学校から帰宅すれば，照明をつけて，お風呂を沸かし，テレビ/ラジオをつけて，ご飯の支度をする．ここで使用されている家電機器は照明，湯沸かし器，テレビ，ラジオ，炊飯器などである．これら家電機器の動作はほぼ毎日変わる事無く繰り返されており，それらをパターン化することが出来る．そのパターン化された機器の動作を事前に予約といった形で登録することが出来れば，毎回の動作の手間が省かれ，時間に余裕が出来る．

HNS を利用したサービスには様々な種類のものがある．しかしながら，HNS 内における家電機器をユーザが自由にリアルタイム制御するサービスは提供されておらず，ユーザが自由に作成できる枠組みも存在しない．

## 2.3 先行研究：サービス指向アーキテクチャに基づく従来家電を対象としたホームネットワーク適応手段の設計と評価

HNS は，様々な家電機器やサービスをシームレスにネットワークに接続し，ユーザの多様化するニーズやライフスタイルに適合するためのシステムとして注目を集めているが，広くは普及していない．そこで，田中らはその原因をネットワーク家電の価格，対応機種限定，アプリケーションレベルの相互接続性の 3 つにあると考え，これらの問題を解決するための HNS フレームワーク (NAIST-HNS) を作成している．そのためのアプローチとして，ネットワークインターフェイスやプロセッサを持たない従来家電を HNS に適応させること，またベンダの異なる複数の家電を単一の HNS に編入させることを挙げている．

具体的にまず，汎用的な赤外線信号を送出可能な Ir-API の構築を行い，その上に家電の機能をサービスとして一般化するサービス層を構築する．これらのサービスを，様々な環境から利用できるように Web サービスとして公開する．

PC に接続可能な家電機器リモコンとして、クロッサム [13] を採用している。サービス層の実装には、Java を用いて、各家電機器ごとにクラスを作成し、家電機器が提供する機能をサービスメソッドとして実装を行っている。最後に、サービス層の各クラスを、Java の SOAP 実装の 1 つである Apache AXIS を用いて Web サービスとして公開している。各家電機器の操作は SOAP-RPC 又は REST で呼び出し可能となっている。

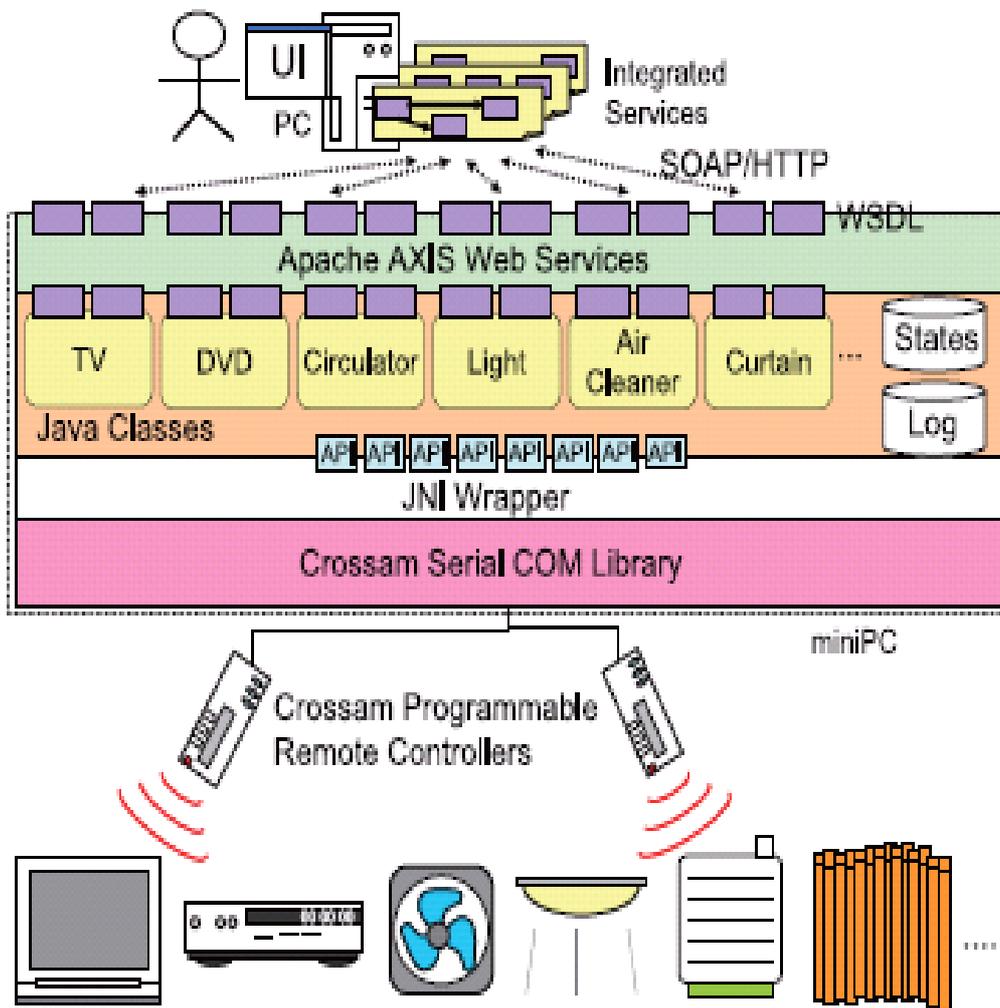


Fig. 2.2. NAIST-HNS

## 第 3 章

# 提案手法

### 3.1 要求

本研究では、リアルタイム家電制御サービスの実現のため、HNS において任意の家電操作を指定した時刻に実行する枠組みを提案することを目的とする。まず、本研究の前提として、既存の HNS が利用できるものとする。

前提 A1: 既存の HNS が利用できるものとする。各家電の機能は、API ( 機器, メソッド, パラメータ ) を通してネットワーク越しに利用できるものとする。

リアルタイム家電制御サービスにおいては、家電操作 ( すなわち API 実行 ) とそれを実行する時刻とが、サービスの基本要素となる。したがって、時刻と家電 API の組を、タスクと呼ぶことにする。リアルタイム家電制御サービスは、複数のタスクの集合で定義される。

これを踏まえて、本論文では以下の要求を満たすシステムをの提案を行う。

要求 R1: システムは、与えられたタスクに基づいて、指定された時刻に家電制御を行う。ここで、システムは複数のタスクを一度に処理できること。

要求 R2: HNS のユーザが自らの生活に合わせたタスクを、システムに対して自由に登録、キャンセルできること。

要求 R1 は、リアルタイム制御の基本的な要求を挙げている。既存の HNS や機器にタイマー機能やスケジュール機能が備わっていなくても、提案システムによって、それらをリアルタイム制御することが必要となる。タスクは、複

数登録されることがあるから，登録されたタスクを並行に監視して，当該時刻にあわせて API を実行する仕組みが必要となる．

一方，要求 R2 は，ユーザが手軽に自分の生活リズムに合わせたタスクを登録できることを述べている．生活リズムはユーザ個人ごとに異なるため，ユーザはなるべく手間のかからない方法で，容易にタスク登録・削除ができることが望ましい．

### 3.2 システム概要

図 3.1 に提案システムの概要を示す．

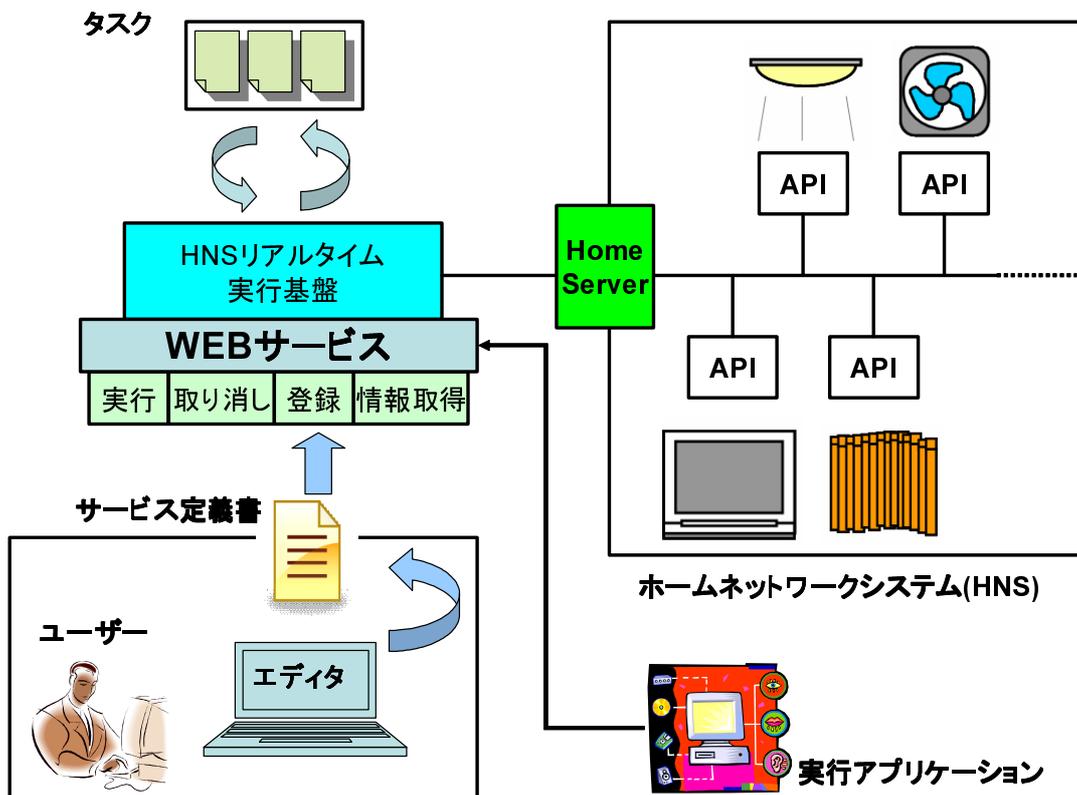


Fig. 3.1. 提案システム

提案システムは，主に 2 つのサブシステムから構成される．まず，HNS リアルタイム実行基盤は，登録されたサービス（すなわち，タスクの集合）を監視し，指定された時刻に HNS のホームサーバ経由で家電 API を呼び出す．ただし，複数のサービスを扱うために，マルチスレッドを用いて実現すること

で、要求 R1 を満たす。このサービス実行基盤は、任意の外部アプリケーションから利用できるように、タスクの登録、キャンセル、実行、ステータス取得といった機能を Web サービスとして公開している。

次に、要求 R2 を満たすために、システムは、HNS リアルタイムサービス作成 GUI を含む。サービス作成 GUI は、ユーザが直感的にタスクを登録できるためのユーザインターフェースである。簡単なマウス操作により、任意の家電操作を、タイムラインに割り当てていくことができる。定義したタスクは、サービス定義ファイルというテキストファイルに保存され、リアルタイム実行基盤の登録 Web サービスを通して、システムに登録される。

なお、リアルタイム実行基盤とサービス作成 GUI は、Web サービスを介して疎結合されている。したがって、実行基盤は、GUI 以外の他のアプリケーションからも実行することができる。

### 3.3 HNS リアルタイム実行基盤

HNS リアルタイム実行基盤は、複数のタスクが書かれたサービス定義を受け取ると、その内容を読み込み、マルチスレッド機構を用いてタスクの監視を行う。

#### 3.3.1 サービス定義

まず、HNS リアルタイム実行基盤で使用するサービス定義について説明する。サービス定義は、複数のタスクから構成される。各タスクは、HNS における対象機器（機器サービス）名、その機器における操作（メソッド）名、メソッドに渡す（パラメータ）名、操作を実行する時刻の 4 要素から構成される。提案システムでは、このサービス定義をある規則に則って書かれたテキストファイルとする。その概要を表 3.1 に示す。なお、タスクが複数ある場合は、「コンマ (,)」で区切って記述する。機器サービス名とは、HNS の機器サービス名の事である。メソッドとは、機器サービスが行うことができる動作のことであり、パラメータ値とは、メソッドの内さらに指定が必要なものの値のことであり、例としてテレビの音量やチャンネルなどが挙げられる。特に指定が無い場合は「ハイフン (-)」と記述する。時間の行に関しては、ユーザの利便性を

Table 3.1. サービス定義の形式

1 行目	機器サービス名
2 行目	メソッド名
3 行目	パラメータ値
4 行目	時間

考え、日付指定方式と時間指定方式の 2 者を採用した。

日付指定方式は、タスクをいつ実行して欲しいかを指定する方式であるのに対し、時間指定方式は、サービスが実行されてから何秒後にタスクを実行するのかを指定する方式である。なお、日付指定方式は、年月日時分秒の 6 つの要素から成り立っており、これらを「スラッシュ (/)」で区切って記述する。

以下にサービス定義の一例を示す。

```
TVService,FLOOR_LIGHT2Service,FANService,CURTAINService
on,setBrightness,on,open
-,5,-,-
20,2008/2/14/18/30/00,15,2008/3/25/4/14/25
```

Fig. 3.2. サービス定義の例

このサービス定義は表 3.2 に示すような以下の 4 つのタスクから成り立っている。

Table 3.2. 図 3.2 の概要

機器名	メソッド名	パラメータ値	時間
TV	on	-	20 秒後
FLOOR-LIGHT2	setBrightness	5	2008 年 2 月 14 日 18 時 30 分 00 秒
FAN	on	-	15 秒後
CURTAIN	open	-	2008 年 3 月 25 日 4 時 14 分 15 秒

### 3.3.2 マルチスレッド機構を用いたタスク監視と API 実行

複数のタスクを処理するために、サービスを構成するタスクひとつひとつに対して新たなスレッドを作成する。HNS リアルタイム実行基盤は、複数のタスクが書かれたサービス定義を受け取ると、その内容を読み込み、書かれているタスク一つに対してスレッドを一つ作成する。そして、作成された複数のスレッドを 2.1.5 で述べたマルチスレッドの技術を用いて処理する。

サービスが実行されると、各スレッドは実行時間が来るまで、もしくは実行フラグが有効な間は待機している。実行フラグとは、そのスレッドが実行待ち状態の間は有効となっているフラグのことで、開始待ち状態または実行終了後は無効となっている。ここで、時間指定方式の場合は時間を待機時間にそのまま利用できるが、日付指定方式の場合は出来ない。そこで、サービスが実行されたときに現在時刻を取得し、実行時間までの時間を計算しそれを待機時間とする。

実行時間が来ると待機をやめ、ホームサーバに対して、当該タスクの家電 API の実行を要求する。

### 3.3.3 Web サービスによる実行基盤の機能公開

次に、HNS リアルタイム実行基盤を Java を用いて Web サービス化する。以下にこのサービスが持つメソッドを示す。

Table 3.3. リアルタイム家電制御サービス

型名	メソッド名	機能
boolean	Init(String)	引数で指定されたファイルを読み込み，サーバーにタスクの登録を行う
boolean	exeTimer()	サービスの実行
TimerBean[]	timerList()	サーバーに登録されている全てのタスクを値として返す
TimerBean[]	waitList()	実行待ち状態のタスクを値として返す
boolean	cancel(int)	(引数で指定された番号 +1) 番目に読み込まれたタスクのキャンセルを行う

型名についてであるが，boolean となっているものに関しては，メソッドが正常に機能すれば true を，正常に機能しなければ false を返す．また，TimerBean[] は独自に定義した TimerBean 型の配列である．なお，TimerBean 型とはタスクが登録された後 (スレッド) の型であり，フィールドとして以下の項目を持つ．

Table 3.4. TimerBean クラスのフィールド

型名	フィールド名	意味
String	app	機器サービス名
String	met	メソッド名
String	param	パラメータ値
String	time	タスクの実行時間
int	rest-time	現時刻からタスクの実行時間までの時間
boolean	flag	このタスクが実行待ち状態であるかどうか

これらのメソッドを用いて，外部から以下のことが可能となる．

タスクの登録： ユーザは，サービス定義が記述されたテキストファイルを用意し，引数にそのテキストファイル名を指定し，Init メソッドを実行することで登録できる．

タスクの実行： タスクの登録を行った後，`exeTimer` メソッドを実行することで登録したタスクの実行を行うことができる．

タスクのキャンセル： 実行開始後に (キャンセルしたいタスクの登録番号-1) を引数に指定し，`cancel` メソッドを実行することでそのタスクはキャンセルされる．

情報取得： 現在登録されているサービス (タスクの) 内容を知りたい場合には，このメソッドを実行する．値として返ってくる `TimerBean` 配列は表 3.4 で示すようなフィールドを持つ．

### 3.3.4 HNS リアルタイムサービス作成 GUI

前節で述べたサービス定義は，テキストエディタを使って作成することも出来る．しかしながら，ユーザーが一つ一つのタスクをより分かりやすく，間違いの無いように作成することは難しい．

そこで，サービス定義を GUI を用いてグラフィカルに行う，HNS リアルタイムサービス作成 GUI を構築する．この GUI は，操作したい機器をアイコンで選び，その機器の操作メソッドを選んで，時間軸上にグラフィカルに配置することで，タスクの定義を行う．ユーザは，自分の意図する操作を時間軸に置くという直感的かつシンプルな操作を繰り返すことで，サービス定義を行うことができる．

作成されたサービス定義は，前節で述べたテキストファイル形式で保存され，登録 Web サービスを通して実行基盤にアップロードされる．

## 第 4 章

# 実装

提案手法に基づいて、システムの実装を行った。

### 4.1 使用した HNS

田中らの方法 [14] に基づいて、我々の実験室に実験用 HNS(CS27-HNS) を構築した。この HNS は、以下の機器から構成されており、各家電は REST 形式の Web サービスで操作可能となっている。

- テレビ：Panasonic
- 扇風機：Pieria メタルリビング扇風機
- フロアライト：オーブス
- テーブルライト：オーブス
- 電動カーテン：Resite
- 空気清浄機：PA-QC13-WX（ナチュラルホワイト）エアブリーズ

### 4.2 HNS リアルタイム実行基盤の実装

提案する HNS リアルタイム実行基盤を、以下の技術を用いて実装した。

- PC：960MB RAM，2.00 GHz，WinXP Pro
- Windows XP SP 2
- Java JDK 5
- Flex SDK 3

- Apache AXIS 2
- Tomcat5.5

### 4.3 HNS リアルタイムサービス作成 GUI

HNS リアルタイムサービス GUI は、Adobe AIR を用いて実装した。図 4.1 に HNS リアルタイムサービス作成 GUI の画面のレイアウトを示す。

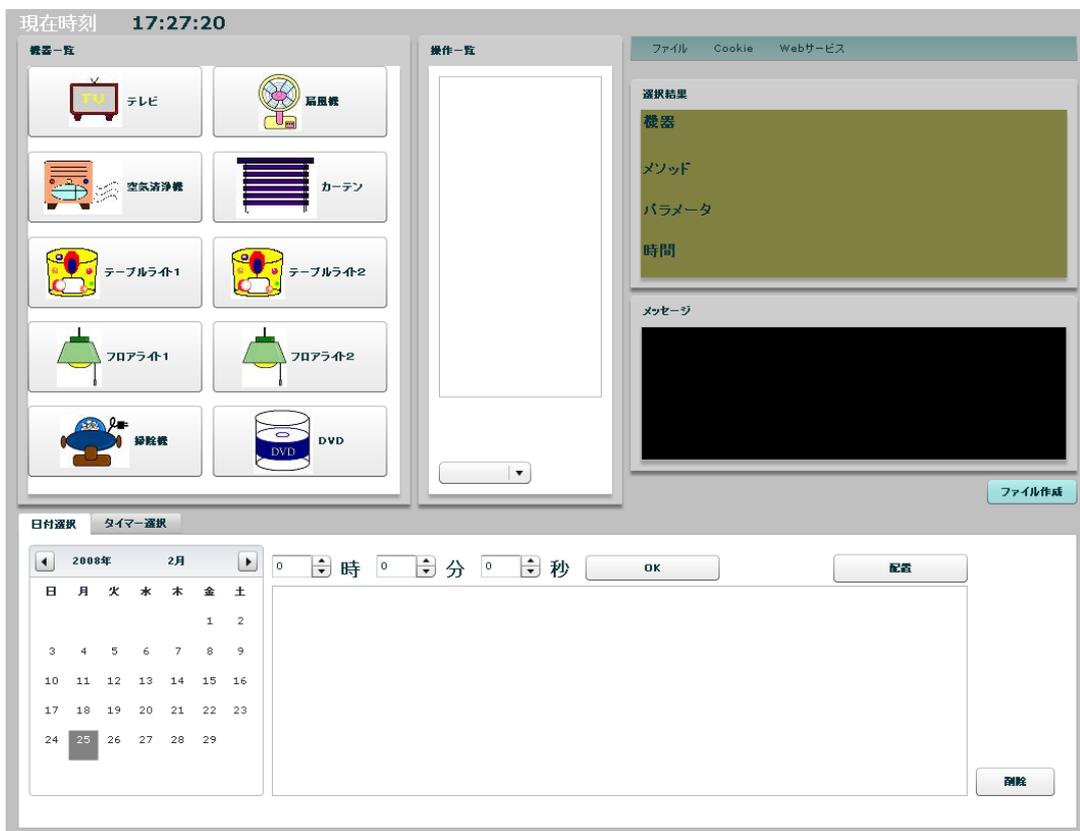


Fig. 4.1. HNS リアルタイムサービス作成 GUI

#### 4.3.1 機能

主な機能を以下に挙げる。

- ファイルの作成
- ファイルの編集
- 履歴の表示

### 4.3.2 操作

#### タスクの作成

1. 機器を選択 .
2. メソッドを選択 .
3. 必要に応じてパラメータを選択 .
4. 時間を選択 .

これでタスクを作成する事が出来る .

#### 時間の選択

##### 日付指定方式

日付指定方式の場合 , カレンダーの日付をクリックし , 動作させたい時間を入力し「OK」をクリックする . その後「配置」をクリックすると完了となり , タスクが生成される . (図 4.2 参照)

##### 時間指定方式

時間指定方式の場合 , 表になっているのでタスクを実行させたい時間欄をダブルクリックすればタスクが生成されるされ , 表に配置される . (図 4.3 参照) ただし , この時間欄の表示単位は秒となっており直感的に分かりにくいと感じたので , 選択された時間の単位を時間/分/秒の単位へと変換した後の値を表示するようにした . この時間欄は「前へ/次へ」のボタンを押すことで , 選択できる値の変更を行うことが出来る . また , 「時間/分/秒」を選択することで , 値の更新を選択した単位時間区切りで変更できる . 例えば , 現在選択できる時間の範囲が 31 ~ 40 の場合で , 「分」を選択し「次へ」を押すと , 選択範囲が 61 ~ 70 へと変更され , 「時間」を選択し「次へ」を押すと 3601 ~ 3610 へと変更される .

#### ファイルの読み書き

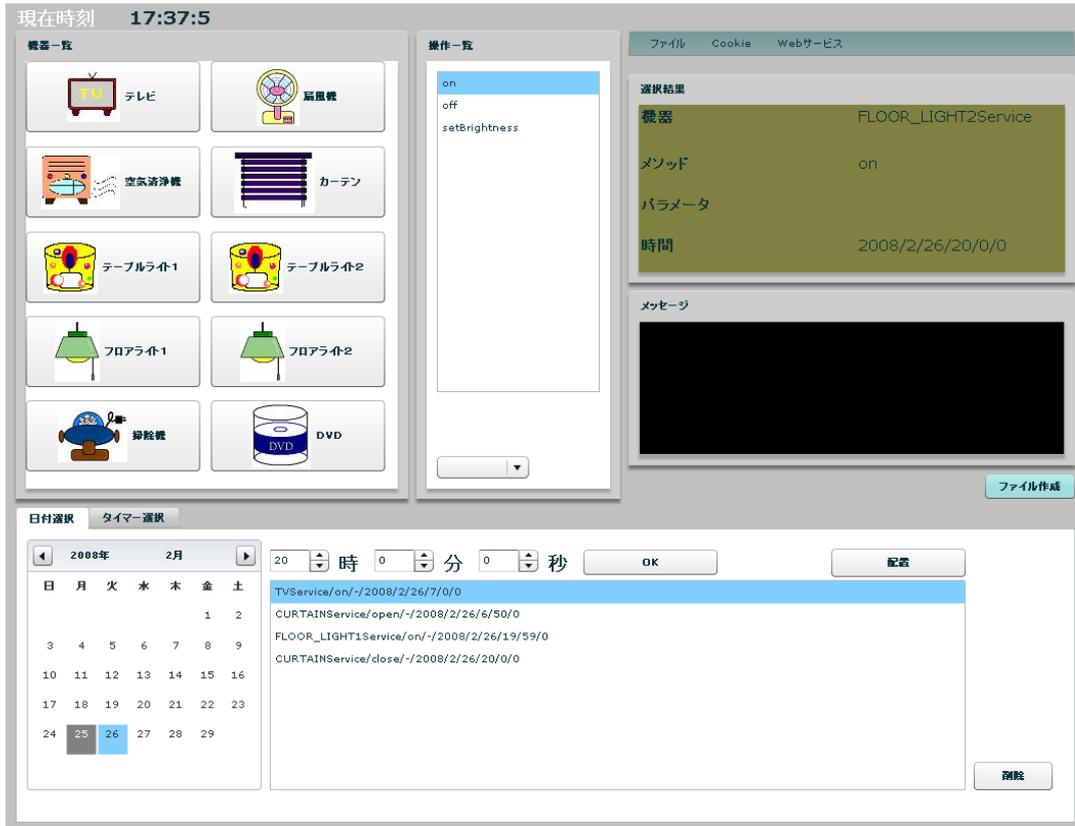


Fig. 4.2. 日付指定方式

「ファイル作成」のボタンをクリックすると、現在編集中的の内容がユーザが指定したファイルへと書き込まれ、サービス定義書が作成される。このアプリケーションは一度作成したファイルを読み込んで編集する事も可能であり、メニューバーより「ファイル ファイルを開く」の順で選択し、編集したいファイルを指定するとその内容が表と日付指定方式の結果一覧に反映される。

### 履歴の表示

メニューバーより「Cookie」を選択し、「保存」を押すと、現時点のデータが取り込まれ、「表示」を押すと前に保存したデータが表示される。履歴を消去したい場合は、「削除」を押すと履歴が消去される。

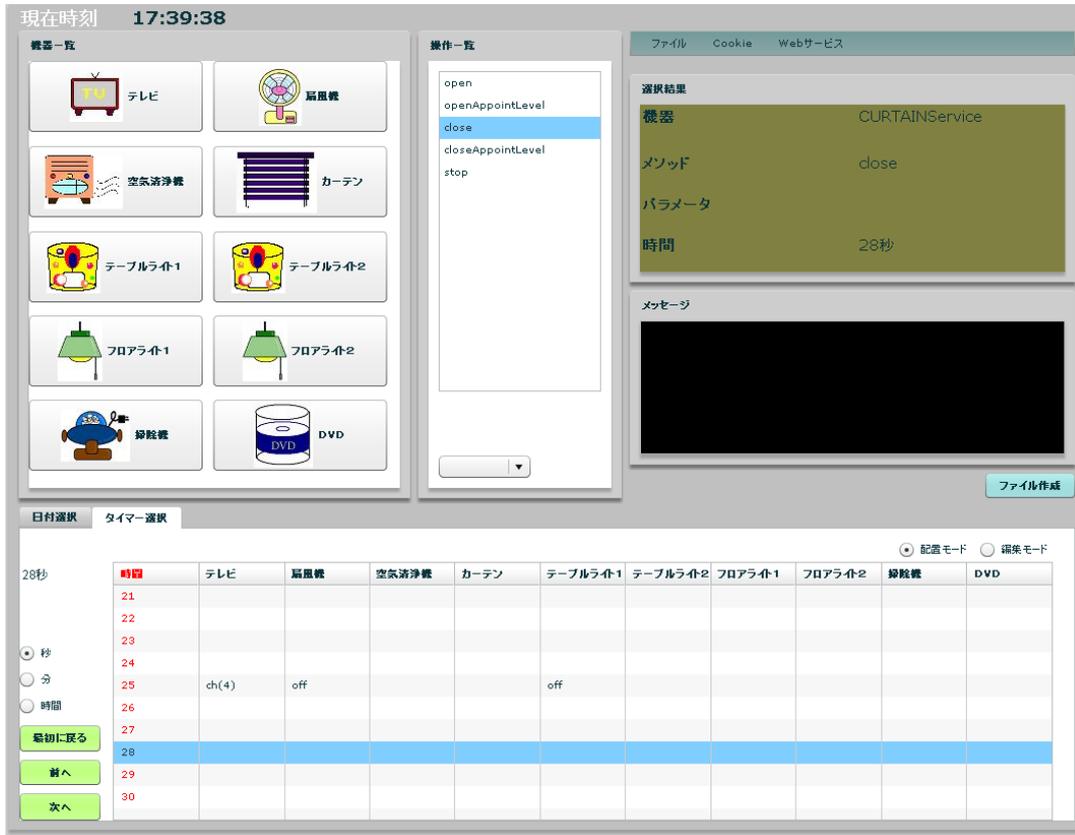


Fig. 4.3. 時間指定方式

## 4.4 HNS リアルタイムサービス実行 GUI

作成したサービスをグラフィカルに実行する実行 GUI も作成した。図 4.4 に HNS リアルタイムサービス実行 GUI の画面のレイアウトを示す。

### 4.4.1 機能

主な機能を以下に挙げる。

- サービスの実行
- タスクのキャンセル
- タスクの情報表示



Fig. 4.4. HNS リアルタイムサービス実行 GUI

#### 4.4.2 操作

##### ファイルの読み込み

サービス定義となるテキストファイルを指定し読み込むと、一つ一つのタスクが分かりやすい形で表示される。(図 4.5 参照)

プロパティとして、機器名、メソッド名、パラメータ値、時間、実行フラグ、残り時間を持つ。残り時間の単位については、日/時間/分/秒を用いた。作成した HNS リアルタイムサービスのメソッドの中で、ファイル読み込みは Init メソッドを、タスクのデータ取得に関しては timerList メソッドを使用している。

##### 実行とキャンセル

「実行」をクリックすると、サーバーに登録されているタスクが全て実行待ち状態になり、指定された時間待機した後、個々のタスクが実行される。タスクをキャンセルする事も可能であり、キャンセルしたいタスクを選び、「キャンセル」をクリックすると、そのタスクはキャンセルされる。実行に関してであ

るが、実行時間が重なるタスクの場合、仕様上、先にサーバーに登録された方から実行されるようになっている。作成した HNS リアルタイムサービスのメソッドの中で、実行は exeTimer メソッドを、タスクのキャンセルに関しては cancel メソッドを使用している。



Fig. 4.5. ファイル読み込み完了

## 第 5 章

# 評価

本章では，3.1 で挙げた 3 つの要求と将来性の指標からリアルタイム家電制御サービスの評価を行う．

### 5.1 評価実験

まず，HNS リアルタイム作成 GUI を用いて幾つかタスクを作成し，その中の幾つかの取り消しを行う．そして，HNS リアルタイムサービス実行基盤が正常に動作するかどうか確かめるために，HNS リアルタイム実行 GUI を使い実際に指定されたタスク通り家電機器が動作するか確かめる．

実験では，表 5.1 に挙げたタスクを用意した．この中で CURTAIN のタスク取り消しを行い，それ以外は実行を行った．その結果，正常に動作することが確認できた．

Table 5.1. 実験で作成したタスク

機器名	メソッド名	パラメータ値	時間
TV	on	-	15 秒後
FLOOR-LIGHT2	on	-	2008 年 2 月 25 日 18 時 10 分 00 秒
FAN	on	-	15 秒後
CURTAIN	open	-	2008 年 2 月 25 日 18 時 12 分 00 秒

表 5.1 に挙げた複数のタスクの正常動作，キャンセルが行えることが確認できたことより，HNS リアルタイムサービス実行基盤は正常動作すると言える．

これより要求 R1 は満たされた。

また，HNS リアルタイムサービス作成 GUI を用いてサービス定義を作成し，それらを HNS リアルタイムサービス実行基盤に登録できたことより要求 R2 は満たされた。

今回作成した HNS リアルタイムサービス実行基盤は Java Web サービスを用いて実装し，汎用的な PC からの使用のみを前提とした。しかし，Web サービスの使用は携帯電話やポータブルゲーム機 [15][16] などからネットワークを通じて行うことが出来る。そこで，将来的に自宅外からでも利用できるようにしたいと考えている。

## 5.2 HNS リアルタイム家電制御サービスの例

提案手法を用いることで，様々な新たなサービスが構築可能となることが予想される。以下に，いくつかの例を挙げる。

### 5.2.1 おはようサービス

次の日の朝が早い場合は，寝る前に目覚まし時計をセットすると思うが，時計ではなく，家電機器をセットするのがこのサービスである。使用する家電機器は，カーテン，エアコン，テレビ（コンポ），トースター，コーヒーメーカーである。例えば，朝 7 時に起きたい場合なら，1 時間前の朝 6 時にエアコンを作動させ，快適な環境を作っておき，10 分前の朝 6 時 50 分にカーテンが開くようにセットし，太陽の眩しい光を浴びて徐々に目が覚めてきたところで，テレビをつけるまたはコンポより音楽を流すようにセットする。それと同時ぐらいに，トースターでパンを焼き，コーヒーメーカーでコーヒーを沸かすようにセットしておけば，スムーズに朝ごはんを食べることが出来る。

### 5.2.2 おやすみサービス

これは，夜寝るときは真っ暗だと眠れないという人を対象としたサービスである。使用する家電機器は，照明，エアコンである。寝るときに，照明をある程度の照度まで落しておき，布団に入った時間から 1 時間ごとに照明の照度を徐々に落とすように設定し，完全に寝付く時間ぐらいに照明を消灯するように

する。エアコンの方は、布団に入ってから 3～4 時間後に切れるようにセットしておく。

### 5.2.3 お出かけサービス

これは、自宅を出る前にあの家電機器の電源を切ったかどうか不安になるような人を対象としたサービスである。使用する家電機器は特に特定しない。具体的には、朝起きてから家を出るまでに動作させた家電機器全ての電源を家を出るぐらいの時間に電源が切れるように設定しておく。

### 5.2.4 空き巣防止サービス

自宅に不在の場合でも誰か居るように見せかけて、空き巣が入るのを未然に防ぐのがこのサービスである。使用する家電機器は、照明、テレビである。具体的には、照明の点灯と消灯を一定時間繰り返し、部屋の明かりを点ける。必要ならばテレビを点けて音を出す。

### 5.2.5 快適帰宅サービス

自宅に帰ってきたときに快適な環境で過ごせるように設定するのがこのサービスである。使用する家電機器は、エアコン、炊飯器、湯沸かし器、空気清浄機である。具体的には、おおよそ帰宅する時間前に冷暖房を起動させ部屋を快適な環境にし、空気清浄機を使って空気を洗浄しておく。また、炊飯器やポットを使いご飯の用意をし、すぐに入れるように湯沸かし器を使ってお風呂も準備しておく。

### 5.2.6 照明連動サービス

何かに連動させて照明の点灯/消灯を繰り返し実行するサービスである。使用する家電機器は、照明のみである。例えば、音楽を流しているときにその音楽のテンポに合わせてリズムカルに照明が点灯するように設定しておけば、照明が音楽と連動して動いているように見える。

## 第 6 章

### まとめ

本論文では，HNS を利用したサービスとしてリアルタイム家電制御サービスを取り上げた．そして，3.1 に挙げた R1，R2 の要求を満たすようなシステムを作成した．本システムを作成するにあたり，Java Web サービスを用いて HNS リアルタイムサービス実行基盤を，AIR を用いて HNS リアルタイムサービス作成 GUI の実装を行った．また，実際に本システムを使いタスクをいくつか作成し，3.1 の 2 つの要求と，将来性の指標から評価した．今後の研究課題として，今回リアルタイム家電制御サービスが汎用的なパソコンからのみ利用可能であるので，携帯電話やポータブルゲーム機からでも使えることができるようにすることを挙げた．これにより，自宅からのみに限られていた利用が，外出先からでも利用できるようになる．

## 謝辞

本研究を進めるにあたり，ご指導，ご協力して頂いた方々にここで感謝の意を表させて頂きたいと思います．本当にありがとうございました．

神戸大学 情報知能工学科 井垣宏 助手には，副指導教員を担当していただき，研究の進め方や，論文の書き方などを詳細にご指導頂きました．本当にありがとうございました．

神戸大学 情報知能工学科 中村匡秀 准教授には，主指導教員を担当していただき，本研究の方向性を示唆して頂き，研究の進め方や，論文の書き方などを詳細に御指導頂きました．本当にありがとうございました．

## 参考文献

- [1] 松下電器産業株式会社 , “ 暮らし安心ホームシステム、暮らし安心・安全盤 ” , <http://www.mew.co.jp/corp/news/0603/0603-3.htm>
- [2] 東芝 , “ ホーム IT システム FEMINITY ” , <http://www3.toshiba.co.jp/femininity/>
- [3] TOSHIBA , “ フェミニティが実現する安心・便利な生活 ” , <http://www3.toshiba.co.jp/femininity/dekiru/index.html>
- [4] ZOJIRUSHI , “ みまもりほっとライン i-PoT ” , <http://www.mimamori.net/>
- [5] M . Nakamura , H . Igaki , H . Tamada , and K . Matsumoto ; Implementing Services of Networked HomeApplication Using Service Oriented Architecture , ” *Proc . 2nd International Conference on Service Oriented Computing(ICSOC2004)* , pp.269-278 , NY , USA , Nov.2004
- [6] 株式会社エーアイコーポレーション , “ TimeSys Linux ” , [http://www.aicp.co.jp/news/20020619\\_gui.shtml](http://www.aicp.co.jp/news/20020619_gui.shtml)
- [7] ITmedia News , “ ロボットの “ 運動神経 ” がコンシューマ機器にも ” , [http://www.itmedia.co.jp/news/0311/14/nj00\\_vxworks.html](http://www.itmedia.co.jp/news/0311/14/nj00_vxworks.html)
- [8] dSPACE , “ 適用分野 ” , <http://www.dspace.jp/ww/ja/jap/home/applicationfields.cfm>
- [9] Adobe , Adobe AIR , <http://www.adobe.com/products/air/>
- [10] 布留川 英一 , “ Adobe AIR プログラミングガイド ” , 株式会社毎日コミュニケーションズ (2007)
- [11] 武蔵工業大学 工学部 機械システム工学科 高機能機械制御研究室 , “ RTMATX によるリアルタイム制御入門 ” ,

- <http://daikoku.cl.mes.musashi-tech.ac.jp/MaTXWorkshop/rtweb/RtMaTXprimer.pdf>
- [12] IT用語辞典 e-Words , “ マルチスレッド ” ,  
<http://e-words.jp/w/E3839EE383ABE38381E382B9E383ACE38383E38389.html>
- [13] SUGIYAMA ELECTRON CO.,LTD , “ 学習機能付マルチリモコン  
「クロッサム 2+USB」 ” , <http://www.sugi-ele.co.jp/top.htm>
- [14] M . Nakamura ,A . Tanaka ,H . Igaki ,H . Tamada ,K . Matsumoto ,  
“ Constructing Home Network Systems and integrated Services Using  
Legacy Home Appliances and Web Services ” , *International Journal  
of Web Services Research* , vol.5 , No.1 , pp.82-98 , January-March  
2008
- [15] 任天堂 ; Nintendo DS ” <http://www.nintendo.co.jp/ds/ds/index.html>
- [16] ソニー・コンピュータエンターテイメント , “ PSP ” ,  
<http://www.jp.playstation.com/psp/>