

Title	ホームネットワーク規格との接続性を考慮したLegacy Deviceホームネットワークの構築に関する研究
Author(s)	浜本, 賢一
Citation	
Issue Date	2005-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1909
Rights	
Description	Supervisor:丹 康雄, 情報科学研究科, 修士

修 士 論 文

ホームネットワーク規格との接続性を考慮した
Legacy Deviceホームネットワークの構築に関する研究

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

浜本 賢一

2005 年 3 月

修 士 論 文

ホームネットワーク規格との接続性を考慮した Legacy Device ホームネットワークの構築に関する研究

指導教官 丹康雄 助教授

審査委員主査 丹康雄 助教授
審査委員 篠田陽一 教授
審査委員 敷田幹文 助教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

310087 浜本 賢一

提出年月: 2005 年 2 月

概 要

近年、白物家電機器を制御することを目的とした ECHONET や AV 機器を制御することを目的とした HAVi など、ホームネットワークを構築するための様々なミドルウェアが提案されている。しかし既存家電機器である LD(Legacy Device) は今後も家庭内に残る可能性が高く、各種ホームネットワーク規格と LD との接続性を提供できることが重要である。HAVi では DCM・FCM といった機能毎にモデル化を行った管理モデル、ECHONET では ECHONET オブジェクトと呼ばれる機器の動作機能・プロファイル・通信定義・公開する機能を規定したオブジェクトにより管理を行うモデルといったように各種ホームネットワーク規格との接続を行うようゲートウェイを実装する際、規格間で異なる家電モデルを統一して扱える新たな家電モデルが必要となる。しかしながら各種規格の様々な機能をこの家電モデルに取り入れると、規格が増える毎に扱う機能が膨大となり管理が複雑となる。

本研究では、LD と各種ホームネットワーク規格における家電を種類別に共通機能を抽出したモデルを用いて、各種規格毎のゲートウェイを実装することにより各種規格との接続性を持った LD によるホームネットワークの構築を行う。

目次

第1章	はじめに	1
1.1	研究の背景	1
1.2	研究の目的	1
1.3	本論文の構成	2
第2章	各種ホームネットワーク規格	3
2.1	ECHONET[6][7]	3
2.2	X-10[8]	6
2.3	HAVi[5][9]	8
2.4	DLNA[7][10]	9
2.5	宅内情報通信・放送高度化フォーラム [7]	10
第3章	提案システム	13
3.1	システムの構成	13
3.2	LD Object	14
3.3	LD Manager	14
3.4	各種 Gateway	14
3.5	機能番号	15
3.6	各クラス間の関係	15
第4章	LD Object	17
4.1	家電クラス [11][12]	17
4.1.1	家電モデルの定義	17
4.1.2	単体家電機器と複合家電機器	22
4.2	システムに必要なプロパティ	25
4.3	電源の状態の定義 [21]	26
4.4	LD Object	28
4.4.1	LD Object クラスの定義	28
第5章	LD Manager	31
5.1	LD Manager クラス	31

第 6 章	各種 Gateway	32
6.1	Gateway クラス	32
6.2	各種規格 Gateway の設計	33
6.2.1	LD における Gateway	33
6.2.2	ECHONET Gateway[6]	44
6.2.3	X-10 Gateway[8]	61
6.2.4	HAVi Gateway[5][9]	62
6.3	規格間における家電制御のシーケンス	69
第 7 章	システムの実装 [20]	70
7.1	実装	70
7.2	家電の制御の検証	71
第 8 章	システムの評価と考察	72
8.1	LD Object に関して	72
8.1.1	家電クラスについて	72
8.1.2	家電における共通機能抽出について	74
8.1.3	家電の機能について	74
8.2	家電の電源の扱いに関して	75
8.2.1	電源の状態について	75
8.2.2	ハード電源とソフト電源について	75
8.3	機能番号について	76
8.4	各種規格との接続性について	76
第 9 章	まとめと今後の課題	77
9.1	まとめ	77
9.2	今後の課題	78
9.2.1	機器の検索	78
9.2.2	セキュリティの検討	78
	謝辞	81
付 録 A	各家電の家電クラスの詳細	82
A.1	冷蔵庫クラス	82
A.2	炊飯器クラス	83
A.3	電気ポットクラス	83
A.4	オープンレンジクラス	84
付 録 B	機能番号表と引数番号表	86

目 次

2.1	ECHONET の通信レイヤ構成の概要	3
2.2	ECHONET アドレス	4
2.3	機器オブジェクト構成例図	5
2.4	ECHONET フレーム	6
2.5	X-10 における伝送フォーマット	6
2.6	Header と Code	7
2.7	デバイスの制御例	9
2.8	宅内フォーラムにおけるホームネットワークアーキテクチャ	11
3.1	システム構成	13
3.2	ゲートウェイ	14
3.3	クラス間の関係	16
4.1	エアコンクラスにおけるクラス図	20
4.2	単体家電クラス	22
4.3	複合家電クラス	22
4.4	家電クラスにおける継承	23
4.5	SystemObject クラスのクラス図	25
4.6	PowerManagement が管理する電源の状態機械	26
4.7	PowerManagement が管理する電源の状態遷移図	27
4.8	PowerManagement Object クラス	27
4.9	LD Object クラス	28
4.10	LD Object における機能番号の処理	30
6.1	ゲートウェイクラス	32
6.2	各種ゲートウェイクラス	33
6.3	LD におけるホームネットワークシステム構成図	34
6.4	LD とやりとりするメッセージタイプ	36
6.5	LD を LD Object へ変換する (ex. エアコン)	37
6.6	LD の登録	39
6.7	LD への応答	40
6.8	他規格から LD を制御する際のシーケンス	41

6.9	LD から他規格の家電を制御する際のシーケンス (他規格から確認通知が返ってこない場合)	42
6.10	LD から他規格の家電を制御する際のシーケンス (他規格から確認通知が返ってくる場合)	43
6.11	ECHONET におけるコールドスタート	46
6.12	ECHONET におけるウォームスタート (1)	47
6.13	ECHONET におけるウォームスタート (2)	48
6.14	ECHONET 機器の登録	53
6.15	ECHONET への機器の登録	56
6.16	ECHONET からの機器の制御	58
6.17	ECHONET に対する家電の制御	60
6.18	X10 機器の登録	61
6.19	アプリケーションと FCM 間の通信	63
6.20	HAVi アプリケーションとコントロールされる機器間の通信	64
6.21	HAVi における機器の登録	66
6.22	HAVi における機器の制御	68
6.23	ECHONET から X10 機器の制御	69
7.1	提案するシステムの実装概要	70
7.2	提案するシステムにおける LD との接続環境	71
8.1	家電クラス	72
8.2	単体家電クラス	73
8.3	複合家電クラス	73
8.4	単体家電クラス	74
8.5	複合家電クラス	74
8.6	On_Off Object クラスと家電クラス間の関係	75

表 目 次

2.1	ファンクションコード	7
3.1	機能番号表	15
4.1	各社におけるエアコンの機能比較 (1)	18
4.2	各社におけるエアコンの機能比較 (2)	19
4.3	エアコンクラスのプロパティ値	21
4.4	エアコンクラスのメソッド値	21
4.5	洗濯機クラスのプロパティ値	23
4.6	洗濯機クラスのメソッド	24
4.7	洗濯乾燥機クラスのプロパティ値	24
4.8	洗濯乾燥機クラスのメソッド値	24
5.1	LD Object アドレス表	31
6.1	機器と対応する部分状態機械	35
A.1	冷蔵庫クラスのプロパティ値	82
A.2	冷蔵庫クラスのメソッド	82
A.3	炊飯器クラスのプロパティ値	83
A.4	炊飯器クラスのメソッド	83
A.5	電気ポットクラスのプロパティ値	83
A.6	電気ポットクラスのメソッド	83
A.7	オーブンレンジクラスのプロパティ値	84
A.8	オーブンレンジクラスのメソッド値	84
B.1	機能番号表	86
B.2	電源の操作・機能実行に関する引数番号表	88
B.3	チャンネル番号に関する引数番号	88
B.4	音声変更に関する引数番号表	88
B.5	オフタイマー・オンタイマー設定時間に関する引数番号	89
B.6	エアコンのモードに関する引数番号表	89
B.7	風向に関する引数番号表	90

B.8 水量に関する引数番号表	91
B.9 レンジ加熱レベルに関する引数番号表	91
B.10 自動メニューに関する引数番号表	92

第1章 はじめに

1.1 研究の背景

近年、高度な情報処理能力とネットワーク接続機能を備えた情報家電と呼ばれる家電同士を接続する家庭内ネットワークの実現が検討され、白物家電を制御することを目的とした ECHONET や AV 機器を制御することを目的とした HAVi などホームネットワーク規格が様々提案されている。これらの規格に基づいたホームネットワークを構築するためには、家電機器の入れ替えや新規取組を行う必要があり、また機能付加による家電機器の価格上昇を考えると、既存家電機器である LD(Legacy Device) は今後も家庭内に残る可能性が高い。よって家庭内ネットワークを構築するには、既存のホームネットワーク規格と LD との接続性を提供できることが重要である。これに対して、電灯線などの既存の伝送路である LC(Legacy Channel) を通じて、LD の制御や複数の LD の連動によるサービスをユーザに提供するシステムが提案されている [4]。こうしたシステムにおいて他のホームネットワーク規格との接続を行うようゲートウェイを実装する際、規格間で異なる家電の管理モデルを統一する新たな管理モデルが必要となる。

1.2 研究の目的

家電を制御する際には、家電の状態を管理することが重要である。各種ホームネットワーク規格は次のような管理モデルを用いて家電を管理している。HAVi では DCM・FCM といった機能毎にモデル化を行っており [5]、ECHONET では ECHONET オブジェクトと呼ばれる機器の動作機能・プロファイル・通信定義・公開する機能を規定したオブジェクトにより管理を行うモデルを用いている [6]。このように規格ごとに管理モデルが異なり、LD によるホームネットワークと他の規格との接続を考えた場合、統一的に扱えるモデルが必要となる。家電の状態は同種の家電において規格間の管理方法は異なるものの、家電の状態の規定に関する違いは少ないので、家電の種類別に LD と他のホームネットワーク規格との間で家電のモデル化を行うことが可能と考えられる。本研究では、LD と各種ホームネットワーク規格における家電の種類別にモデル化し、各種規格毎のゲートウェイを実装することにより各種規格との接続性を持った LD によるホームネットワークの構築を行う。

1.3 本論文の構成

本論文は以下の構成になっている。

- 第1章 … 研究の背景と目的について説明する。
- 第2章 … 本研究の対象である各種ホームネットワーク規格について説明する。
- 第3章 … 本論文で提案する各種ホームネットワーク規格と接続性を考慮した Legacy Device ホームネットワークを構築するシステムについて説明する。
- 第4章 … 提案システムにおける家電モデルである LD Object について説明する。
- 第5章 … 提案システムにおける LD Object の管理や各種規格の家電の検索を行う LD Manager について説明する。
- 第6章 … 提案システムにおける各種ホームネットワーク規格とのゲートウェイの設計および機器の登録・制御のシーケンスについて説明する。
- 第7章 … 提案システムの実装に関する環境および接続の検証について説明する。
- 第8章 … 提案システムの評価および考察を行う。
- 第9章 … 本研究における全体のまとめおよび今後の課題を述べる。

第2章 各種ホームネットワーク規格

本章では、既存のホームネットワーク規格に関する概要をまとめる。現在策定が進められている ECHONET、X-10、HAVi、DLNA、宅内情報通信・放送高度化フォーラムについてそれぞれ概説する。

2.1 ECHONET^{[6][7]}

ECHONET(Energy Conservation and Homecare Network) は、エコーネットコンソーシアムが仕様の策定を行っている、家庭内の様々な白物家電製品や住宅設備を相互接続するためのホームネットワーク規格である。

ECHONET 機器の通信レイヤは図 2.1 に示すようにアプリケーションソフトウェア、通信ミドルウェア、下位通信ソフトウェアの3階層に分けられる。ECHONET 規格では、通信ミドルウェアおよび下位通信ソフトウェアの仕様を規定している。

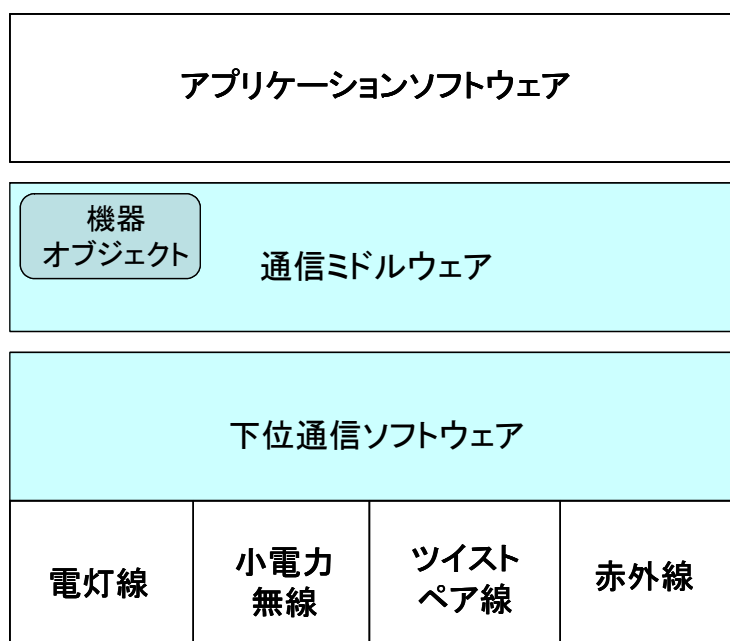


図 2.1: ECHONET の通信レイヤ構成の概要

通信ミドルウェア部は、次の2つの機能を持つ。

- 異なる伝送プロトコルを隠蔽して、独自の ECHONET アドレスを管理するなど、上位のアプリケーションに対して1つのネットワークとして見せる。
- デバイスをオブジェクトとして管理し、操作やモニタのための API を提供する。

下位通信ソフトウェア、電灯線、無線、赤外線などの伝送メディア特有の通信プロトコル処理を行うソフトウェアであり、主に OSI 参照モデルのレイヤ 1 , 2 に相当する通信処理を行う。

ECHONET アドレスは、図 2.2 に示すように1バイトの NetID と1バイトの NodeID からなる。NodeID はサブネット内で、ECHONET ノードを一意に識別する識別子で、MAC アドレスから変換され、一意に対応付けされるものである。NetID はサブネットの識別子で、各々の ECHONET ノードにおいて、自身の NetID は ECHONET ルータの保持する情報により設定されるものである。

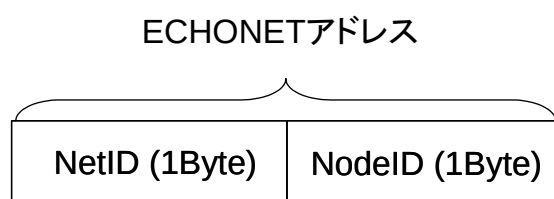


図 2.2: ECHONET アドレス

ECHONET において、家電の制御情報などのやり取りは、通信ミドルウェアレイヤで定義されている ECHONET オブジェクトを操作することにより行われる。ECHONET で規定されている ECHONET オブジェクトは、大きく以下の4つに分類されている。

- 機器オブジェクト
- プロファイルオブジェクト
- 通信定義オブジェクト
- サービスオブジェクト

機器オブジェクトは、機器の持つ「機器としての動作機能」を規定しているものである。このオブジェクトは図 2.3 に示すように機器ごとに持つ動作機能をプロパティとして持つ。ECHONET を介して ECHONET ノードを制御したい他の ECHONET ノードは、この機器オブジェクトを操作(書き込み/読み出し)することにより、この ECHONET ノードの機能の制御や状態の確認を行うこととなる。

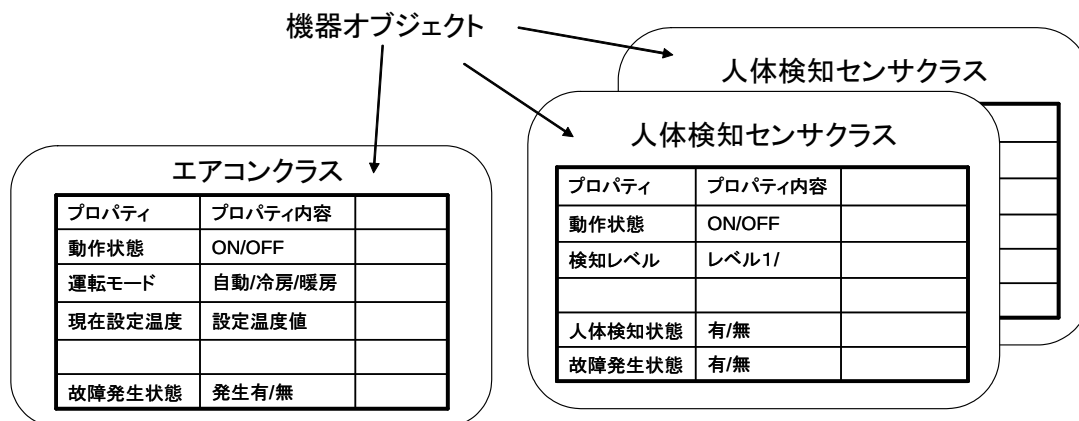


図 2.3: 機器オブジェクト構成例図

プロファイルオブジェクトはECHONET ノードの動作状態や、メーカ情報、機器オブジェクトリストなど ECHONET ノードとしてのプロファイルの情報を、アプリケーションソフトウェア及び他の ECHONET ノードが操作 (書き込み/読み出し) することを目的として規定するものである。

通信定義オブジェクトは機器オブジェクト、プロファイルオブジェクト、サービスオブジェクトの通信上の動作を操作することを目的として規定するオブジェクトの総称であり、「エアコン通信定義オブジェクト」、「ノードプロファイル通信定義オブジェクト」など個々の機器オブジェクトとプロファイルオブジェクト、およびサービスオブジェクトの各クラスに対し規定する。

サービスオブジェクトはサービスミドルウェアの機能に基づき、ネットワークに対し後悔する機能をモデル化しクラス仕様を規定する。このオブジェクトは ECHONET 機器相互で、通信を介してのサービスミドルウェアの操作を行うことを目的としているものである。

ECHONET は、オブジェクトの操作を図 2.4 に示す形式の電文を送受信することによって行われる。ECHONET 電文ヘッダに続いて、送信元/相手先アドレスを指定するフィールドがあり、続く ECHONET データの中では、送信元/相手先オブジェクト、プロパティ、サービス (ここでサービスとは、プロパティに対する読み出しや書き込み (設定)、通知を指定するものである)、設定プロパティ値を指定する。

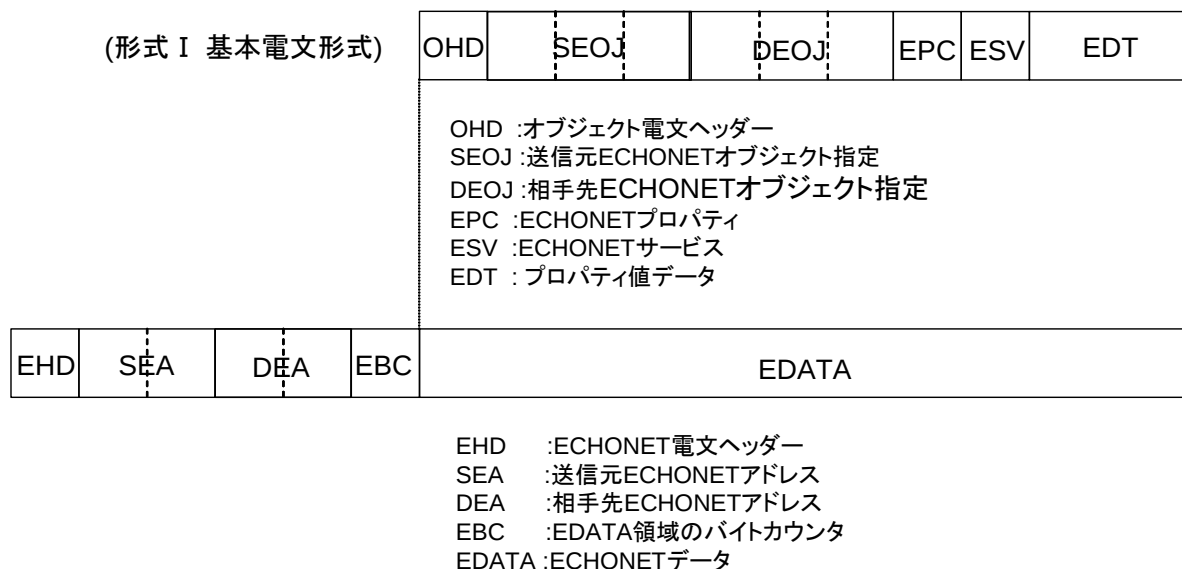


図 2.4: ECHONET フレーム

2.2 X-10^[8]

X-10 は、伝送媒体として家庭内の電灯線を用い、X-10 モジュールを通して家庭内の様々な機器の電源、もしくは照明機器を制御する規格である。X-10 における伝送は Housecode と Device Code、ファンクションコードの伝送である。これらの伝送フォーマットは図 2.5 に示す通りである。

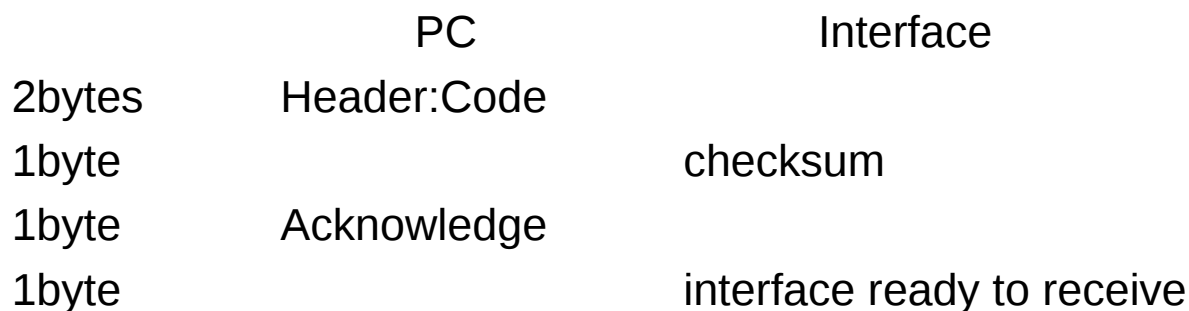


図 2.5: X-10 における伝送フォーマット

Header と Code の詳細を図 2.6 に示す。Header にある Number of Dims は 0 ~ 22 まで数値を取り、明るさを示している。2 ビット目はインタフェースが同期を維持できることを保証するために常に”1”をセットする。F/A は次のバイトがファンクション (1) であるかアドレス (0) であるかを示す。E/S は次のバイトが拡張された伝送 (1) か標準の伝送 (0)

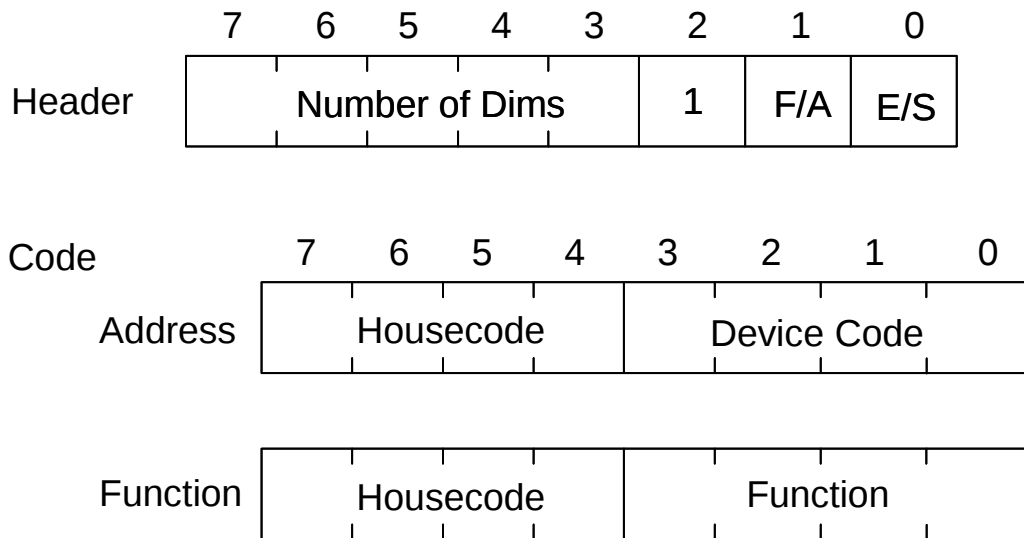


図 2.6: Header と Code

表 2.1: ファンクションコード

Function	ファンクションコード
All Units Off	0000
All Units On	0001
On	0010
Off	0011
Dim	0100
Bright	0101
All Lights Off	0110
Extended Code	0111
Hail Request	1000
Hail Acknowledge	1001
Pre-set Dim(1)	1010
Pre-set Dim(2)	1011
Extended Data Transfer	1100
Status On	1101
Status Off	1110
Status Request	1111

であるかを示している。Code にある HouseCode と Device Code は X-10 におけるアドレスを示している。Function は表 2.1 に示すファンクションコードを示している。

X-10 における伝送ははじめに PC からインタフェースに対して Header:Code の 2 バイトを送る。次にインタフェースは (Header+Code) & 0xff で計算されるチェックサムを PC に返す。チェックサムが正しければ、PC は確認通知である 0x00 の値をインタフェースに返す。チェックサムが正しくなければ、再度 Header:Code の組をインタフェースに送り、チェックサムを待つ。一度 X-10 における伝送が行われると、インタフェースは自身が 'ready' 状態にいることを示すために PC へ '0x55' の値を送る。

2.3 HAVi_{[5][9]}

HAVi(Home Audio Video Interoperability) は家庭内の AV 機器をメーカーや機種にとらわれることなく相互接続するためのホームネットワーク規格である。HAVi アーキテクチャはソフトウェアエレメントと呼ばれるオブジェクトで構成される。

- 1394 Communication Media Manager - 1394 上で他のソフトウェアエレメントが非同期転送、アイソクロナス転送を行えるようにする
- Messaging System - ソフトウェアエレメント間にメッセージ伝送サービスを提供する
- Registry - ソフトウェアエレメントを検索するためのディレクトリサービスを提供する
- Event Manager - オブジェクトやホームネットワークの状態の変化であるイベントを通知するサービスを提供する
- Stream Manager - ネットワーク上を流れる AV や他のメディアのリアルタイム伝送を管理する
- Resource Manager - リソースの共有やアクションのスケジューリングを容易にする
- DCM - デバイスを制御するために使われるソフトウェアエレメントである
- DCM Manager - FAV もしくは IAV 上に DCM コードユニットのインストールとアンインストールを行うサービスを提供する

HAVi の制御モデルはコントローラとコントロールされるデバイスとに区別される。コントローラはコントロールされるデバイスにとってホストとして動作するデバイスである。このコントローラは DCM(Device Control Module) と呼ばれる。DCM はアプリケーションからデバイスを制御する API を提供する。

HAVi ではデバイスを FAV(Full AV device)、IAV(Intermediate AV device)、BAV(Base AV device)、LAV(Legacy AV device) の 4 つのカテゴリに分けられる。FAV は HAVi を構

成する全てのソフトウェアエレメントを持つものである。IAV は一般的に FAV よりもコストが低く、リソースに制限があるものである。JAVA ランタイム環境を提供しないのでホームネットワーク内で任意のデバイスのコントローラとして振舞うことができない。しかしネイティブコードをサポートするので一部のデバイスのコントローラとしては振舞うことができる。BAV はホストとしては振舞えないが、アップロード可能な JAVA バイトコードを提供するので FAV によってバイトコードを通じて、また IAV によってネイティブコードを通じて制御される。LAV は HAVi に対応していないデバイスを指す。このデバイスは独自のプロトコルを用いて制御しているので、ホームネットワーク内で制御するにはネイティブコードをサポートしている FAV や IAV が必要となる。図 2.7 は HAVi アーキテクチャにおいてデバイスを制御する例を示している。各デバイスは IEEE1394 で接続されており、IAV や FAV にあるアプリケーションが LAV や IAV を制御するには FAV もしくは IAV に組み込まれた DCM を用いて行う。

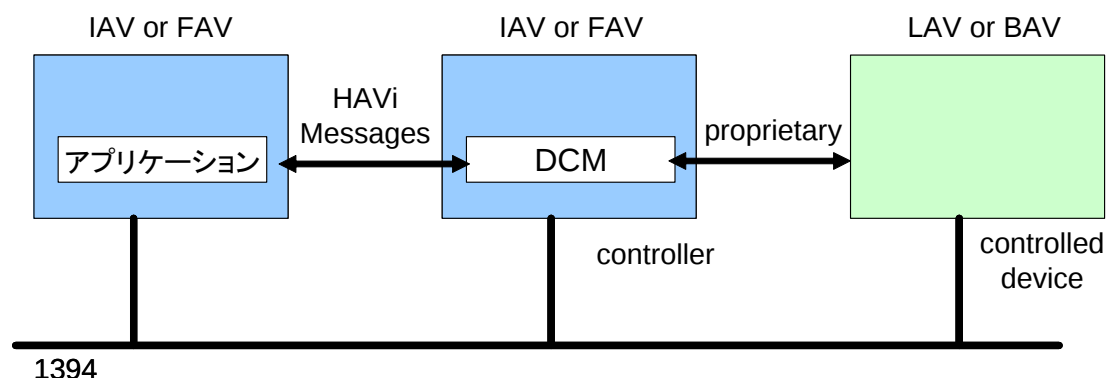


図 2.7: デバイスの制御例

2.4 DLNA^{[7][10]}

DLNA(Digital Living Network Alliance) は、家電・パソコン・モバイルなどの機器間でデジタルコンテンツ(音楽、写真、ビデオなど)を家庭内で簡単に共有するため、業界標準技術に基づいたオープンな相互接続互換性を構築するための技術的な設計ガイドラインを策定することを目的都市、2003 年 6 月に設立された非営利団体である。

DLNA 機器は

1. DMS(Digital Media Server) : メディアの入力、記録、配信の能力を有するデバイスで、必要なコンテンツ保護機能を備えることが求められる。DMS は多くの場合、次の DMP の能力を兼ね備え、高度な HI やメディアの管理・集積機能を有するインテリジェントな機器となる。

(例)STB、PVR、パソコン、チューナ、カメラ

2. DMP(Digital Media Player)：メディアの再生、表示を行うことのできるデバイスである。

(例) テレビ、ステレオ、プリンタ、PDA

の2つのクラスに分かれ、機能が定義されている。フォーマットは、各メディアクラスにより次のような標準技術が定義されている。

1. IMAGE メディアクラス：JPEG [必須]
GIF,PNG,TIFF [オプション]
2. AUDIO メディアクラス：Linear PCM (2ch) [必須]
AAC,AC-3,ATRAC3plus,MP3,WMA9 [オプション]
3. VIDEO メディアクラス：MPEG-2 [必須]
MPEG-1、MPEG-4、WMV9 [オプション]

また、フレームワークは、次のような標準技術が使用される。

1. 物理・リンク層：IEEE802.3i/u および IEEE802.11a/b/g
2. ネットワーク層：IPv4
3. 機器発見：UPnP(Universal Plug and Play) の Device Architecture Version 1.0
4. 機器制御：UPnP DCP(Device Control Protocol) の AV Version 1.0
5. トランスポート層：HTTP(Hyper Text Transfer Protocol)

ガイドラインに準じて設計された機器は、相互接続試験、認証プログラム、ロゴプログラムが適用される予定である。

更に 2005 年に向け、メディアフォーマットに AV/C と JPEG 2000 を、ネットワークに IEEE802.11e/i と IPv6 を、また著作権保護コンテンツでの互換性技術などを検討している。

2.5 宅内情報通信・放送高度化フォーラム [7]

宅内情報通信・放送高度化フォーラム (以下宅内フォーラムと呼ぶ) は ITU-T に提案されたホームネットワーク規格である。宅内フォーラムの基本アーキテクチャは、次のネットワークの構築を目的としている。

1. 宅内における AV 系、白物家電系、パソコン系などの多様な情報端末機器、家電機器を相互に自由に接続し、操作できる環境を提供し、家電機器の組み合わせ自由度を高めることにより、利便性を飛躍的に向上するネットワーク
2. 宅外の通信・放送のネットワークと宅内における情報端末機器、家電機器との間を自由に接続し、安心して利用できるネットワーク利用型サービスインフラの一部を構成するネットワーク
3. 異なる規格に基づく複数のネットワークの共存が可能で、将来の新しい規格にも対応が可能な柔軟なネットワーク

このようなネットワークを構築するために、宅内フォーラムの基本アーキテクチャでは図 2.8 に示すように、固有プロトコルドメインと IP ドメインという二つのドメインをホームネットワークに導入している。

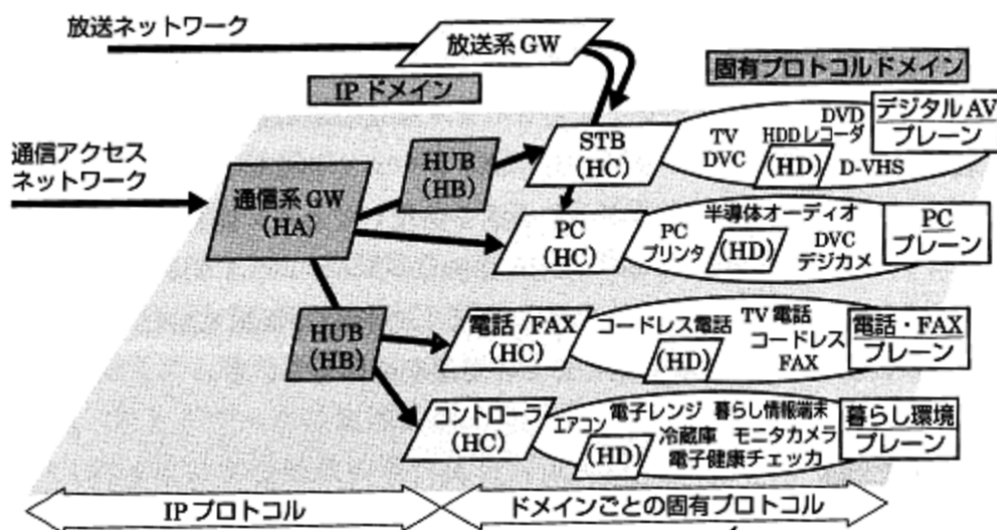


図 2.8: 宅内フォーラムにおけるホームネットワークアーキテクチャ

固有プロトコルドメインは、用途別ネットワークに対応するものであり、用途や接続される機器の種別により複数のプレーンに分類される。各々のプレーンでは、そのプレーンの用途、機器に応じた独自に定められたプロトコルを用いて機器の相互接続が実現される。代表的なプレーンには、デジタル AV プレーン、暮らし環境プレーン、PC プレーン、電話・FAX プレーンがある。

デジタル AV プレーンは、デジタル TV、HDD/DVD レコーダなど、デジタル AV 機器が属するネットワークである。IEEE1394 で規定されるプロトコルを用いて、相互操作や映像などのコンテンツ伝送が実現されている。

暮らし環境プレーンは、エアコン、電子レンジなど白物家電と呼ばれる機器や各種センサが属するネットワークである。ECHONET 規格で定められているプロトコルを用いて、機器の遠隔操作やモニタリングなどの機能が提供される。

PC プレーンはパソコンとパソコン周辺機器が属するネットワークである。USB や無線 LAN により機器が接続される。UPnP は PC プレーンにおけるプラグアンドプレイを実現するための仕様として提案されたものであったが、最近ではデジタル AV 機器などへの適用も検討されている。

電話・FAX プレーンは、電話、FAX サービスを提供する機器が属するネットワークである。このプレーンの主な用途は Voice over IP (VoIP) であり、音声の通話品質と秘匿性を保証するために、QoS や IPSec などの機能が重要になる。

IP ドメインは固有プロトコルドメインの各プレーンを相互に接続するとともに、宅外ネットワーク (通話アクセスネットワーク、放送ネットワーク) に接続されるネットワークであり、IP を基本プロトコルとして定めている。宅外のネットワーク利用型サービスインフラの一部を構成するネットワークという観点から、ホームネットワークの標準プロトコルはインターネットの標準プロトコルである IP に定められている。

第3章 提案システム

本章では、家電の種類ごとに機能を抽象化した家電モデルを用いて各種ホームネットワーク間を相互接続するシステムを提案し、システムの構成、LD Object、LD Manager、ゲートウェイについて説明する。

3.1 システムの構成

本研究で提案するシステムの構成を図 3.1 に示す。本システムでは、各種規格における家電を LD Object としてモデル化し、この LD Object を通じて家電を制御する。LD Object から家電を制御する際は、各種規格のゲートウェイを介して行われる。LD Manager は、各種規格において家電の検索を行い、システム内に LD Object の生成を行う。

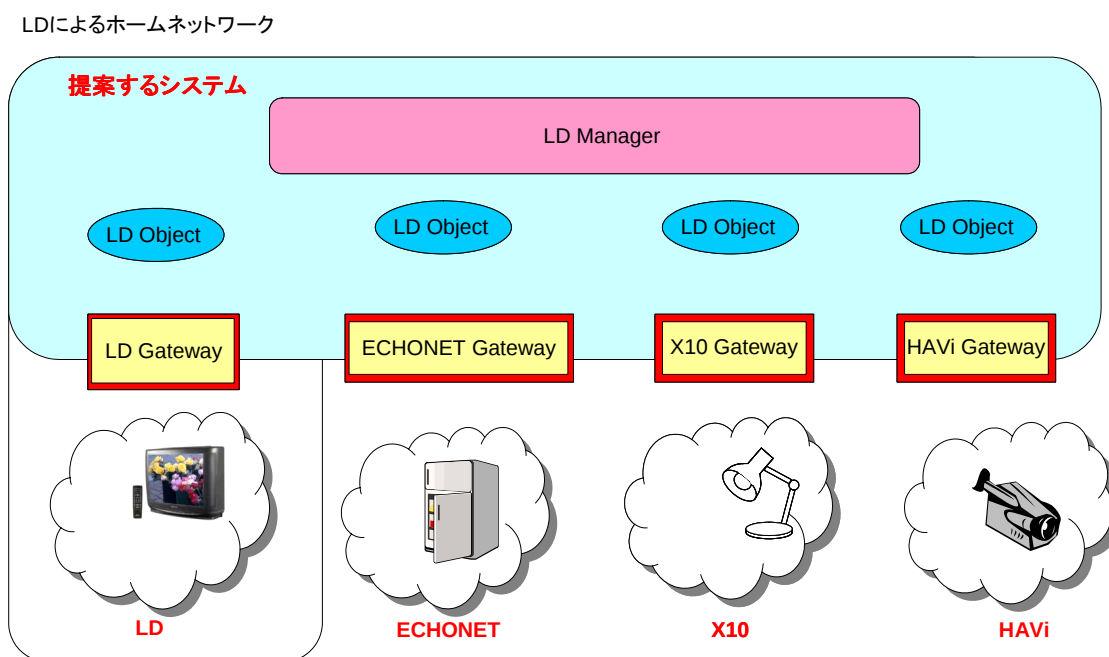


図 3.1: システム構成

3.2 LD Object

LD Object は、LD ネットワークにおいて LD や各種規格における家電を区別することなく扱うことを可能にしたものである。機種固有の機能も含めた家電の機能を洗い出し、同種の家電に分類し共通したものをその種類の家電の共通機能とし、LD オブジェクトのプロパティ・メソッドとして定義する。規格間を相互接続する際、この LD オブジェクトを通じて家電の制御を行う。

3.3 LD Manager

LD Manager は各種規格における家電の検索を行い、その結果に基づいてシステム内に LD Object を生成し、その際にアドレスも付与を行う。また他規格から家電の検索に対してシステム内に存在する LD Object のリストを返す役割も持つ。

3.4 各種 Gateway

Gateway は各種規格と本システムとのゲートウェイを表す。本システムと各規格の間にゲートウェイを実装することで、図 3.2 に示すように、各種規格間にゲートウェイを実装するよりも少ない工数で済む。

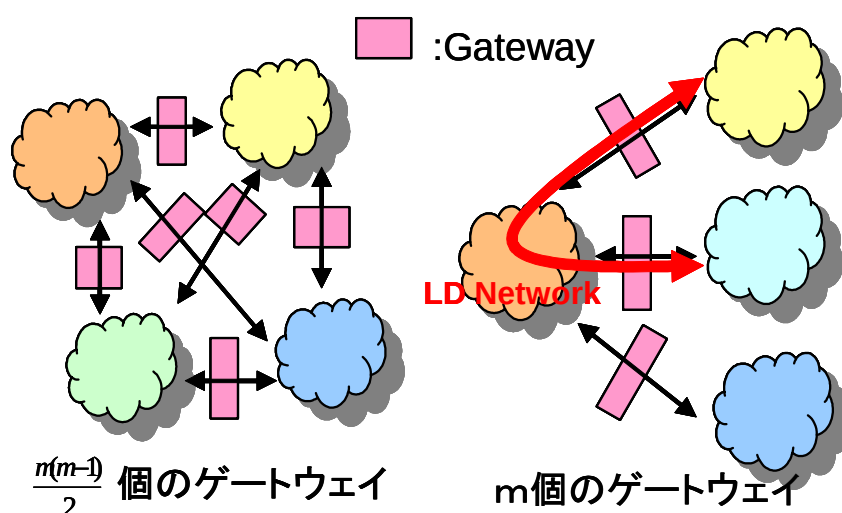


図 3.2: ゲートウェイ

3.5 機能番号

機能番号はシステム内でやり取りされ、各オブジェクトに対する命令を指定する番号である。表に機能番号表を示す。システム内でやり取りされる命令は全て機能番号を用いる。やり取りされる機能番号は、機能番号と機能番号に対する引数の対で用いる。ある機能に対する引数は引数番号表より求める。

表 3.1: 機能番号表

機能	機能番号	引数
LD Object の検索	0x00	-
LD Object の応答	0x01	LD Object のアドレス + 家電の種類
他規格の家電の検索	0x02	-
他規格の家電の応答	0x03	家電の種類 + 規格
・	・	・
・	・	・
・	・	・

各種ゲートウェイは機能番号を各種規格で定義されている命令に変換するための機能番号変換表をそれぞれが持つ。システムでのやり取りの例として、他の規格から LD Object に対して電源の On を行う命令がきた場合を挙げる。他の規格からゲートウェイに電源の On という命令が来ると、機能番号変換表より機能番号：電源の操作 (表 3.1 を参照)、引数：On (表 ?? を参照) に変換し、LD Object にこれらの値を渡す。LD Object はこれらの値がくると、機能番号表と引数番号表を用いて適切な制御を行う。機能番号と引数番号の詳細は付録 C に示す。

3.6 各クラス間の関係

LD Object、LD Manager、各種ゲートウェイのクラス間の関係を図 3.3 に示す。LD Gateway は LD とのゲートウェイを、X10 Gateway は X10 のゲートウェイを示している。各クラス間でやり取りされる情報は機能番号とそれに対する引数をやり取りする。各種ゲートウェイのクラスは LD Manager と家電の情報のやり取りを行う。また LD Object クラスとは家電の制御情報のやり取りを行う。LD Manager クラスは LD Object クラスの生成・管理を行う。

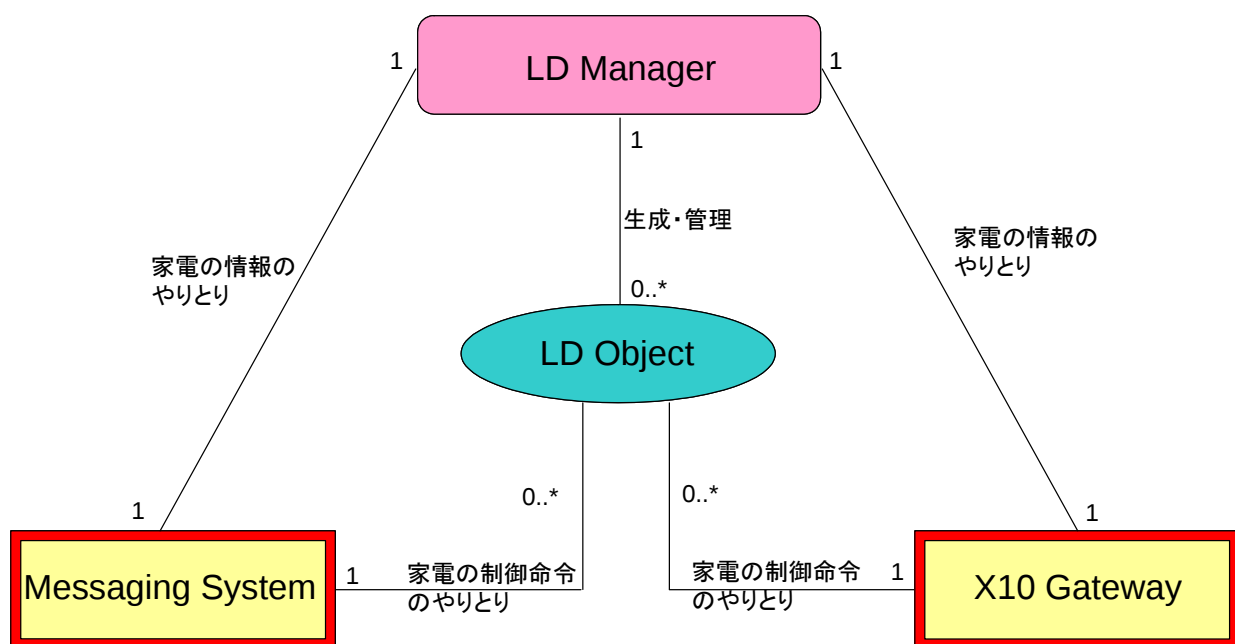


図 3.3: クラス間の関係

第4章 LD Object

本章では、家電の種類ごとに機能を抽象化した家電モデルである LD Object の詳細について説明する。

4.1 家電クラス [11][12]

4.1.1 家電モデルの定義

本システムでは同種の家電毎に状態を規定するモデルを扱う。全ての家電の機能を家電モデルに取り入れると、扱う規格が増える毎に扱う機能が膨大となり管理が複雑となる。よって本システムで扱う家電モデルは、家電の種類ごとに共通機能を抽出し、モデル化を行う。

同種の家電における共通機能の抽出は次の方法を行う。まず家電の全機能を洗い出し、家電メーカー各社と ECHONET で定義している機器オブジェクトがどの機能を持つかを明確にする。機能ごとに各社がその機能を採用しているかを見ていく。そしてその割合が高ければ、その機能を共通機能として定義する。

表 4.1 と表 4.2 は、エアコンにおいての各社の機能比較を行った表である。表中の○はその機能を搭載していることを示し、×はその機能を搭載していないことを示している。は上位機種のみには搭載されている機能であることを示している [13][14][?][16][17][18][19]。

表 4.1: 各社におけるエアコンの機能比較 (1)

	ECHONET	DAIKIN	SHARP	HITACHI
機能状態 (運転/停止)	○	○	○	○
冷房	○	○	○	○
暖房	○	○	○	○
除湿	○	○	○	○
加湿	×	○	×	×
フルパワー	○	○	○	○
風向 (上下)	○	○	○	○
風向 (左右)	○	○		
スイング (上下)	○	○		○
スイング (左右)	○	○		
風量変更	○			○
温度設定	○	○	○	○
湿度設定	○			
空気清浄	○		○	
切タイマー	○	○	○	○
入タイマー	○	○	○	○

表 4.2: 各社におけるエアコンの機能比較 (2)

	TOSHIBA	FUJITSU	National	mitsubishi
機能状態 (運転/停止)	○	○	○	○
冷房	○	○	○	○
暖房	○	○	○	○
除湿	○	○	○	○
加湿	×	×	×	×
フルパワー	○	○	○	○
風向 (上下)	○	○	○	○
風向 (左右)				○
スイング (上下)	○	○		○
スイング (左右)				
風量変更	○	○	○	○
温度設定	○	○	○	○
湿度設定	×	×	×	○
空気清浄	○	○		○
切タイマー	○	○	○	○
入タイマー	○	○	○	○

表 4.1 と表 4.2 を用いて、エアコンにおける共通機能を定義し、エアコンクラスを定義する。エアコンクラスにおけるクラス図を図 4.1 に示し、各プロパティおよびメソッド内容を表 4.3、表 4.4 に示す。

エアコン
-運転モード -温度設定値 -風向 -風量 -スイング状態 -フルパワー状態 -空気清浄機能状態 -オンタイマー予約時間 -オフタイマー予約時間
+運転モード変更 +温度変更 +風向変更 +風量変更 +スイングOn/Off +フルパワーOn/Off +空気清浄On/Off +オンタイマー予約時間設定 +オフタイマー予約時間設定

図 4.1: エアコンクラスにおけるクラス図

- 運転モードは、冷房・暖房・除湿の3種類のモードがある。運転モード変更メソッドでモードを切り替えた後、運転モードプロパティに設定したモードの値をセットする。
- 温度設定値は、運転モードに対する設定温度をセットするプロパティである。この温度を目標値として、運転が行われる。
- 風向は、エアコンの風向の上下角度の位置を設定するプロパティである。風向は5段階あり、それぞれ”上”、”上中”、”中”、”下中”、”下”と定義する。
- 風量は、エアコンの風量を設定するプロパティである。風量は8段階で調節を行う。
- スイングは、上下角度に風向がスイングする機能である。スイングのプロパティがOnの場合に、その機能が使用されていることを表す。

表 4.3: エアコンクラスのプロパティ値

プロパティ名称	プロパティ内容
運転モード	冷房・暖房・除湿のいずれかの値
温度設定値	運転モードに対する温度設定値
風向	上下角度の風向
スイング状態	スイング On/Off
風量	風量
フルパワー状態	フルパワー On/Off
空気清浄状態	空気清浄 On/Off
オンタイマー予約時間	オンにするための予約時間
オフタイマー予約時間	オフにするための予約時間

表 4.4: エアコンクラスのメソッド値

メソッド名称	メソッド内容
運転モード変更	運転モードを変更する
温度変更	設定温度を変更する
風向変更	風向を変更する
スイング On/Off	スイングを On もしくは Off にする
風量変更	風量を変更する
フルパワー On/Off	フルパワーを On もしくは Off にする
空気清浄 On/Off	空気清浄機能を On もしくは Off にする
オンタイマー予約時間設定	オンタイマーを設定する
オフタイマー予約時間設定	オフタイマーを設定する

- フルパワーは、急激に部屋を暖めるもしくは涼しくするための機能である。フルパワーのプロパティが On の場合に、その機能が使用されていることを表す。
- 空気清浄は、エアコンに備わっている空気清浄機能の状態を表すプロパティである。このプロパティが On の場合に、その機能が使用されていることを表す。
- オフタイマー・オンタイマー予約時間は、それぞれオンタイマーとオフタイマーが開始される時間をセットするプロパティである。それぞれの予約時間設定のメソッドを行うことによって、予約が開始される。

「電気ポットクラス」や「洗濯機クラス」といったその他の各家電クラスの仕様は、付録においてそれぞれプロパティを規定する。

4.1.2 単体家電機器と複合家電機器

家電機器の中には、複数の家電機能を持つものもあり、その中には片方の電源が落ちていてももう一つの機能を使える家電や機能は複数持っているが同時には使えないという家電も存在する。本システムでは、家電機器を単体家電機器と複合家電機器の二種類に分類する。単体家電クラスは一つの家電クラスからなるものであり、複合家電クラスは複数の家電クラスからなるものである。単体家電クラスのクラス図を図 4.2、複合家電クラスのクラス図を図 4.3 に示す。単体家電クラスの例として TV クラスがあり、複合家電クラスの例として TV と VTR のクラスから構成されるテレビデオクラスがある。単体家電クラスと複合家電クラスを総称して家電クラスと呼ぶ。

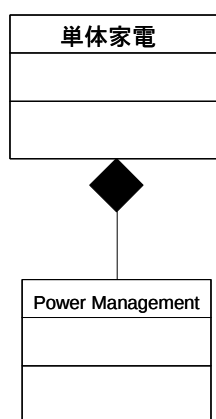


図 4.2: 単体家電クラス

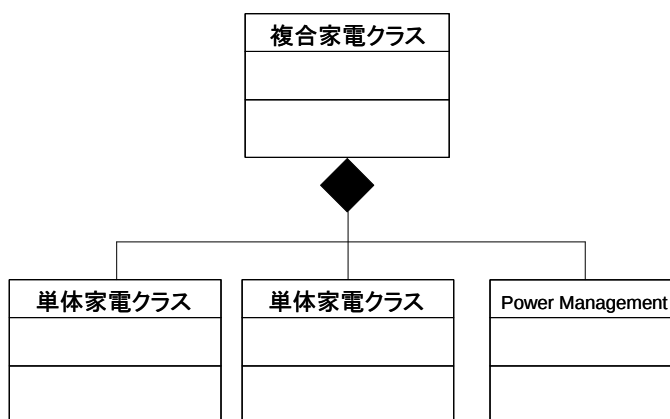


図 4.3: 複合家電クラス

単体家電クラスと複合家電クラスは後述する電源の状態を扱う Power Management クラスを持つ。複合家電クラスは単体機器クラスを複数保有し、単体機器クラスごとに Power Management クラスを持っている。複合家電機器はこれらの Power Management クラスを管理するために Power Management クラスを持つ。

また単体家電クラスの中には、一つの単体家電クラスに機能を追加して別の家電クラスを構成するものもある。しかし追加した機能のみから単体家電クラスとして構成することができない場合、複合家電クラスとして扱わない。そこで単体家電クラスから継承して、別の単体家電クラスを作るという構成をとる。図 4.4 は継承して出来る新たな単体家電クラスの例として、洗濯機クラスと洗濯乾燥機クラスの例を示している。また、表 4.5、表 4.6 に提案するシステムにおいて、共通機能を抽出して定義した洗濯機クラスのプロパティおよびメソッドを、表 4.7、表 4.8 に洗濯機クラスにおけるプロパティおよびメソッドに加えて追加される洗濯乾燥機クラスのプロパティおよびメソッドを示す。

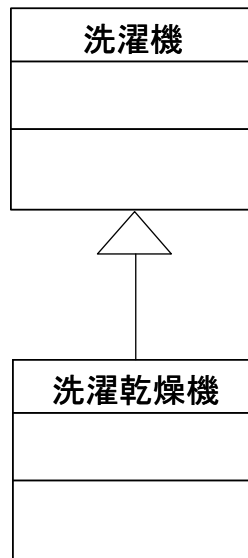


図 4.4: 家電クラスにおける継承

表 4.5: 洗濯機クラスのプロパティ値

プロパティ名称	プロパティ内容
洗濯コース	標準・スピード・毛布・ドライ・つけおき・念入り・漕洗浄
水量	水量指定
洗い	時間指定
すすぎ	回数指定
脱水	時間指定
残り時間	洗濯残り時間
予約時間	洗濯開始予約時間

- 洗濯コースは、標準・スピード・毛布・ドライ・つけおき・念入り・漕洗浄の7つのコースがある。この洗濯コースを設定し、洗濯実行のメソッドを行うと、設定したコースで洗濯を開始する。
- 水量は、洗濯する際の水量を指定するプロパティである。メーカー毎に水量が異なるが、設定できるレベルは4種類と同じであるので、本モデルでは、“最多”、“多”、“少”、“最少”の4段階に設定した。
- 洗いは、洗濯する際の洗いの時間を指定するプロパティである。洗いの時間は、MMの形式で単位を分とする時間をとる。

表 4.6: 洗濯機クラスのメソッド

メソッド名称	メソッド内容
洗濯実行	選択されている洗濯コースを実行する
水量変更	水量を変更する
洗い変更	洗いの回数を変更する
すすぎ変更	すすぎの回数を変更する
脱水変更	脱水の回数を変更する
予約指定	設定されている予約時間で予約を開始する
ふろ水設定	洗濯の際にふろ水を利用する

- すすぎは、洗濯する際のすすぎの回数を指定するプロパティである。すすぎの回数は1～9回のいずれかの値をとる。
- 脱水は、洗濯する際の脱水の時間を指定するプロパティである。脱水の時間は、MMの形式で単位を分とする時間をとる。
- ふろ水設定は、洗濯の際にふろ水を利用するメソッドである。

表 4.7: 洗濯乾燥機クラスのプロパティ値

プロパティ名称	プロパティ内容
乾燥コース	標準・漕乾燥

表 4.8: 洗濯乾燥機クラスのメソッド値

メソッド名称	メソッド内容
乾燥実行	乾燥を実行する

- 乾燥コースは、である。このコースを選択して乾燥実行のメソッドを行うと、このコースで乾燥を開始する。

洗濯乾燥機クラスは、洗濯機クラスに乾燥機能を追加したものである。乾燥機能を持つ単体家電クラスとして衣類乾燥機クラスがあるが、漕乾燥といった洗濯乾燥機クラスに特化した乾燥機能があるため、洗濯機と衣類乾燥機から構成される複合家電機器として見なすことができない。よって洗濯乾燥機クラスは洗濯機クラスから継承して、すなわち乾燥機能を追加してできあがる単体家電クラスとして扱う。

4.2 システムに必要なプロパティ

家電クラスは、家電の機能のみを定義しただけで、家電のクラスのインスタンスを提案するシステム内に生成するには、システムに特化したプロパティを定義しなければならない。そのプロパティを持つものとして、System Object クラスを定義する。System Object クラスはシステムに存在するために必要なプロパティのみを定義しているだけなので、実際にシステムにインスタンスが生成される際は、後述する家電クラスと System Object クラスから継承してできあがる LD Object クラスを用いる。図 4.5 に System Object クラスを示す。SystemObject クラスが持つプロパティとして次のものがある。

- 家電の種類
- 規格名
- アドレス

家電の種類は、LD Object を生成する際に、家電クラスを特定するためのプロパティである。LD Manager においてこのプロパティ値を用いる。規格名は System Object クラスのインスタンスがどの規格における家電を示しているかを特定するためのプロパティである。インスタンスが規格に対して制御メッセージを送信する際、このプロパティ値からゲートウェイを特定し、ゲートウェイに制御メッセージを送信する。アドレスはシステム内で System Object クラスのインスタンスを一意に識別するためのプロパティ値である。これらのプロパティが指定されて、LD Manager がこのクラスのインスタンスを生成する。

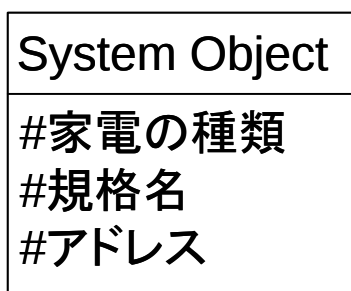


図 4.5: SystemObject クラスのクラス図

4.3 電源の状態の定義 [21]

家電の電源のスイッチは、人間の手で電源の状態を切り替えるハード的なスイッチとリモコンなどで電源の状態を切り替えるソフト的なスイッチの二種類がある。本システムで扱う家電の電源はソフト的なものを扱い、ハード的に電源が切れているものはシステムに存在しないものとして扱う。家電の電源の状態として、On と Off の二種類を扱うものの他に家電の種類によっては On と Off 以外に、予約などを行い、電源を制御できない状態を表す Stand-by を持つ家電も存在する。Stand-by 状態は基本的に家電の操作を受け付けられないが、例外として Standby を解除する操作は受け付ける。この三種類の電源の状態を扱うオブジェクトとして Power Management Object を定義する。電源の状態は状態機械で表され、その状態機械を図 4.6 に示す。また状態遷移図を図 4.7 に示す。Power Management Object が管理する電源の状態は On ・ Off ・ Stand-by の三種類あり、初期状態は Off である。入力集合 Σ は、On 命令 ・ Off 命令 ・ 予約設定 ・ 予約開始 ・ 予約解除の五通りである。On 状態へは、Off 状態からの On 命令、Stand-by 状態からの予約開始のそれぞれが入力されることにより遷移する。Off 状態へは、On 状態からの Off 命令、Stand-by 状態からの解除のそれぞれが入力されることにより遷移する。Stand-by 状態へは、On 状態 ・ Off 状態からの予約設定が入力されることにより遷移する。このように状態遷移を行うことで、On と Off しか電源の状態を持たない家電、Stand-by 状態を持つ家電の両方にも対応できる。

$$M = (Q, \Sigma, \delta, q_0, F)$$

- $Q = \{ON, OFF, STAND-BY\}$
- $\Sigma = \{ON \text{ 命令}, OFF \text{ 命令}, \text{予約設定}, \text{解除}, \text{開始}\}$
- 状態遷移関数
 - $\delta(ON, OFF \text{ 命令}) = OFF$
 - $\delta(ON, \text{予約命令}) = STAND-BY$
 - $\delta(OFF, ON \text{ 命令}) = ON$
 - $\delta(OFF, \text{予約設定}) = STAND-BY$
 - $\delta(STAND-BY, \text{開始}) = ON$
 - $\delta(STAND-BY, \text{解除}) = OFF$

図 4.6: PowerManagement が管理する電源の状態機械

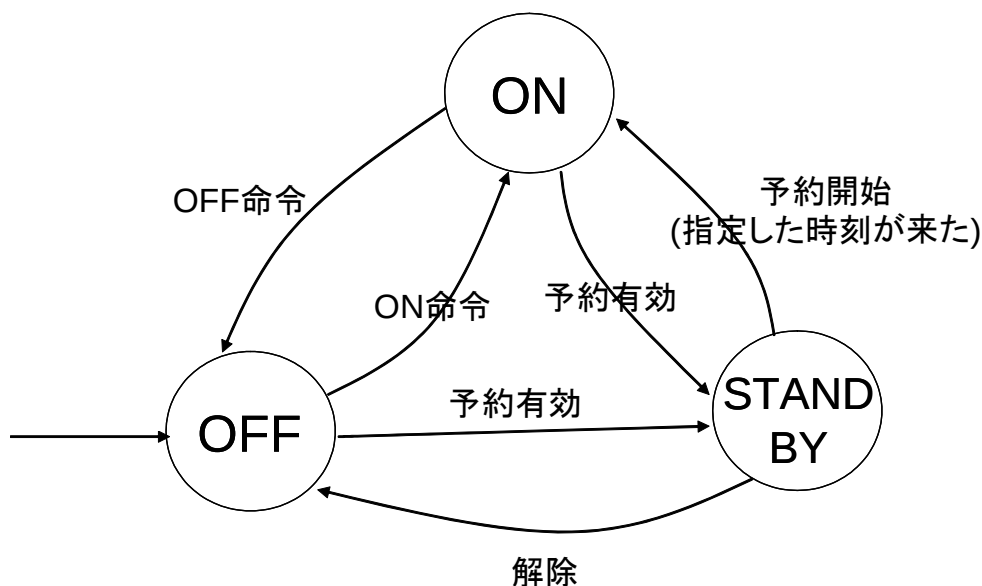


図 4.7: PowerManagement が管理する電源の状態遷移図

PowerManagement Object は、複合家電機器におけるそれぞれの単体家電機器の電源をそれぞれ管理する機能も持つ。すなわち、複合家電クラスであるテレビデオにおいては TV と VTR の各クラスがそれぞれ持つ PowerManagement Object をテレビデオが持つ PowerManagement Object が管理を行う。このことで、家電機器としての排他制御や、片方の電源が切れていても、もう片方の電源が On であるならば、その家電の機能は使用可能であるといったことが定義できる。

PowerManagement Object クラスのクラス図を図 4.8 に示す。単体家電機器を扱う場合は、各クラスの PowerManagement Object のインスタンスおよび各クラスの電源の状態のプロパティは使用しない。状態のプロパティは電源の状態を表し、ON、OFF および STAND-BY の三種類の状態を定義する。この状態の遷移は、メソッドにある状態遷移関数に基づいて遷移される。複合家電を扱う場合は、まず複合家電機器を構成する単体家電

PowerManagement Objectクラス
-状態 +各クラスのPowerManagerment Objectのインスタンス -各クラスの電源の状態
+状態遷移関数 +各クラスのPowerManagement Objectの登録

図 4.8: PowerManagement Object クラス

クラスの PowerManagement Object を複合家電クラスが持つ PowerManagement Object に登録を行う。そのためのメソッドとして”各クラスの PowerManagement Object の登録”がある。そして各単体家電クラスへの電源の状態の操作のメッセージが複合家電クラスに通知されると、単体家電機器のインスタンスにその状態を渡し、単体家電機器のインスタンスの中で状態遷移を行う。そして状態遷移の結果を元に、各クラスの電源の状態のプロパティを変更する。複合家電クラスが持つ”状態”は、全ての単体家電クラスの電源の状態が OFF の場合、状態が OFF に遷移され、どれか一つでも電源の状態が ON であるならば、ON の状態を示すことになる。また複合家電クラスにおける STAND-BY 状態は、各クラスの電源の状態が STAND-BY 状態であるときに起こる状態である。

4.4 LD Object

4.4.1 LD Object クラスの定義

LD Object は各規格の家電を提案するシステム内にインスタンス化するためのクラスである。LD Object クラスは図 4.9 に示すように単体家電クラスと複合家電クラスの総称である家電クラスと System Object クラスの二種のクラスを多重継承してできるクラスである。LD Object は LD Manager によりシステム内に生成される。その際に与えられるパラメータとして家電の種類・規格名・アドレスがある。そのため System Object クラスを継承する。

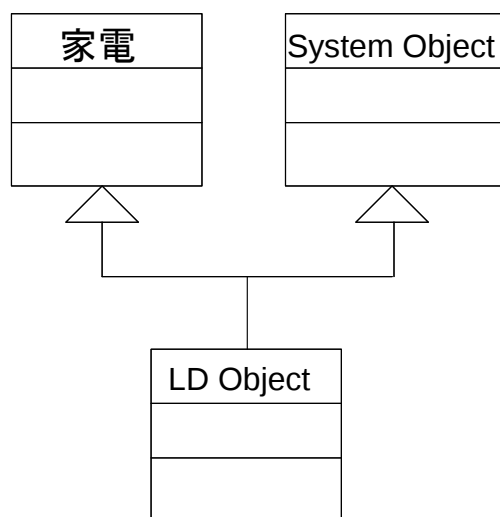


図 4.9: LD Object クラス

多重継承して出来るクラスは家電のクラスが TV クラスならば TV オブジェクトクラス、VTR クラスならば VTR オブジェクトクラスと家電の種類によってできあがるクラ

スが異なる。よってTVオブジェクトやVTRオブジェクトの総称としてLD Objectを定義する。

LD Objectは送られてくる機能番号を処理して、規格毎の家電の制御を行う。図4.10に機能番号が送られてきた時の処理を示す。

1. 各種規格におけるゲートウェイから家電を制御するために、LD Objectに対して機能番号が届く。
2. LD Objectは自身が持つプロパティの中に、機能番号に相当する機能を持っているかどうか確認する。機能番号に相当する機能を持ってない場合、その機能番号は破棄する。
3. LD Objectが機能番号に相当する機能を持っている場合、自身が持つプロパティから規格名を確認し、家電を制御するメッセージに対して応答を返信する規格であることを確認する。応答を返信しない規格である場合、機能番号とその引数からプロパティ値を変更する。そしてゲートウェイに対して機能番号とその引数を送信し、家電の制御を行う。
4. 制御に対する応答を返信する規格である場合、プロパティ値の変更を保留し、通知された機能番号をその規格のゲートウェイに送信し、家電の制御を行う。
5. 制御メッセージを送信した規格の家電から応答がゲートウェイで受信されると、機能番号に変換され、LD Objectに通知される。
6. LD Objectがその機能番号を受け取ると、保留していたプロパティ値の変更を行う。その際、プロパティ値の変更を行う機能番号が電源の操作に関するものかどうかを確認する。電源の操作に関する機能番号である場合、自身が持つPowerManagement Objectの状態遷移関数を元に状態を遷移させる。その他の機能番号ならば、引数に基づいてプロパティ値の変更を行う。

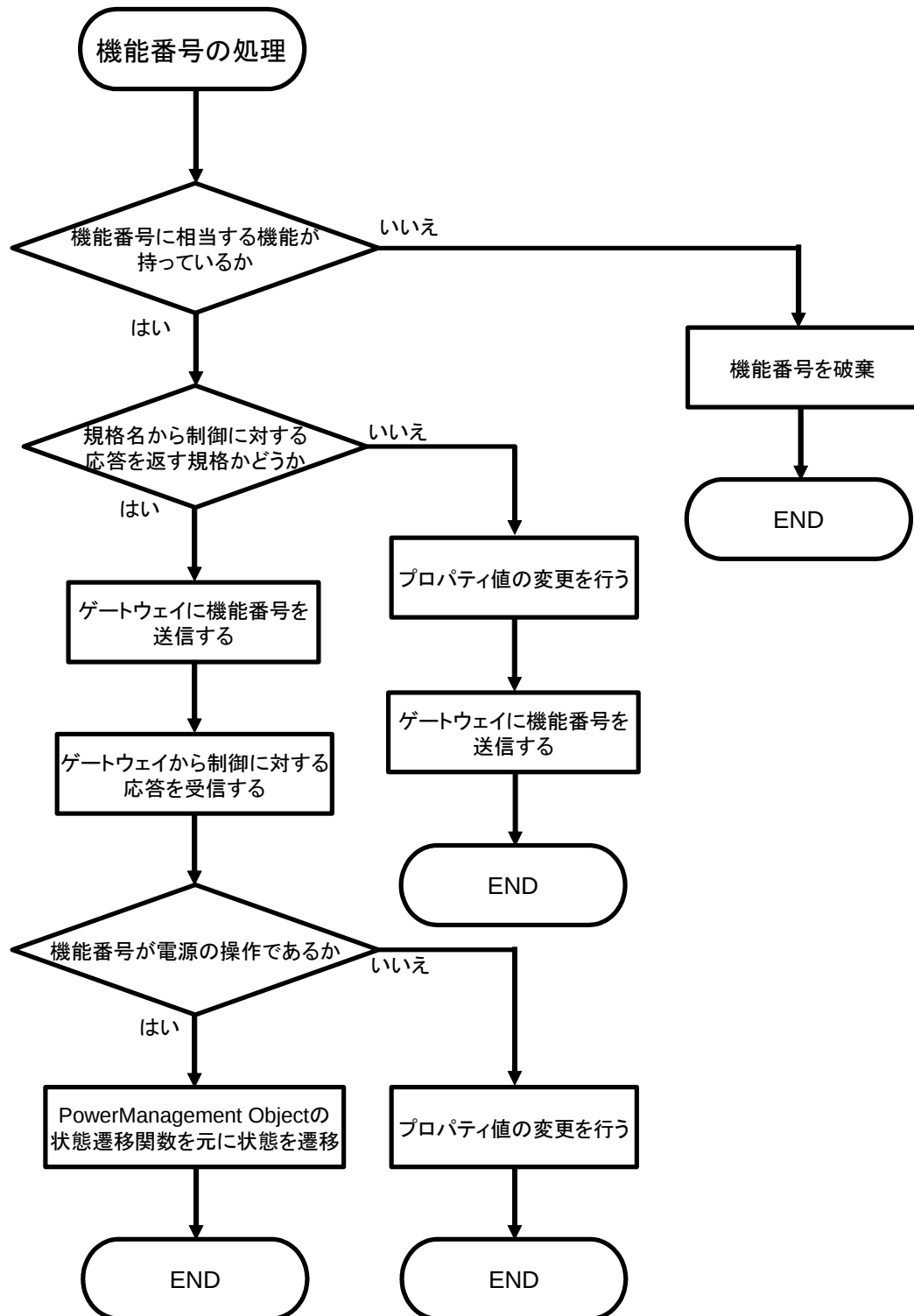


図 4.10: LD Object における機能番号の処理

第5章 LD Manager

本章では、LD Object の管理・生成を行う LD Manager について説明する。

5.1 LD Manager クラス

LD Manager は次の機能を持つ。

- 各種ホームネットワーク規格における家電の検索
- LD Object の生成
- LD Object のアドレスの生成
- システムに存在する LD Object の問合せに対する応答

LD Object の生成は、はじめに LD Manager が他規格に対して家電の検索を行う。LD Manager が各種ゲートウェイに対して他規格の家電の検索の機能番号 (3.1) を送り、ゲートウェイは各種規格に検索メッセージを送る。各種規格から家電の情報を受け取ると、ゲートウェイは他規格の家電の応答の機能番号とそれに対する引数 (家電の情報+規格名) に変換し、LD Manager に送る。LD Manager はこの機能番号と引数を受け取ると、まず LD Object のアドレスを生成する。そして LD Object のアドレスと引数として受け取った家電の情報と規格名を用いて LD Object を生成する。

表 5.1: LD Object アドレス表

LD Object のアドレス	家電の種類
0x01	TV
0x02	VTR
0x03	エアコン

システムに存在する LD Object の問い合わせに対する応答は、他規格から家電の検索のメッセージを各種ゲートウェイが受け取り、この検索に相当する機能番号を LD Manager に送られる。LD Manager がこれらの情報を受け取ると、LD Object アドレス表にある LD Object のアドレスと家電の種類の組を引数とする他規格から家電の応答の機能番号をゲートウェイに送り、問い合わせに応答する。

第6章 各種 Gateway

本章では、提案するシステムと各種ホームネットワーク規格とをつなぐゲートウェイの詳細、また LD Object の登録、家電の制御のシーケンスについて説明する。

6.1 Gateway クラス

ゲートウェイは図 6.1 に示す Gateway クラスで表される。Gateway クラスが持つプロパティとして、アドレス変換表・機能番号変換表を持つ。アドレス変換表は LD Object のアドレスと各種規格で定義されているアドレスと対応表である。機能番号変換表は機能番号を各種規格で定義されている制御命令に変換するための表である。Gateway クラスが持つメソッドは家電を LD Object として登録する際に LD Object のアドレスと各種規格で定義されているアドレスとを対応づける。また ECHONET から X10 を制御する際に X10 の機器のアドレスを ECHONET アドレスとして見せるといったように他規格の機器に対して規格固有のアドレスを生成する。また機能番号を各種規格で定義されている機能に変換を行う。またその逆も行い、各種規格に送信する。家電の情報・制御情報を LD Manager や LD Object に送る。

Gateway
#アドレス変換表 #機能変換表
#規格固有のアドレスを生成() +LDオブジェクトのアドレスを登録() +機能番号を規格固有の命令へ変換() +規格固有の命令を機能番号へ変換() +LD Managerに家電の情報を送る() +LDオブジェクトのアドレスと対応する規格固有のアドレスを返す() +規格固有のアドレスと対応するLDオブジェクトのアドレスを返す() +命令を送信()

図 6.1: ゲートウェイクラス

各種規格のゲートウェイは図 6.2 に示すように Gateway クラスから派生して作られる。各種規格のゲートウェイは各々が独自の機能を持ち、次節以降で詳細を示す。

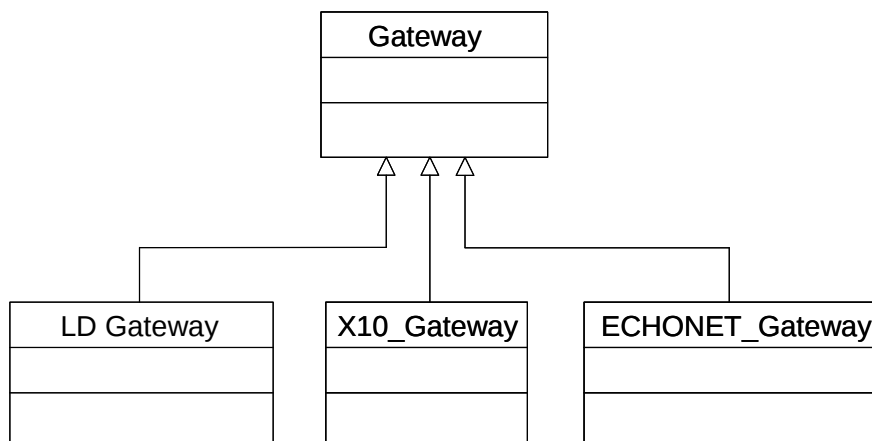


図 6.2: 各種ゲートウェイクラス

6.2 各種規格 Gateway の設計

6.2.1 LD における Gateway

LD におけるホームネットワークの概要 [2]

LD を情報家電に見立てて、ホームネットワークを構築するためのシステムを図 6.3 に示す。リモコンによる LD への制御情報や温度などのセンサといった LD の動作に関係する環境情報を監視するものを観測デバイスと定義する。そして観測デバイスによって状態を観測されるデバイスを被観測デバイスと定義する。観測デバイスはこれらの情報を監視し、得られた情報を処理ユニットで処理し、被観測デバイスの内部状態を取得する。そして、取得した内部状態に基づき、処理ユニットにおいて被観測デバイスの制御が可能であるリモコンを用いて機器の制御を行うことで、連動動作など様々なサービスを実現する。

- 被観測デバイス
状態を観測されるデバイスであり，TV, カーテン, エアコンなどの LD が含まれる．
- 観測関連デバイス
状態を観測する関連デバイスであり，処理ユニット，フィルタ，センサなどが含まれる．状態の管理，信号の切り換え，アナログ信号の取得・処理などを行う．

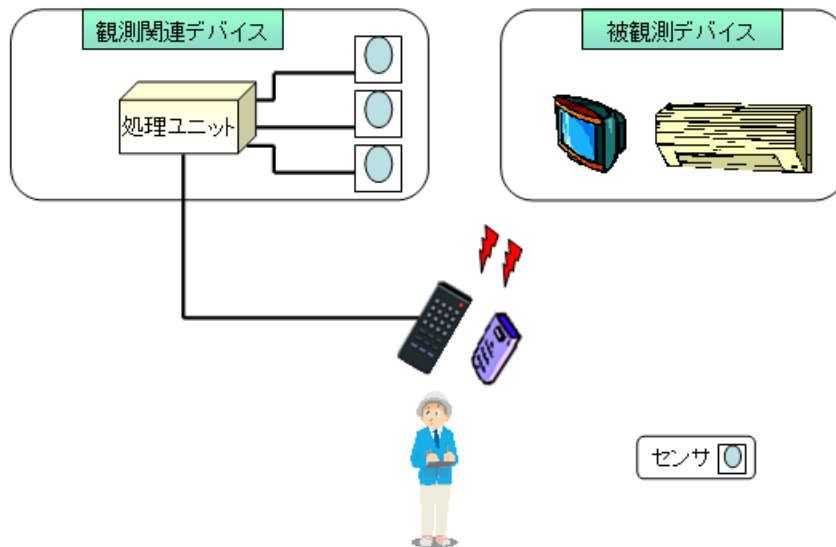


図 6.3: LD におけるホームネットワークシステム構成図

処理ユニットは、LD を用いたホームネットワークにおいてすべての中心となる機器であり、家庭内に唯一存在する。処理ユニットは、

- 機器情報の管理
- 機器制御の実行
- サービス情報の掲示
- LD に対する命令の捕捉
- センサの管理
- センサ情報の処理
- 他システムとの通信

を行う。処理ユニット内に、機器の連動動作など高度なサービスを持たせることにより、機器の相互運用が可能となる。また、環境型センサと LD を組み合わせたサービスを実現することも可能である。

LD によるホームネットワークシステムでは、LD を複数の部分状態機械の集合体としてとらえ、複数の部分状態機械が並行して動作する状態機械としてモデル化を行っている。これにより状態機械のモデル化が容易になり、同様の機能を持つ複数の LD で部分状態機械の流用が可能となる。部分状態機械の状態を機器の持つ状態としてとらえ、部分状態機械が提供する機能を機器の持つ情報とすることで、各機器を機器資源として扱うことが可能となる。

表 4.1 に代表的な機器と機器が持つ代表的な部分状態機械を示す。表にない機器についても部分状態機械を組み合わせることで、モデル化を行っている。例えば、複数の機能が組み合わさっているテレビデオのような機器は、複数の機器が組合わさって構成されていると考え、両方の機器の部分状態機械をすべて保持する状態機械を構成する。テレビデオの場合、TV と VTR が組み合わさって構成されていると考えるので、保持する部分状態機械は、power/tuner/amp/aux/player・recorder/となる。また、ラジオやコンボなどオーディオ機器は、power/amp/aux などから構成されることができる。

appliance	部分状態機械
TV	power/tuner/amp/aux/
VTR	power/tuner/amp/aux/player・recorder/
DVDplayer	power/tuner/amp/aux/player・recorder/
DVDrecorder	power/tuner/amp/aux/player・recorder/
CDplayer	power/tuner/amp/aux/player・recorder/
CDrecorder	power/tuner/amp/aux/player・recorder/
エアコン	power/temp/humid/mode_drive/mode_angle/mode_wind/
送風機	power/temp/humid/mode_drive/mode_angle/mode_wind/
換気扇	power/temp/humid/mode_drive/mode_angle/mode_wind/
加湿器	power/temp/humid/mode_drive/mode_angle/mode_wind/
雨戸	power/open_close/illumination
カーテン	power/open_close/illumination
ブラインド	power/open_close/illumination
シャッター	power/open_close/illumination
ランプ・蛍光灯	power/illumination
電気ポット	power/

表 6.1: 機器と対応する部分状態機械

LD Gateway の設計

LD Gateway とは、LD と本システムとを接続するゲートウェイのことをいう。LD と家電の情報をやりとりするメッセージタイプを図 6.4 に示す。

inquiry

Message Type

response

Message Type	LD ID	Appliance	Maker	Model	State Machines
-----------------	-------	-----------	-------	-------	-------------------

control

Message Type	LD ID	State
-----------------	-------	-------

notify

Message Type	LD ID	State
-----------------	-------	-------

図 6.4: LD とやりとりするメッセージタイプ

メッセージタイプは、inquiry・response・control・notify の 4 種類があり、それぞれ家電の問合せ・問合せに対する応答・家電の制御・制御に対する応答を表す。LD Gateway が inquiry メッセージタイプを受け取ると、システムに存在する LD Object を LD で定義されている家電オブジェクトに変換し、response のメッセージタイプとして LD に返す。また control メッセージタイプを受け取ると、LD Object に対して制御メッセージを送り、制御が完了すると notify メッセージタイプを返す。

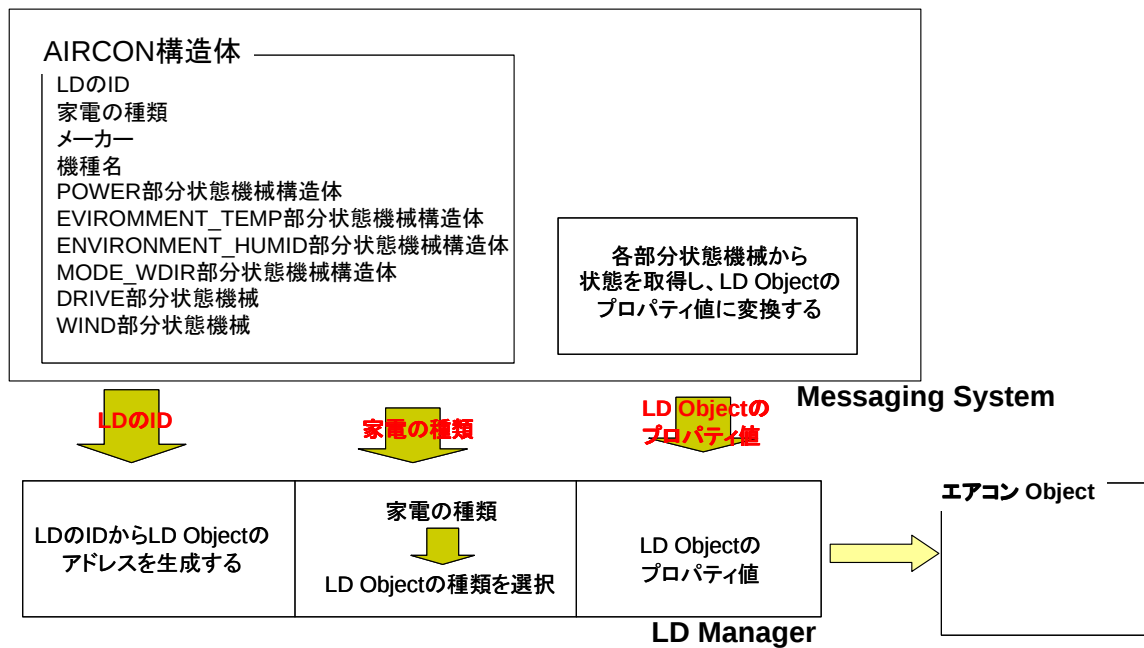


図 6.5: LD を LD Object へ変換する (ex. エアコン)

提案システムとのシーケンス

LD の登録のシーケンスを図 6.6 に示す。

1. LD の登録は、LD Manager が LD の検索をすることで行われる。まず LD Manager が LD Gateway に対して検索を表す機能番号を送る。
2. LD Gateway では機能番号変換表を元に inquiry メッセージタイプに変換し、LD に送信する。
3. LD から応答である response が返ってくると、response メッセージの中にある部分状態機械の情報から家電の種類・家電の状態に変換する。
4. この情報を LD Manager に送る。LD Manager ではこれらの情報を元に LD Object の生成を行う。
5. LD Object のアドレスを生成する。そしてそのアドレスと家電の情報から LD Object を生成する。
6. 生成したアドレスが LD Gateway に送られてくるので、アドレス変換表に登録する。

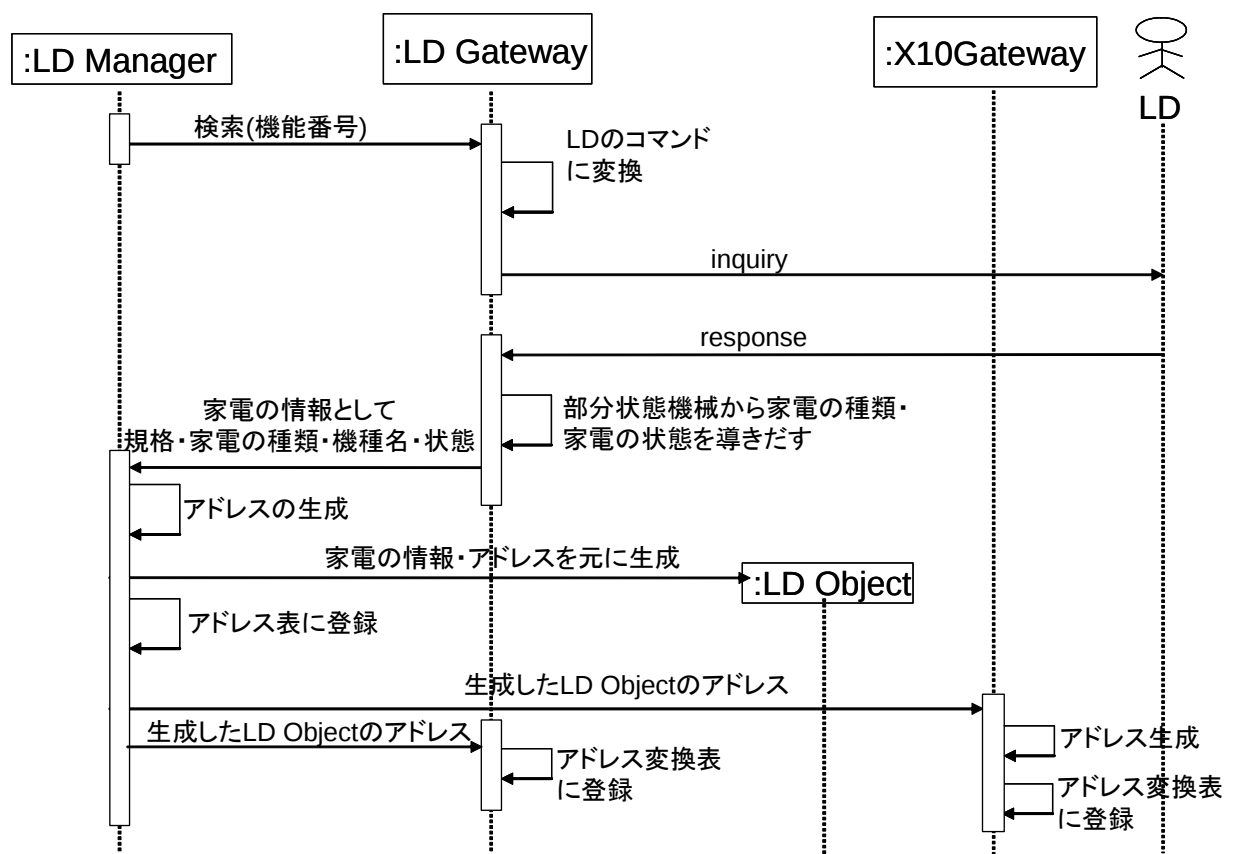


図 6.6: LD の登録

LD から inquiry メッセージが届いた時のシーケンスを図 6.7 に示す。

1. LD から LD Gateway に inquiry が届くと、機能番号変換表を元に機能番号に変換する。
2. システムに存在する LD Object のリストを取得するためにその機能番号を LD Manager に送る。
3. LD Manager はこの機能番号が届くと、LD Object アドレス表にある LD Object のアドレスを LD Gateway に返す。
4. LD Gateway は LD Object のアドレスを指定して、LD Object に対して家電の情報を問い合わせる。
5. 家電の情報が届くと、その情報を元に LD で定義されている部分状態機械を生成する。部分状態機械と家電の情報から response メッセージタイプを生成し、LD に送信する。

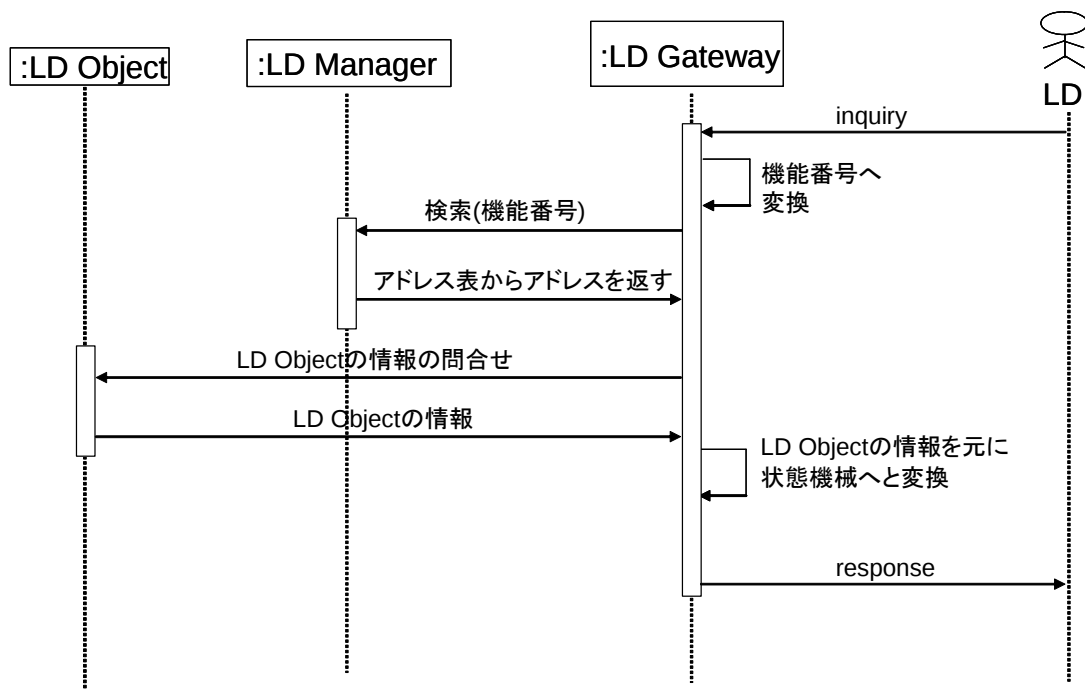


図 6.7: LD への応答

他規格から LD を制御する時のシーケンスを図 6.8 に示す。

1. 他の規格から LD の制御が行われるとその規格のゲートウェイから LD Object に機能番号とそれに対する引数が届く。
2. LD Object はその情報が届くと、実際に LD を制御するために LD Gateway に対して機能番号とその引数を送る。
3. LD Gateway はその情報から control メッセージタイプを生成し、LD に送信する。
4. LD で実際に制御が行われ、control に対する確認通知である notify メッセージタイプが LD Gateway に送られる。
5. notify メッセージタイプが届くと、家電の制御が行われたことを LD Object に伝えるために機能番号に変換する。
6. LD Object にその機能番号が届くと LD Object はプロパティ値を変更する。

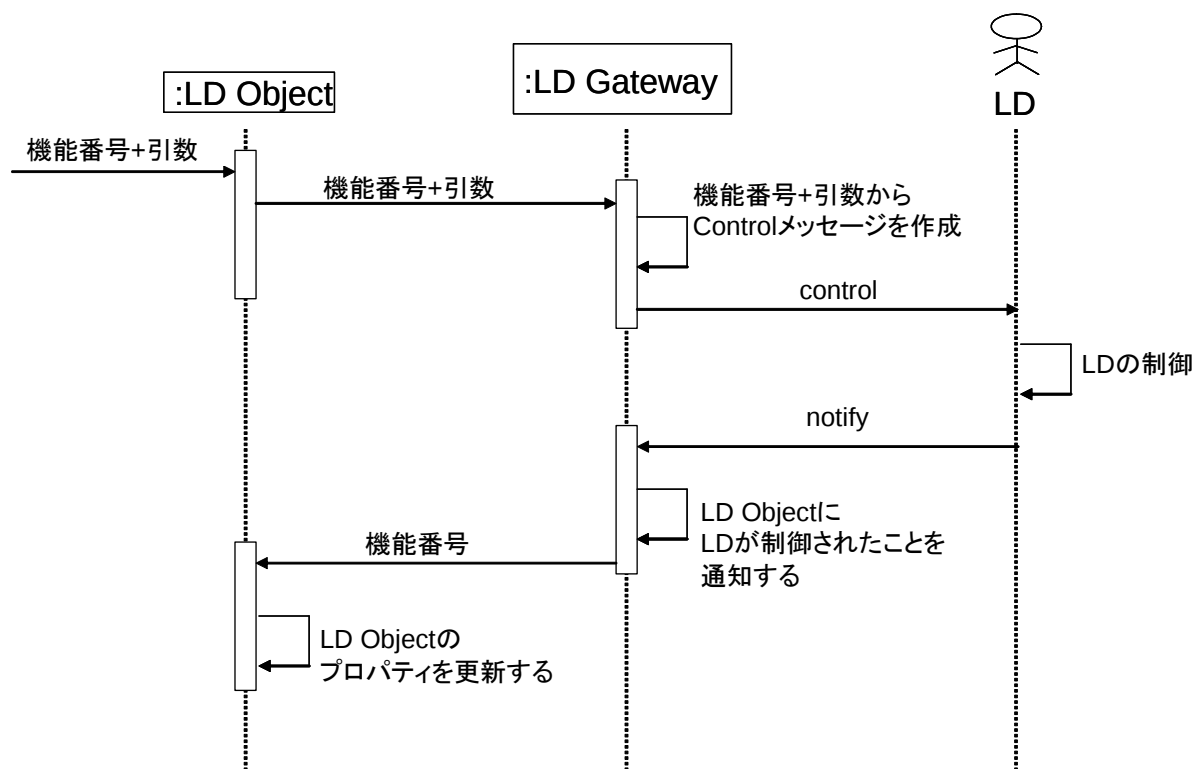


図 6.8: 他規格から LD を制御する際のシーケンス

LD から他規格の家電を制御する際のシーケンスを図 6.9、図 6.10 に示す。他規格によって家電が制御された場合に確認通知が返ってくるものと返ってこないものと考えられるので、このシーケンスも二通り考えられる。図 6.9 は他規格から確認通知が返ってこないものを示している。

1. LD から control メッセージが届くと、control メッセージタイプの中の State から機能番号と引数に変換する。
2. アドレス変換表から LD Object のアドレスを指定し、機能番号と引数を送る。
3. LD Object は機能番号と引数を他規格のゲートウェイに送り、ゲートウェイからの応答を待たずに LD Object のプロパティ値の変更を行う。
4. ゲートウェイは機能番号と引数を受け取ると機能番号変換表から規格独自のメッセージに変換し、メッセージを送信する。

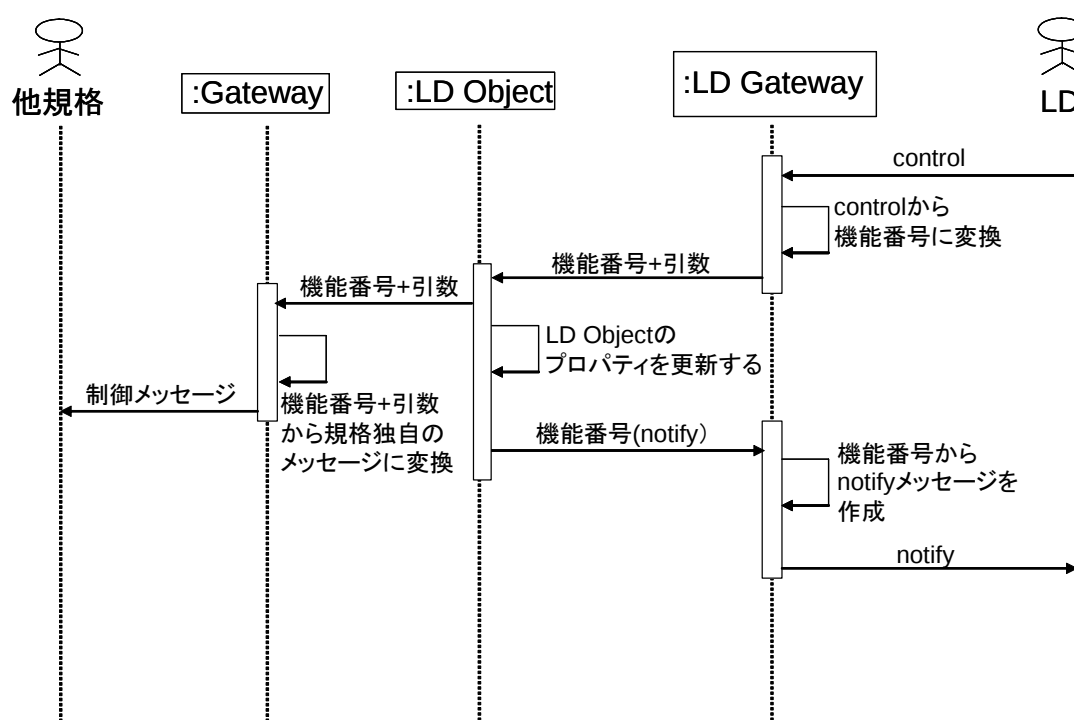


図 6.9: LD から他規格の家電を制御する際のシーケンス (他規格から確認通知が返ってこない場合)

次に他規格から確認通知が返ってくるシーケンスを図 6.10 に示す。

1. 図 6.9 の時と同様に、LD Object が機能番号と引数を受け取る。LD Object は機能番号と引数をゲートウェイに送るが、他規格から確認通知が来るまでプロパティ値の変更を保留する。
2. ゲートウェイは機能番号と引数を規格独自のメッセージに変換・送信を行い、確認通知を待つ。
3. 確認通知がゲートウェイに届くと、機能番号変換表を元に機能番号に変換し、LD Object へ通知する。
4. LD Object は通知を受け取ると、保留していたプロパティ値の変更を行う。
5. LD に対して確認通知を行うために通知に対する機能番号を LD Gateway に送る。
6. LD Gateway は機能番号を notify メッセージタイプに変換し、LD に送信する。

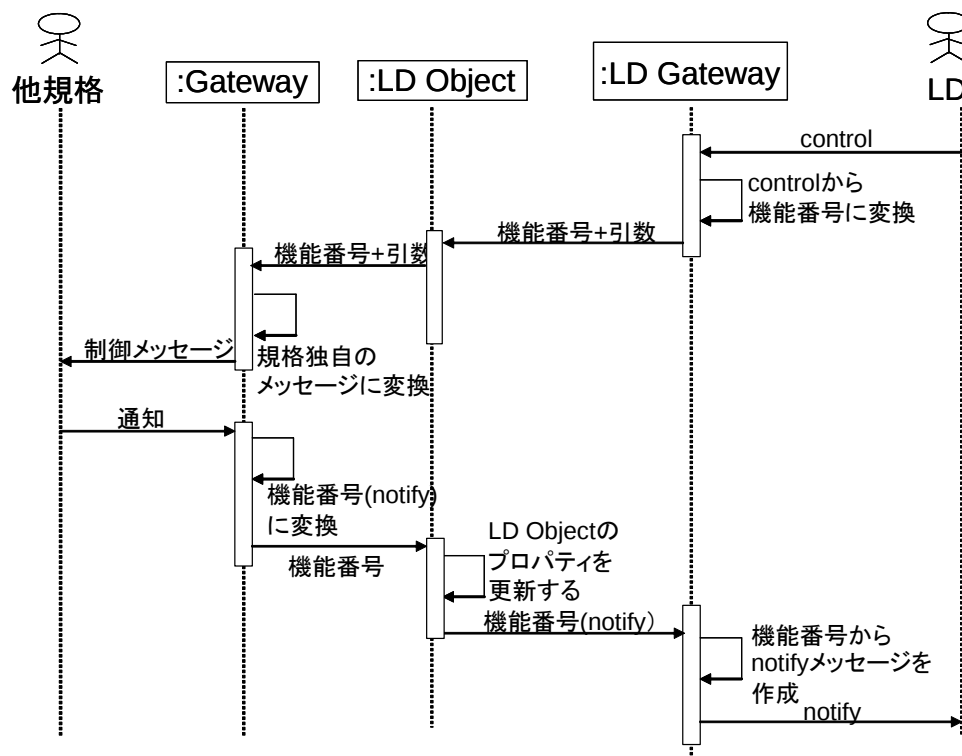


図 6.10: LD から他規格の家電を制御する際のシーケンス (他規格から確認通知が返ってくる場合)

6.2.2 ECHONET Gateway[6]

本節では、ECHONETにおけるゲートウェイの設計を行う。まずECHONETに関する機器の立ち上げ、制御に関するシーケンスの詳細を示し、続いて提案するシステムとのゲートウェイを設計する。

ECHONETにおける機器登録・制御に関するシーケンス

ECHONETでは、ECHONETオブジェクトを操作することで機器の制御や情報のやり取りを実現している。提案するシステムとのゲートウェイを設計するため、ECHONETノードの起動・ECHONETにおける機器の制御といったECHONETで規定されている機器制御をするためのプロセスを以下より示す。

・ECHONETオブジェクト制御のシーケンス

ECHONETにおける機器制御は、ECHONETオブジェクトのプロパティに対するサービス (ESV : ECHONET サービス) 指定によって行われる。ここでいうサービスとは、EPC(ECHONET プロパティ) に対する操作を指定するものである。ECHONETオブジェクトは機器オブジェクト・プロファイルオブジェクト・通信定義オブジェクト・サービスオブジェクトの四種類あり、制御電文をECHONETノードに送信する際には、ECHONETアドレスの他に、ECHONETオブジェクトの種類を指定し、そのオブジェクトに対するサービスとプロパティを指定する。

・サービス内容に関するシーケンス

サービスとは、EPC(ECHONET プロパティ) に対する操作を指定するものである。操作としては、「要求」・「応答」(応答/不可応答)・「通知」の三種類がある。さらに「応答として、指定されたプロパティが存在するときの「応答」と、指定された「要求」対象のプロパティが存在しない場合及び指定のサービスが処理できない場合の「不可応答」がある。「応答」は、応答を必要とする「要求」を受けての返信の位置づけとして、宛先ECHONETオブジェクトにより指定されたオブジェクトが存在する場合には、「応答」か「不可応答」(指定された処理を受理できないか、或いはオブジェクトは存在するがプロパティが存在しない、或いはプロパティは存在するが指定の配列要素が存在しない場合の応答) を返すものとする。応答不可な「要求」の場合、及び指定されたオブジェクトが存在しない場合には、「応答」は行わないものとする。「通知」は、自発的に自身のプロパティの情報を送信するものと、通知要求の応答として送信するものがあるが、区別は行わない。

・ECHONETノード立ち上げ時のシーケンス

ECHONETノードの立ち上げは、まず自己を認識・規定するためのECHONETアドレスの取得から始まる。ECHONETアドレスは、NodeIDとNetIDから構成され、NodeIDはMACアドレスから変換され、一意に対応付けられ通信ミドルウェアが動作開始する時

点で取得済みとなる。NetID は、サブネット内でユニークとなるようにルータからふられるアドレスである。NetID の取得処理は、以下の二通りを規定している。

- コールドスタート時のシーケンス
- ウォームスタート時のシーケンス

・ コールドスタート時のシーケンス

ECHONET ノードにおいて NodeID の設定が完了した後、ECHONET ルータから NetID を取得するため、図 6.11 に示すシーケンスを行う。送信元 ECHONET アドレスは NetID を指定無しのデフォルト値 (0x00) を指定し、宛先 ECHONET アドレスを自サブネット内の全ノードに対して一斉同報するように指定する。宛先 ECHONET オブジェクトはルータプロファイルオブジェクトを指定する。サービスを読み出し要求を指定し、読み出すプロパティは NetID プロパティを指定する。これらの値から電文を生成し、一斉同報を行う。受信待ちのタイムアウトである T2(60s：設計指針) 以内に、起動済みの ECHONET ルータから、NetID が返ってくる場合、NetID の設定を行う。そして起動済みの ECHONET ルータをデフォルトルータとして、ルータの ECHONET アドレスを設定する。NetID と NodeID から ECHONET アドレスの設定が完了すると、ECHONET ノードはドメイン内に通知を行う。その際、宛先 ECHONET アドレスをドメイン内に一斉同報するためのアドレスを指定する。送信元 ECHONET オブジェクトはノードプロファイルオブジェクトを指定、サービスとして通知を指定、プロパティ値をインスタンス変化クラスアナウンスプロパティを指定する。これらの値から電文を生成し、一斉同報を行う。NetID の応答に関して T2 を経過しても受信がない場合、NetID として 0x00 を設定する。

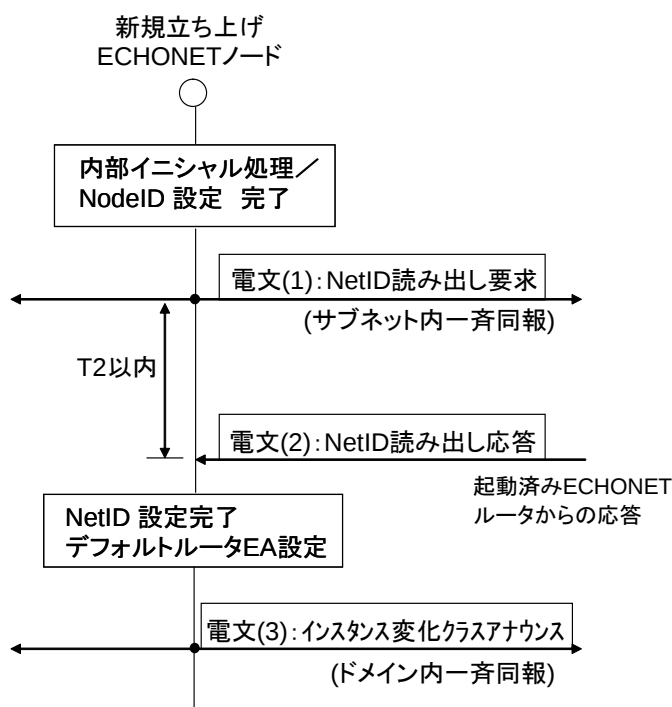


図 6.11: ECHONET におけるコールドスタート

・ウォームスタート時のシーケンス

ECHONET アドレス等の情報を保持した状態で、ECHONET から切り離されるか或いは休止していた状態から、ECHONET に再接続或いは再起動した場合に、ECHONET ノードはウォームスタートを行う。ウォームスタートする場合、図 6.12 と図 6.13 に示すシーケンスを行う。送信元 ECHONET アドレスとして既に保持している ECHONET アドレスを指定し、宛先 ECHONET アドレスとしてデフォルトルータを指定する。宛先 ECHONET オブジェクトはルータプロファイルオブジェクトを指定する。サービスは読み出し要求を指定し、読み出すプロパティは NetID を指定する。これらの値から電文を生成し、デフォルトルータへ送信する。その結果として返信される NetID が自身の持つ NetID 情報と一致している場合、図 6.12 に示すシーケンスを行う。NetID 読み出し要求に対する応答がない場合、図 6.13 に示すシーケンスを行う。返信された NetID の情報が自身の持つ情報と一致しない場合、ECHONET ノードはコールドスタート時の処理を行う。NetID が返信され、自身の持つ NetID と一致した場合、図 6.12 に示すようにインスタンス変化クラスアナウンスをドメイン内に一斉同報する。その際、宛先 ECHONET アドレスとしてドメイン内一斉同報を指定する。また、送信元 ECHONET オブジェクトとしてノードプロファイルオブジェクトを指定し、サービスとして通知し、通知するプロパティはインスタンス変化クラスアナウンスを指定する。

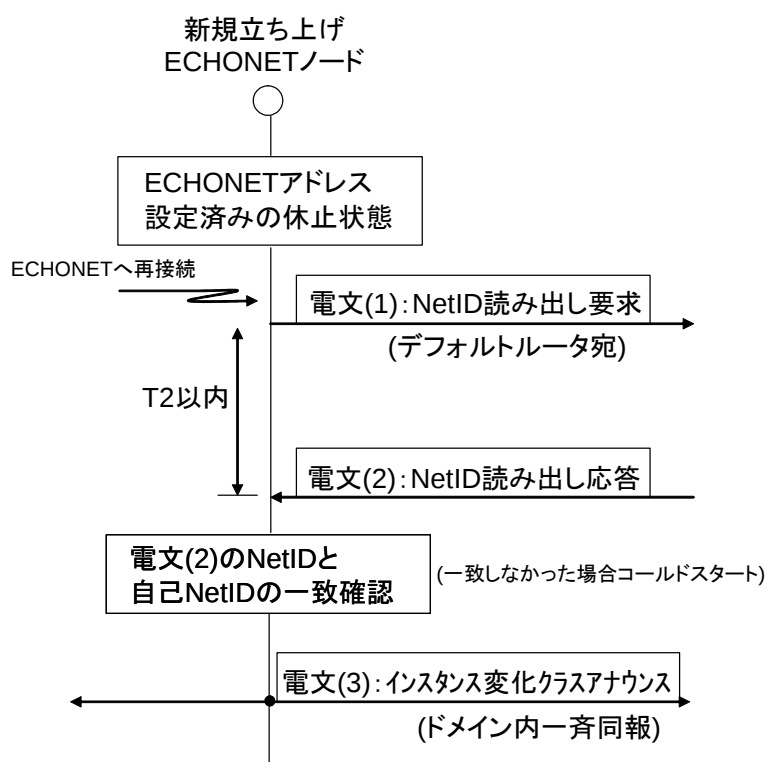


図 6.12: ECHONET におけるウォームスタート (1)

デフォルトルータから NetID 読み出し要求に対する応答が T2 を経過しても返ってこない場合、図 6.13 に示すシーケンスを行う。T2 経過しても返信がない場合、デフォルトルータと同じ電文をサブネット内に一斉同報する。サブネット内のルータから返信が会った場合、返信された NetID と自身の持つ NetID が一致することを確認する。NetID が一致する場合、宛先 ECHONET アドレスはドメイン内一斉同報を指定し、宛先 ECHONET オブジェクトはノードプロファイルオブジェクトを指定する。またサービスとして通知を指定し、通知するプロパティをインスタンス変化クラスアナウンスを指定する。これらの値から電文を生成し、一斉同報を行う。ルータからの返信がない場合は、自身が保持している NetID で起動を行う。返信された NetID と自身が保持する NetID が一致しなかった場合、コールドスタートを行う。

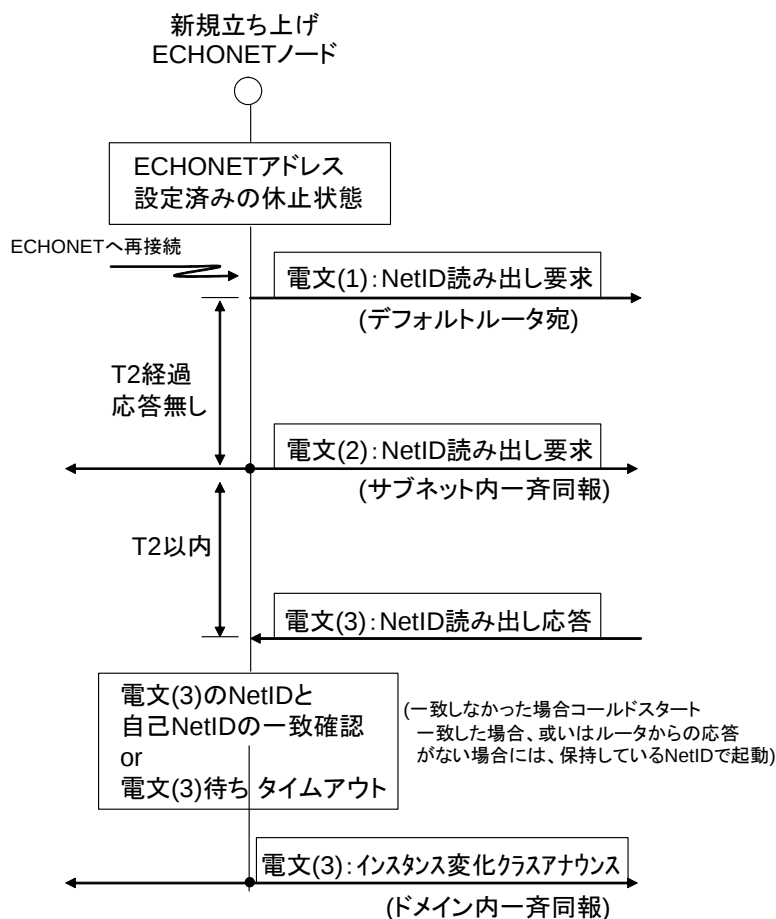


図 6.13: ECHONET におけるウォームスタート (2)

・システムへの登録

ECHONET における家電の登録は当該 ECHONET ノードのアプリケーション情報が ECHONET アドレスとともに記録保持されることをいう。いずれかの ECHONET ノードに連動情報、またはインスタンスリストとして生成される。

1. ECHONET アドレスの認識
2. 当該 ECHONET ノードの機能認識 (ノードプロファイルクラス情報から機能、個体識別情報を取得)
3. 通信相手として他機器インスタンスリストへの登録または、連動情報 (トリガ、アクションのいずれか) として登録

或いは

1. 機器の設定手段、ネットワークを介した設定器などによるインスタンスリスト、連動情報への登録

など、いずれかのプロセスを経てシステムへの登録となる。

提案システムとのシーケンス

ECHONET ゲートウェイは前述の ECHONET における機器の起動、制御のシーケンスを元に ECHONET ゲートウェイを設計する。ECHONET ゲートウェイが持つ機能として、LD Manager および LD Object から通知される機能番号とその引数から ECHONET 電文を生成し ECHONET への通知、また ECHONET 電文から家電の情報や ECHONET アドレスの抽出を行い、これらの情報から機能番号と引数に変換し、LD Manager および LD Object への通知、また ECHONET アドレスと LD Object のアドレスとを対応づけるためにアドレス変換表を用意し、電文の生成および受信の際のアドレスの変換を実装する。

ECHONET ゲートウェイが起動すると、まず各種規格の家電を仮想的に ECHONET 内の機器として見せるため、すなわちサブネット内に存在するように見せるため、ECHONET ルータから NetID を取得する。このシーケンスは前述のノードのコールドスタートのシーケンスに基づいて行う。

ECHONET ゲートウェイでは ECHONET アドレスと LD Object のアドレスと変換し、情報のやり取りを行うためアドレス表が必要となる。ECHONET ゲートウェイが持つアドレス変換表についての詳細を示す。ECHONET においてオブジェクトを制御する際、特定するものとして ECHONET アドレスと ECHONET オブジェクトがある。ECHONET オブジェクトは、“クラスグループコード”、“クラスコード”、“インスタンスコード”によって構成され、一意に識別される。クラスグループコードはクラスをグループ化したものである。クラスコードはクラスグループコード単位に規定され、クラスを表すコードである。ECHONET では、クラスグループコードとクラスコードの組み合わせでクラスを一意に識別する。インスタンスコードは実際に ECHONET オブジェクトが存在する場合

の実体を表したコードである。同一クラスのオブジェクトが1つのノードに複数存在する場合は、複数のインスタンスが存在することになるので、このインスタンスコードを用いて一意に識別する。他規格における LD Object は、機器オブジェクトのインスタンスは唯一となる。よって他規格における LD Object に対応した機器オブジェクトは、“クラスグループコード”、“クラスコード”、“インスタンスコード (0x01)”として規定する。またノードプロファイルオブジェクトに関しても、他規格における LD Object において一つ存在すればいいので、他規格における LD Object に対応するノードプロファイルオブジェクトの値は 0x0EF01 として規定する。ECHONET 内に存在する機器についてのインスタンスはインスタンスコードの識別を行う。

ECHONET アドレス	機器オブジェクト	LD Object のアドレス
0x0511	0x013001	0x01
0x05A0	0x01340D	0x02
0x05B2	0x026010	0x03
0x05B5	0x029001	0x04
0x05CC	0x03B203	0x05

次に提案システムにおける ECHONET 機器の登録のシーケンスを図 6.14 に示す。ECHONET 機器の登録は LD Manager が ECHONET に対して ECHONET 機器の検索を行うことで始まる。LD Manager が ECHONET ゲートウェイに対して他規格の家電の検索の機能番号 (0x02) を送る。ECHONET ゲートウェイはこの機能番号を受け取ると、ECHONET に対して家電の検索を行うため、次のようなシーケンスを行う。

1. LD Manager から ECHONET ゲートウェイに対して他規格の家電の検索の機能番号 (0x02) を送る。
2. ECHONET ノードの検索を行う。ノードの検索は、各ノードに対して ECHONET アドレスを通知するように、ドメイン内にこの電文を一斉同報する。その際に指定する値として、宛先 ECHONET アドレス (DEA) はドメイン内一斉同報を示す 0xFF を、宛先 ECHONET オブジェクト (DEOJ) はノードプロファイルオブジェクトを示す 0x0EF01 を、ECHONET サービス (ESV) は読み出し要求を示す 0x62 を、指定する ECHONET プロパティ (EPC) は ECHONET アドレスを示す 0xE0 をそれぞれ指定する。
3. ECHONET アドレスの読み出し応答として各ノードから次のような電文が返信される。送信元 ECHONET オブジェクト (SEOJ) としてノードプロファイルオブジェクトを示す 0x0EF01、ESV として通知を示す 0x73、EPC として各ノードの ECHONET アドレスが電文に含まれ返信される。

4. ECHONET ゲートウェイは次に各ノードに対してノードが保有する機器オブジェクトのリストを問い合わせる。その際に指定する値として、DEA は返信された ECHONET アドレスを、DEOJ はノードプロファイルオブジェクトを示す 0x0EF01 を、ESV は読み出し要求を示す 0x62 を、指定する EPC は自ノードとしてノードプロファイルを除く、公開しているクラスのリストを表す自ノードクラスリスト S を示す 0xD7 をそれぞれ指定する。
5. 機器オブジェクトの読み出し応答として各ノードから次のような電文が返信される。送信元 ECHONET アドレス (SEA) として各ノードの ECHONET アドレス、SEOJ としてノードプロファイルオブジェクトを示す 0x0EF01、ESV として通知を示す 0x73、EPC としてノードが保有するクラスリストが返信される。
6. ECHONET ゲートウェイでは、ノードが保有するクラスリストから機器オブジェクトを抽出し、その機器オブジェクトを示す値から家電の種類を特定する。
7. ECHONET ゲートウェイは、LD Manager に対して他規格の家電の応答の機能番号 (0x03) とその機能番号に対する引数として、機器オブジェクトから特定した家電の種類と規格名 (この場合 ECHONET) を指定して通知する。
8. LD Manager は家電の種類と規格名が通知されると、まず LD Object の生成を行うため、LD Object のアドレスを生成する。そして家電の種類、規格名およびアドレスから LD Object を生成する。
9. LD Manager は、生成した LD Object のアドレスを ECHONET ゲートウェイに対して通知する。ECHONET ゲートウェイは LD Object のアドレスを取得すると、アドレス変換表に ECHONET アドレスと LD Object のアドレスを追加する。LD Manager は自身が保有する LD Object アドレス表に LD Object のアドレスと家電の種類を追加する。
10. 次に LD Object は、家電の状態を問い合わせるために、家電の状態の問い合わせの機能番号 (0x05) とその引数として LD Object が持つ状態の種類をそれぞれ指定して、状態の数だけ ECHONET ゲートウェイに通知する。
11. ECHONET ゲートウェイは各ノードが保有する機器オブジェクトに対して、家電の状態を問い合わせる。ECHONET ノードを特定するため、LD Object のアドレスからアドレス変換表を用いて ECHONET アドレスを取得する。指定する値として、DEA は各ノードの ECHONET アドレス、DEOJ は各機器オブジェクトを表す値、ESV は読み出し要求を示す 0x62、EPC は家電の状態から機器オブジェクトが持つ家電のプロパティに変換しその値をそれぞれ指定する。
12. 各状態の応答として各ノードから次の電文が返信される。SEA として各ノードの ECHONET アドレス、SEOJ として指定した機器オブジェクト、ESV として 0x73、

EPC として指定した家電のプロパティ、ECHONET データ (EDT) として家電のプロパティ値が返信される。

13. ECHONET ゲートウェイはそれぞれ返信された ECHONET における家電のプロパティを提案するシステムにおける家電の状態に変換する。そして LD Object のプロパティ値変更の機能番号 (0x04) とその引数として家電の状態の種類と変換した家電の状態を LD Object に通知する。
14. LD Object は ECHONET ゲートウェイから通知された家電の状態を用いて LD Object のプロパティを変更する。

また、ECHONET において新たな機器が追加された場合、すなわち ECHONET ノードがコールドスタート或いはウォームスタートを行った場合、図 6.11、図 6.12、図 6.13 に示したように、ECHONET ノードからインスタンスの構成に変化のあったクラスを列挙しているインスタンス変化クラスのプロパティが通知されるので、この電文から SEOJ とインスタンス変化クラスに含まれるクラスを抽出し、前述の 7 以降の処理を行う。

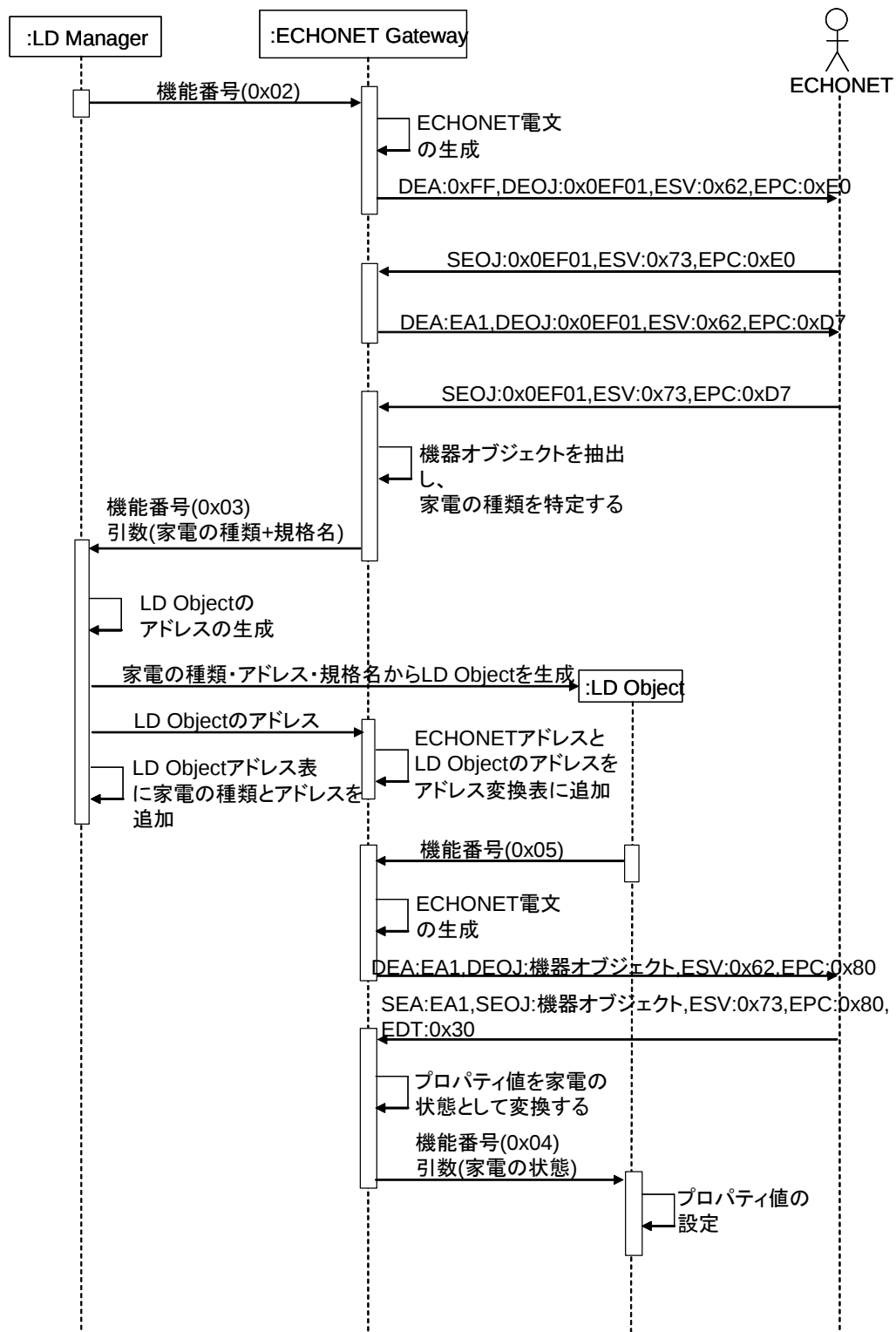


図 6.14: ECHONET 機器の登録

次に ECHONET からの家電の問合せを示す。ECHONET における家電の登録は、いずれかの ECHONET ノードに連動情報、またはインスタンスリストとして生成される。提案するシステムにおいて、連動情報を定義してないので、インスタンスリスト、すなわち LD Object の種類に対応した ECHONET オブジェクトのインスタンスを ECHONET に登録する。図 6.15 に家電の登録のシーケンスを示す。

1. ECHONET から提案するシステムに対して、ECHONET アドレスを認識するため、次の電文が ECHONET ゲートウェイに通知される。DEOJ としてノードプロファイルオブジェクトを示す 0x0EF01、ESV として読み出し要求を示す 0x62、EPC として ECHONET アドレスを示す 0xE0 をそれぞれ指定された電文が届く。
2. ECHONET ゲートウェイはアドレス変換表に登録されている ECHONET アドレスを返信するが、アドレス変換表に登録されていない LD Object がないか、LD Object の検索の機能番号 (0x00) を用いて、LD Manager に問い合わせる。
3. LD Manager は、LD Object の応答の機能番号 (0x01) とその引数として自身が持つ LD Object アドレス表に登録されている LD Object のアドレスと家電の種類を ECHONET ゲートウェイに通知する。
4. ECHONET ゲートウェイは、自身が持つアドレス変換表にない LD Object のアドレスがある場合、まず NetID から ECHONET アドレスを生成する。また家電の種類から機器オブジェクトのコードを生成する。そしてこれらと LD Object のアドレスをアドレス変換表に登録する。
5. ECHONET ゲートウェイは ECHONET からの応答として次の電文を生成する。SEOJ としてノードプロファイルオブジェクトを示す 0x0EF01、ESV として通知を示す 0x73、EPC としてアドレス変換表に登録されている ECHONET アドレスをそれぞれ指定し、ECHONET へ通知する。
6. 次に当該ノードの機能認識として個体識別情報を取得するための電文が届く。DEA として各 ECHONET アドレス、DEOJ としてノードプロファイルオブジェクトを示す 0x0EF01、ESV として読み出し要求を示す 0x62、EPC として自ノードクラスリスト S を示す 0xD7 が電文に指定され、通知される。
7. ECHONET ゲートウェイではアドレス変換表に登録されている機器オブジェクトのコードを指定して、電文を ECHONET に通知する。
8. 次に ECHONET から機器オブジェクトを指定して、家電の状態を問い合わせる電文が届く。その電文には、DEA として家電を指定する ECHONET アドレス、DEOJ として機器オブジェクト、ESV として読み出し要求を示す 0x62、EPC として問い合わせる家電の状態 (図 6.15 には電源の状態を問い合わせる 0x80) が指定されている。

9. ECHONET ゲートウェイは指定された ECHONET アドレスをアドレス変換表から LD Object のアドレスに変換する。そしてその LD Object に対して家電の状態を問い合わせる機能番号 (0x05) を通知する。
10. LD Object はこの機能番号が届くと、自身が持つ家電の状態を ECHONET ゲートウェイに返信する。その際用いる機能番号は、家電の状態の問合せに対する応答 (0x06) を用い、その引数として LD Object のプロパティ名と家電の状態を用いる。
11. ECHONET ゲートウェイは家電の状態を返信する際、次の電文を生成する。まず LD Object のアドレスからアドレス変換表を用いて ECHONET オブジェクトに変換し、SEA として指定する。SEOJ として機器オブジェクト、ESV として通知を示す 0x73、EPC として問い合わせのあった家電の状態の種類、EDT として家電の状態 (図 6.15 には電源の状態を問い合わせた結果、EDT には電源の状態が On である 0x30) を指定する。この電文を ECHONET に通知する。

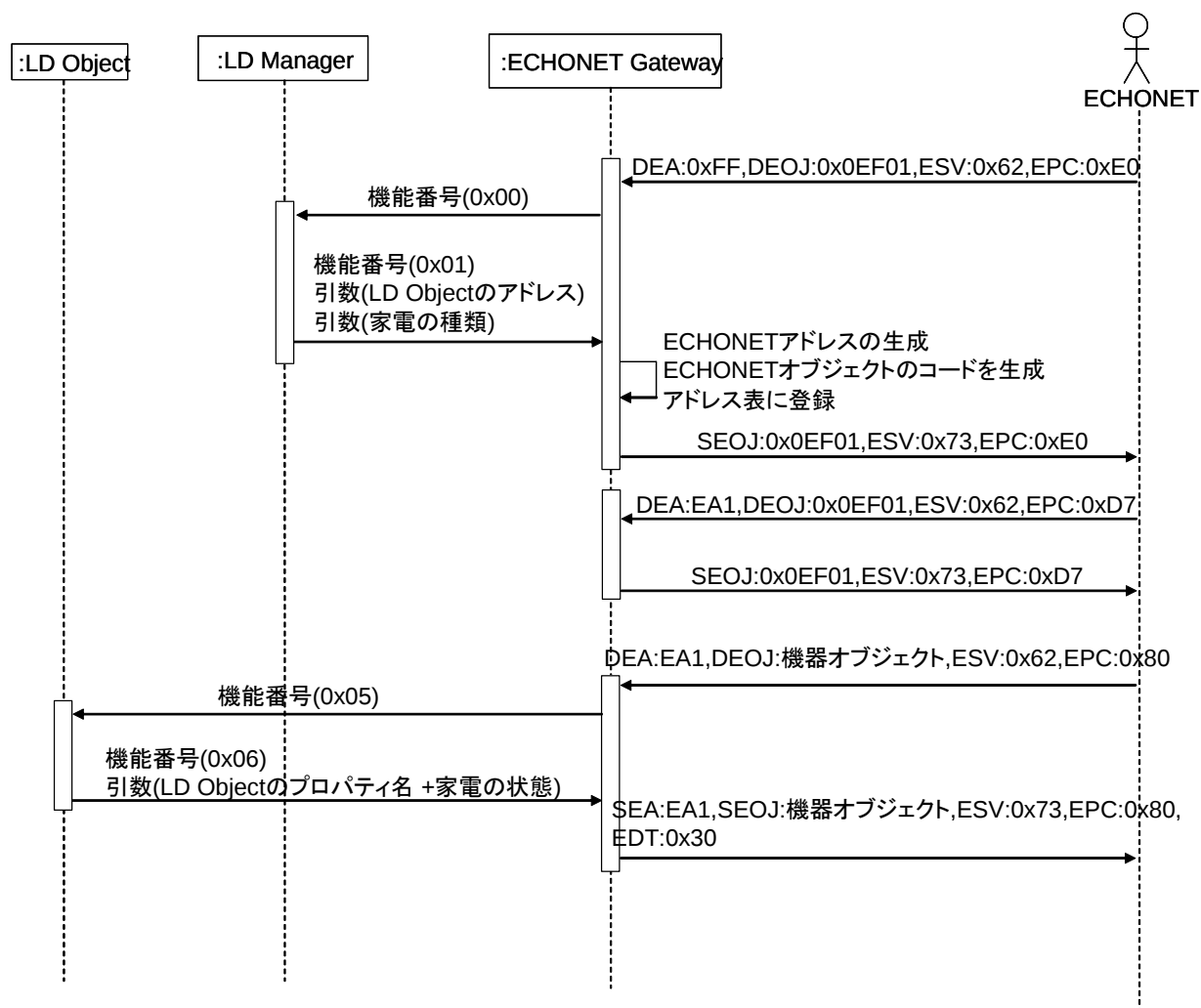


図 6.15: ECHONET への機器の登録

次に ECHONET からの家電の制御に対するシーケンスを図に示す。ECHONET からの家電の制御は ECHONET ゲートウェイに対して書き込みサービスを指定した電文が通知されることによって行われる。ECHONET からの家電の制御におけるシーケンスを図 6.16 に示す。

1. ECHONET から家電を制御するための電文が届く。DEA として制御したい家電の ECHONET アドレス、DEOHJ として機器オブジェクト、ESV として応答有の書き込み要求を示す 0x61、EPC として制御したい家電の状態の種類 (図では電源の状態を示す 0xE0) を、EDT として制御した家電の状態 (図では電源の On を示す 0x80) が指定された電文が ECHONET ゲートウェイに届く。
2. ECHONET ゲートウェイでは、この電文から ECHONET アドレスを取り出し、アドレス変換表を用いて LD Object のアドレスへ変換する。また EPC と EDT から機能番号へ変換する。
3. ECHONET ゲートウェイで変換された LD Object のアドレスを用いて、制御したい LD Object を指定し、家電を制御する機能番号とその引数を通知する (図 6.16 では機能番号として電源の操作を示す 0x10、その引数として On を示す 0x01 をそれぞれ指定する)。
4. LD Object に家電を制御する機能番号と引数が通知されると、実際に家電を制御するために、LD Object に指定されている規格名から通知された機能番号と引数をその規格に対するゲートウェイに通知する。LD Object のプロパティの変更は家電を制御された後に行うため、制御完了通知が来るまで保留しておく。(制御完了通知を返さない規格においては、機能番号を送った後に、LD Object のプロパティ値を変更する。)
5. 家電が実際に制御されると、その規格のゲートウェイに完了通知が届く。そのゲートウェイから LD Object に対して制御完了を表す機能番号 (0x07) が返信される。この機能番号が返信されると、LD Object で保留していたプロパティ値の変更を行う。
6. LD Object のプロパティ値を変更した後、ECHONET に対しても応答を行うため、ゲートウェイに対して制御完了を表す機能番号 (0x07) を返信する。
7. ECHONET ゲートウェイに制御完了を表す機能番号が通知されると、次の電文を生成し、ECHONET ノードに返信する。LD Object のアドレスからアドレス変換表を用いて ECHONET アドレスに変換し、SEA として指定する。また SEOJ として機器オブジェクト、ESV として書き込み要求に対する応答である 0x71、EPC として制御した家電の状態をそれぞれ指定し、返信する。

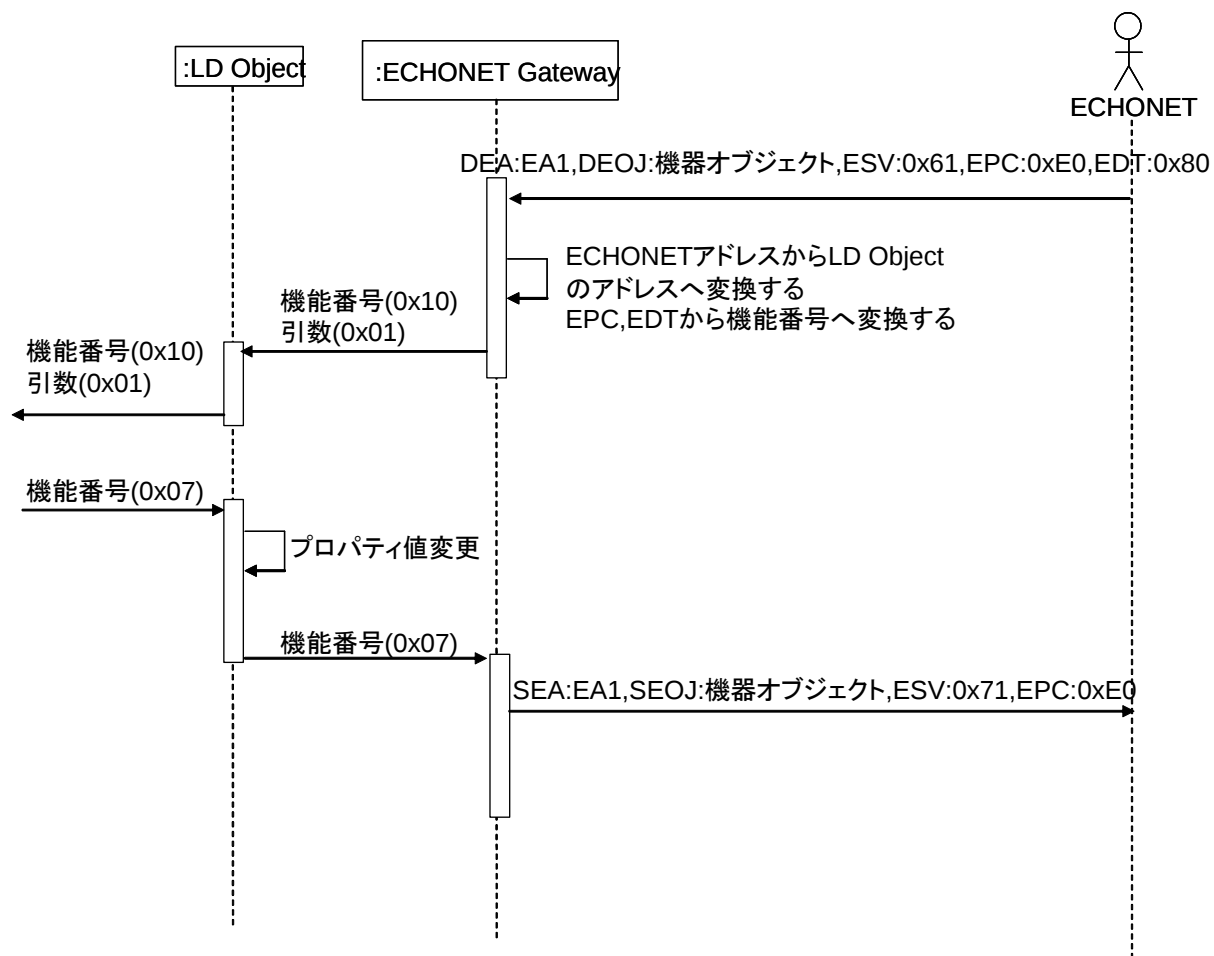


図 6.16: ECHONET からの機器の制御

次に ECHONET に対する家電の制御について示す。ECHONET における機器の制御は機器オブジェクトに対して書き込みサービスを行うことで実行される。ECHONET に対する家電の制御におけるシーケンスを図 6.17 に示す。

1. ある規格から ECHONET 機器を制御するために提案するシステムに制御メッセージが届く。そのメッセージをゲートウェイで受信すると、ゲートウェイは LD Object に対して制御したい家電の状態の機能番号が通知される (図 6.17 では、電源を On にする操作が指定されている)。
2. LD Object にこの機能番号が届くと、ECHONET ゲートウェイに対してこの機能番号とその引数を通知する。ECHONET においては制御に対する応答が返信されるので、LD Object のプロパティ値の変更は保留しておく。
3. ECHONET ゲートウェイにこの機能番号が届くと、ECHONET に対して制御メッセージを送るため、次の電文が生成される。まず LD Object のアドレスからアドレス変換表を用いて ECHONET アドレスに変換する。そして DEA としてこの ECHONET アドレスを指定する。また ECHONET アドレスに対応した機器オブジェクトを DEOJ として指定する。ESV として書き込み要求を示す 0x61、EPC として制御したいプロパティ名、EDT として制御したいプロパティ値をそれぞれ指定し、ECHONET に送信する。
4. ECHONET に電文を送信する際に、ESV を 0x61 として指定しているので、制御が完了すると次に示す電文が返信される。SEOJ として機器オブジェクト、ESV として書き込み要求に対する応答である 0x71、EPC としてプロパティ名をそれぞれ指定されている。
5. ECHONET ゲートウェイではこの電文から、ECHONET アドレスを抽出し、アドレス変換表から LD Object のアドレスに変換する。そして LD Object に対して、制御完了を示す機能番号 (0x07) を通知する。
6. LD Object にこの機能番号が届くと、保留していたプロパティ値の変更を実行する。

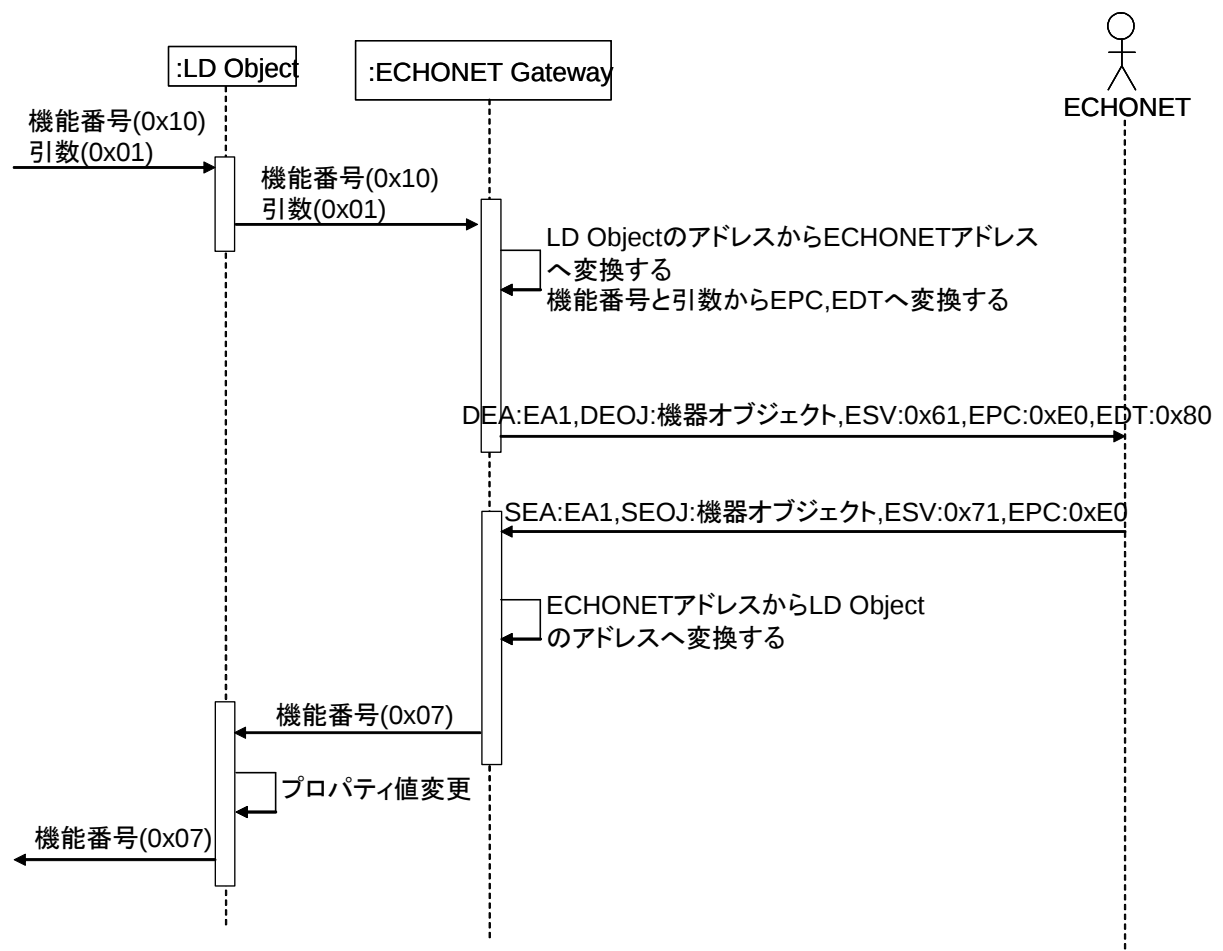


図 6.17: ECHONET に対する家電の制御

6.2.3 X-10 Gateway[8]

X-10 ゲートウェイは、Housecode と Device Code からなるアドレスを LD Object に対して設定および家電機器の電源や照明の操作を行うメッセージの生成を行い、X-10 モジュールに送信する機能を持つ。

提案システムとのシーケンス

X10 機器の登録のシーケンスを図 6.18 に示す。

1. システムのユーザが X10 機器のアドレスである Housecode と DeviceCode、家電の種類を入力し、LD Manager に伝える。
2. LD Manager はアドレスと家電の種類を受け取ると、LD Object を生成するために LD Object のアドレスを生成する。
3. LD Object のアドレスと家電の種類を元に LD Object を生成する。
4. LD Manager は LD Object アドレス表にアドレスを登録する。
5. LD Manager は Housecode、DeviceCode と LD Object のアドレスを X10 ゲートウェイに送り、X10 ゲートウェイのアドレス変換表に登録する。

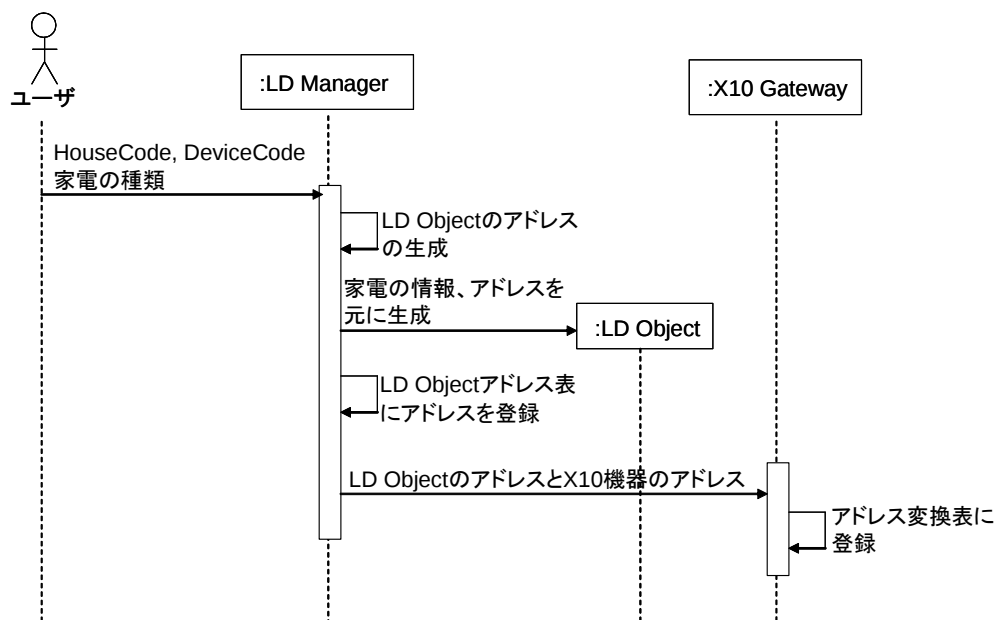


図 6.18: X10 機器の登録

6.2.4 HAVi Gateway[5][9]

本節では、HAViにおけるゲートウェイの設計を行う。まず HAVi におけるソフトウェアエレメントの登録、機器制御に関するシーケンスの詳細を示し、続いて提案するシステムとのゲートウェイを設計する。

HAVi における機器登録・制御に関するシーケンス

HAVi の FAV または IAV デバイスにおけるソフトウェアのブートストラップの実行中は以下の規則に従わなければならない。

- システムコンポーネントは、ローカル Registry に登録されたとき、Ready 状態になる。システムコンポーネントの登録は、独自の方法で実現される場合がある (すなわち、HAVi における Messaging Sytem を用いない場合)。
- HAVi における Messaging System は、全てのシステムコンポーネントの準備が整った時のみ、Ready 状態になる。
- Messaging System が Ready である場合、SystemReady イベントを生成する。
- いったん SystemReady イベントが生成されると、Messaging System は入力メッセージを受け付けることができ、ローカルソフトウェアエレメントはそのプロシージャコール API を利用できる。

Registry は、CMM1394 のメソッドである GetGUIDList を用いて GUID のリストを得ることができ、また IAV や FAV の GUID のリストを抽出し (実装依存の機構を用いる)、ネットワーク上にある他の Registry SEID を得ることが出来る。ネットワークを通じて使用される非システムソフトウェアエレメントは全て、ローカル Registry を通して登録する。DCM Manager は DCM Manager のプロトコルに従って、DCM をインストールする。この段階で、どのソフトウェアエレメントも入力メッセージを受け付けることができるため、デバイスは、ネットワーク内で他のデバイスとのインターオペラビリティを確保できる。

次に BAV または LAV のネットワークへの接続について示す。BAV や LAV デバイスが 1394 バスに接続されると、各 FAV や IAV の CMM は、NetworkReset イベント (または NewDevices イベント) をローカルに生成する。DCM Manager が、事前に NetworkReset イベントを登録していた場合、このイベントを受け取る。そして CMM1394 の GetGUIDList メソッドを用いて、ネットワーク上の新たなデバイスや、離れたデバイスの GUID を得る。各 DCM Manager は、新たに接続されたデバイスの SDD を読み、このデバイスが新たな BAV や LAV のいずれかであるかを検知する。そして DCM Manager のプロトコルに従い、1 つの DCM Manager がこの新たなデバイスを制御する権利を取得し、このデ

バースに DCM をインストールする。新たなデバイスの新たな DCM コンポーネントは、Messaging System から SEID を得た後、Registry に登録する。

次にアプリケーションと FCM との通信について図 6.19 に示す。IAV1 のアプリケーションが VCR を検索すると仮定する。各デバイスの DCM や FCM は Registry に登録されているものとする (1)。このアプリケーションは、VCR の FCM を問い合わせるため Registry にメッセージを送る (2)。Registry は、この条件に適合する FCM の SEID のリストを返す (3)。IAV2 の FCM が、このリストのメンバであると仮定する。IAV1 のアプリケーションは、IAV2 の FCM と通信可能で、コマンドを送信することができる。ターゲット FCM はコマンドを受信すると、VCR の API の定義に従って、レスポンスを送信できる。IAV1 のアプリケーションは、Messaging System を用いて、VCR の Play コマンドを送信する (4)。Message Layer は、メッセージ内にコマンドと SEID をカプセル化し、このメッセージを IAV2 の GUID とともに CMM へ送信する。CMM は IEC61883/1394 パケットを用いて、IAV2 の GUID を表す 1394 ネットワークアドレスとともにメッセージを送信する (5)。IAV2 の Messaging System は、メッセージを受信し、宛先 SEID をチェックする。そして、宛先のソフトウェアエレメントに対応するコールバック関数を呼び出す。IAV2 の FCM は、コマンドと、ソースのソフトウェアエレメントの SEID を受信する (6)。FCM は要求された動作を実行し、VCR の API で指定されたようにレスポンスを返す (7,8,9)。

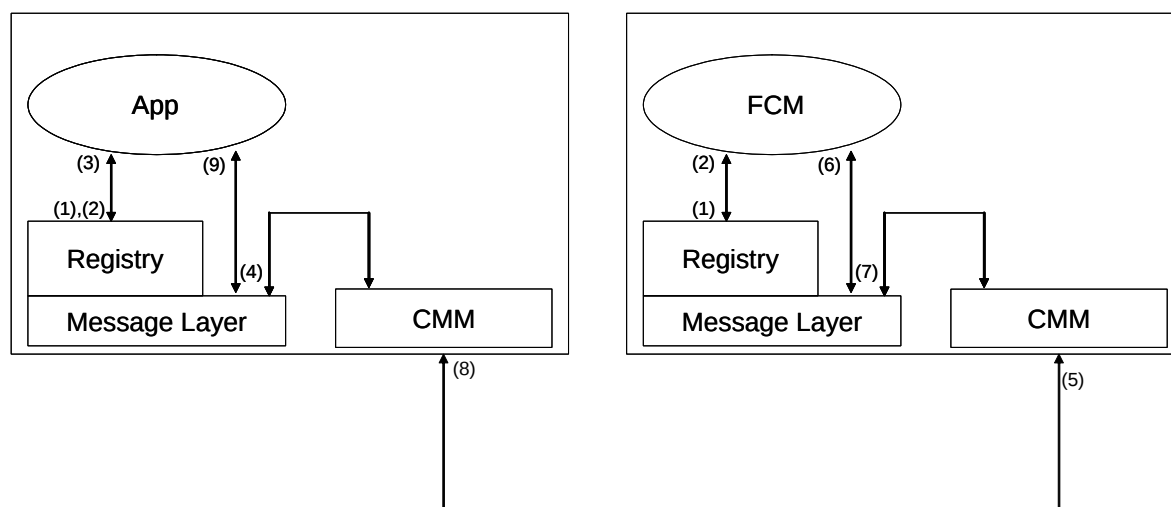


図 6.19: アプリケーションと FCM 間の通信

次に HAVi に対応していない機器 (LAV) における制御を図 6.2.4 に示す。

1. アプリケーションはターゲットとなる機器の動作を要求するメッセージを HAVi メッセージングを使って DCM へ送る。
2. DCM はターゲット機器への要求を中継する。DCM とターゲット機器間のプロトコルは HAVi によって明確にされていないので、ターゲット機器の製造元によって決

定される。

3. ターゲット機器は DCM へ応答する。

4. DCM は HAVi メッセージを使ってアプリケーションへ応答する。

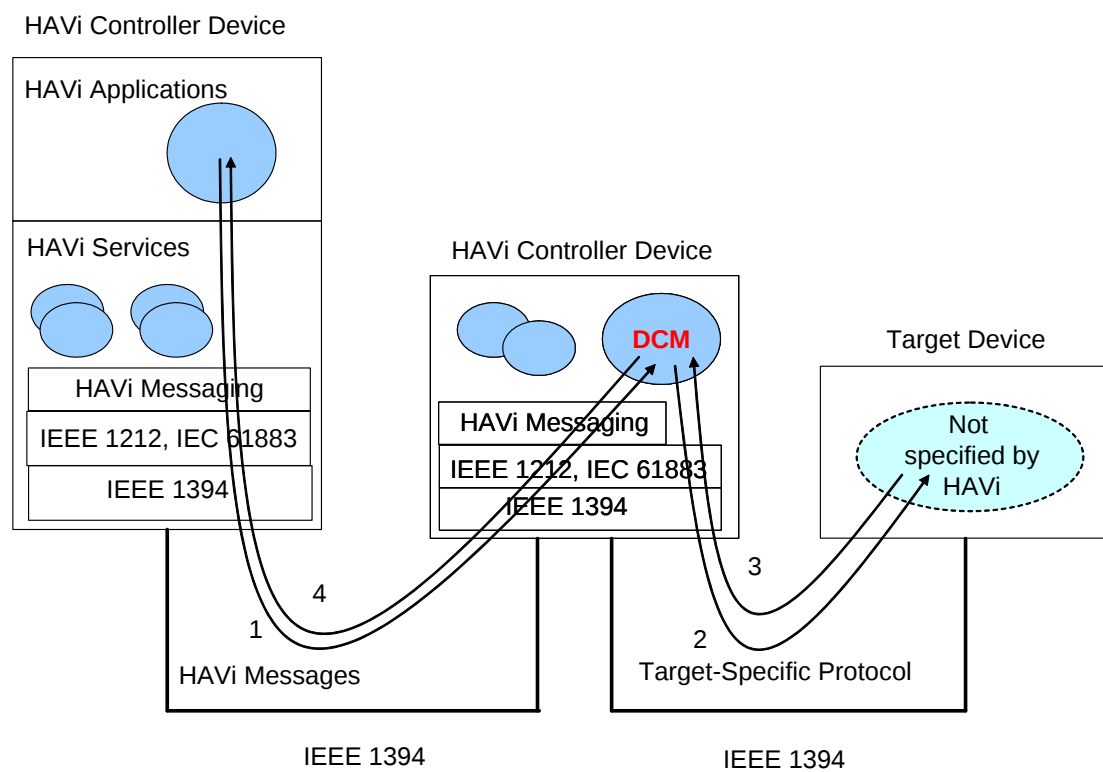


図 6.20: HAVi アプリケーションとコントロールされる機器間の通信

提案システムとのシーケンス

HAVi ゲートウェイは、他規格の家電を表している LD Object を仮想的に DCM および FCM と見せることによって実現する。

HAVi の機器の登録のシーケンスを図 6.21 に示す。

1. LD Manager が HAVi ゲートウェイに対して検索に相当する機能番号を送る。
2. HAVi ゲートウェイでは IAV もしくは FAV に実装されている Registry に対して HAVi に存在する DCM の SEID を要求するメッセージを生成し、送信する。
3. Registry から DCM の SEID が返ってくると、DCM に対して DCM に関連する FCM の SEID を問い合わせる。
4. DCM から FCM の SEID が返ってくると、各 FCM に対して FCM の種類・状態を問い合わせる。
5. 全ての FCM の種類と状態を取得すると、これらの情報からシステムで定義している家電の種類・状態に変換する。
6. HAVi ゲートウェイはこれらの家電の種類と状態、規格名を LD Manager に送る。
7. LD Manager はこれらの情報を受け取ると、LD Object を生成するために LD Object のアドレスを生成する。そして家電の種類・状態・アドレス・規格名から LD Object を生成する。
8. LD Manager は HAVi ゲートウェイに生成した LD Object のアドレスを送る。
9. アドレスを受け取った HAVi ゲートウェイは LD Object のアドレスと、DCM と FCM の SEID をアドレス変換表に登録する。
10. また LD Manager は LD Object アドレス表に生成した LD Object の家電の種類とアドレスを LD Object アドレス表に登録する。

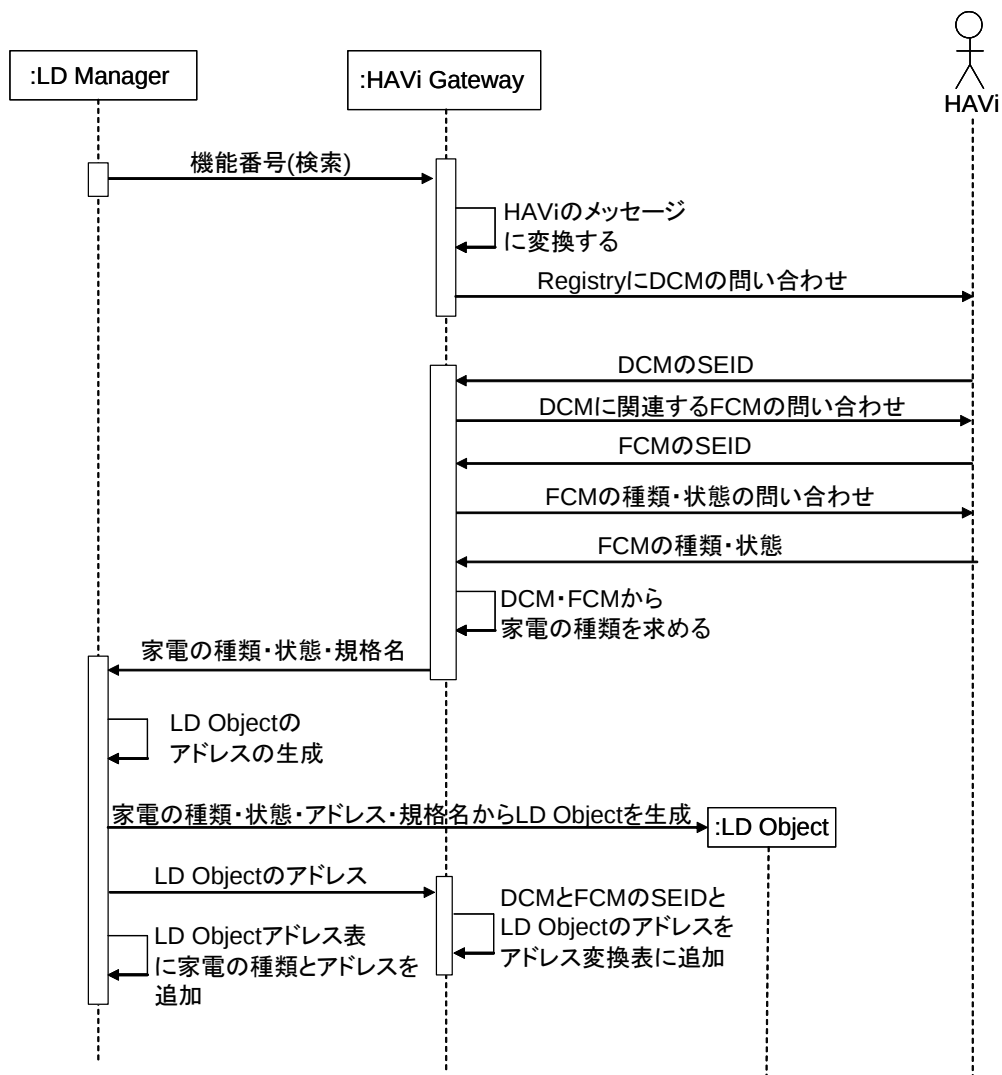


図 6.21: HAVi における機器の登録

次に HAVi における機器制御のシーケンスを図 6.22 示す。このシーケンスでは、電源の操作の制御について示している。

1. 他規格から規格名のプロパティとして HAVi が指定されている LD Object に対して機器の制御のメッセージが届くと、LD Object は HAVi ゲートウェイに対して機能番号 (0x10) を送信する。LD Object は機能番号を送信すると、プロパティ値の変更を保留しておく。
2. HAVi における機器制御は、DCM および FCM をホストしている機器に対して制御メッセージを送信することで行われるので、HAVi ゲートウェイはその機器に対して制御メッセージを送信する。この機能番号は電源の操作に相当し、HAVi において電源の操作は DCM によって行われるので DCM に対して HAVi メッセージを送信する。
3. HAVi において機器の制御が行われると、それに対するレスポンスが返信され、HAVi ゲートウェイで受信する。
4. HAVi ゲートウェイでは、そのメッセージを受信すると LD Object に対して機能番号 (0x07) を送信する。
5. LD Object はその機能番号が届くと、保留していたプロパティ値の変更を行う。
6. LD Object は他規格のゲートウェイに対して、制御に対する応答である機能番号 (0x07) を送信する。

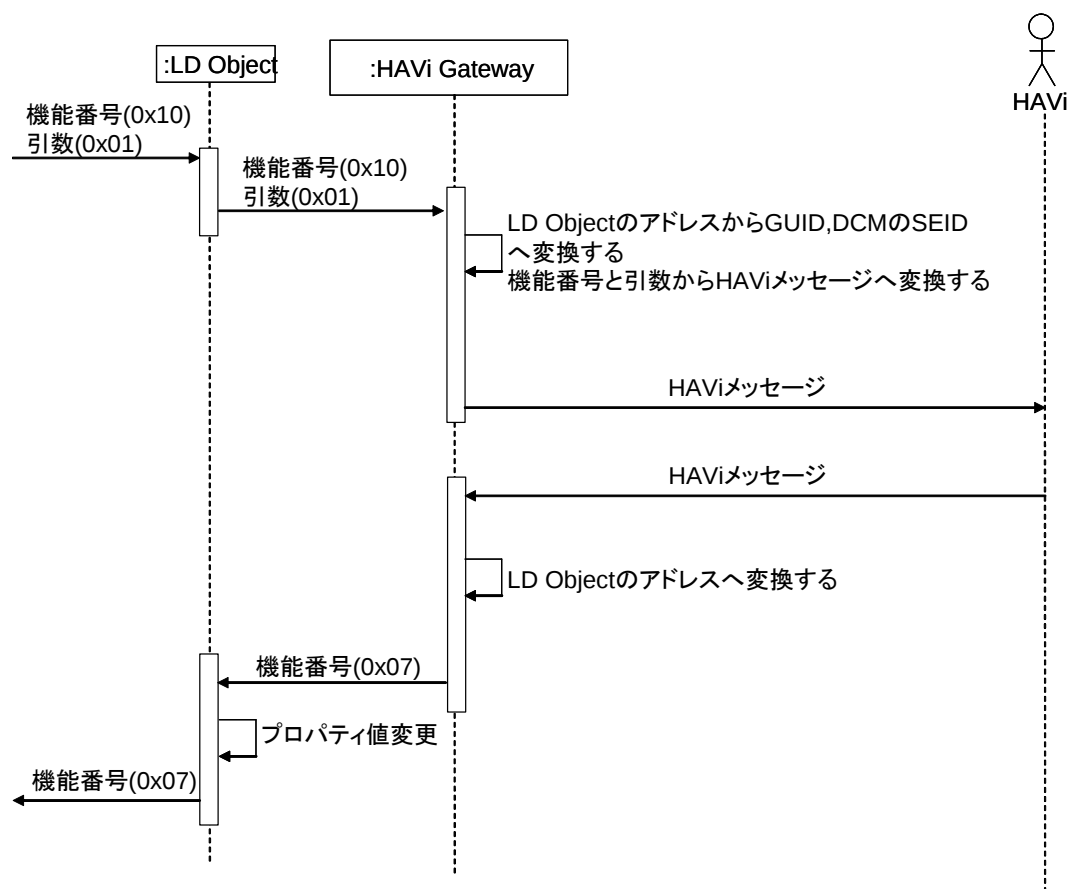


図 6.22: HAVi における機器の制御

6.3 規格間における家電制御のシーケンス

ECHONET から他規格の機器を制御するシーケンスを図6.23に示す。図ではECHONET から X-10 の機器を制御する例を示している。

1. ECHONET から ECHONET ゲートウェイに対してプロパティ値の書き込みを指定する ECHONET 電文が届くと、機能変換表を元に機能番号へ変換する。
2. LD Object へプロパティ値の変更を行う機能番号を送信する。X-10 では家電を制御した際に確認通知が返ってこないため、LD Object はプロパティ値の変更を行う。
3. 実際に家電の制御を行うため LD Object は X-10 ゲートウェイに対して先ほど送られてきた機能番号を送る。
4. X-10 ゲートウェイではアドレス変換表から LD Object に対応する X-10 のアドレスに変換し、また機能番号変換表を元に送られてきた機能番号を固有の命令に変換し、X-10 の機器に命令を送る。

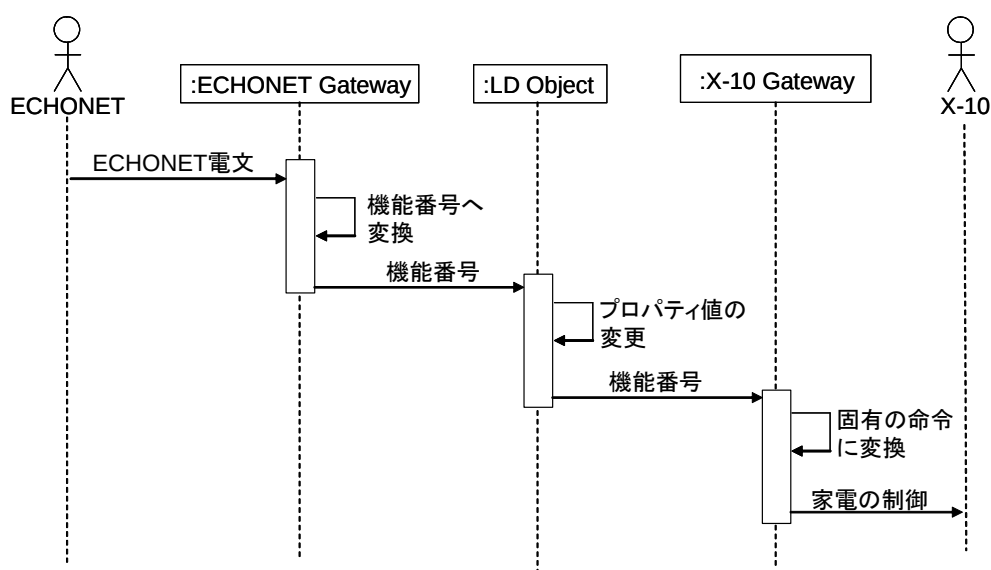


図 6.23: ECHONET から X10 機器の制御

第7章 システムの実装^[20]

本章では、提案するシステムの実装の詳細について説明する。

7.1 実装

提案するシステムにおいてホームネットワーク規格との接続の検証を行うための実装を行った。実装した提案システムを図 7.1 に示す。システムは Ethernet インタフェースを備えたデスクトップ PC 上に Ethernet と電灯線とのインタフェース (PLEBR10) を接続し、Linux 上に実装した。

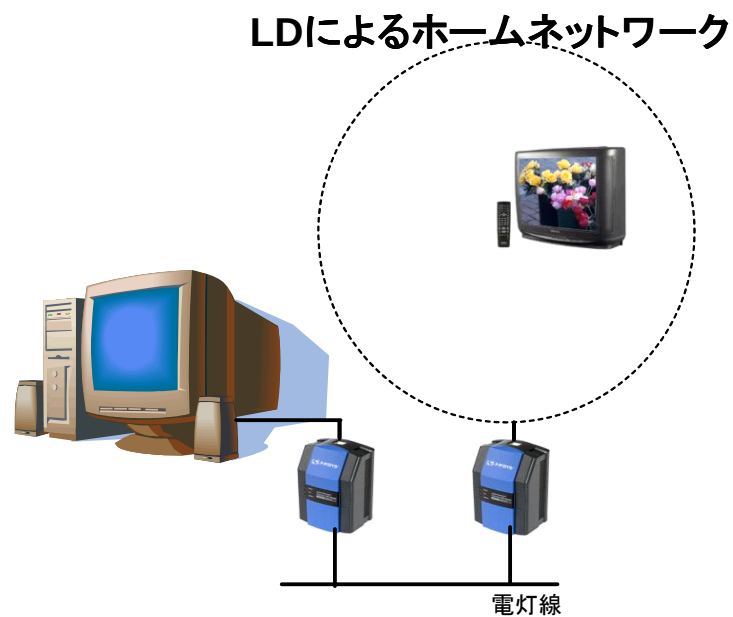


図 7.1: 提案するシステムの実装概要

提案するシステムと LD との接続環境を図 7.2 に示す。PC と LD システムには電灯線と Ethernet のブリッジである PLEBR10 を接続し、それぞれの IP アドレスを 192.168.0.1 と 192.168.0.100 に設定した。LD としての TV は LD システムでは赤外線で制御する。

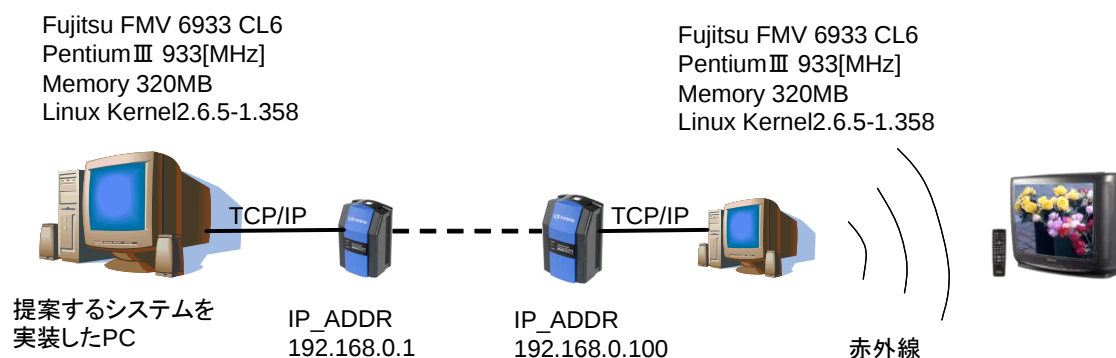


図 7.2: 提案するシステムにおける LD との接続環境

7.2 家電の制御の検証

提案システムの検証を行うため、LD によるホームネットワークに対して、LD の検索および制御について行った。検証のために行った動作は次の通りである。これらの動作を行った結果、LD の検索および制御の実現が確認できた。

- LD によるホームネットワークに対して、LD の検索を行うため、inquiry のメッセージタイプを送信し、response が返信されるかどうかの検証
- 返信された response に対して、LD Object を生成し、LD を制御するために control メッセージタイプを送信し、notify が返信されるかどうかの検証

第8章 システムの評価と考察

8.1 LD Object に関して

8.1.1 家電クラスについて

本研究では、提案するシステムの家電モデルとして、単体家電機器・複合家電機器を定義する家電クラスとシステムに存在するために必要なプロパティを持つ System Object クラスの二種から継承される LD Object クラスを定義した。これに対して、様々な家電モデルの定義の仕方が考えられる。

図 8.1 における設計では、システムに必要なプロパティを持つクラスを基底クラスとして全ての家電クラスが継承されるといった方式を用いている。

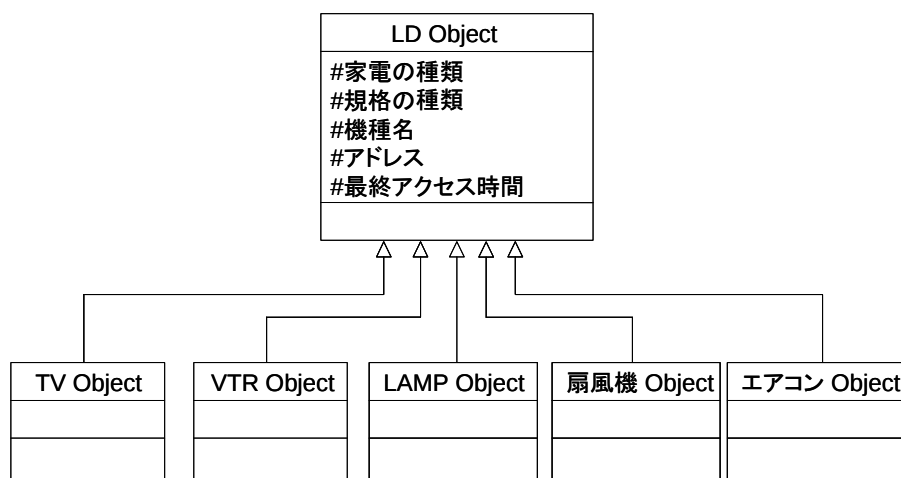


図 8.1: 家電クラス

しかしながら Sony の DESR-7500[22] は、ゲーム機の機能と DVD Recorder の機能を持った複合家電機器であるが、ゲーム機の機能を使用している際は、DVD を視聴することができないといったように、複数機能を持っているが、同時に使用することができない機器の設計、またはテレビデオのように TV の電源は OFF になっているが、VTR の録画機能の使用は可能であるといったように複数機能を持っている家電において、片方の電源が OFF の状態であっても、もう一方の電源が ON であればその機能を使用することができる機器といったことが図 8.1 に示す定義方法を用いた場合、設計できなくなる。よって

家電を単体家電機器と複数の単体家電機器から構成される複合家電機器に分類した方式を用いることによって前述の機能を満たす設計が可能となると考えられる。

図 8.2 と図 8.3 に単体機器と複合機器を区別した定義方法を示す。TV や洗濯機といった単体家電クラスは System Object クラスに相当するクラスを基底クラスとして継承されるものである。複合家電クラスも同様に System Object クラスに相当するクラスから継承される。複合家電機器は単体家電機器を管理するものとしてクラス化し、複合家電機器クラスを定義する。そして図における TV と VTR から構成される TeleVideo クラスといった複合家電機器は複合家電機器クラスを基底クラスとして継承されるクラスであると定義する。

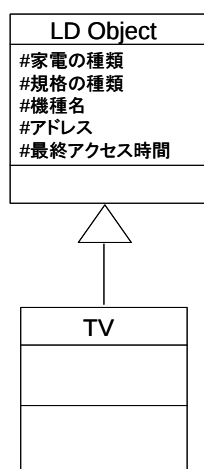


図 8.2: 単体家電クラス

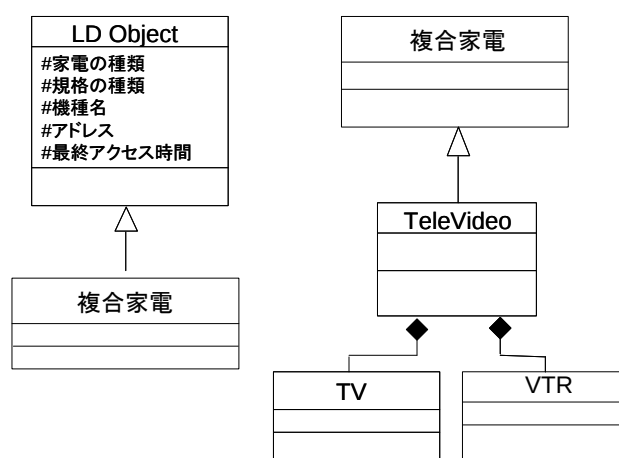


図 8.3: 複合家電クラス

これらの二種類の定義方法は、システムに存在するために必要なプロパティを持つ System Object クラスに相当するクラスを基底クラスとしてそれぞれ家電クラスを定義している。これに対して、家電クラスを基底クラスとして継承されるクラスを LD Object(TV Object および TeleVideo Object) を設計する。

図 8.5 の方式では、例えば TV と VTR が 2 台から構成されるテレビデオが存在した場合、新たにクラスを定義し直さなければならない。一方図 8.3 では、TeleVideo クラスが VTR を 2 つ保持するという方法で解決することができる。また図 8.2 および図 8.3 の方式では、新たな家電が出てきても基底クラスは変わらないが、図 8.4 および図 8.5 の方式では基底クラスを替えなければならない。

よってこれらのことから、図 8.2 および図 8.3 の方式の方が、拡張性が高いことが考えられる。本研究で提案した家電モデルは、この方式を元に再設計を行った。家電の種類やアドレスといったシステムに必要な性質をプロパティとするクラス (System Object クラス) として独立して定義し、家電クラスとこのクラスを継承して LD Object クラスを定義する方式を提案した。この家電モデルを用いることで、新たな家電機器が開発されても図 8.3 で示した拡張性を維持したまま、家電クラスを定義することが可能であると考えられる。

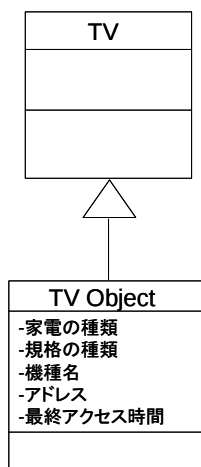


図 8.4: 単体家電クラス

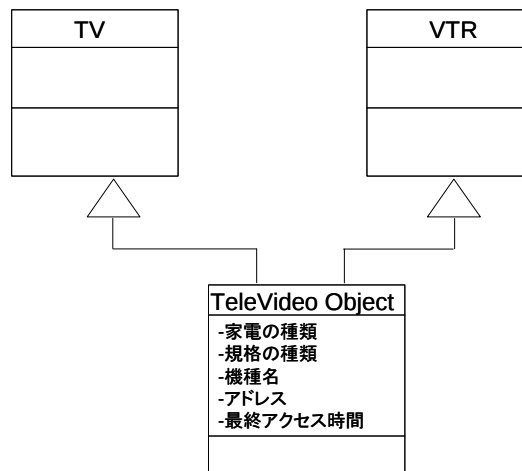


図 8.5: 複合家電クラス

8.1.2 家電における共通機能抽出について

家電クラスを定義するために家電の機能の抽出を行うが、この方法として家電の全ての機能を抽出する方法と、家電ごとの共通機能を抽出する方法の二種類が挙げられる。機種によらず全ての機能を抽出し家電クラスの定義を行った場合、全ての制御メッセージに応じて、制御を行うといったことが可能となる。しかし新たな機種が増えるたびに、また扱う規格が増えるたびにおいても、その規格が扱う機能を追加しなければならないため、家電クラスを再定義しなければならない。そして再定義が行われるたびに、LD Object としての機能が増えるので、扱う機能数が膨大となり、状態の管理が複雑となりうる。そのため本研究では、家電ごとに共通機能を抽出し、この共通機能を元に家電クラスの定義を行った。この家電クラスを用いることで、扱うことができない機能があるものの、ユーザがよく利用する機能を中心に抽出しているので、この家電クラスを用いても、ほぼ要求が満たすことができると考えられる。

8.1.3 家電の機能について

家電の機能として、DVD レコーダにおける EPG を用いた番組録画の機能などのように UI を用いて家電の操作を行う機能がある。これはリモコンの再生ボタンを押すと再生が開始されるという機能とは異なり、UI を用いてワンクッションおいた家電の制御を行っているため、実際に行った機能が不明となりうる。本研究では、リモコンの再生ボタンを押すと、再生の開始を、チャンネル番号を押すとチャンネルの変更が行われるといったように物理的に機能が直結しているものを扱うこととし、UI を必要とする動作については対象外とした。UI を必要とする操作に関して、標準化がなされていないため、HTTP ベースを用いて行うなど、今後の動向を見つつ、アプローチを行うというのが今後の課題としてあげられる。

8.2 家電の電源の扱いに関して

8.2.1 電源の状態について

全ての家電機器は、電源の状態として、ON もしくは OFF のいずれかの状態を持つ。ここで ON という状態は家電の制御が可能であり、OFF という状態は家電の制御が不可能であることを示す。このことから全ての家電クラスは、電源の状態 (ON もしくは OFF) を制御することが可能であるという性質を持つものとして、電源の状態を扱うクラスから継承してできるクラスとみなすことができる。ここで電源の状態を扱うクラスを On_Off Object クラスとして定義する。よって家電クラスと On_Off Object クラス間の関係は図 8.6 のように示される。

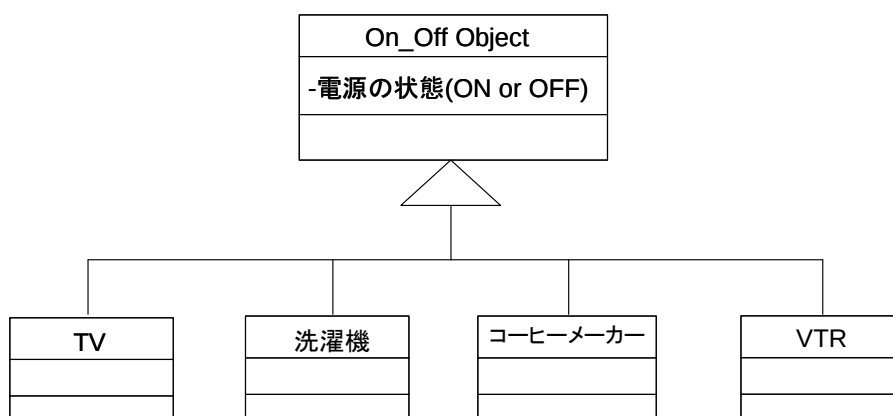


図 8.6: On_Off Object クラスと家電クラス間の関係

しかし家電機器の中には、VTR やラジカセのように予約録画設定を行っている場合、家電の制御ができなくなる機種が存在する。この状態は、家電の制御はできないが、予約が設定されているだけで、解除をすることで再度家電の制御可能となるため、全ての家電の制御ができないわけではないので、OFF の状態にあるとは言えず、この On_Off Object では実現できない。

一方本論文で提案した PowerManagement Object については ON と OFF の状態に加えて予約設定中であるという STAND-BY 状態を考慮に入れた設計を行っているため、前述の制御が可能と考えられる。

8.2.2 ハード電源とソフト電源について

本研究ではハード電源が OFF の場合、家電がシステムに存在しないこととして扱い、ソフト電源の操作のみを扱う。LD によるホームネットワークにおいてはハード電源も含めた電源管理を行っているが、ハード電源を扱わない規格も存在している。しかし、ほと

んどの規格においてソフト電源を扱っていることから、ソフト電源を共通機能として定義しても、ほぼ要求を満たすことが可能であると考えられる。

8.3 機能番号について

提案するシステムにおいてオブジェクト毎にやり取りされる機能番号の定義として、1 バイトと固定している。下位 4bit を家電の種類ごとの機能として定義しているが、炊飯器クラスにおけるメソッドである炊飯実行という機能は炊飯器しか持ち合わせていないので、冗長となってしまっている。また上位 4bit はシステムの操作および家電毎の操作を識別するための部分となっているが、4bit と固定してしまっていることから上限に限りがある。これらのことから拡張性に乏しくなっている。よって、本来は家電の種類と機能の組み合わせを組み合わせで XML 形式などを用いることが望ましい。

8.4 各種規格との接続性について

本システムと各種規格との間にゲートウェイを実装すると、各種規格間のゲートウェイを個々に実装するよりも、ゲートウェイを実装するよりも少ない工数で済む。一方、規格間における家電モデルが類似している場合において、図 3.2 の左記にある提案するシステムのゲートウェイを介して相互接続するよりも、図 3.2 の右記にあるような個々に実装したゲートウェイを介して相互接続する方が扱うことが可能な機能数が多くなるなど規格間の関係がより強固になる可能性が考えられる。本研究では、家電の操作という基本的な部分をモデル化し、幅広い規格を相互接続するように設計を行っている。図 3.2 の左記よりも規格間の関係が薄くなるが、幅広く家電の操作を行うことが可能になるといえる。

第9章 まとめと今後の課題

9.1 まとめ

本研究では、各種ホームネットワーク規格と接続性を考慮した LD ホームネットワークを提案し、各種規格間の家電機器をモデル化する LD Object や LD Object を管理するための LD Manager および提案するシステムと各種規格間のゲートウェイの設計し、評価を行った。

本研究で提案する家電モデルでは、単体家電機器と複数の単体家電機器から構成される複合家電機器の二種類に家電を分類し、単体家電クラスと複合家電クラスを定義した。これらのクラスを定義する際、家電の機能の抽出を、家電ごとの共通機能を抽出する方法を用いて行った。また、本研究では家電の電源のスイッチを、ソフト電源を用い、ハード電源が OFF である場合は、電源が切れているものはシステムに存在しないものとして扱い、この電源の状態を扱うクラスとして PowerManagement Object クラスを提案した。単体家電クラスと複合家電クラスはこの Power Management Object クラスを持つ。また、家電クラスは、家電の機能のみを定義しただけであるので、家電のクラスのインスタンスを提案するシステム内に生成するために、システム内で必要なプロパティを持つクラスである System Object クラスを定義した。システム内にインスタンス化する際は、家電クラスと System Object クラスの二種のクラスを多重継承される LD Object クラスを用いるよう定義した。

次に各種規格における家電機器を LD Object としてシステムに生成する LD Manager の設計、また提案するシステムと各種規格とのゲートウェイを設計し、各種規格の家電機器をシステムに登録するシーケンス、各種規格からの家電検索シーケンス、家電機器の制御シーケンスおよび相互接続シーケンスを示した。そして提案するシステムにおける各種規格との接続を検証するため、LD によるホームネットワークに対して LD の検索および制御について実装を行い、実現を確認した。最後に、家電クラスの定義、家電の電源の扱いおよび各規格との接続性における考察を行った。

9.2 今後の課題

9.2.1 機器の検索

提案するシステムでは機器の検索において、家電の情報・アドレスをユーザに入力してもらう必要があるシーケンスがある。これは家電機器が自身の存在を知らせる仕組みを持たない規格において発生する。今後、ユーザに対してこのようなシーケンスを伴わない仕組みを設計する必要がある。

9.2.2 セキュリティの検討

提案するシステムでは、ホームネットワークを使用可能な範囲に関する設計がされていない。よって宅内もしくは宅外からにおいても家族のみが扱うことができるようにするための認証およびセキュリティに関する仕組みの検討を行う必要がある。

参考文献

- [1] 浜本 賢一, 丹 康雄, 各種ホームネットワーク規格との接続性を考慮した Legacy Device ホームネットワークの管理モデル, 平成 16 年度電気関係学会北陸支部連合大会
- [2] 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, 北陸先端科学技術大学院大学 修士論文 (2005)
- [3] 松崎寛之, 丹 康雄, レガシーデバイスによるホームネットワーク構築のための AV 機器状態取得方法の提案 平成 16 年度電気関係学会北陸支部連合大会
- [4] 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, 北陸先端科学技術大学院大学 修士論文 (2003)
- [5] HAVi The HAVi Specification Ver1.1 <http://www.havi.org>
- [6] ECHONET CONSORTIUM, エコーネット規格書 Ver2.11, ECHONET CONSORTIUM.
- [7] 丹康雄 監修, 宅内情報通信・放送高度化フォーラム 編, ユビキタス技術 ホームネットワークと情報家電, オーム社 (2004).
- [8] x10, cm11a-protocol, <http://www.x10.com>
- [9] Rodger Lea, Simon Gibbs, Ravi Gauba, Ram Balaraman, HAVi Example by Example, Prentice Hall PTR (2002)
- [10] DLNA, <http://www.dlna.org>
- [11] 竹政 昭利, はじめて学ぶ UML, ナツメ社 (2003)
- [12] J. ランボー, M. ブラハ, W. プレメラニ, F. エディ, W. ローレンセン 著, 羽生田栄一 監訳, オブジェクト指向方法論 OMT モデル化と設計, プレンティスホール, トッパン (1995)
- [13] <http://www.daikin.co.jp>
- [14] <http://www.sharp.co.jp>

- [15] <http://www.hitachi.co.jp>
- [16] <http://www.toshiba.co.jp/>
- [17] <http://jp.fujitsu.com/>
- [18] <http://national.jp/>
- [19] <http://www.mitsubishielectric.co.jp/>
- [20] <http://www.linksys.com>
- [21] 都倉信樹, オートマトンと形式言語, 昭晃堂 (1995)
- [22] <http://www.sony.jp>

謝辞

本研究を進めるにあたり、研究の方向性について指針を与えてくださり、また熱心なご指導・ご助言をいただきました丹康雄助教授に深く感謝致します。また本研究に関して多くの有意義なご意見を頂きました博士課程の中田潤也氏、牧野義樹氏に心から感謝いたします。

さらに、LD 班として互いに切磋琢磨して研究に励んできた松崎寛之氏に感謝致します。そして、励まし合いながら、研究生活・学生生活を過ごしてきた丹研究室の皆様に深く感謝致します。

最後に、研究生活を様々な面で支えてくださった家族に感謝致します。

付 録 A 各家電の家電クラスの詳細

A.1 冷蔵庫クラス

表 A.1: 冷蔵庫クラスのプロパティ値

プロパティ名称	プロパティ内容
一気冷凍	一気冷凍実行状態
一気製氷	一気製氷実行状態
クイック脱臭	脱臭実行状態
冷蔵室温度	冷蔵室計測温度
冷凍室温度	冷凍室計測温度

表 A.2: 冷蔵庫クラスのメソッド

メソッド名称	メソッド内容
一気冷凍実行	急速に冷凍を行う
一気製氷実行	急速に製氷を行う
クイック脱臭実行	脱臭を実行する

- 冷蔵室温度は冷蔵室の温度を の単位で示す。プロパティの値域は-127～126 までの範囲をとる。
- 冷凍室の温度は冷凍室の温度を の単位で示す。プロパティの値域は-127～126 までの範囲をとる。
- 一気冷凍は、食品を素早く冷やす際に用いる機能である。一気冷凍実行状態は On(1) もしくは Off(0) の値をとる。
- 一気製氷は、氷を早く作る際に用いる機能である。一気製氷実行状態は On(1) もしくは Off(0) の値をとる。
- クイック脱臭は、素早く脱臭を行う際に用いる機能である。クイック脱臭実行状態は On(1) もしくは Off(0) の値をとる。

A.2 炊飯器クラス

表 A.3: 炊飯器クラスのプロパティ値

プロパティ名称	プロパティ内容
予約時間	炊飯を開始する予約時間

表 A.4: 炊飯器クラスのメソッド

メソッド名称	メソッド内容
炊飯実行	炊飯を実行する
予約指定	予約時間を元に炊飯の予約を開始する

- 予約時間は、炊飯の予約を行うための時間である。予約指定のメソッドを実行することによって、予約が開始される。
- 炊飯実行は、炊飯を実行するメソッドである。

A.3 電気ポットクラス

表 A.5: 電気ポットクラスのプロパティ値

プロパティ名称	プロパティ内容
沸騰・保温状態	沸騰中もしくは保温状態を示す
ロック状態	出湯ロック状態
保温温度	保温温度計測値
沸騰モード	通常沸騰・クエン酸洗浄モード

表 A.6: 電気ポットクラスのメソッド

メソッド名称	メソッド内容
ロック解除	出湯ロックもしくはロックを解除する
保温温度設定	保温温度を設定する
沸騰開始	沸騰実行

- ロック状態はポットの蓋がロックされている状態を示しており、ロック状態 (1) もしくはロック解除 (0) の値をとる。
- 保温温度設定は保温温度設定値を 98、90、80 のいずれかの温度に設定する。
- 沸騰モードは通常沸騰もしくはクエン酸洗浄モードを表す。
- 沸騰実行は沸騰モードに設定されているモードで沸騰を開始する。

A.4 オープンレンジクラス

表 A.7: オープンレンジクラスのプロパティ値

プロパティ名称	プロパティ内容
加熱モード	レンジ加熱・レンジ解凍・オープン・グリル・トースト
レンジ加熱レベル設定値	加熱レベル (W)
オープン加熱温度設定値	加熱温度
加熱時間設定値	加熱時間
加熱残り時間	残り時間
自動メニュー設定値	使用中の自動メニュー番号

表 A.8: オープンレンジクラスのメソッド値

メソッド名称	メソッド内容
レンジ加熱レベル変更	レンジ出力を変更する
自動メニュー実行	自動メニューを実行する

- 加熱モードは、レンジ加熱・レンジ解凍・オープン・グリル・トーストの 5 つのモードをとる。
- レンジ加熱レベルは、レンジ出力を設定するためのプロパティである。メーカーごとに加熱レベルは異なるものの各メーカーとも 4 段階の値を取り、本モデルではその 4 段階の値をとるものとする。
- オープン加熱温度は、オープン使用時に設定する加熱温度を表す。
- 加熱時間設定は、レンジを使用する際加熱時間を設定する。設定時間は MM:SS で表される。残り時間はこの時間と経過時間との差で表される。

- 自動メニューは、レンジに食品を入れて、一つのキーを押すことで加熱が始まる機能を指す。自動メニューを実行する際は、パラメータとしてメニューを選択する。

付 録 B 機能番号表と引数番号表

表 B.1: 機能番号表

機能	機能番号	引数
LD Object の検索	0x00	-
LD Object の応答	0x01	LD Object のアドレス + 家電の種類
他規格の家電の検索	0x02	-
他規格の家電の応答	0x03	家電の種類 + 規格
LD Object のプロパティ値変更	0x04	LD Object のプロパティ名 + 値
家電の状態の問い合わせ	0x05	家電の状態の種類
状態の問い合わせに対する応答	0x06	LD Object のプロパティ名 + 値
家電の制御に対する完了通知	0x07	-
電源の操作	0x10	電源の状態
音量変更	0x11	音量値
Mute 実行	0x12	On/Off
チャンネル変更	0x13	チャンネル番号
画面表示	0x14	-
音声変更	0x15	主 or 副音声設定値
オフタイマー設定	0x16	設定時間
オンタイマー設定	0x17	設定時間
入力切替	0x18	-
再生	0x20	-
停止	0x21	-
早送り	0x22	-
巻き戻し	0x23	-
一時停止	0x24	-
録画	0x25	-
予約設定	0x26	予約時間
エアコンのモード変更	0x30	モード
温度変更	0x31	温度
次ページへ続く		

機能	機能番号	引数
風向変更	0x32	風向
フルパワー	0x33	On/Off
スウィング	0x34	On/Off
風量変更	0x35	風量
リズム風量	0x36	On/Off
照明明るさ	0x40	明るさ (%)
一気冷凍	0x50	On/Off
一気製氷	0x51	On/Off
炊飯実行	0x60	-
沸騰実行	0x70	-
蓋ロック	0x71	On/Off
保温値設定	0x72	保温値
洗濯実行	0x80	-
水量変更	0x81	水量値
洗い変更	0x82	回数
すすぎ変更	0x83	回数
脱水変更	0x84	回数
乾燥実行	0x85	-
レンジ加熱レベル変更	0x90	加熱レベル
加熱時間設定	0x91	時間
自動メニュー実行	0x92	メニュー番号

- LD Object の検索の機能番号は他の規格からシステムに家電の検索があった場合に起こる番号である。その応答としての機能番号が LD Object の応答で、LD Object のアドレスと家電の種類を引数として送られる。これらは各種ゲートウェイと LD Manager において発行される番号である。
- 他規格の家電の検索の機能番号は他規格の家電を検索するために LD Manager で発行され、各種ゲートウェイに送られる。その応答として他規格の家電の応答の機能番号があり、各種ゲートウェイにおいて各種規格で定義されている家電の情報をシステムで定義している家電の情報に変換し、これを引数として LD Manager に送られるものである。
- LD Object のプロパティ値の変更は、洗濯の残り時間といった家電の機能としては実行されないが、プロパティ値が変更される際に扱う機能番号である。
- 電源の操作の機能番号は LD Object に送ることによって LD Object が持つ Power-Management オブジェクトに対して電源の操作を行う。この機能番号は表 B.2 に示す引数と同時に送られる。

表 B.2: 電源の操作・機能実行に関する引数番号表

電源の状態	値
On	0x01
Off	0x02
Stanby	0x03

- 音量の変更の機能番号は、TV や CD Player などの音量を変更するための機能番号である。この機能番号の引数として音量値をとり、0～50 までの範囲をとる。
- Mute 実行の機能番号は、TV や CD Player などに Mute を実行するための機能番号である。この機能番号は、(音量の変更の機能番号+引数 0) に相当し、実行すると、音量の値が 0 となる。
- チャンネル変更の機能番号は、TV などのチューナのチャンネルを変更するための機能番号である。この機能番号の引数として表 B.3 に示すチャンネル番号をとる。

表 B.3: チャンネル番号に関する引数番号

チャンネル	値
1～12	0x01～0x0C
BS-5	0x0D
BS-7	0x0E
BS-9	0x0F
BS-11	0xA0

- 画面表示の機能番号は、TV において現在のチャンネル番号を表示するものである。
- 音声変更の機能番号は、主音声・副音声・主+副音声の切り替えを行うものである。この機能番号は表 B.4 に示す引数番号とともに送られる。

表 B.4: 音声変更に関する引数番号表

音声	値
主音声	0x01
副音声	0x02
主+副音声	0x03

- オフタイマー設定の機能番号は、TVなどの電源をオフにする機能番号である。表 B.5 に示す時間設定値を引数にとる。
- オンタイマー設定の機能番号は、TVなどの電源をオンにする機能番号である。表 B.5 に示す時間設定値を引数にとる。

表 B.5: オフタイマー・オンタイマー設定時間に関する引数番号

タイマー時間	値
30	0x01
60	0x02
90	0x03
120	0x04

- 入力切換の機能番号は、外部入力に切り替える機能番号である。この機能番号を実行する毎に外部入力切り替わる。
- 再生・停止・早送り・巻き戻し・録画・一時停止の機能番号は、VTRなどの機器を再生・停止・早送り・巻き戻し・録画・一時停止を行う機能番号である。
- 予約設定の機能番号は、VTRなどの録画に関して予約設定を行う機能番号である。引数として、時間値をとり、YYYY/MM/DD hh:mm の形式をとる。
- エアコンのモード変更の機能番号は、冷房・暖房・除湿のモードに切り替える機能番号である。この機能番号は、表 B.6 に示す引数番号とともに送られる。

表 B.6: エアコンのモードに関する引数番号表

モード	値
冷房	0x01
暖房	0x02
除湿	0x03

- 温度変更の機能番号は、エアコンなどの設定温度値を変更する機能番号である。引数として 0 ~ 50 までの値をとる。
- 風向変更の機能番号は、エアコンの風向の上下角度を変更するための機能番号である。引数として表 B.7 に示す値をとる。
- フルパワーの機能番号は、エアコンなどのフルパワーを On もしくは Off にするための機能番号である。引数として表 B.2 に示した On または Off の値をとる。

表 B.7: 風向に関する引数番号表

風向	値
上	0x01
上中	0x02
中	0x03
下中	0x04
下	0x05

- スウィングの機能番号は、エアコンなどのスウィングを On もしくは Off にするための機能番号である。引数として表 B.2 に示した On または Off の値をとる。
- 風量変更の機能番号は、エアコンなどの風量を変更するための機能番号である。風量は 8 段階で調節ができ、引数として 1(最小) ~ 8(最大) の値をとる。
- リズム風量の機能番号は、リズム風量を実行するための機能番号である。この機能番号の引数として表 B.2 に示す On と Off の値をとる。
- 照明明るさの機能番号は、照明の明るさを変更する機能番号である。この機能番号の引数として、明るさの割合をとり、0 ~ 100(%) の範囲の値をとる。
- 一気冷凍の機能番号は、冷蔵庫の機能としての急速冷凍を実行するための機能番号である。この機能番号の引数として表 B.2 に示す On と Off の値をとる。
- 一気製氷の機能番号は、冷蔵庫の機能としての急速製氷を実行するための機能番号である。この機能番号の引数として表 B.2 に示す On と Off の値をとる。
- 炊飯実行の機能番号は、炊飯器の炊飯を実行するための機能番号である。
- 蓋ロックの機能番号は、電気ポットの蓋をロックもしくはロックを解除するための機能番号である。この機能番号の引数として表 B.2 に示す On と Off の値をとる。On は蓋をロックし、Off は蓋のロックを解除するものとする。
- 保温値設定の機能番号は、炊飯器や電気ポットなどの保温値を設定するための機能番号である。この機能番号の引数として、0 ~ 100() までの値をとる。
- 洗濯実行の機能番号は、洗濯機の洗濯コースで指定されているコースで洗濯を実行するための機能番号である。
- 水量変更の機能番号は、洗濯機の水量を 4 段階で設定するための機能番号である。この引数として表 B.8 に示す値をとる。

表 B.8: 水量に関する引数番号表

水量	値
最多	0x01
多	0x02
少	0x03
最少	0x04

- 洗い変更の機能番号は、洗濯機の洗いの時間を変更するための機能番号である。この機能番号の引数として時間を取り、MM の形式で単位は分で表される。
- すすぎ変更の機能番号は、洗濯機のすすぎの回数を変更するための機能番号である。この機能番号の引数として回数を取り、1～9 までの範囲の値をとる。
- 脱水変更の機能番号は、洗濯機の脱水の時間を変更するための機能番号である。この機能番号の引数として時間を取り、MM の形式で単位は分で表される。値の範囲は 1～9 である。
- 乾燥実行の機能番号は、洗濯乾燥機における乾燥を実行する機能番号である。
- レンジ加熱レベル変更の機能番号は、レンジ出力を変更するための機能番号である。この機能番号の引数として表 B.9 に示す値をとる。

表 B.9: レンジ加熱レベルに関する引数番号表

レンジ加熱レベル	値
最強	0x01
強	0x02
弱	0x03
最弱	0x04

- 加熱時間設定の機能番号は、レンジを使用する際に加熱時間を設定するための機能番号である。この引数として加熱時間を取り、MM:SS の形式で入力する。
- 自動メニュー実行の機能番号は、自動メニューを実行するための機能番号である。この引数として、表 B.10 に示す値をとる。

表 B.10: 自動メニューに関する引数番号表

自動メニュー	値
牛乳	0x01
お酒	0x02
グラタン	0x03
パン	0x04
ピザ	0x05
解凍	0x06