

Title	ホームネットワークにおけるユーザインタフェースの 統一的なAPI提供に関する研究
Author(s)	中村, 太一
Citation	
Issue Date	2007-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/3613
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における
ユーザインタフェースの
統一的な API 提供に関する研究

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

中村 太一

2007 年 3 月

修 士 論 文

ホームネットワークにおける ユーザインタフェースの 統一的な API 提供に関する研究

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

510072 中村 太一

提出年月: 2007 年 2 月

概 要

近年、各種家電のホームネットワーク規格の制定が行われ、規格に準じた製品も発売されるなどホームネットワークを構築する環境は年々整いつつある。そのような、ホームネットワーク環境下における新たなサービスとして、家電同士の連係動作や家電情報を統一的に管理し異常を検知する研究などが行われている [1][2]。しかし、ホームネットワーク環境では様々な UI が混在していると考えられ、そのような UI 一つ一つを考慮するサービス開発は開発者にとって大きな負担になると考えられる。そこで本研究では、ホームネットワークにおいて利用される UI を調査しそれぞれ扱うことができる入出力情報の定義し、その入出力情報から UI を統一的に扱うことができる API の提案を行う。その後、その API を利用し作成したアプリケーションがどのような振る舞いをするか検討し、本研究の成果について考察を行った。

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	ホームネットワーク	3
2.1	ホームネットワークとは	3
2.2	ホームネットワーク規格	3
2.2.1	ECHONET[3]	3
2.2.2	IEEE1394[4]	4
2.2.3	DLNA[5]	4
2.2.4	HAVi[6]	5
2.2.5	UPnP[7]	5
2.3	レガシーデバイスホームネットワーク	6
2.4	各種ホームネットワーク規格のまとめ	7
第3章	ユーザインタフェース	8
3.1	ホームネットワークにおけるユーザインタフェースとは	8
3.2	現在利用されているユーザインタフェース	8
3.2.1	リモートコントローラー（リモコン）	8
3.2.2	テレビ	9
3.2.3	GUI	9
3.2.4	マウス	9
3.2.5	携帯電話	10
3.2.6	家電本体	10
3.2.7	iPod	10
3.2.8	電子楽器	11
3.2.9	アロマジュール [11]	11
3.3	研究が行われているユーザインタフェース	12
3.3.1	水晶玉ディスプレイ (Magicscape)[12]	12
3.3.2	WaterScape[13]	12
3.3.3	Tactile driver	13

3.3.4	PHANTOM(ファントム)[14]	13
3.3.5	ミラーインタフェース [15]	13
3.3.6	ホームネットワークロボット [16]	14
3.3.7	Hug Shirt[17]	15
3.4	ホームネットワークにおけるユーザインタフェースのまとめ	15
第 4 章	入出力情報の定義	16
4.1	ユーザインタフェースそれぞれの入出力情報の定義	16
4.2	入力情報	16
4.2.1	基本入力情報	16
4.2.2	単純入力情報	16
4.2.3	高度入力情報	19
4.3	出力情報	21
4.3.1	パターン 1	21
4.3.2	パターン 2	25
4.3.3	パターン 3	26
4.3.4	パターン 4	28
4.4	それぞれのパターンにおける出力例	29
4.4.1	パターン 1 での出力例	29
4.4.2	パターン 2 での出力例	29
4.4.3	パターン 3 での出力例	31
4.4.4	パターン 4 での出力例	31
第 5 章	プライオリティ	32
5.1	API 利用時のプライオリティ	32
5.2	ユーザ・アプリケーション単位のプライオリティ	33
第 6 章	API の設計・提案	35
6.1	API の位置づけ	35
6.2	API を利用するために必要となる情報	36
6.3	入出力情報を扱う API の定義	36
6.3.1	基本入力情報	36
6.3.2	単純入力情報	37
6.3.3	高度入力情報	38
6.4	出力情報の API の定義	39
6.4.1	パターン 1	39
6.4.2	パターン 2	39
6.4.3	パターン 3	40
6.4.4	パターン 4	42

第 7 章	提案した API を利用したアプリケーション例と評価	43
7.1	アプリケーション例	43
7.1.1	テレビ会議システム	43
7.1.2	高齢者用住宅システム	44
7.1.3	有用性のまとめ	45
7.2	開発手順の考察	46
7.2.1	テレビ会議システム	46
7.2.2	API を利用した場合	46
7.2.3	API を利用しなかった場合	46
7.2.4	手順の比較	46
第 8 章	今後の課題	49
8.1	家電の入出力情報の定義	49
8.2	新しい入出力情報の出現	49
8.3	情報の定義	49
8.4	情報の型	49
8.5	非同期のユーザからの入力	50
第 9 章	まとめ	51
付 録 A	仕様 API 一覧	55
A.1	入力情報	55
A.2	出力情報	56
A.3	プライオリティ設定	57
A.4	その他	58

目 次

4.1	パターンの外部構造	22
4.2	パターンの内部構造	22
4.3	パターン 1 で” こんにちは”と出力	29
4.4	パターン 2 で” こんにちは”と視覚情報を用いて出力	30
4.5	パターン 2 で” こんにちは”と聴覚情報を用いて出力	30
5.1	プライオリティの決定方法	34
6.1	本研究におけるレイヤ構造	35
7.1	テレビ会議システムの代替的实现例	44
7.2	提案 API を利用した場合の手順例	48
7.3	提案 API を利用しなかった場合の手順例	48

表 目 次

4.1	デバイスの入出力情報 1	17
4.2	デバイスの入出力情報 2	18
4.3	基本入力情報	20
4.4	意向リスト	23
4.5	パターン 2	25
4.6	パターン 3,4	27
6.1	”datatype”とポインタの型の値 (入力情報)	37
6.2	識別子と情報の関係	41
6.3	”datatype”とポインタの型の値 (出力情報)	41

第1章 はじめに

1.1 研究の背景

将来的にホームネットワークを利用するユーザ層の幅や利用できるサービスの種類は多岐にわたると考えられる。そのような環境下で用いられる UI (ユーザインタフェース) は、リモコン、家電本体に取り付けられたボタンやダイヤルなどのように、従来の家庭内で用いられてきたものに加え、将来的には、PDA のような端末の GUI を用いて指示を与える UI が存在すると考えられる。また、ミラーインタフェースや音声入出力デバイスや人感センサーを搭載したネットワークロボットなど、今現在も研究が行われている UI も多数あり、ホームネットワークという環境下では多種多様な UI 混在していると予想される。

そのような複数の UI が混在する環境化において、ユーザにあるサービスの提供を行う場合、特定の UI に限定するのではなく、複数の UI の中から時間や場所などのユーザの状況に応じた UI を選択、あるいは組み合わせて提供し、ユーザとのインタラクションを実現するのが望ましいと考えられる。

しかし、複数の UI の利用を考えたアプリケーションの作成は、開発者側にとって大きな負担となることが予想される。

1.2 研究の目的

本研究の目的は、ホームネットワークにおいて利用される UI を統一的に利用するための API の定義を行うことである。

本研究ではまず、ホームネットワークにおいて用いられると考えられる UI を実用化されており普段の生活で使われているような身近なものから、現在も研究が行われており、将来的な実用化が見込まれるものまで幅広く調査・検討を行った。さらに、それぞれの UI の持つ機能の分析・分類を行い、UI 側とアプリケーション側でやりとりされる入出力情報を統一的に定義し単一の API として提案した。

提案した API を開発者が利用することによって、UI の動的選択を実現し、UI の種類に依存しないアプリケーションの作成を行うことが出来る。また、ユーザの場所や時刻などの状況に応じた UI の選択の実現や、組み合わされた複数の UI を利用したサービスの提供が可能となる。

1.3 本論文の構成

本論文は以下のような構成になっている

- 第一章
本研究の目的と背景。
- 第二章
現在のホームネットワークに関する規格の説明を行う。
- 第三章
将来的にホームネットワークの環境内で利用されるユーザインタフェースの考察を行う。
- 第四章
それぞれのユーザインタフェースで取り扱われる入出力情報を定義し、更にそれぞれを統一的に定義する。
- 第五章
複数のユーザやデバイス、アプリケーションがある場合、どのような形でプライオリティの設定を行うか考察する。
- 第六章
入出力情報の定義から API の作成を行う。
- 第七章
定義した API を利用したアプリケーションの作成例を示し、この API の有用性について考察する。
- 第八章
本研究の今後の課題を述べる。
- 第九章
本研究における全体のまとめを述べる。

第2章 ホームネットワーク

2.1 ホームネットワークとは

現在、ホームネットワークとはルータやハブなどを利用しコンピュータ間の通信を行う家庭内で構築した LAN のことを示すことが多い。しかし、本研究におけるホームネットワークとは、コンピュータだけではなく家電機器や住宅設備間をネットワークを介して様々な情報のやりとりを行うことによって、機器の連係動作や外部からの操作などを実現するネットワークのことを示す。

本章では現在提案されている各種ホームネットワーク規格についての説明を行う。

2.2 ホームネットワーク規格

2.2.1 ECHONET[3]

ECHONET とはエコーネットコンソーシアムが提唱している、主に設備系のネットワーク、そしてシステム開発を目的としたネットワークの規格のことである。

ECHONET 機器の通信レイヤは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、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。

アプリケーションソフトウェアは、コントローラなどにおいて、システムに接続される機器を遠隔制御するソフトウェアや、エアコンや冷蔵庫などの個別の機器においてその機器本体の機能を実現するソフトウェアである。

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、上位アプリケーションに対して一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は、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。

また、ドメインの外部とは、ECHONET ゲートウェイを介して接続できるようになっており、ECHONET 内にあるコントローラから直接機器を制御することができるだけでなく、遠隔操作による制御も可能になっている。このゲートウェイを利用することによって、利用者自身による遠隔操作のほかに、ガス漏れ通知などのリモート保守を行うこともできる。

2.2.2 IEEE1394[4]

IEEE1394 とは、AV 機器やコンピュータを接続する高速シリアルバス規格であり、アップルコンピュータ社が開発を先導し、IEEE によって標準化された。IEEE1394 は動画や音声を含む様々なデータをやりとりできるため、FireWire、i.LINK、DV 端子などの複数の名称が使われている。

特徴として、USB 同様プラグ&プレイおよびホットプラグに対応しているが、USB とは異なり、バス上にホスト機器を必要とせず、機器から機器へと接続するだけでデータ転送が可能になっている。そのため、多くの IEEE1394 対応機器は、ポートを 2 つ備えている場合が多い。また、伝送速度として 100Mbps、200Mbps、400Mbps の 3 種類を提供している。転送モードとして、動画や音声などのリアルタイム通信に適したアイソクロナス (Isochronous) 通信と、静止画やテキストなどを想定したアシンクロナス (Asynchronous) 通信の二つが用意されている。

2.2.3 DLNA[5]

DLNA とは、「Digital Living Network Alliance」の略で、ホーム・ネットワークでデジタル AV 機器同士やパソコンを相互に接続し、動画、音楽、静止画像のデータを相互利用する仕様を策定するため、業界標準に基づいた技術的な設計ガイドラインを査定することを目的とし、2003 年 6 月に設立された営利団体である。

DLNA は、コンテンツを提供するサーバーを DMS (digital media server)、再生するクライアントを DMP (digital media player) と呼び、DMS と DMP の間の接続条件を定めたものである。この規格に準拠した機器同士なら、単に線をつなぐだけで特別な設定なしに相互接続してコンテンツを共用できるようになる。

また、DLNA は新しいプロトコルを一から開発したものではなく、すでにある以下のような標準規格や汎用のプロトコルを組み合わせで作られている。

1. 物理層・リンク層

IEEE802.3i/u, 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 Architecture1.0

4. 機器制御

UPnP DCP(Device Control Protocol) の AV Version1.0

5. トランスポート層

HTTP(Hyper Text Transfar Protocol)

2.2.4 HAVi[6]

HAVi (Home Audio/Video Interoperability) は、デジタルオーディオ、デジタルビデオ等の家庭用 AV 機器をネットワークに接続して、相互運用を可能にするための標準プロトコルである。ホームネットワーク上で異なるベンダー製品の相互運用を可能にすることを目指しており、Jini(Java Intelligent Network Infrastructure) とともに相互接続可能になるはずなのだが、製品という形ではまだ登場していない。

HAVi では、下位層として IEEE1394 インターフェイス (iLINK または FireWire と呼ばれることもある) を使用して機器間を接続する。さらに上位インターフェイスとして、HAVi の API (Application Program Interface) やミドルウェアを規定している。これらにより HAVi ネットワークでは、ネットワーク上に存在する機器が自動的に検出・登録され、それら機器からなる分散ネットワーク環境が構築される。このネットワークに参加した機器は、他の機器 (他メーカーの機器) の機能を利用したり、逆に自分が持つ機能を他の機器に提供したりすることができるようになる。また各機器間での機能調整、アプリケーションやユーザー・インターフェイスの機器へのインストール、異なるブランドによる機器間での相互操作性などが実現される。

2.2.5 UPnP[7]

UPnP(Universal Plug and Play) は、マイクロソフトを中心に 1999 年 10 月 18 日に設立された、UPnP フォーラムが査定する仕様である。インターネットで標準となっている技術を基盤とし、ネットワークにつなぐだけで複雑な操作や設定作業を伴うことなく機能することを目指して作られている。

UPnP の仕様は、大きく分けて、アドレッシング、ディスカバリー、ディスクリプション、コントロール、イベント通知、プレゼンテーションが定義されている。UPnP では、各種機能を持ったデバイスと、そのデバイスに対して制御を行なうコントロールポイントの 2 者を定義している。デバイスは、自身の持つ機能を XML 形式で公開しており、コントロールポイントはそれを参照して、必要な機能を利用できる。

- アドレッシング

機器がネットワークに参加したさいに、アドレスを取得する方法について規定されている。通常はDHCPを用いるが、DHCPによるアドレス配布を利用できない場合、AutoIPの仕組みを用いてアドレスを決定する。

- ディスカバリー

SSDPというプロトコルを用いて、ネットワーク上のデバイスの検出を行なう。具体的には、ネットワークに参加したデバイスは、マルチキャストパケット(NOTIFYメソッド)の送出行ない、コントロールポイントはそれを受け取り、デバイスを検出する。また、コントロールポイント側からマルチキャストパケット(MSEARCHメソッド)を用いて問い合わせを行ない、デバイスが応答するモデルもある。

- ディスクリプション

デバイスが提供できる機能や情報を記述したXMLファイルである。デバイス自身が持つサービスなどについて記述したデバイスディスクリプションと、各サービスが持つアクションなどについて記述したサービスディスクリプションの2種類がある。

- コントロール

サービスの持つ機能呼び出すアクションと、デバイスの状態変数を問合せるクエリーがある。

- イベント通知

デバイスに対して、特定の状態変数を指定してイベント購読要求を行なうと、その状態変数の値が変化するたびに、イベントが通知される。

- プレゼンテーション

ウェブブラウザから、デバイスの状態の確認や、制御ができる。

2.3 レガシーデバイスホームネットワーク

レガシーデバイスホームネットワーク(LDHN)とは、既存の家電機器であるレガシーデバイスを用いて作成されたホームネットワークであり、家電をホームネットワーク対応機器に買い換えなくても、それらのサービスを提供できるシステムである。

現在のレガシーホームネットワークシステムは、レガシーデバイスを複数の部分状態機械からなる複合状態機械として捉え、状態機械に対する入力である赤外線信号を受け付けるセンサを配置し、レガシーデバイスの内部の状態情報を補足し、機器の状態情報を管理する。また、その制御は学習リモコンを用いて制御コマンドを送信し機器の制御を行う。

2.4 各種ホームネットワーク規格のまとめ

本章では、ホームネットワークで利用される通信規格についての調査を行った。どの通信規格も様々な伝送メディアを利用し情報家電や住宅設備の連携を目指していたが、いずれもユーザインタフェースの領域までを考慮した規格はなかった。そこで本研究において提案を行う API は、これらの通信規格に上位層に位置し、統一的に UI を扱えるものとして定義する。

第3章 ユーザインタフェース

3.1 ホームネットワークにおけるユーザインタフェースとは

ホームネットワークと呼ばれる環境下においては、従来まで用いられたインタフェースに加え、いままでにないような様々な種類のインタフェースが混在すると考えられる。この章ではホームネットワークにどのようなインタフェースが用いられるかを現在と将来的な両面から考察する。

3.2 現在利用されているユーザインタフェース

3.2.1 リモートコントローラー（リモコン）

現在、家庭内で最も身近で利用されているインタフェースの1つはリモコンであると考えられる。テレビ放送の多様化やAV機器の多様化に伴い、リモコンには複数の機器の多くの操作を行うことが求められており、種々の機能の操作信号をプリセットしたリモコンや学習リモコンなどが存在する。

学習リモコンとは、複数のリモートコントロール装置が発生させる操作信号を記憶（学習）して、一つのリモートコントロール装置で複数の機器を操作する事を可能にするための装置である。現在では、ネットワークに対応した機種[8]も発売されており、Webアプリケーションの画面で単純に機種を選び、目的にあった“アクティビティ”をリストから選択するだけで、高機能なマクロプログラムが出来上がるという製品も発売されている。

また、赤外線通信機能を持つ携帯電話では、テレビなどのリモコン操作ができるものも存在する。携帯端末などの赤外線通信の規格IrDAに準拠した赤外線ポートを利用して、リモコンの信号をエミュレートできるソフトも存在する。

最近では、任天堂から発売されWiiに付属するWiiリモコン[9]ではモーションセンサーを搭載することによって、振る、ひねる、指すといった直感的な動作の認識を可能にしている。

ホームネットワークという環境においても、液晶画面やGUIなどといった形の変更や機能の増加はあるかもしれないが、直感的にユーザが操作の指示を行えるリモコンは、ユーザとデバイスを繋ぐ一番身近なインタフェースであると考えられる。

3.2.2 テレビ

現在、最も技術の発達が著しい家電機器の1つとしてテレビがあげられると考えられ、特に2003年からの地上デジタル放送の開始と相まって薄型テレビの普及が進んでいる。薄型テレビの現在の主流は液晶テレビとプラズマテレビだと考えられ、一般的に液晶テレビは小画面～大画面（～65V型）、プラズマテレビは大画面のみ（32V型～103V型）の製品である。この2つは外見はほとんど変わらないが構造は全く異なり、一長一短がある。液晶テレビは最大消費電力が低く長寿命であるが、年間消費電力量が高く反応速度が遅い。プラズマテレビは、大画面であり年間消費電力が低いが、最大消費電力が高く小型化が困難なため、パーソナル用途には不向きである。その他の薄型テレビとしてはリアプロジェクションテレビや有機ELDテレビなどがあげられる。

ネットワーク機能としては近年、DLNAに対応し本体に10BASE-T/100BASE-TX × 1ポートを装備した製品も発表されている[10]。

3.2.3 GUI

GUI(Graphical User Interface)はコンピュータグラフィックスとポインティングデバイスを用いて、直感的な操作を提供するユーザインターフェースである。GUIでは画面上に、ウィンドウ、アイコン、ボタンといったグラフィックが表示され、ユーザはそれらの中から目的の動作を表すグラフィックスをマウスなどのポインティングデバイスで選択する。

現在、GUIを利用し操作を行うデバイスはPDAやニンテンドーDSなどが存在する。将来的には、リモコンや携帯電話などでGUIを利用し、指などを利用し情報を出力するデバイスが登場することも考えられる。

3.2.4 マウス

マウスは、手に持って移動させることによってボールや赤外線などを利用したセンサで移動を検知し、2次元座標をコンピュータに伝えるデバイスである。近年では、Bluetoothや独自方式の無線による「コードレスマウス」が次第にシェアを高めつつあり、特定のディジタイザ上のみで使用可能なマウス型デバイスといったものも存在する。マウスに備え付けてあるボタンは、Macintoshでは1つまたは4つ、PC/AT互換機では2つから5つ、UNIXマシンでは3つのボタンがついていることが多い。ボタンだけでなく、ホイールやトラックボールが表面に付いているものもあり、1次元座標の取得も可能である。

コンピュータを操作する他のデバイスとしては、タッチパネルや「Windows Vista」に搭載された音声認識などが考えられるが、マウスより操作性に優れているとは言えず、将来的にもコンピュータを操作する主要なデバイスはマウスであると考えられる。

3.2.5 携帯電話

携帯電話の端末には、アンテナ、スピーカー、マイクと、これらを制御する電子回路と、入力のためのボタンと、電源から成っているが、機能の増加からパーツは増える傾向にある。最近の端末ではディスプレイを搭載しており、液晶や無機 EL、有機 EL、発光ダイオードなどさまざまな素材が利用されている。アンテナは初期よりほとんど露出していたが、2003 年頃からは内蔵型の傾向が多くなり、現在の端末のアンテナはほとんどが内蔵型である。

出始めた頃は純粹に電話をする機能だけを搭載していたが、メールやカメラ機能などが搭載され、2006 年 4 月 1 日からは携帯機器を対象とする地上デジタルテレビジョン放送の「ワンセグ」が開始され、テレビの視聴も可能になった。

当初は通話機能だけであった携帯電話だが、音声通話のデジタル化により端末全体がデジタル化し、これによりデジタルネットワークへの接続が可能となった。デジタルネットワークの中でも、世界的に普及しているインターネットへの接続が早くから行われ、携帯電話でインターネット網にアクセスできるようになった。さらに、従来では携帯電話では携帯電話を対象にして作られた簡素な HTML による Web ページしか表示できなかったものが、近年ではフルブラウザが実現し、パソコン向けに作成されたコンテンツの閲覧が可能となった。また、NTT ドコモは 2007 年 3 月にもパソコン向けのサイトを定額で利用できるプランの開始を発表したことから、大手携帯キャリア 3 社の定額サービスが出揃ったことになり、今後ますます携帯電話のパソコン化が一段と加速すると考えられる。

このように、携帯電話は従来の電話専用端末から携帯情報端末機器として変化しており、家庭外からホームネットワークを操作するデバイスとして携帯電話を利用すると考えられる。

3.2.6 家電本体

家電本体にボタンが備え付けてあることも多く、そういったものを使ってユーザとのインタラクションをとることも多い。また現在、家電が動作してるかしてないかを確認すること自体もユーザとの情報のやり取りの 1 つだとも考えることが出来る。

3.2.7 iPod

iPod(アイポッド)はアップルコンピュータによって開発・製造及び販売されている携帯型デジタル音楽プレイヤーである。現在アップル社より発売されている iPod は第五世代であり、QVGA 2.5 インチ カラーディスプレイを搭載し、音楽の再生だけでなく MPEG-4 ビデオ、および H.264 ビデオや写真を閲覧することが可能である。すべての iPod はデザインが一貫しており、大きな液晶画面を備え、その下にはホイールがある。このホイール

を用いて選曲、音量調整、早送り、巻き戻し、画像・動画閲覧などすべての操作を直感的に行える。

また、2007年1月9日には携帯電話機能を搭載したiPhoneを発表した。iPhoneでは、キーパッドを廃しタッチパネル主体としたデザインが特徴的であり、電子メール、メッセージャー、Safari、タッチパネル、Wi-Fi (IEEE802.11b/g/n)、Bluetooth 2.0などの機能を備えるとされている。

3.2.8 電子楽器

電子楽器（でんしがっき）とは、電子回路による演算によって波形信号を作り出す楽器であり、電子回路の演算により、音の3要素である音程・音量・音色を制御することができる。エレクトーンや電子ピアノ、シンセサイザーなどが一般的によく知られている。

近年では、IEEE1394に対応した機器も開発されており、プロオーディオ機器やパソコンなどとのネットワーク構築を容易にする製品も発売されている。

3.2.9 アロマジュール [11]

アロマジュールとは、パソコンとUSB接続して使う香り発生端末であり、6種類のアロマオイルをセットし、ソフトウェア上でレシピをブレンドすることで、お気に入りの香りを何度も再現できる。ブレンド比率を1パーセント単位で調整できるため、初心者でも簡単にレシピ通りの香りがブレンドでき、WEB上に公開されているレシピのデータをダウンロードし、自分のパソコンに取り込んでその香りを再現することもできる。また、オリジナルの香りレシピを開発し、そのデータをパソコン内に保存したものを、メールに添付ファイルとして送信することによって、ネットワークを介した香りの共有も可能となる。

ユーザが利用するデバイスは主に、視覚や聴覚、触覚を利用して情報を受け取るものが主だが、このデバイスは嗅覚を利用する。いまはまだ、一般家庭に普及しているとは言い難いのが現状だが、例えばテレビ画面の動画と連動した香りの出力など、家電機器の連携サービスの中で利用される機会がくると考えることもできる。

3.3 研究が行われているユーザインタフェース

ここでは、まだ製品として市場には出回っていないが研究が行われているユーザインタフェースについて考察する。ここに書いたユーザインタフェース全てが製品化を目的として研究されているわけではないかもしれないが、それぞれのユーザインタフェースが持つ特徴、目的などを調べることによって、将来的にユーザインタフェースを介して、どのような情報がやり取りされるかを考察する。

3.3.1 水晶玉ディスプレイ (Magicscape)[12]

Magicscape とは日立製作所が開発した水晶玉をイメージさせる情報表示機器であり、ボタンは一切無く、手で包み込むようにジェスチャー操作し、透明球の内部に表示された映像を「没入感」と共に楽しむことが出来る。水晶玉はアクリルの半球を貼り合わせたものであり、接着面で下にあるディスプレイ画像が全反射するので、覗き込むと画像が見える仕組みとなっている。センサが台座についており、手をかざすと中を浮遊しているアイコン画像がそちらへ流れ、囲むようにすると寄ってきて、息をふっと吹きかけると消える。水晶玉は目の前にあるのだが焦点距離は遠く、画像はだいたい1m先にあるように見える。

使い方の例としては、カフェでの人待ちや美容院での待ち時間などの暇つぶしに軽く利用することが考えられる。

3.3.2 WaterScape[13]

WaterScape とは日立製作所が開発した情報端末であり、簡単で直感的なインタフェースを実現するために、本体外表面から一切のボタンを排除してある。本体内部に加速度センサを搭載し、ボタン操作の代替手段として本体を傾けたり、振ったりするジェスチャによる操作コマンドより、テキスト、動画、静止画、音楽といったコンテンツを、楽しみながらブラウジングすることが可能である。デザインコンセプトとしては、“深海に漂う情報の泡を舷窓からながめるような別世界感”をイメージの手がかりとし、思わず眺めつづけてしまう緩やかな画面の動きと、宙に浮いている円盤のような外観形状により、神秘的で飽きのこない情報環境を演出している。

仕様用途としては、自宅のPCに配信される好み情報はクレードル（データ転送用充電器）を介して Waterscape 本体に転送され、泡形状に加工されたコンテンツとして現れる。ユーザは、浮力で浮かび上がってくる泡を想像することで、初めての機器操作でも、容易に閲覧を楽しむことができる。

3.3.3 Tactile driver

Tactile driver とは日立製作所が製作したタッチパネルであり、押したときのボタンの重さや沈み込み量が自由に設定でき、本物のボタンを押したような感触が感じられる。タッチパネルを押す力を検出し、それに応じてパネル自体を前後に駆動することにより、本物のボタンの動きを忠実に再現し、ボタンごとに押す重さや、沈み込み量が変わるので、ボタンの役割に応じて設定が可能となっている。確実に押せたことが感触でフィードバックされるので、操作の確実性が向上期待できる。

仕様用途としては、銀行 ATM や駅券売機など、公共情報端末の操作デバイスとして利用することや、様々なモノの柔らかさや形状を表現する手段としても利用が期待できる。

3.3.4 PHANTOM(ファントム)[14]

PHANTOM(ファントム) とは、ハプティック (触覚) 機器と呼ばれるものであり、このデバイスを利用することによってコンピュータ上の仮想オブジェクト (CG モデル) に触れることができる。利用方法はまず、3 次元空間でロボット・アーム先に付いているペンを使い、コンピューター上の仮想オブジェクト上の形状や表面の感触のトレースを行う。そうやって仮想オブジェクトに触れることによって形状や質感など動きに応じた触覚情報をリアルタイムにユーザ側の指先にフィードバックする。更新レートも非常に高く約 1000hz で処理されるので、精密なモデルを再現することができる。既にインターネットを介しての触覚情報の伝達も成功しており、ネットワーク上での感覚情報のやり取りも期待できる。

利用方法としては、ネットショッピングなどにおいて購入前に衣の感覚を確かめることや、医療の現場において医学生が手術で腫瘍を取り除く練習を行う際、力の入れ具合を正確に感じる事が出来るなど、様々な利用方法が考えられる。

3.3.5 ミラーインタフェース [15]

ミラーインタフェースとは、自分の映像、遠隔地の映像、およびアプリケーション画面を、各々半透明にして重ね合わせることで、遠隔地間で疑似的な共有空間を生成し、その中で互いにインタラクティブな操作を可能とする、新しい双方向型の UI である。

この UI を利用することによって、2 地点の映像を重ねた共有空間の中でインタラクションが可能であるとともに、アプリケーションの表示を共有し、双方からそれに対する操作が可能となる。これにより、あたかも双方のユーザが同じ空間にいるような状況を実現しており、共同作業向きの会議空間の構築が可能となった。映像中の物体やアイコンとのインタラクションでは、カメラで撮影された自己像を、映像中の物体像に“タッチ”させる (画像上で重ねる) ことで実現する。これによって、遠隔の家電機器の操作や物体情報の表示があたかも触ったかのように可能となる。

また、画像認識を用いた手法により、ユーザが持つマーカの位置をリアルタイムに認識できるため、画面に直接触らずに物体やアイコンへの“タッチ”が実現できる。マーカ位置の認識は、あらかじめマーカの色を登録し、色認識処理等により実行する。これにより、従来のタッチパネルと異なり、画面から離れた位置から画面上のアイコンを選択できます（タッチレスなタッチパネルの実現）。さらに、物体に位置認識用の赤外線（IR）タグを装着すれば移動物体でも追跡し、その移動先の位置で“タッチ”することが可能となる。

このUIを利用することによって、空間共有型TV会議、操作を教える遠隔教育や遠隔指導、公共の場での情報端末、また見守りや監視等、さまざまな分野への応用が期待出来る。

3.3.6 ホームネットワークロボット [16]

近年、ASIMO や AIBO などといった様々な形のロボットが研究されているが、ホームネットワークという観点から注目したいのは、日本電気株式会社が開発した PaPeRo である。

PaPeRo は家庭内におけるユーザーのパートナーとしての利用を目的とし、インターネット経由で入手したニュースや天気予報の読み上げ、伝言メモ、テレビのリモコン操作などのほか、占いや健康チェックなどといった補助的な機能を備えている。そのような機能をささえるための高度な音声認識、画像認識、動画認識能力といった技術も持っており、音声認識機能では、複数のマイクを備えたことで音が到達する時間差から音源方向を検出する仕組みを利用し、ロボットが発話中でも音声認識が行えるなど、対話性能を向上させている。画像認識でも、OCR 機能を搭載したことで、手書き文字も認識することが可能となっている。更に画像認識で最も重要なのが、人の顔をどのように認識するかということであったが、認識のためにはある程度の光源や距離を必要としていたため、ユーザーとの距離によっては認識できないことも多々あった。そういった問題を解決するために、動画認識能力を利用することによって、目的のユーザーと離れていれば身体の動きを検知してそれが何か判別可能な距離まで近づくことを可能にしている。

それらの高度な技術を利用し、接する相手の対応から自身の口調や声質、動き方を変化させる機能も特徴として挙げることができる。数種類のキャラクター設定が用意されており、前述したユーザーの反応による自律変化と合わせることで、ダンスを褒めてあげるとダンス好きになるなど、まるで感情があるかのように振る舞うことが出来る。そのような振舞いもユーザとのインタラクションの一つとして方向付けされている。

このように、家庭内での利用を目的とするロボットは将来的にホームネットワークにおける重要なデバイスとして普及すると考えることが出来る。

3.3.7 Hug Shirt[17]

”Hug Shirt”とはBluetoothと”HugMe”というJavaアプリを利用し、抱きしめられる感覚を再現できるシャツであり、TIME 誌の 2006 年”Best Inventions”を受賞した商品である。

このシャツを着用した場合、シャツに埋め込まれているセンサーで腕の圧力、肌の温度、心臓の鼓動などをデータ化して携帯電話を経由して相手に送信し、シャツに埋め込まれた各種作動装置で、抱きしめる感覚を再現する。

3.4 ホームネットワークにおけるユーザインタフェースのまとめ

本章では、現在利用されているものから、将来的に利用が考えられるものまで幅広くユーザインタフェースの調査を行った。現在、家庭内で最も UI として利用されているリモコンは、ホームネットワークという環境下においても主要な UI であると考えられる。一方、”Magicscape”や”Waterscape”など現在研究が行われているインタフェースはユーザに与える情報量は少ないが、直感的に、操作方法を考えずに扱うことが出来る。これはホームネットワークという、多岐にわたるユーザ層が利用する環境において非常に重要なことであると考えられ、このような形式のユーザインタフェースがホームネットワーク内で利用されることは十分考えられる。また、ネットワークを利用することによって遠隔地との情報のやり取りを行えることから、”ミラーインタフェース”や”Hug Shirt”などといった、距離感を感じさせないサービスを行うためのデバイスが利用されることが考えられる。

このように、ホームネットワーク内には様々な種類の UI が異なる利用目的を持ちながらも、1つのネットワークでつながれた環境下に混在していると考えられる。

第4章 入出力情報の定義

本章ではまず、ホームネットワークにおける UI が扱う入出力情報の定義の分析を行い、入力情報、出力情報それぞれの統一的な定義を検討を行う。

4.1 ユーザインタフェースそれぞれの入出力情報の定義

ここでは、先の章で出てきた UI を中心にそれぞれがもつ入力情報、出力情報の定義をおこなった。同じ種類のデバイスであっても、製品によって持つ情報の誤差が生じると考えられるが、ここでは最低限そのデバイスが持つであろう入出力情報を定義している (表 4.1, 表 4.2)。

ホームネットワークという環境下におけるインタフェースは常に新しい形のものが、研究、開発されつづけると考えられる。そのような UI を本提案 API の一部として利用するためには、新しい種類のインタフェースの入出力情報を定義し、既存のシステムの中に組み込み必要がある。

4.2 入力情報

続いて表 4.1、表 4.2 における入力情報を元に、API 作成の為の入力情報の定義を行った。

4.2.1 基本入力情報

まず、基本的な入力情報の定義として体の部位を考慮した基本入力情報の定義を行った (表 4.3)。ホームネットワークにおいて、入力情報は体の一部を利用して入力されることが多いと考えられる。しかし、同じ種類の情報の入力が行われても入力する部位によって情報の正確性に違いがあると考えられる。そこで、情報を入力する際に利用する体の部分をメインクラスとして分類を行い定義した。

4.2.2 単純入力情報

YES、NO などの単純な情報の入力を受け付けるものを単純入力情報と定義した。YES、NO という単純答えの入力があれば、デバイス側がどのような形でユーザに情報の入力を

デバイス	操作方法	入力情報	出力情報
リモコン	ボタン	ON/OFF	-
Wii コントローラ	ボタン 十字キー ジェスチャ	ON/OFF 2次元座標 加速度 3次元座標	-
テレビ	ボタン	ON/OFF	静止画 動画 文字 音声
テレビ会議端末	ボタン マイク カメラ	ON/OFF 音声 静止画 動画	文字 静止画 動画 音声
ディスプレイ	ボタン	ON/OFF	静止画 動画
スピーカ			音声
マウス	ボタン 操作	ON/OFF 2次元座標	-
携帯電話	ボタン マイク カメラ	ON/OFF 音声 静止画 動画	文字 静止画 動画 音声 振動
固定電話	ボタン マイク	ON/OFF 音声	音声
家電本体	ボタン	ON/OFF	動作 LED
ipod(第四世代)	ダイヤル ボタン	1次元座標 ON/OFF	音声 文字
電子楽器	タッチ	ON/OFF	音声 光
アロマジュール	-	-	香り

表 4.1: デバイスの入出力情報 1

デバイス	操作方法	入力情報	出力情報
水晶玉ディスプレイ Magicscape	ジェスチャ(手) 息	3次元情報 圧力	文字 静止画
Waterscape	振る 傾ける	加速度 傾き	文字 静止画 動画
Tactile driver	タッチ	2次元座標 圧力	静止画 圧力
ファントム	タッチ		圧力
ミラーインタフェース	ジェスチャ	2次元座標	文字 静止画 動画 音声 合成映像
ホームネットワークロボット	マイク ジェスチャ	音声 ON/OFF	音声 ジェスチャ表現 表情表現
Hug shirt	ジェスチャ	温度	温度 圧力 振動

表 4.2: デバイスの入出力情報 2

行わせるかは問わない。

4.2.3 高度入力情報

基本入力情報より抽象度を高くしたものを高度入力情報として定義した。これらの入力情報は、基本入力情報において定義されている入力情報に加え、識別子や出力情報など、複数の情報を組み合わせることによって情報の入力が行えるものである。

- ボタン入力

ボタン入力は、ON/OFF+識別子を組み合わせることによって実現される

- 文字列・数字列入力

GUI を利用し画像 (出力情報)+2 次元座標 (入力情報) などを利用することが考えられる

- メニュー入力

複数の選択肢から、一つ又は複数の項目の選択を行う

音声 (出力情報)+音声 (入力情報) の組み合わせなどが考えられる

maintype	subtype	datatype	例 (操作方法 デバイス)
口 (1)		音声 (1)	喋る マイク
目 (2)	視線 (1)	虹彩 (1)	見つめる カメラ
	眼球運動 (2)	2次元座標 (1)	眼球運動 カメラ
手 (3)	指 (1)	ON/OFF(1)	操作 ボタン
		1次元座標 (2)	操作 ダイアル
		2次元座標 (3)	操作 タッチパネル
		3次元座標 (4)	指差し センサ
		圧力 (5)	触る タッチパネル
		指紋 (6)	触る センサ
	手 (2)	2次元座標 (1)	手を動かす センサ
		3次元座標 (2)	手を動かす センサ
		ON/OFF(3)	手を叩く センサ
足 (4)		圧力 (1)	踏む センサ
その他 (体全体)(5)	動き (1)	静止画 (1)	動き カメラ
		動画 (2)	動き カメラ
		ON/OFF(3)	動き センサ
		経度緯度 (4)	動き GPS
	体温 (2)	熱 () (1)	動き センサ

表 4.3: 基本入力情報

4.3 出力情報

提案する API において、開発者が指定する情報の抽象度を基準として4つのパターンを作成した。最も抽象度が低い出力パターンとは、現在使用されている API のように全ての出力情報の指定を行う。逆に、出力する形式やデバイスの指定などを行わず、ユーザーに伝えたい意向などの情報だけを指定する場合を最も抽象度の高いパターンとした。これらの間にある抽象度の出力として、さらに2つのパターンを定義した。出力形式のパターンを分けたことにより、開発者は情報の出力の抽象度を段階的に選ぶことができ(図4.1)、ユーザの状況と伝えたい情報に最も適した UI を利用して情報の出力を行える。

アプリケーションからそれぞれのパターンに渡された情報は、適切な処理を行いアプリケーションに戻される。これらのやり取りを繰り返すことによって、情報の抽象度を下げていき、デバイスから情報の出力を行う。パターンは1から4まで存在しているが、内部の処理では必ずしもすべてのパターンを経由していくことはなく、情報の性質やユーザの状況にしたがって動的に処理が行われる(図4.2)。

- パターン 1
伝えるデバイス、手段を問わずに表示する(意向を表示させる)
- パターン 2
デバイスは問わないが、指定した形式(感覚)で表示する
- パターン 3
デバイスを限定し、表示する
- パターン 4
デバイスを限定し、指定した形式で表示させる

4.3.1 パターン 1

パターン1では伝えるデバイス、手段を問わずに出力を行う。一番デバイス側からは遠く、ユーザの意向のみを受け取り情報の出力を行う。伝える意向を表の形として整理し、意向リストと定義した。この表における”mainsection”と”subsection”を指定することによってユーザに対する情報の出力を行うこととする。

意向リストの”outdata”欄にある情報に対応する音声、静止画、動画情報などは予め用意されているものとする。

1. maintype 挨拶

意向として挨拶を伝える場合のタイプ

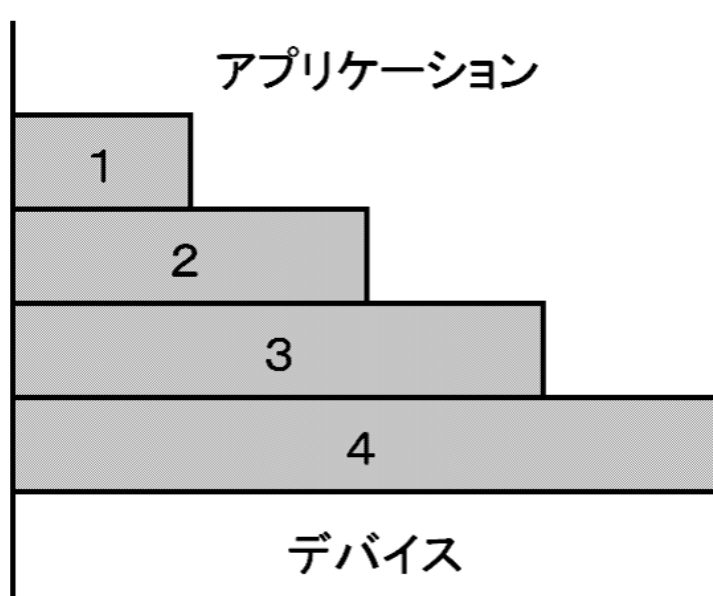


図 4.1: パターンの外部構造

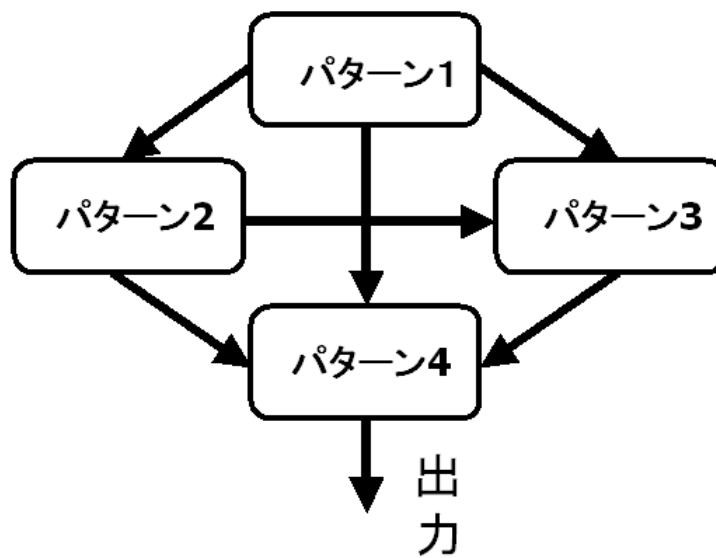


図 4.2: パターンの内部構造

maintype*	subtype*	outdata
挨拶 (1)	はじまり (1)	おはよう こんにちは こんばんは
	終わり (2)	さようなら
	感謝 (3)	ありがとう
警告 (2)	気象 (1)	気象地震警報など
	火災 (2)	火災警報
	ガス漏れ (3)	ガス漏れ警報警報
	水漏れ (4)	水漏れ警報
	侵入者検知 (5)	侵入者警報
単純 (3)	OK(1)	OK
	NO(2)	NO
	エラー (3)	エラー
ユーザ定義 (4)	-	-

表 4.4: 意向リスト

- subtype はじまり
セッションの開始を意向として伝える場合に利用するタイプ
”outdata”にある3つのデータのうち、どれを出力するかは時間によって適切に判断される
- subtype 終わり
セッションの終わりを意向として出力する際に選択する
- subtype 感謝
意向として感謝を出力する際に選択する

2. maintype 警告

ユーザーに対して警告を意向として伝えるタイプ

- subtype 気象
大雨や大雪など気象に関する警報の意向を出力する
- subtype 火災
家庭内で火災に関する警報の意向を出力する
- subtype ガス漏れ
家庭内でガス漏れが起きている時に警報を出力する

- subtype 水漏れ
家庭内で水漏れの恐れがある際に警報を出力する
- subtype 侵入者検地
家庭内に未確認の侵入者があった際に警報を出力する

3. maintype 単純

単純な受け答えを意向として伝えるタイプ

- subtype OK
OK という単純な意向を出力する際に選択する
- subtype NO
NO という単純な意向を出力する際に選択する
- subtype エラー
エラーを出力する際に選択する

4. maintype ユーザ定義

上の3つでは足りない情報をユーザ定義として定義するタイプとした

maintype*	datatype(*)	デバイス例
視覚 (1)	文字 (1)	テレビ
	静止画 (2)	ディスプレイ
	動画 (3)	ディスプレイ
	光 (4)	LED
	拳動 (5)	ホームネットワークロボット
聴覚 (2)	音声 (1)	スピーカ
味覚 (3)	味 (1)	味覚出力デバイス
嗅覚 (4)	香り (1)	アロマージュール
触覚 (5)	圧力 (1)	ファントム
	温度変化 (2)	Hug shirt
	電気刺激 (3)	スマートタッチ
	振動 (4)	携帯電話

表 4.5: パターン 2

4.3.2 パターン 2

ここでは、五感に関する情報を元にそれぞれの分類を行った。この出力パターンを利用する場合は、”maintype”(感覚情報)の指定を行う必要がある。更に、”datatype”の指定も行い情報の出力を行うことが出来る。

1. maintype 視覚視覚情報を出力する

- datatype 文字
文字として情報を出力する
- datatype 静止画
静止画として情報を出力する
- datatype 動画
動画情報として情報を出力する
音声情報も付加している場合は、音声情報の出力も行う
- datatype 光
光情報として出力を行う (モールス信号など)
- datatype 拳動
ロボットなどが拳動を示すことによって情報の出力を行う

2. maintype 聴覚

聴覚情報を出力する

- datatype 音声
音声を情報として出力する

3. maintype 味覚

味覚情報を出力する

- datatype 味
味を情報として出力する

4. maintype 嗅覚

嗅覚情報を出力する

- datatype 香り
香りを情報として出力する

5. maintype 触覚

触覚情報を出力する

- datatype 圧力
圧力として情報を出力する
- datatype 温度変化
温度情報を出力する
- datatype 電気刺激
情報を電気刺激として出力する
神経信号なども電気刺激情報として出力する
- datatype 振動
情報を振動として出力する

4.3.3 パターン 3

パターン 3 では、デバイスを限定し情報の出力を行う。そのデバイスが持つ出力情報 (datatype) を指定する必要がある。表 4.6 における "maintype" のみ指定する。

maintype*	datatype(*)
テレビ (1)	静止画 (1) 動画 (2) 文字 (3) 音声 (4)
携帯電話 (2)	文字 (1) 静止画 (2) 動画 (3) 音声 (4) 振動 (5)
テレビ会議端末 (3)	静止画 (1) 動画 (2) 音声 (3)
ipod(第四世代)(4)	音声 (1) 文字 (2)
アロマジュール (5)	香り (1)
水晶玉ディスプレイ (6) Magicscape	文字 (1) 静止画 (2)
Waterscape(7)	文字 (1) 静止画 (2) 動画 (3)
Tactile driver(8)	静止画 (1) 圧力 (2)
ファントム (9)	圧力 (1) 3次元座標 (2)
ミラーインタフェース (10)	文字 (1) 静止画 (2) 動画 (3) 音声 (4) 合成映像 (5)
ホームネットワークロボット (11)	音声 (1) 拳動 (2) 表情表現 (3)
Hug shirt(12)	温度 (1) 圧力 (2) 振動 (3)

表 4.6: パターン 3,4

4.3.4 パターン 4

パターン 4 では、デバイスと指定した情報での出力を行う。このパターンの形が、普段使う API の形に一番近いと考えられる。表 4.6 における”maintype”と”datatype”を指定する。

また、このことはパターン 3 でも言えることだが、デバイスを指定して情報の出力 (パターン 3,4) を行った後すぐに情報の入力を求めた場合、その指定したデバイスに求められた入力情報を処理することが可能ならば、指定されたデバイスで処理を行うこととする。これは、開発者が意図したデバイスでの処理を望んでいる状況にもかかわらず、処理を行うデバイスの変更を阻止するためである。

4.4 それぞれのパターンにおける出力例

それぞれのパターンにおいて”こんにちは”という情報をユーザに伝える例を示す。ユーザの周りには”リモコン”、”テレビ”、”ディスプレイ”、”ネットワークロボット”の4種類であり、その環境はどのパターン出力でも変わらないとする。

4.4.1 パターン1での出力例

パターン1では意向リスト(表4.4)の、”あいさつ”・”はじまり”を選択することによって出力される。”こんにちは”という意向を出力するために、文字や動画、音声などが用意されているので、出力されるデバイスは図4.3で示すとおり、”テレビ”、”ディスプレイ”、”ネットワークロボット”の3種類が考えられる。どのデバイスが採用されるかは、ユーザのプライオリティに関する設定で決定される。

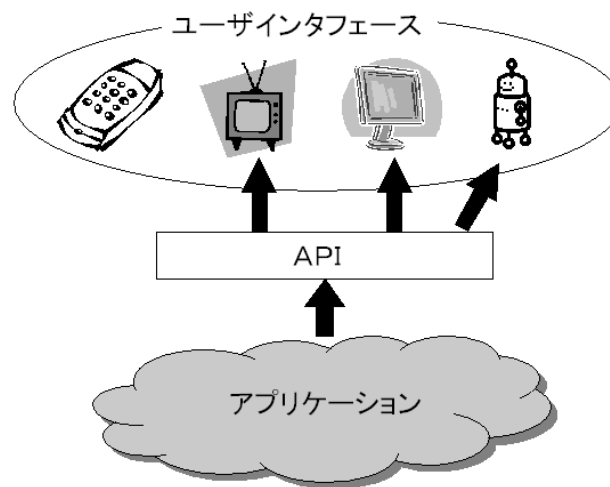


図 4.3: パターン1で”こんにちは”と出力

4.4.2 パターン2での出力例

パターン2ではどの五感を利用するか指定し、情報の出力を行う。まず最初の例として、視覚情報で”こんにちは”と出力を行う決定をしたとする(表4.5)。その場合、”こんにちは”と文字での出力が決定された場合利用されるデバイスは”テレビ”又は”ディスプレイ”の2種類が考えられる(図4.4)。

さらに、聴覚情報での”こんにちは”という出力を決定した場合は音声で扱えるデバイスの出力が行われる。この場合に利用されると考えられるデバイスは、”テレビ”又は”ネットワークロボット”の2種類となる(図4.5)。

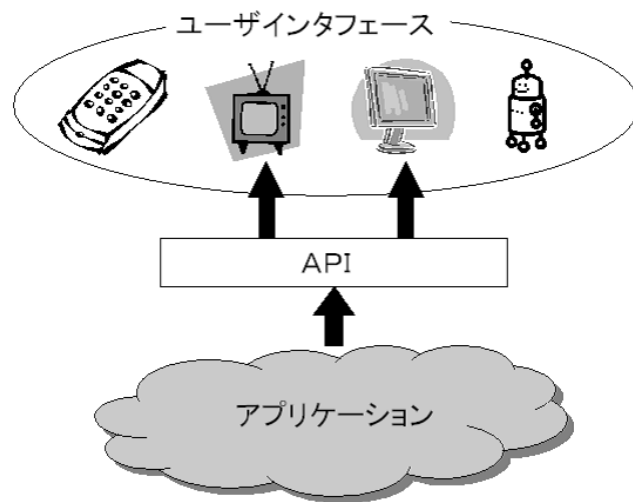


図 4.4: パターン 2 で”こんにちは”と視覚情報を用いて出力

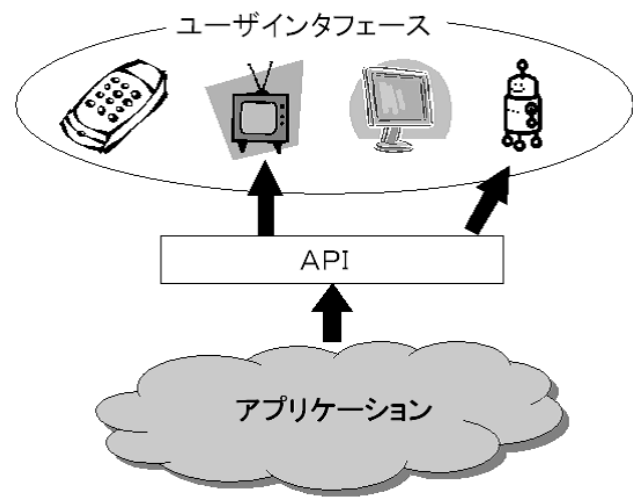


図 4.5: パターン 2 で”こんにちは”と聴覚情報を用いて出力

4.4.3 パターン 3 での出力例

パターン 3 ではデバイスの指定を行い、情報の出力を行う (表 4.6)。例えば、テレビでの”こんにちは”という情報出力を選択した場合、文字で出力されるか、音声で出力されるかはそのデバイスにつけられたプライオリティによって判断される。なお、動画や静止画などで”こんにちは”という情報が出力可能な場合 (データとして保持していた場合) はそれらの情報を用いて情報の出力を行うことも可能となる。

4.4.4 パターン 4 での出力例

パターン 4 ではデバイスと出力情報形式の選択を行い出力する (表 4.6)。この場合、開発者が指定したデバイスにおいて、指定した情報形式で適切に出力が行われる。

第5章 プライオリティ

ホームネットワークの環境下において、サービスを行う際に複数ある選択肢の中から一つを選ぶことは、頻繁に発生すると考えられる。本章ではそのような状況はどのような時に起こり、それを解決するためのプライオリティ設定の考察を行う。

5.1 API 利用時のプライオリティ

定義した API を利用した場合に必要なプライオリティの考察を行う。まず、情報の出力時にパターンが 1 から 4 に順番に遷移したと仮定し、どのプライオリティが必要になるか考察する。最初に、パターン 1 からパターン 2 に遷移する際には、パターン 2 においてどの感覚情報で出力するか決定する必要性が生じる。さらにその決定した感覚内において、どの表 4.5 のどの”datatype”からの出力を行うかを考える必要性もでてくる。

続いて、パターン 3 に遷移するが、ここではどのデバイスでの出力を行うか決定する必要性がでて来る。例えば、表 4.5 の”datatype”を文字と選択していた場合、文字の出力を扱える複数のデバイスの中でどれから出力を行うか決定する必要がある。さらに同じ種類のデバイスが 2 つ以上があった場合、選択が必要となることから、デバイス毎のプライオリティも必要になると考えられる。これはパターン 3 でも同じことが言えるが、直接パターン 4 を指定した場合でも複数の同じ種類のデバイスがあった際、それぞれのプライオリティが必要になるとわかる。

このことから、パターン 2 ~ 4 にはそれぞれ以下の部分のプライオリティが必要になることがわかる

- パターン 2

表 4.5 における”maintype”(感覚) と”datatype”

- パターン 3

表 4.6 における”maintype”(デバイス) と”datatype”

- パターン 4

複数の同一種類のデバイスがあった際、デバイス毎の優先順位

以上のことから、情報の出力を行う際、以下のプライオリティを保持する必要があると考えられる。

- 感覚情報のプライオリティ
- ホームネットワーク内にあるデバイス全ての出力情報 (4.5,4.6 における”datatype”) のプライオリティ
- デバイスの種類ごとのプライオリティ
- ホームネットワーク内にある全てのデバイスのプライオリティ

ここで着目したいのはデバイスのプライオリティである。デバイス選択は情報を出力する際に必要なだけでなく、情報の入力を行う時にも必要になると考えられる。さらに、ユーザそれぞれの生活規則に基づいたプライオリティの設定も必要になってくると考えられる。例えば、家事など作業を行う際は音声での出力を優先し、休憩時間には文字での出力を優先するといった時間帯に応じたプライオリティも必要になってくると思われる。そういった状況を考慮し、時間帯に応じたプライオリティの設定も可能にしておく。

5.2 ユーザ・アプリケーション単位のプライオリティ

続いて考えられるプライオリティはユーザ毎のプライオリティである。ホームネットワークという環境下では複数ユーザが同時に生活していることが考えられ、同じ環境下でサービスを受ける際、どちらのユーザのプライオリティを優先するか決定する必要がある。さらに、サービスを行うアプリケーションが優先したいデバイスを保持していることも考えられる。そこで、それぞれのプライオリティ決定を行うために以下のようなプライオリティを定義した。

- ユーザ間プライオリティ
 - 複数のユーザが同じ環境下に存在した場合、どのユーザのプライオリティを利用するか決定する
- 基本プライオリティ
 - デフォルトとして定義されているプライオリティ
 - 先に出てきた、情報の入出力を行う際に必要なプライオリティ全てを保持している
- アプリケーションプライオリティ
 - そのアプリケーション利用する際に考慮するプライオリティの設定
 - 基本プライオリティとの変更部分（考慮したい部分）のみの情報を設定しておく

- ユーザプライオリティ

- ユーザ毎に異なる部分のプライオリティの設定を、基本プライオリティと異なる部分だけ保持する
- なお、ゲストなどホームネットワーク外からの利用者の場合は基本プライオリティに沿って処理を行う

それぞれの優先度は、ユーザプライオリティ > アプリケーションプライオリティ > 基本プライオリティ、とした。処理の流れは図 5.1 に示した。

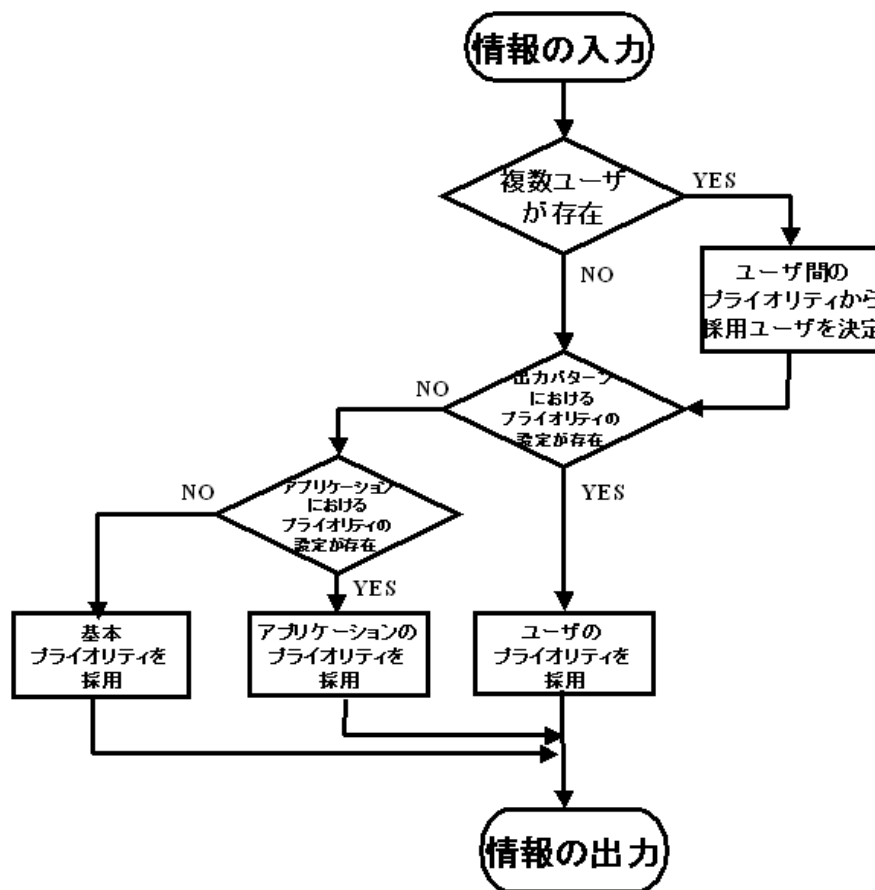


図 5.1: プライオリティの決定方法

第6章 APIの設計・提案

この章では、先の章で定義した入出力情報を元に API の定義を行う。

6.1 APIの位置づけ

提案する API が利用されるシステムの構成を図 6.1 に示す。提案する API は UI とアプリケーションの間に実装される。アプリケーションは複数のサービスや、コンテキストから構成される。アプリケーションの下位には UPnP と LDHN が存在している。UPnP が複数のホームネットワーク規格の上位に存在している理由は、UPnP がそれぞれの規格の上位にあることによって、それぞれの規格のデバイスが IP で認識、制御することが可能となるからである。それぞれの規格と UPnP の間には各々、ゲートウェイが存在し規格間の差異を吸収しているものとする。

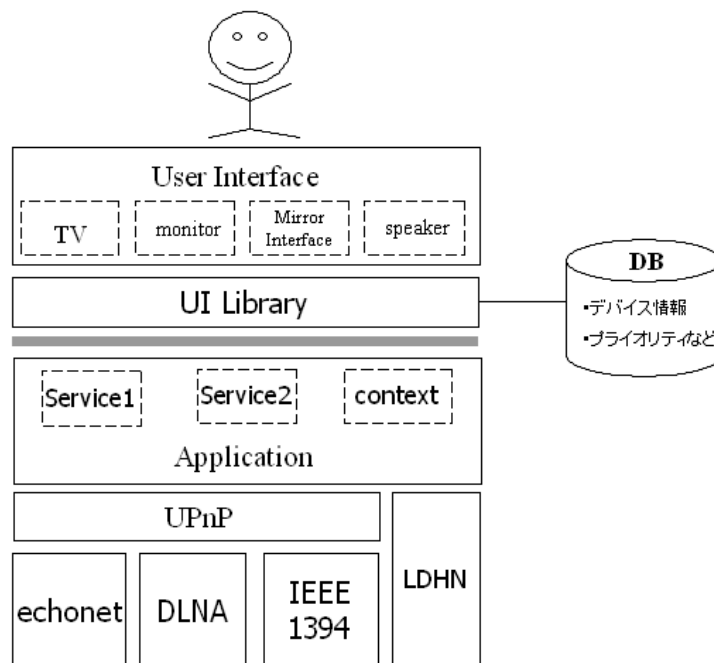


図 6.1: 本研究におけるレイヤ構造

6.2 APIを利用するために必要となる情報

ホームネットワークでは、ユーザの移動やデバイスの購入などによる環境変化が頻繁に発生するものと考えられる。そういった環境下で、提案する API を利用するためにはどのような情報を把握する必要があるかを考察する。

- ホームネットワーク内にあるデバイス
ホームネットワーク内にあるデバイス全ての情報を把握する必要がある。更に、それぞれに固有の識別子を登録し管理する。
- ホームネットワーク内で利用できる入出力情報
ホームネットワーク内にあるデバイスそれぞれが扱える入出力情報を管理する必要がある。特に同じ種類のデバイスであっても扱う入出力情報の違いが生じることも考えられるので、デバイスそれぞれの識別子との対応をとる。
- ユーザ環境
今現在、ユーザが利用できるデバイスを把握する必要がある。
- プライオリティ
先の章で考察した各種プライオリティを保持する必要がある。

6.3 入出力情報を扱う API の定義

6.3.1 基本入力情報

```
void *inputdata(int maintype, int subtype, int datatype)
```

- int maintype
表 4.3 の”maintype”の () 内の数字を指定する
- int subtype
表 4.3 の”subtype”の () 内の数字を指定する
なお、数字が存在しない場合は”0”を指定する
- int datatype
表 4.3 の”datatype”の () 内の数字を指定する 3

呼び出した側でポインタの開放を行う必要があり、値の取得に失敗した場合は NULL を返す。”datatype”対応する戻り値の型は表 6.1 のようになっており、予め呼び出す側で定義する必要がある。

使用例

この API を利用して指を利用した 2 次元の座標の情報を得る場合、表 4.3 から”maintype”は”3”、”subtype”は”1”、”datatype”は”3”となる。この場合、表 6.1 より戻り値は要素数 2 の int 型の配列のポインタとなる。

```
int *axis;  
axis = inputdata(3,1,3);  
//axis[0],axis[1] にそれぞれ座標が入力される  
free(axis);
```

datatype	ポインタの型
ON/OFF	int 型
1 次元座標	int 型
2 次元座標	要素数 2 の int 型の配列
3 次元座標	要素数 3 の int 型の配列
経度緯度	要素数 2 の int 型の配列
圧力	int 型
熱	int 型
指紋	char 型
虹彩	char 型
音声	char 型
静止画	char 型
動画	char 型

表 6.1: ”datatype” とポインタの型の値 (入力情報)

6.3.2 単純入力情報

```
int inputSimpleData()  
OK なら 0、NO なら 1 を返す
```

使用例

OK なら 0、NO なら 1 が戻り値として返ってくる。

```
int num;  
num = inputSimpleData()
```

6.3.3 高度入力情報

1. int inputButtom()

戻り値はそれぞれ表 6.2 に対応した識別子を戻す。エラーの場合は-1 を返す。

使用例

戻り値は、対応する識別子の値が入る。

```
int num;  
num = inputButtom();
```

2. char *inputDataChar()

文字列の入力を受けつける。戻り値として文字列のポインタを返す。エラーの場合は NULL を返す。

使用例

文字列の入力を受けつける。エラーの場合は NULL を返す。

```
char *str;  
str = inputDataChar();
```

3. int inputDataNum()

数字列の入力を受け付ける。戻り値として数字列を返す。エラーの場合は-1 を返す。

使用例

数字列の入力をうけつける。エラーの場合は-1 を返す。

```
int num;  
num = inputDataNum();
```

4. char *inputMenu(char *menudata,int selectNum)

メニュー入力を受け付ける。選択肢ごとを”,”で区切った文字列アドレスを munudata にポインタとして渡す。selectNum は選択肢の中から何項目選ぶかを指定する。戻り値は、選択された項目の文字列のポインタとなる。複数の項目が選択された場合は、”,”で繋げて返ってくる。

使用例

”red”, ”blue”, ”yellow”の中から一つ選択する。戻り値は、選択されたデータの文字列のポインタとなる。

```
char *menudata,*str;  
menudata= "red,blue,yellow";  
str = inputMenu(menudata,1)
```

6.4 出力情報の API の定義

6.4.1 パターン 1

```
int putPattern1(int maintype,int subtype)
```

- int maintype
表 4.4 の”maintype”の () の数字を指定する
- int subtype
表 4.4 の”subtype”の () の数字を指定する

成功した場合は 0 を返し、失敗ならそれ以外の値を返す。

使用例

パターン 1 で”こんにちは”と出力する例を示す。表 4.4 より”maintype”は 1、”subtype”は 1 を指定することによって、”はじまり”を意味する意向を指定した。”こんにちは”と出力されるかは時間に依存する。正しく出力された場合は 0 が戻り値として返される。

```
int num;  
num = putPattern1(1,1);
```

6.4.2 パターン 2

```
int putPattern2(int maintype,int datatype,void *data,int time)
```

- int maintype
表 4.5 の”maintype”の () の数字を指定する

- int subtype

表 4.5 の”datatype”の指定を行うことが出来る
指定する場合は”datatype”の () の数字を指定する
指定しない場合は 0 を入れる

- void *data

表 6.3 の”datatype”に対応する型のポインタを指定する

- int time

出力する時間間隔の指定を行う
必要ないときは”0”を指定する

成功した場合は 0 を返し、失敗ならそれ以外の値を返す。

使用例

パターン 2 で”こんにちは”という情報を視覚情報で伝える例を以下に示す。適切に処理が行われたら 0 が戻り値として返される。

```
int num;
char *str;
str = "こんにちは"
num = putPattern2(1,0,str,0);
free(str);
```

6.4.3 パターン 3

```
int putPattern3(int maintype,void *data,int time)
```

- int maintype

表 4.6 の”maintype”の () の数字を指定する

- void *data

表 6.3 の”datatype”に対応する型のポインタを指定する

- int time

出力する時間間隔の指定を行う
必要ないときは”0”を指定する

成功した場合は 0 を返し、失敗ならそれ以外の値を返す。

識別子の値	対応する情報
1	電源 ON/OFF
2	音量 UP
3	音量 DOWN
4	画面サイズ
5	画面表示
6	入力切替
7	CATV
8	音声切り替え
9	消音
10	オンタイマー
11	オフタイマー
12	メニュー
13-24	1-12

表 6.2: 識別子と情報の関係

datatype	ポインタの型
文字	char 型
静止画	char 型
動画	char 型
光	char 型,int 型
拳動	int 型
熱	int 型
音声	char 型
味	int 型
香り	int 型
圧力	int 型
温度	int 型
電気刺激	int 型
振動	int 型
合成映像	要素 2 の char 型の配列
表情表現	int 型

表 6.3: "datatype" とポインタの型の値 (出力情報)

使用例

パターン3で、“こんにちは”とテレビを用いて出力する例を示す。表4.6より”maintype”は1となる。静止画、動画、文字、音声で出力されるかどうかは、ユーザのプライオリティによって決められる。正しく出力された場合は0が戻り値として返される。

```
int num;
char *str;
str = "こんにちは"
num = inputPattern3(1,str,0);
free(str);
```

6.4.4 パターン 4

```
int putPattern4(int maintype,int datatype,void *data,int time)
```

- int maintype
表 4.6 の”maintype”の () の数字を指定する
- int datatype
表 4.6 の”datatype”の () の数字を指定する
- void *data
表 6.3 の”datatype”に対応する型のポインタを指定する
- int time
出力する時間間隔の指定を行う
必要ないときは”0”を指定する

成功した場合は0を返し、失敗ならそれ以外の値を返す。

使用例

パターン4で、“こんにちは”とテレビの文字を利用して出力する例を示す。表4.6より”maintype”は1となり、subtypeは”3”となる。正しく表示された0が戻り値として返される。

```
int num;
char *str;
str = "こんにちは"
num = inputPattern4(1,3,str,0);
free(str);
```

第7章 提案したAPIを利用したアプリケーション例と評価

提案APIを利用したアプリケーション例や、開発者の負担軽減などの考察を行い、本研究において提案したAPIの評価を行う。

7.1 アプリケーション例

提案APIを利用したアプリケーションの作成例を示し、APIの有用性の考察を行う。現在利用されているアプリケーションをこのAPIを利用し発展させた形や、将来的に実現されると考えられるアプリケーションをこのAPIを利用することによって、どのような機能の提供が行えるか考察する。

7.1.1 テレビ会議システム

ホームネットワーク内におけるテレビ会議システムを提案APIを利用し作成した場合、どのような効果が得られるか考察する。

現行のテレビ会議システム

現行のテレビ会議システムはメーカー間の規格は統一され機種に関わりなく多地点会議は可能である。しかし、テレビ会議システムを利用する為にはカメラを装備した専用の機器を購入する必要がある。PCを利用したテレビ会議システムも存在するが、PC間のみの通信が可能のため拡張性は今のところない。

提案APIを利用した場合

提案APIを利用した場合におけるテレビ会議システムを考える。まず、現行どおりに専用の端末を利用したい場合はパターン4(表4.6)のテレビ会議端末を指定する。

さらに、テレビ会議システムの実現には入出力情報としてそれぞれ

- 入力情報
音声、動画
- 出力情報
音声、動画

が必要だと考えられる。この入出力情報の定義から、過不足なく入出力情報を満たすものを表 4.1 から探すと携帯電話がある。このことから、パターン 4 で携帯電話を指定することによってテレビ会議端末の代わりとして携帯電話を利用できる。

また、開発者が基本入力情報として”音声”と”動画”（表 4.3）。パターン 2 の出力情報として”音声”と”動画”の指定を行ったとする（表 4.5）。このようにデバイスの指定を行わなかった場合はそれぞれユーザの環境に適したデバイスが選択される。例えば、ディスプレイが近くにあった場合にはディスプレイの出力情報の利用や、マイクがあった場合はマイクの音声入力を利用することでの実現が考えられる（図 7.1）。

以上のように、デバイスの指定をせずに情報の入出力を行うことはユーザの環境にあったデバイスを動的に選択することだけでなく、既存のデバイスを利用し複雑なサービスの代替的实现なども可能になると考えることができる。

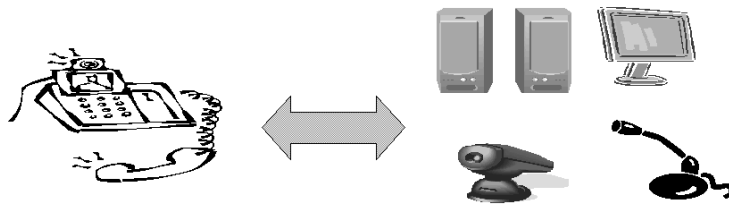


図 7.1: テレビ会議システムの代替的实现例

7.1.2 高齢者用住宅システム

近年、少子高齢化が社会的問題になってきており 2025 年には全人口の約 30 %が高齢者 (65 歳以上) になると言われている [18]。そのような社会環境において、ホームネットワーク技術は距離を越えたサービス提供を行えることなどから非常に有用であると考えられる。その中で、提案 API によって与えることができる機能の考察を行う。

提案 API を利用した場合

情報の入出力という観点からそれぞれ考察する。

- 情報の入力

高齢者の方が利用するという観点から、複雑な操作方法を正しく理解し、使いこなすということは期待できないと思われる。このことから、高齢者の方が利用するためにはわかりやすく、直感的に操作できる必要があると思われる。

本提案 API では基本情報入力は体のどの部分を利用するかという視点から定義しており、この定義を利用することにより、直感的に利用するアプリケーションの作成がやりやすいと思われる。

- 情報の出力

高齢者になると目や耳の衰えがひどく、日常生活に支障している方はめずらしくない。このような方々に対して情報伝達を行うためには、その方が受け取ることの出来る情報形式で伝達を行う必要があると考えられる。

そこで、本研究における提案 API を利用することによって、目が悪い方には音声で情報の伝達を行い、逆に耳の悪い方には視覚情報として出力するなど、その人の環境に合った伝達手段を選ぶことが出来る。更に、視覚や聴覚などでは困難な場合は振動モーター付きベッドなどを新しいUIとして開発し、振動を起こすことによって情報伝達を行うなど、今はまだ発売されていないようなUIへの将来的な応用も可能になると考えることが出来る。

7.1.3 有用性のまとめ

これらのアプリケーション例により、本研究における提案 API の有用性は以下のようなものが考えられる。

- ユーザの環境に合ったデバイスの利用

高齢者用住宅システムの例より、ユーザの希望する情報形式での出力シナリオを示した

- 既存のデバイス機能を代替的に利用

テレビ会議システムの例より、専用端末がなくても既存のデバイスを組み合わせて代替的な機能の実現例を示した

- 幅広いデバイスに対応し、将来的に利用されるデバイスにも対応

高齢者用住宅システムより、将来的なデバイスに対する利用例を示した

7.2 開発手順の考察

先のセクションでは、ユーザ側から見た API の有用性について考察してきたが、ここでは開発者側からの API に対する評価を行いたいと思う。

7.2.1 テレビ会議システム

先のセクションでテレビ会議システムを既存の複数の UI を使うことで、代替的な実現する例を示した。ここでは、その複数 UI を利用して代替的にテレビ会議システム実現する場合、本提案 API を利用した開発手順と利用しない場合の開発手順の違いについて考察する。なお、提案 API を利用した場合も利用しなかった場合も、代替的に利用するデバイスは図 7.1 の形になるとする。

7.2.2 API を利用した場合

まず、テレビ会議システムに必要な入出力情報は

- 入力情報
音声、動画
- 出力情報
音声、動画

なので、それぞれ 4 つの API を呼び出すこととなる。音声と動画はそれぞれ基本入力情報を利用し、音声と動画の出力はパターン 2 を利用する (図 7.2)。

7.2.3 API を利用しなかった場合

API を利用しない場合は、利用する入出力情報毎にユーザが利用できる環境かどうか調べる必要があり、利用できるデバイスが存在した場合、それぞれ利用を開始することになると思われる。処理のばらつきなどにより、図 7.3 のような順番で処理が行われるとは限らないが、最終的な手順の合計は変化しないと思われる。

7.2.4 手順の比較

API を利用した場合とそうでない場合の手順の比較を行いたいと思う。やはり、API を利用していない場合はユーザの環境に利用できるデバイスが存在するか調べる必要性があるため手順の数は多くなっている。今回は入出力情報をそれぞれ違うデバイスで担当するという前提で行ったことから、大きな差がついたとも考えられるが、提案 API を利用し

た場合にはプライオリティなどの考慮も同時に行ってくれるという点を含めて考えると、提案 API を利用した方が効率は良いと言える。

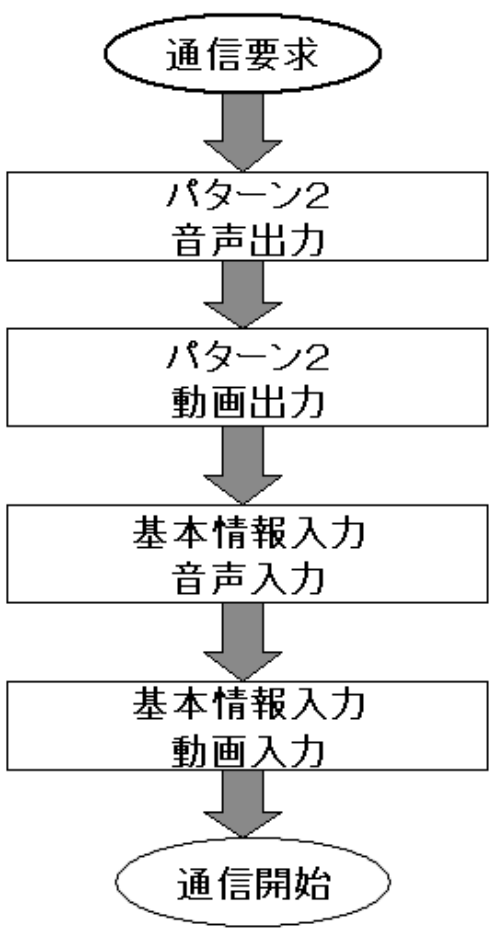


図 7.2: 提案 API を利用した場合の手順例

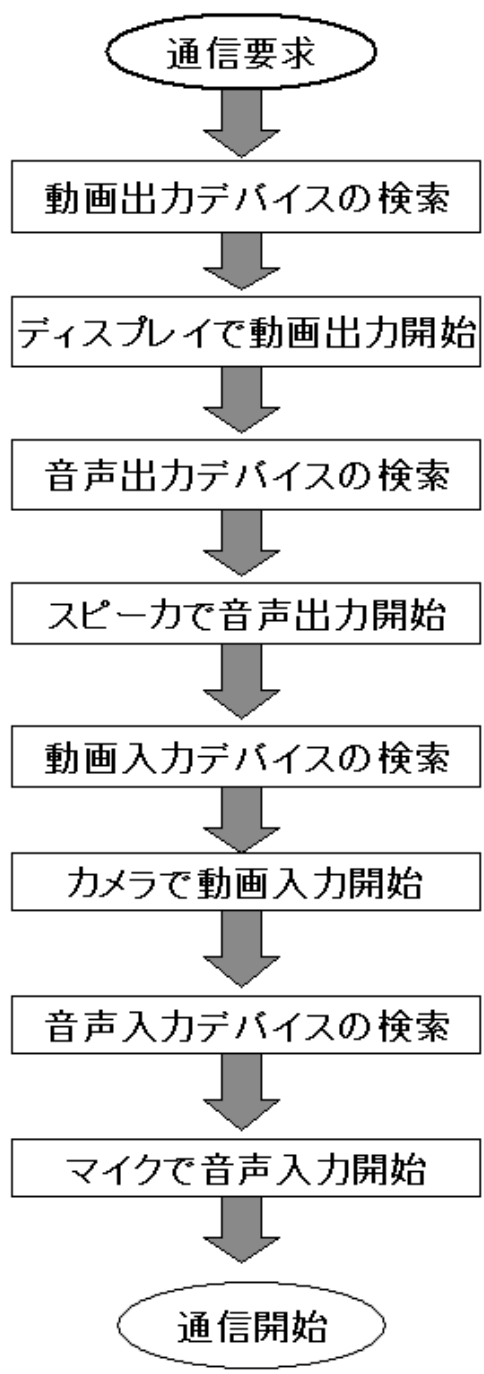


図 7.3: 提案 API を利用しなかった場合の手順例

第8章 今後の課題

8.1 家電の入出力情報の定義

今回、家電の入出力情報の定義は製品各自の機能を考慮せず統一的に定義を行った。しかし、製品の高性能化が進むにつれ、それぞれの製品は機能による差別化が進むと考えられる。そういった状況になった場合、それぞれの製品間における入出力情報のみを定義すると製品独自の機能を考慮出来ない可能性も出てくることが考えられる。そのような状況を避けるため、より細かい家電ごとの分類が必要になることが考えられる。

8.2 新しい入出力情報の出現

本研究においては、現在から将来的なものまで幅広くりようされるであろう UI を調査し、それぞれの入出力情報の定義をおこなったが、新たな情報の定義が出現することは十分考えられる。そのような場合、今までの入出力情報の統一的な定義に加える必要がでてくるが、そうなった場合どのように既存のアプリケーションとの協調性や API の利便性を崩さずに加えることが出来るか考察する必要がある。

8.3 情報の定義

本研究では、人と機器、機器と人という間での入出力情報の定義を行った。しかし、ホームネットワーク内では機器と機器の情報のやり取りを考慮する必要がでてくると考えられる。現に、リモコンからの情報出力はパターン 1 ~ 4 外での例外的な定義となった。今後の課題としては、機器と機器の間で定義する情報のやり取りをより深く考察し、統一的な定義方法を検討する必要があると考えられる。

8.4 情報の型

今回、情報の入出力における音声や動画、静止画の型を `char` と定義しているが `char` 型だけを渡せば音声や動画、静止画扱える保証はない。特に、音声という情報一つを考えてみても `MIDI` と `wav` では全く情報の性質が異なる。また、電話を利用する時に生じる音声

の出力とファイルとして保存してある音声の出力も全く異なる処理を行う必要があると考えられる。

8.5 非同期のユーザからの入力

デバイスから情報の入力を求める際、必ずしもユーザはすぐに情報を入力するとは限らない。そういった、非同期のユーザからの入力をどのように受け取るか考える必要がある。

第9章 まとめ

本研究では、ホームネットワーク内における UI(ユーザインタフェース) を統一的に扱うことが出来る API の定義を行った。

そのためにまず、既に利用されている UI から将来的に利用が見込まれる UI まで幅広く調査を行い、それぞれの UI が持つ入出力情報の定義した。更に、その入出力情報の統一的な定義を行った。入力情報では、情報を入力する際に利用する体の部分を基にした定義を行った。出力情報では、ユーザの状況と伝えたい情報に適した UI を動的に選択できるように、抽象度に応じて 4 つのパターンを作成した。

続いて、入出力情報の定義を利用し API を作成し、それら API を利用することによって作成できるアプリケーション例を示し、提案 API を利用することによって UI の動的な選択や多種類のデバイスに対して汎用性があることを確認した。また、そういった性質を利用することによって、ユーザの環境にあった UI の掲示や、環境内にある複数デバイスの機能を組み合わせてユーザに提供することで、複雑なサービスを代替的に実現することが可能となった。

謝辞

本研究を進めるに当たり、終始熱心なご指導をいただいた主指導教官である丹康雄助教授に深く感謝いたします。また、研究に関して多くのご意見をください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。さらに励ましあいながら学生生活を過ごしてきた丹研究室の皆様に深く感謝いたします。

参考文献

- [1] 出村哲也
センサを用いたレガシーデバイスホームネットワークの連携サービスシステムに関する研究, 北陸先端科学技術大学院大学情報システム学専攻修士論文, 2006
- [2] 増田耕一
ホームネットワークにおける異常状態のモデル化とその検知手法に関する研究, 北陸先端科学技術大学院大学情報システム学専攻修士論文, 2006
- [3] ECHONET CONSORTIUM
The eCHONET Specification Version 2.11, ECHONET CONSORTIUM, 2002
- [4] IEEE1394
online available at <http://www.1394ta.org/>
- [5] DLNA
online available at <http://www.dlna.org>
- [6] HAVi
online available at <http://havi.org>
- [7] UPnP
online available at <http://upnp.org/>
- [8] Logicoool Harmony
online available at <http://www.logitech.com/index.cfm/products/features/harmony>
- [9] 任天堂 Wii リモコン
online available at <http://www.nintendo.co.jp/wii/features>
- [10] SONY , BRAVIA
online available at <http://www.sony.jp/event/DLNA/>
- [11] Aromageur
online available at <http://www.aromageur.com/index.html>

- [12] HITACHI , Magicscape
online available at <http://hhil.hitachi.co.jp/products/magicscape.html>
- [13] HITACHI , WaterScape
online available at <http://hhil.hitachi.co.jp/products/waterscape.html>
- [14] Thomas H Massie , J Kenneth Salisbury
MIT AI-Lab, Probing Virtual Environments with the PHANToM,
- [15] 細谷英一, 北端美紀, 佐藤秀則, 原田育生, 小野澤晃
ミラーインタフェースを用いた双方向型インタラクティブコミュニケーションの実
現, 信学総大 05,A-16-20,p296,2005
- [16] NEC ロボット開発センター
online available at <http://www.incx.nec.co.jp/robot/robotcenter.html>
- [17] Cute Circuit , Hug Shirt
online available at <http://www.cutecircuit.com/>
- [18] 内閣府政策統括官 , 高齢社会白書
online available at <http://www8.cao.go.jp/kourei/whitepaper/index-w.html>

付 録 A 仕様 API 一覧

A.1 入力情報

inputdata

書式

```
void *inputdata(int maintype, int subtype, int datatype)
```

詳細

基本情報入力を受け付ける

inputSimpleData

書式

```
int inputSimpleData()
```

詳細

OK や NO などの単純な情報の入力を受け付ける

inputButtom

書式

```
int inputButtom()
```

詳細

ボタン入力を受け付ける

inputDataChar

書式

```
char *inputDataChar()
```

詳細

文字列の入力を受けつける

inputDataNum

書式

int inputDataNum()

詳細

数字列の入力を受け付ける

inputMenu

書式

char *inputMenu(char *menudata,int selectnum)

詳細

メニュー入力を受け付ける

A.2 出力情報

putPattern1

書式

int putPattern1(int maintype,int subtype)

詳細

パターン 1 で情報の出力を行う

putPattern2

書式

int putPattern2(int maintype,int datatype,void *data,int time)

詳細

パターン 2 で情報の出力を行う

putPattern3

書式

int putPattern3(int maintype,void *data,int time)

詳細

パターン 3 で情報の出力を行う

putPattern4

書式

```
int putPattern4(int maintype,int datatype,void *data,int time)
```

詳細

パターン 4 で情報の出力を行う

putButtom

書式

```
int putButtom(int dist)
```

詳細

リモコンを用いて情報の出力を行う

A.3 プライオリティ設定

setPri1

書式

```
int setPri1(int userid,int prinum)
```

詳細

ユーザ間のプライオリティを設定

setPri2

書式

```
int setPri2(int appid,int prinum)
```

詳細

アプリケーション間のプライオリティを設定

setPrilist

書式

```
int setPrilist(int userid,int appid,int senses,int datatype,int device,int deviceid,int prinum)
```

詳細

プライオリティ要素ごとのプライオリティの設定

setPritime

書式

```
int setPritime(int userid,int appid,int senses,int datatype,int device,int deviceid,int prinum,  
int starthour,int startmin,int endhour,int endmin)
```

詳細

時間ごとのプライオリティを設定

A.4 その他

setViewlist1

書式

```
int setViewlist1(int subtype,int subtypenum,char *subtypename)
```

詳細

意向リストのユーザ定義、”subtype”に項目を追加

setViewlist2

書式

```
int setViewlist2(int subtypenum,char *outdataname,int starthour,  
int startmin,int endhour,int endmin)
```

詳細

意向リストのユーザ定義、指定”subtype”の”outdata”に項目を追加

devStat

書式

```
int devStat(int devid,int turn)
```

詳細

デバイスの電源管理

setDevice

書式

```
int setDevice(char *devicename)
```

詳細

デバイス情報を追加