



2007年度 修士論文

携帯電話を利用するドラッグ&ドロップモデル
を基盤とした情報家電ネットワーク上の
データフロー制御

早稲田大学大学院 理工学研究科
情報・ネットワーク専攻

三井 悟

学籍番号	3606U104-2
提出日	2008年2月4日
指導教授	中島 達夫 教授

Control of Data Flows Based on Drag-and-Drop between Appliances Using a Camera Phone

Satoru MITSUI

Thesis submitted in partial fulfillment of
the requirements for the degree of
Master in Information and Computer Science

Student ID 3606U104-2

Submission Date February 4, 2008

Supervisor Professor Tatsuo Nakajima

Department of Computer Science
Graduate School of Science and Engineering
Waseda University



概要

本論文では、情報家電ネットワーク環境における情報家電間の通信を携帯電話から制御するインタラクションモデルを提案する。近年リモートアクセス可能な情報家電が増えてきており、今後は今まで以上に情報家電間の通信が増えてくると予想される。本論文で開発するシステムは、携帯電話上でグラフィックユーザインタフェースのドラッグ&ドロップモデルを拡張した家電選択手法を用いており、ユーザが情報家電間のデータのやり取りや方向を理解できるよう工夫がなされている。これを利用することで、多機能な家電が多く存在する状況下でもユーザが家電操作を迷わないシステムを目指した。評価実験により本論文で提案するインタラクションモデルが、ユーザに学習性の高さを使いやすさをもたらすことが示された。

Abstract

In this thesis, I propose an interaction model to be able to control the connection between information appliances in the next-generation home network environment by using a camera phone. More information appliances which have the remote accessibility there are, more connections between information appliances are expected. This system adopts the extended direct manipulation technique based on drag-and-drop model in graphic user interface and allows users realize data flows and those directions between information appliances. By using this system, users are not confused about appliances' manipulations even in the environment which has many multifunctional information appliances. The evaluation experiment showed that the proposed interaction model provide the usability and the ease of learning to users.

目次

概要	iii
Abstract	iv
図一覧	iv
表一覧	v
第1章 序論	1
1.1 背景	1
1.2 課題	2
1.3 目的	5
1.4 本論文の構成	5
第2章 関連研究および家電連携コントローラの要件	6
2.1 複数デバイスの接続を行う研究	6
2.1.1 Pick-and-drop	6
2.1.2 mediaBlocks	6
2.1.3 InfoStick	7
2.1.4 Touch-and-connect	7
2.1.5 tranStick	7
2.2 データフローに関する研究	8
2.2.1 DataTiles	8
2.3 GUI上のドラッグ&ドロップを用いることの弊害	8
2.4 家電連携コントローラの要件	9
第3章 家電連携の特性に基づくインタラクションの検討	10
3.1 家電連携のシナリオ	10
3.1.1 家電連携の対象となる機器	10

3.1.2	具体的なシナリオ	11
3.2	ドラッグ&ドロップベースのインタラクションの検討	12
3.2.1	ドラッグ&ドロップの概要	12
3.2.2	ドラッグ&ドロップのデータフローモデル	13
3.2.3	家電連携の特徴	15
第4章	システムの設計と実装	19
4.1	設計	19
4.1.1	アプリケーションシナリオ	19
4.1.2	携帯電話を選択した理由	20
4.1.3	インタラクションモデル	21
4.1.4	連携しない家電選択時の挙動	27
4.2	実装	28
4.2.1	利用技術	28
4.2.2	アーキテクチャ	30
4.2.3	サービスアプリ UI クラスの抽象化	32
第5章	ユーザスタディ	35
5.1	実験概要	35
5.2	実験環境・被験者	35
5.3	実験手順	36
5.4	評価方法	36
5.5	実験結果と考察	37
5.5.1	日常の作業の頻度について	37
5.5.2	実験の内容について	38
第6章	結論と将来課題	40
	謝辞	41
	参考文献	42

図一覧

1-1	情報家電ネットワークのイメージ [13]	2
1-2	想定されるデジタル情報家電の利用例 [11]	3
3-1	情報家電の分類	11
3-2	GUI上でのドラッグ&ドロップの例	12
3-3	ドラッグ&ドロップできない場合	13
3-4	右クリックドラッグ&ドロップ	14
3-5	ドラッグ&ドロップのデータフロー	14
3-6	コンテンツ連携	15
3-7	コンテンツ連携のデータフローモデル	16
3-8	コンフィグレーション連携	17
3-9	コンフィグ連携	18
4-1	インタラクションモデル	22
4-2	2種類のモードの切り替え	23
4-3	コンテンツ連携モードの表示	23
4-4	コンフィグレーション連携モードの表示	23
4-5	Sony Type X ビデオステーション	24
4-6	X ビデオステーションのアイコン	24
4-7	X ビデオステーションの機能を表示	24
4-8	再生可能な動画のリスト	25
4-9	移動元デバイスを選択した状態	25
4-10	ディスプレイの機能	25
4-11	動画再生中のディスプレイ	25
4-12	動画再生中のコントロールUI	25
4-13	時計デバイスのアイコン	26
4-14	時計デバイスの機能（移動元選択時）	26

4-15 時計デバイスの時刻データダイアログ	26
4-16 時計デバイスの機能（移動先選択時）	26
4-17 時計の同期	26
4-18 時計デバイス（右は時刻がずれている）	27
4-19 連携できない場合	27
4-20 Nokia N93	29
4-21 VisualCodes	29
4-22 アーキテクチャ	31
5-1 実験環境その1	36
5-2 実験環境その2	36

表一覧

4-1 VisualCode ID と UDN の関連	32
5-1 PC 上でドラッグ&ドロップをどのように利用するか	37
5-2 携帯電話によるタグの読み取り頻度	38

第1章

序論

本章では、本研究に至る背景と目的、本論文の構成について述べる。

1.1 背景

近年、大容量ハードディスクレコーダーや自動温度調節可能なエアコンなど、高性能で様々な機能を持ったデジタル製品が増えてきている。これらは一般的に情報家電と呼ばれている。経済産業省基本戦略報告書「e-Life イニシアティブ」[10]によると、『「情報家電」とは、携帯電話、携帯情報端末（PDA）、テレビ、自動車等生活の様々なシーンにおいて活用される情報通信機器及び家庭電化製品等であって、それらがネットワークや相互に接続されたものを広く指す。』とある。最近では、パソコンやサーバなどのPC機器のみならず、デジタルテレビやDVDレコーダーなどのAV家電や、冷蔵庫やエアコンなどの白物家電も情報家電化してきており、それらの持つネットワーク機能により、多くの情報家電がPCなどからリモートアクセス、リモート操作が可能となっている。今後もその流れは継続する見込みで[2]、将来的には以下の図1-1のように、多種多様な情報家電が自宅に存在して相互接続可能な情報家電ネットワーク[13]（ホームネットワークとも呼ばれる）が現実のものになるだろう。

ところで、相互接続と単純にいても接続する家電によって様々なデータをやり取りすることが考えられる。図1-2は、デジタル情報家電のネットワーク化に関する調査研究会による情報家電ネットワークの利用例である。例えば図左上の地域安全システムの例では、監視カメラやビデオカメラによる静止画や動画、ドアや天井に取り付けられたセンサなどの情報を様々な場所に送り、図中央の家電コントロールシステムの例では、時計の時刻の情報を照明に送ったり、温度情報をエアコンに伝えたりしている。これら以外にも、ソニーではDLNA*に対応した製品を数多く

*DLNAはホームネットワークを利用してPCや情報家電を相互接続するための技術仕様を策定す

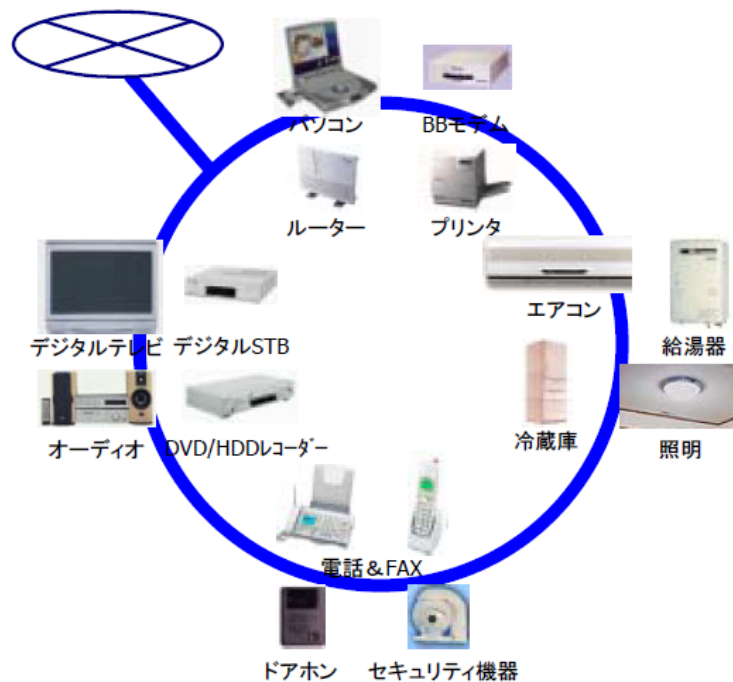


図 1-1: 情報家電ネットワークのイメージ [13]

扱っており、ハードディスクレコーダーに撮り貯めたテレビ番組を LAN を通して離れたテレビにも送ることができる[†]。このように様々な組織、企業で情報家電ネットワーク化の動きは促進されているのである。

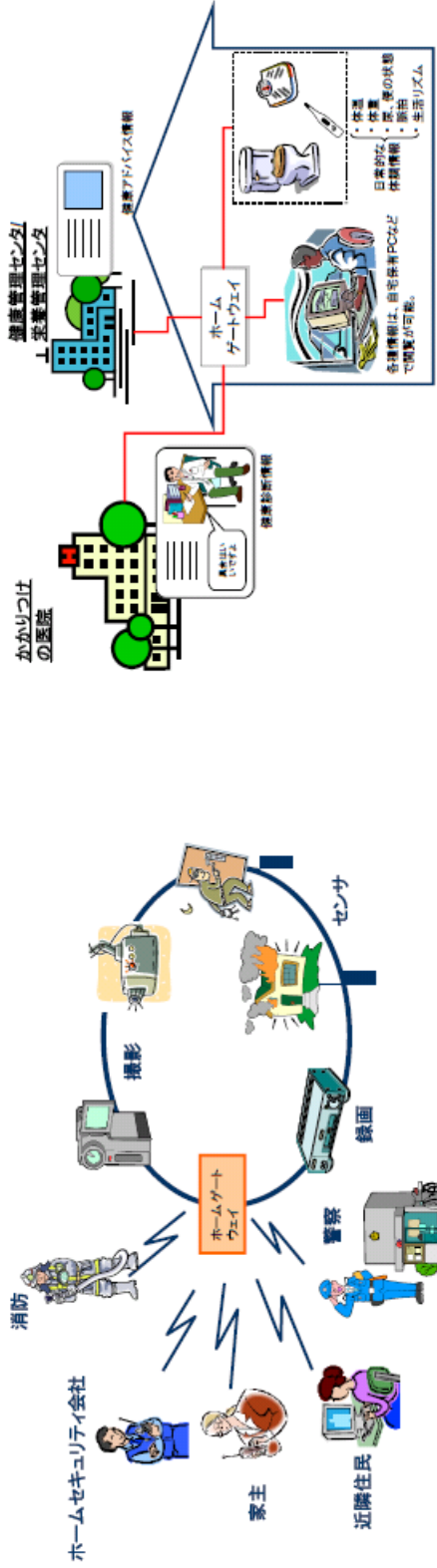
1.2 課題

こうしたシナリオを広く普及するためには、乗り越えなければならないいくつかの問題が存在する。ユビキタス技術 ホームネットワークと情報家電 [14] によると情報家電ネットワークは以下の要件を満たさなければならない。

1. 簡単に接続できて利用できること
2. 相互に接続できること
3. 安心して利用できること

る業界団体である。

[†]Sony のルームリンクという製品で可能である。



遠隔医療診断システム

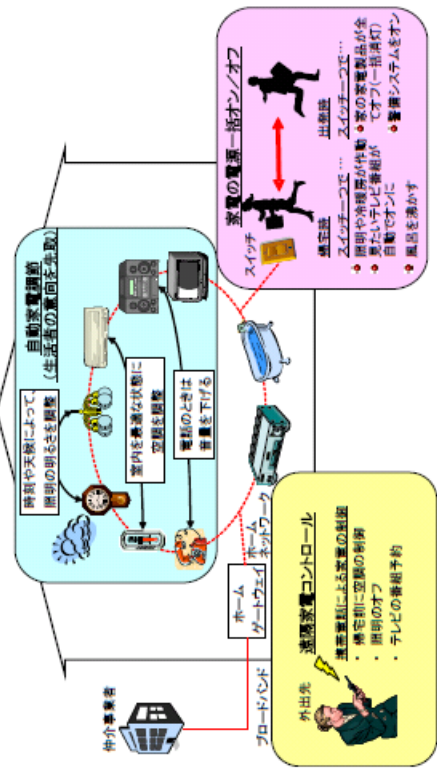


図 1-2: 想定されるデジタル情報家電の利用例 [11]

2や3は徐々に研究が進められている。相互接続規格として白物家電にはECHONET[†]規格が策定され、AV家電ではIEEE1394やDLNAなどが設立されるなどの動きがあるが、1に関してはまだ未発展な状態である。PCからネットワーク上のIP名かコンピュータ名を探し出し操作する方法もあるが、初心者ユーザには非常に使いづらいため、万人に操作しやすい新しいインタフェースの開発が進められている。

その流れの一つとして、複数の情報家電を接続する方法の一つとしてドラッグ&ドロップの概念を採用する傾向がある。ドラッグ&ドロップはグラフィカルユーザインタフェース（以下GUI）上の概念であり、アイコンやファイルをマウスのボタンを押しながら移動させることでそれらを運ぶインタラクションである。これは2.4で述べるShneidermanが提唱した直接操作手法（direct manipulation technique）[8]を満たしており、初心者のユーザにも利用しやすい操作手法である。暦本が開発したPick-and-drop[5]が実世界の情報家電の相互連携にこの概念を取り入れた。直接操作手法を実世界で応用するのは非常に効果的である。しかしながら、GUI上のドラッグ&ドロップをそのまま実世界志向のインタラクションとして取り入れるのは難しい。なぜなら一つには、ファイルや情報家電などのオブジェクトを選択し運んでいる状態を表したり、どの情報家電が操作できて、どのオブジェクトがどの情報家電と対応するのか、などといったフィードバック情報を視覚化するのが難しいことが挙げられる。その結果、ユーザは情報家電間で流れるファイルや何らかのデータの流れ（以後データフロー）を認識できなくなる恐れがある。表示するディスプレイを吟味した上で、さらにオブジェクトを直接選択できる方法も考えなければならない。二つ目には、情報家電同士の通信と連携がとても複雑になってきていることである。例えば、ある情報家電が同時に複数の情報家電につながる可能性もある。先ほどの図1-2の地域安全システムの例など、複数の情報が関連していることでセキュリティを高めている。最後に、実世界で情報家電をドラッグ&ドロップと言っても、扱うデータはファイルだけではなく様々なものが存在する。情報家電を相互に接続するという事は、データの流れが発生するという事である。データを選択して送るだけではなく、データがどのタイミングで、どこからどこに送られるのか、実際に情報家電同士が接続されるのはいつなのかといった、接続された情報家電のデータフローの方向や条件を指定する必要もあるだろう。これまでの製品や研究では以上の状況を十分に解決できていない。

[†]ECHONETは家庭内の電灯線や無線を利用したネットワークの規格で、家電機器の制御を行うためのものである。

1.3 目的

そこで、本研究では実世界上で情報家電ネットワークのデータフローを制御するための新しい直接操作手法を提案する。これは既存のドラッグ&ドロップモデルを元に開発し、ビジュアルフィードバックを得ながら、情報家電同士をコントロールし、連携した動作をさせたり設定をより柔軟に行うことを可能とするものである。この手法をカメラ付き携帯電話で実現することで、ユーザは二つの情報家電をそれぞれに貼り付けられたビジュアルタグをカメラを通して見ることで直感的に選択することが可能となる。これを利用するユーザが制御する情報家電のデータフローを自然と認識できることを期待できるインタフェースを開発する。

1.4 本論文の構成

本論文の構成は以下の通りである。

第2章 関連研究および家電連携コントローラの要件

家電やデバイス进行操作、連携させる関連研究や、データフローモデルについて参考となる研究について述べる。また、ドラッグ&ドロップモデルをそのまま模倣する弊害について述べ、家電連携を行うコントローラに必要な要件について述べる。

第3章 家電連携の特性に基づくインタラクションの検討

情報家電連携における主なシナリオについて記述する。加えて情報家電の特性を述べ、それに基づいた実世界志向のドラッグ&ドロップモデルを定義する。

第4章 システムの設計と実装

本システムのインタラクションモデルや、実装に関して記述する。

第5章 ユーザスタディ

ユーザスタディの概要や実験環境、評価方法について説明する。結果とそれについて考察も加える。

第6章 結論と将来課題

本研究の結論と将来課題について述べる。

第2章

関連研究および家電連携コントローラ の要件

本章では，家電やデバイス进行操作，連携させる関連研究や，データフローモデルについて参考となる研究について述べる．本章の最後に，実世界上の情報家電の接続，制御（以降，家電連携）を行うためのコントローラとしての要件を挙げる．

2.1 複数デバイスの接続を行う研究

2.1.1 Pick-and-drop

Pick-and-drop[5] は複数コンピュータ環境での直接操作技法を実現した代表的な研究で，ソニーコンピュータサイエンス研究所の暦本が開発した．タッチペン型のデバイスを利用して，携帯型コンピュータやディスプレイなどに存在するデータやオブジェクトをつまみ上げるようにして別のデバイスにデータを移動することが出来る．ドラッグ&ドロップの概念を拡張しながらも，ペンでつまむというメタファを利用することで実世界志向のドラッグ&ドロップを実現している．データの移動だけではなく，携帯ディスプレイでペンの属性を選択し，パブリックディスプレイで絵を描いたり，紙媒体のものから ID 認識をすることでデジタルデータを取り出したりすることも応用例として挙げられている．しかし，統一されたデータフローモデルを持たないので，ユーザがどのようなデータを移動させているのか認識するのに支障をきたす可能性がある．

2.1.2 mediaBlocks

mediaBlocks[9] は MIT の Ullmer らが開発したタンジブルユーザインタフェースである．PC やプロジェクタなどの操作対象のデバイスに設置されたスロットに，電

子タグをつけた木のブロックを挿入することで、データを別のデバイスに持ち運ぶことが出来るというものである。例えば、ディスプレイで描いた絵のデータをプリンタのロットに入れるとその絵が印刷されるという仕組みである。ブロック自体にデータは入らないが、IDによりネットワーク上でデータが管理され、あたかもブロックにデータが入っているように感じられる。しかし、木のブロックはディスプレイを持たないため、その内部にどのようなデータが存在し、それをデバイスに挿入すると何が起こるのかを適切にユーザにフィードバックできておらず、データフローの認識が困難である。

2.1.3 InfoStick

InfoStick[4]は、慶応義塾大学 SFC 研究所の神武らが開発した家電間の連携を行うためのデバイスである。CCD カメラやディスプレイを持ったハンドヘルドデバイスを利用して、コンピュータ間でファイルの移動や、ディスプレイへの表示、プリンタへの出力を行うことが出来る。InfoStick は家電が持つデータを保持していて、それを別のデバイスに渡し、機能を選択することで連携を行う。例えば、紙にある ID を認識しデータを取得する。次にプリンタを選択し、メニューから印刷を選択すると、先ほどのデータが印刷される。これは 3.2.1 で述べる右クリックのドラッグ&ドロップとほぼ同じインタラクションである。InfoStick はファイルなどを操作するインタフェースを備えているが、継続的に接続が行われるタイプの家電の操作はできない。

2.1.4 Touch-and-connect

Touch-and-connect[3]は、複数デバイス間の通信の確立を、直接操作で指定して行うことが出来るフレームワークである。デバイスはプラグボタンと、ソケットボタンの二つを持ち、接続元と接続先をこれらを一致させることにより決定する。また、複数人環境の場合でも、ボタンを押している間排他制御を行うことにより、誤接続を防ぐことができる。接続した機器は継続的に入力デバイスからの命令を受け付ける。これらの概念は本論文で提案する“コンフィグレーション連携”の参考としている。

2.1.5 tranStick

tranStick[12]は、ソニーコンピュータサイエンス研究所の綾塚らが開発した、メモリースティック*を利用した複数デバイス間の接続方法である。メモリースティック

*メモリースティックはソニーが中心となって開発した小型メモリーカード規格である。

クを別々のPCに繋ぐと、それらが仮想的に接続され、同じデータを共有できるようになる。メモリースティックはファイルやデータを直接コントロールするのではなく、接続環境を直接操作で確立している。これらの概念は“コンフィグレーション連携”の参考となっている。

2.2 データフローに関する研究

2.2.1 DataTiles

DataTiles[6]は、暦本，Ullmerらが開発した、様々な機能を持つ透明なタイルを組み合わせることで連携を行うタンジブルユーザインタフェースである。アプリケーションタイル、コンテナタイル、ポータルタイル、パラメタタイルなどの数種類の機能を持つタイルを組み合わせ、隣接したタイルの操作を行うことができる。隣接するタイルやタイルの種類によって流れるデータが異なる様子は、本論文のデータフローモデルの構築の参考となっている。

2.3 GUI上のドラッグ&ドロップを用いることの弊害

実世界のデバイスに対して、ドラッグ&ドロップのインタラクションモデルをそのまま持ち込もうとする傾向は、いくつかの関連研究で述べたとおりである。しかし、この傾向はユーザビリティの観点からするとあまり好ましくない。その理由の一つは、家電をドラッグして別の家電にドロップしても、どのようなデータが転送されるのか、何が起こるのかユーザにはわからず、推測を元にしか判断できないことである。それは、GUIのドラッグ&ドロップと違い、ドラッグやドロップされる対象が、様々な機能を持つ家電だということに理由がある。家電の持つ機能やデータなどを適切に表示できなければ、ユーザはドラッグ&ドロップする対象がわからず負担を強いられるだけだろう。

もう一つの問題は、ドラッグやドロップを行う際に、フィードバックを表示する場所が実空間上には存在しないという状況の違いである。ユーザは簡単にデータフローを認識できる必要がある、つまりどちらがデータの移動元のデバイスで、どちらがデータの移動先のデバイスなのか、どのようなデータが転送されるのかといった情報を実空間上でも示さなければならない。これらを適切に表示できる、何らかのフィードバックを与える機器が必要である。

2.4 家電連携コントローラの実要件

以上のことから，家電連携を行うコントローラに必要な要件が3つ導き出される．以下にそれらを述べる．

ビジュアルフィードバック

情報家電の数やその機能は継続的に増大している．これらの情報家電から，他の情報家電との接続可能性を含めたビジュアルフィードバックを得ることが重要である．しかし，ディスプレイを持たない情報家電も数多く製造されている．よって，家電連携コントローラがこれらの情報家電のディスプレイの代わりにフィードバックを提供する必要がある．

直接操作インタフェース

Shneiderman により提唱された Direct manipulation[8] の概念は，実世界志向のインタラクションを構築する際に非常に有用である．直接操作インタフェースの特徴を簡単に述べると，以下の四点になる．

1. 操作対象と，操作自体を視覚的に表示すること
2. 迅速で，逐次的かつ可逆的な操作を持つこと
3. 操作結果の即時的な視覚化（フィードバック）
4. コマンド入力に代わるポインティング操作と選択操作

こうした直接操作の原則を参考にすると，家電連携コントローラはポインティングなどの手法により情報家電を選択でき，その結果をフィードバックすることが望ましい．

シンプルなモード選択

情報家電は多くの動作モードや遷移を持ち，しかも，家電連携の際はそれを移動元デバイスと，移動先デバイスで複数選択しなければならないため，ユーザを悩ませる可能性がある．ほとんどの場合，ユーザはいくつかの機能しか使わないため，ユーザが使いやすく，設定可能なようにシンプルなモードと遷移を提供することが重要である．

第3章

家電連携の特性に基づくインタラク ションの検討

この章では、本論文で述べている家電連携の具体的なシナリオについて幾つか例を挙げる。また、GUI上のドラッグ&ドロップを元に、家電連携にふさわしいインタラクションについて検討を行う。

3.1 家電連携のシナリオ

情報家電にも様々な種類があり、豊富な機能を持つ機器が増加している。そのため、本研究でどのようなものを情報家電と定義し、実世界志向のドラッグ&ドロップインタラクションで対応するのか以下で明らかにしていく。

3.1.1 家電連携の対象となる機器

情報家電の定義としてネットワークに接続できることを 1.1 にて述べたが、IP ネットワークに接続可能なものと、非 IP ネットワークに接続可能なものが存在する。更に家電の種類により、AV 家電、設備・白物家電、情報通信家電の 3 種類に大きく分けることが出来る。その分類を元に主な家電をリストアップした(図 3-1)。

近年急速にネットワーク化が進んでいるのは AV 家電で、DLNA や UPnP*の普及が大きく貢献している。これらの規格に互換性のあるハードディスクレコーダーやディスプレイ、デジタルカメラ、オーディオなどの家電が今も生産されている。情報通信家電は、PC や携帯電話などを含み、これらの中には情報家電を操作することが出来るものもある。また、設備系家電・白物家電は冷蔵庫、エアコン、時計や蛍光灯、セキュリティやヘルスケアを目的とした各種センサなどが存在する。設備・白物家電は ECHONET という規格が最も一般的であるが、本研究では全ての家電を同

*UPnP は PC や家電などの機器を自動的に発見・操作できる技術仕様である。 <http://www.upnp.org/>

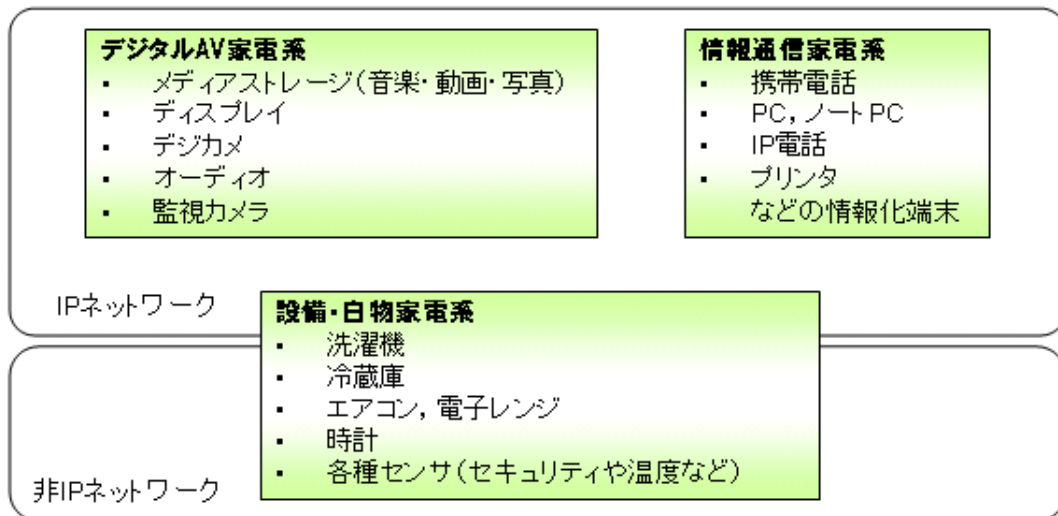


図 3-1: 情報家電の分類

ーネットワーク上で扱えるよう設備系・白物家電を利用したシナリオでも UPnP を用いることにしている。

3.1.2 具体的なシナリオ

これらの家電が相互に連携して動作するシナリオは以下のようなものがある。

- ハードディスクレコーダーから映像を選択し、ディスプレイで再生する
- 音楽データをオーディオに転送し再生する
- 監視カメラ映像を元に携帯電話にメールを送る
- 温度センサでエアコン温度を調節する
- 時刻によって部屋の明るさを調節する
- 家の時計のアラームを一度に修正する
- デジカメの画像をディスプレイに映しながら保存する

これらは一例であり、様々な家電が相互に連携する可能性を秘めている。

3.2 ドラッグ&ドロップベースのインタラクションの検討

ドラッグ&ドロップを基にしたインタラクションを構築するため、まずはGUI上で行うドラッグ&ドロップモデルを採りあげ、それを家電連携においてどのように応用できるか考察する。

3.2.1 ドラッグ&ドロップの概要

ドラッグ&ドロップとは、アプリケーション間やアプリケーション内でマウスを利用してデータやファイルを移動させるインタラクションである。一連の動作は以下の通りになる。

1. 移動元の場所に行き、データを選択する（クリック）
2. ボタンを押して掴み、対象場所まで移動する（ドラッグ）
3. 移動先の場所、もしくはアプリケーションを確定し、ボタンを離す（ドロップ）
4. （右クリックでドラッグした場合のみ）オペレーションを選択する

以上のような順番でインタラクションを行う。移動することができるデータやオブジェクトは、ファイルやディレクトリ、テキストやセルなどのデータ構造、アプリケーションやウィンドウ内のアイコンやウィジェットなどが挙げられる（図3-2）。

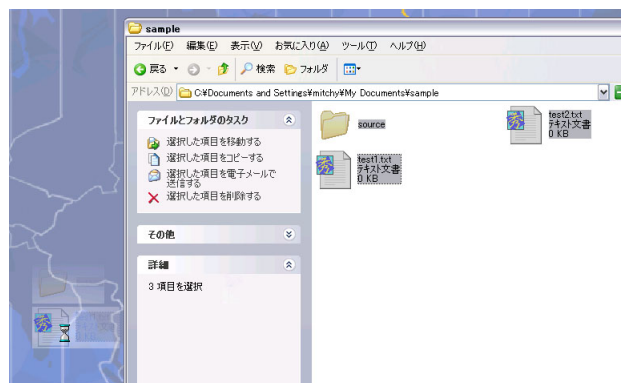


図 3-2: GUI 上でのドラッグ&ドロップの例

ドラッグ&ドロップは既存の GUI 環境では非常に多く利用されるインタラクションであり、初心者やマウスを利用するユーザにとって非常に使いやすいと言える。

3.2.2 ドラッグ&ドロップのデータフローモデル

データフローを考える際に重要なことは、データフローの発生するタイミングや移動元・移動先、データの種類である。ドラッグ&ドロップを行う対象は通常ほとんどファイルであり、フォルダ移動やショートカット作成などに使われている（節5.5）ため、扱うデータはファイル、移動元・移動先はフォルダやデスクトップということになる。また、タイミングは常にユーザが行った直後に即時的に起こる。選択している状況や移動している軌跡などが全てデスクトップに表示され、ユーザは常に状況が分かるため、ドラッグ&ドロップは理解しやすいユーザインタフェースとなっている。

ドラッグ&ドロップの優れた点は、移動先がデータをサポートしていない場合にドロップできない旨を表示する点である（図3-3）。これにより、ユーザはドラッグ&ドロップが不可能なデータの流れを学習し、適切な移動先やアプリケーションを選択できるようになる。

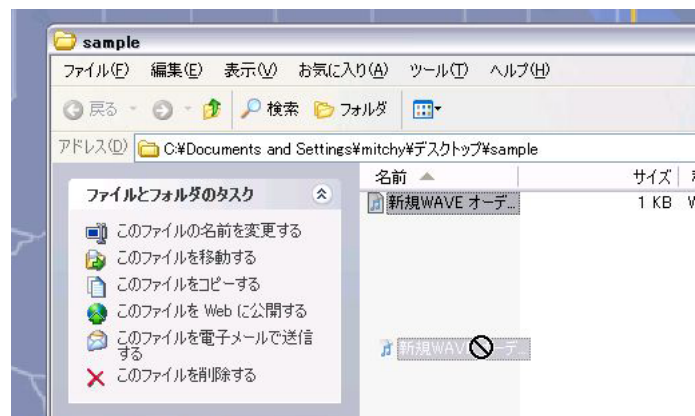


図3-3: ドラッグ&ドロップできない場合

また、右クリックによるドラッグ&ドロップも存在する。これはドロップ後にメニューが現れ、機能を選択できるようになっている。例えば、ファイルを選択し、デスクトップなどに右クリックでドラッグを行い、ボタンを離すと、ファイルの移動、コピー、ショートカットの作成、などが選択できる（図3-4）。

これらをまとめると、ドラッグ&ドロップのデータフローモデルは以下の図のようになる（図3-5）。

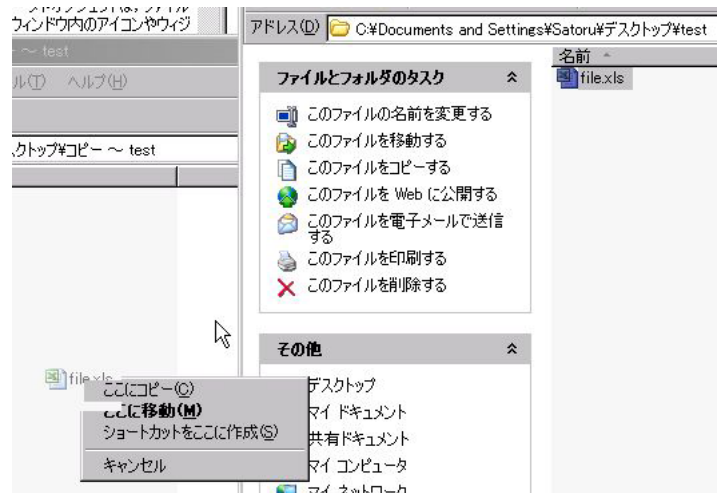


図 3-4: 右クリックドラッグ&ドロップ

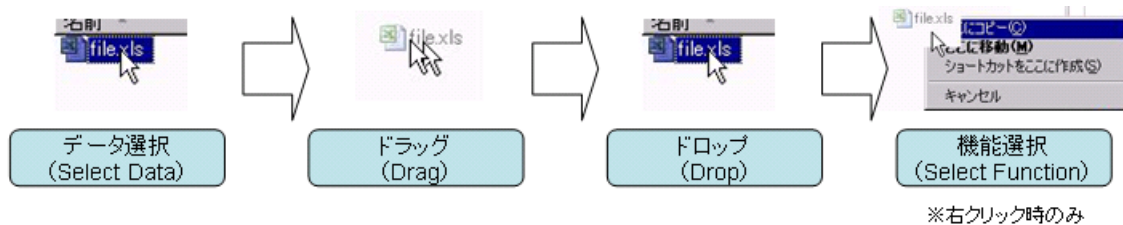


図 3-5: ドラッグ&ドロップのデータフロー

3.2.3 家電連携の特徴

節 3.1.2 で挙げられたシナリオには2種類に区分できる。それを本論文では，“コンテンツ連携”，“コンフィグレーション連携”と定義している。ここではその特徴について述べる。

コンテンツ連携

情報家電でも GUI と同じようにファイルなどのオブジェクトを扱うことは多い。例として，ハードディスクレコーダーに保存されたテレビ番組のファイルを，離れた別のディスプレイにストリーミングなどで転送し再生することが挙げられる。このようにユーザがある情報家電に保存されているファイルオブジェクトを別の情報家電上に移動させたり，コピーさせたり，その家電上で実行する動作を行うことを“コンテンツ連携”と定義する。コンテンツ連携はユーザの操作が即座に画面に反映され，データの転送後は処理が終わってしまう一時的なものを対象とする。

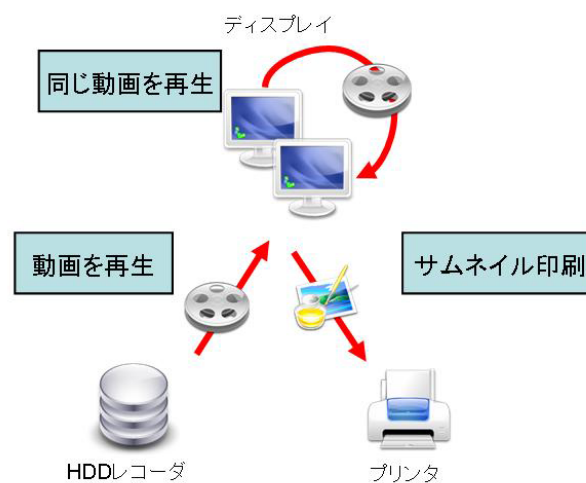


図 3-6: コンテンツ連携

コンテンツ連携の例

- ハードディスクレコーダーから映像を選択し，ディスプレイで再生する
- 音楽データをオーディオに転送し再生する
- ディスプレイのスクリーンショットをプリンタで印刷する

- デジカメの画像をディスプレイに映す
- 移動元の家電から，移動先の家電にデータをコピーする

コンテンツ連携のデータフローモデル

コンテンツ連携のデータフローはGUIのドラッグドロップに比較的似ている．データとなる対象は動画・写真などを含むファイル全般であり，移動元の家電から移動先の家電にファイルを渡すという概念となる．ユーザが操作した直後に連携が起こるのはドラッグ&ドロップと同じである．

しかし，いかにそれらのファイルを選択し，移動先に渡すかという手順はドラッグ&ドロップのデータフローモデルとは異なる（図3-7）．

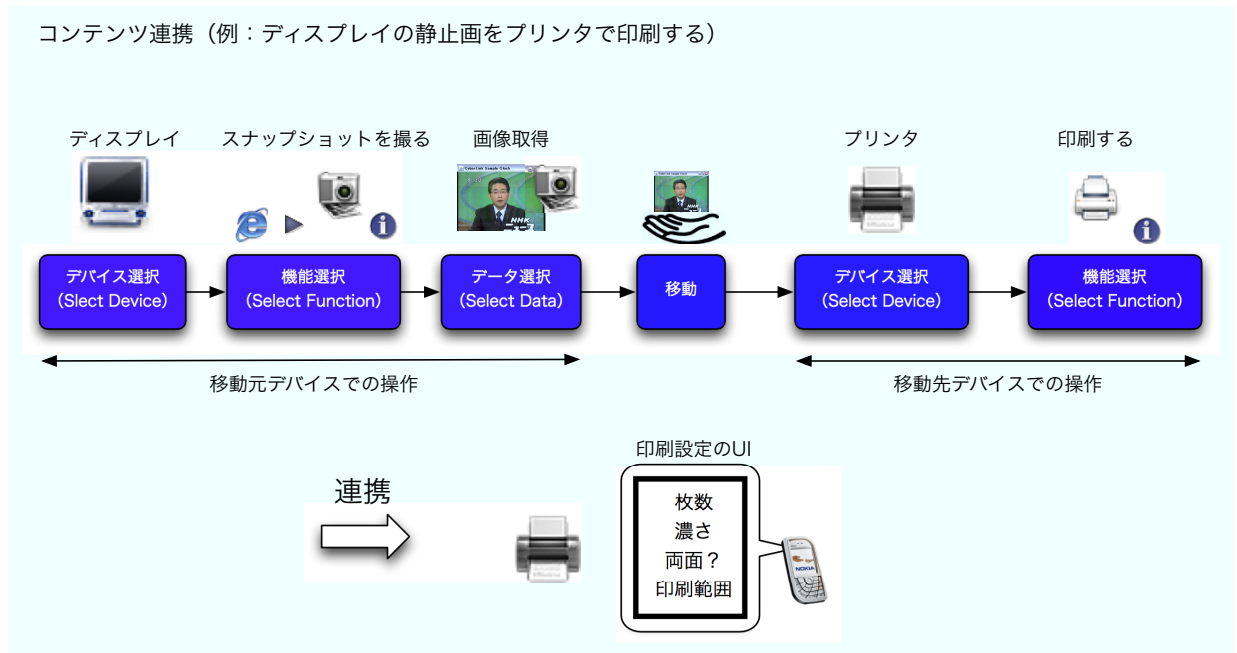


図 3-7: コンテンツ連携のデータフローモデル

実世界における情報家電連携の場合，選択するフェーズがGUIのドラッグとドロップ以外にも多くなる．データの選択のみならず，デバイスの選択，機能の選択を行う必要があるためである．機能選択の後にデータ選択を行うのは，コントローラから扱えるデータは家電の持つ機能によって異なると考えられるからである．例えば，図の例で仮に“スナップショットを撮る”機能を選択せず，左のアイコンの“Web ページからデータを選ぶ”を選択していれば，次に選択するデータは画像ではなくなるかもしれない．そのため，このようなデータフローモデルとなっている．連携後に表示される画面は，その連携をコントロールするためのユーザインタフェースであ

る．例えば，動画を再生する連携を行った場合，早送りや巻き戻しなどの機能を備えたユーザインタフェースが画面に表示されるようになっている．

コンフィグレーション連携

家電連携では，コンテンツ連携のような即時性を持ったものばかりでなく，特定の時間にイベントが発生したり，また継続的に接続が行われているものもある．例えば，あるテレビ番組をハードディスクレコーダーに録画したい場合は指定した時間に録画が開始されるし，図 1-2 の地域安全システムは継続的にデータをやり取りしているだろう．家電連携を行うコントローラはこれらの特徴を持つ家電連携もサポートする必要がある．これらの特徴を持つ家電連携を「コンフィグレーション連携」と定義する．コンフィグレーション連携は継続的・自動的にデータを転送する接続に対して行い，連携時にはユーザは操作を行わないという特徴を持っている．コントローラはコンフィグレーション連携の確立や解除などを行えるようにする．つまり，情報家電間で流れるデータのパスを作成するのがコンフィグレーション連携である．

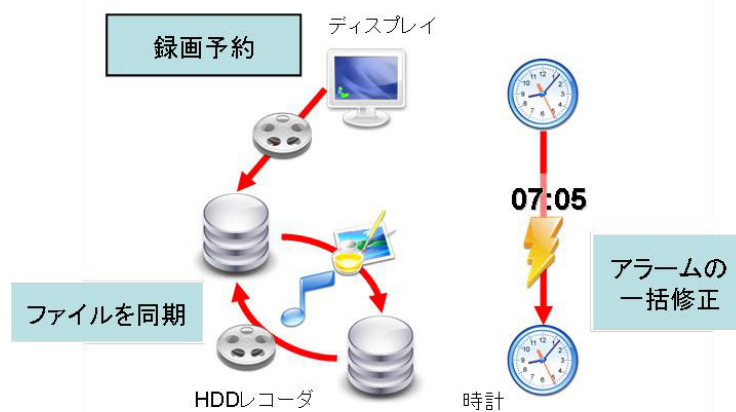


図 3-8: コンフィグレーション連携

コンフィグレーション連携の例

- ディスプレイで再生する動画をハードディスクレコーダーで録画予約する
- あるメディアストレージから別のストレージにファイルを同期する
- 監視カメラ映像を元に携帯電話にメールを送る

- 温度センサでエアコン温度を調節する
- 時刻によって部屋の明るさを調節する
- 家の時計のアラームを修正する

コンフィグレーション連携のデータフローモデル

コンフィグレーション連携のデータフローモデルはGUIのドラッグ&ドロップのデータフローモデルと大きく異なる。それは、連携が継続的・自動的に発生するため、即座に連携が開始するわけではないという点に起因する。また、移動するデータの対象として、ファイルやセンサなどのデータ、時刻などの何らかの数値などがある。移動元の家電から、移動先の家電にそれらのデータを渡し続けるというのがコンフィグレーション連携のデータフローモデルである。

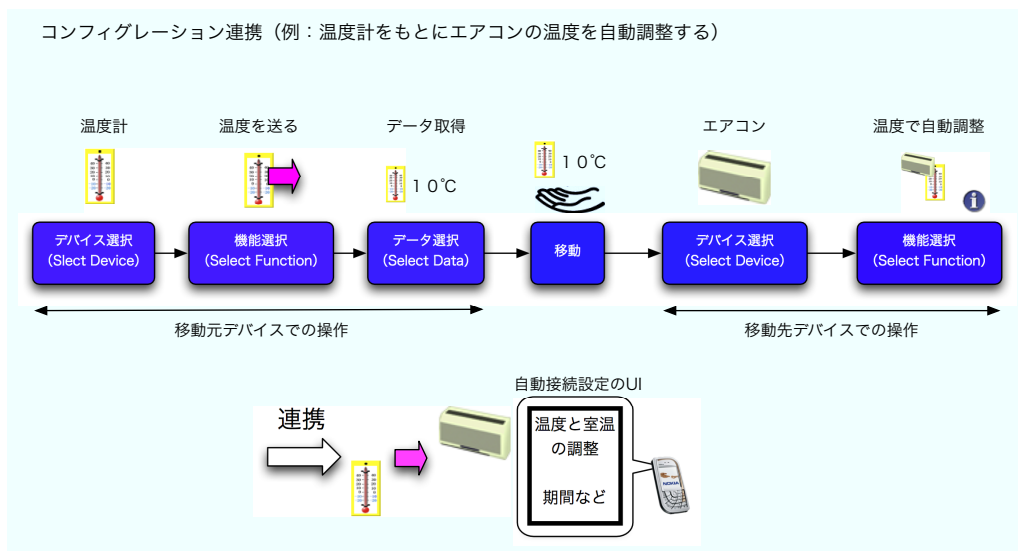


図 3-9: コンフィグ連携

連携にいたるまでのフェーズ自体は、コンテンツ連携のものとは大差はない。同じように移動元デバイスを選択し、機能選択、データ選択を行い、移動先のデバイスと機能を選択するように考案し、ユーザが異なる2種類の連携を行う際に混乱をすることがないようにした。コンフィグレーション連携の連携確立後のユーザインタフェースは、連携後にいつ、どのように連携を行うか設定をする画面を表示し、そこで設定することで自動的に家電間でデータの移動が行われる。

第4章

システムの設計と実装

本章では、2.4章の検討事項を元に構築した家電連携コントローラ“DataFlowController”のシステム設計、実装に関して記述する。

4.1 設計

本節では本システムの設計にあたって、基本となるアプリケーションシナリオの作成、コントローラとしてカメラ付き携帯電話を採用した理由、本システムのインタラクションモデルについて述べる。

4.1.1 アプリケーションシナリオ

DataFlowControllerで操作できる家電連携の典型的なシナリオを考案した。以下のシナリオをコンテンツ連携、コンフィグレーション連携の代表的な例と定義して、これらに対応できるように各家電のユーザインタフェースやUPnP通信機能などをあらかじめ組み込んだ実装を行う。

コンテンツ連携シナリオ

1. ハードディスクレコーダーにある動画を、ディスプレイで再生する
2. ディスプレイで再生している動画を、別のディスプレイでも再生する
3. ディスプレイに映った動画をキャプチャし、その静止画をプリンタで印刷する

コンフィグレーション連携シナリオ

1. ある時計のデータを定期的に送り、別の時計の時刻と同期させて修正する

2. ハードディスクレコーダーの複数ファイルを，ディスプレイに渡してループ再生を行う
3. ハードディスクレコーダーとPCなどのメディアストレージを繋いで，動画の同期を行う

節 1.1 や節 1.2 で述べた相互接続のための規格は，依然として完全に統一されていない状況であり，シナリオの対象となる製品化された機器を全て取り揃えてシステムを実装することは難しい．そのため本研究では，情報家電の代わりとして同等の機能を提供するアプリケーションをPC上で実装することにした．上記シナリオのディスプレイ，時計，PCメディアストレージはPC上で開発を行う．

コンテンツ連携に関してはマルチメディアコンテンツである動画や音声ファイルを持つ情報家電を制御するシナリオが自然と多くなる．この理由は，GUI上のドラッグ&ドロップと比較すると，アイコンをドラッグしてアプリケーションにドロップすることと，マルチメディアコンテンツを選択して別の情報家電に渡すことは非常に似ているためである．いずれの場合も，ユーザが明示的に制御を行い，指定するファイルやコンテンツにユーザの意識が注がれる．このような状況にあるため，マルチメディア家電を利用するシナリオをコンテンツ連携の代表的な例として扱う．

一方，コンフィグレーション連携は継続的な連携のため，設定後はあまり操作を伴わないシナリオとなっている．AV家電だけではなく，設備家電・白物家電などでもこうしたシナリオは非常に多いと考えられる．

4.1.2 携帯電話を選択した理由

節 2.4 で挙げた要件を全て満たすデバイスとして携帯電話を選択した．以下に記載する三つが主な理由である．

一つには，携帯電話は幅広く普及しており，ユーザにとって扱い慣れているデバイスであることが挙げられる．フィードバックを表示できることに加え，ユーザが日々利用しており，そのフィードバックを違和感なく受け取れるため，コントローラとして相応しいと言える．また，画像や文字などを実世界の画像の上に重ね合わせる*技術である拡張現実感 (Augmented Reality) [1] の手法も利用することが出来る．

二つ目の理由として，多くの携帯電話が CCD カメラを持ち，画像解析による ID 認識を行うことが可能ということである．DataFlowController では，ビジュアルタグをカメラ付携帯電話で認識することにより，情報家電の発見や操作のトリガーとする．情報家電にタグを貼り付けそれをカメラを通して見ることで，ユーザは実空間

*スーパーインポーズと言う

上でドラッグ&ドロップを行っているように感じ、直接操作の感覚を得られるのではないかと考える。

最後の理由は、携帯電話が元々ハンドヘルドデバイスとして非常にシンプルに設計されていることが挙げられる。小さな画面と限られたボタンによるインターフェースでは複雑な操作を行えない。それゆえに、多機能な情報家電をシンプルに表示し操作しなければならないという要件は必然的に満たすことになる。

また将来的には、ホームネットワーク内で操作するだけでなく、屋外でも家電と同様の機能を持ったパブリックディスプレイなどを制御できる可能性があるため、携帯電話は適していると言える。

これらの理由から、本研究ではカメラ付携帯電話を家電連携コントローラとして採用することにした。

4.1.3 インタラクションモデル

情報家電は今や非常に多くなってきており、家電連携のシナリオ（節 3.1.2）で述べたシナリオ以外にも、多くの利用例が存在する。そのため、いかなる家電が情報家電ネットワーク上に存在しても対応できるような一貫したインタラクションモデルである必要がある。そのために、コンテンツ連携・コンフィグレーション連携ともにほぼ同一のデータフローモデルを 2.4 章で作成し、連携後のインタラクションのみ異なるユーザインタフェースになるようにした。この二つのモードをシステム起動時に切り替えるようにして、DataFlowController のインタラクションモデルを図 4-1 のように定めた。

まず最初に、ユーザはコンテンツ連携モードか、コンフィグレーション連携モードかを選択する。ソフトキーボタンを押して、図 4-2 のような画面で切り替えを行う。切り替えを行うと、画面上部のアイコンが切り替わる（図 4-3, 4-4）。以下、コンテンツ連携モードおよびコンフィグレーション連携モードそれぞれの遷移について述べる。

コンテンツ連携モード ハードディスクレコーダーである X ビデオステーション（図 4-5）の動画をディスプレイで再生するシナリオの場合、まず移動元デバイスである X ビデオステーションに貼り付けられたビジュアルタグをカメラを利用して読み取る。このビジュアルタグは解析され、ビジュアルタグと紐づけられた家電がネットワーク上に存在する場合、図 4-6 のように当該家電を示すアイコンが表示される。ここで、ボタンを押して選択すると、X ビデオステーションの機能を表示するダイアログが出現する（図 4-7）。その後、“動画を選ぶ”を選択すると、動画ファイル

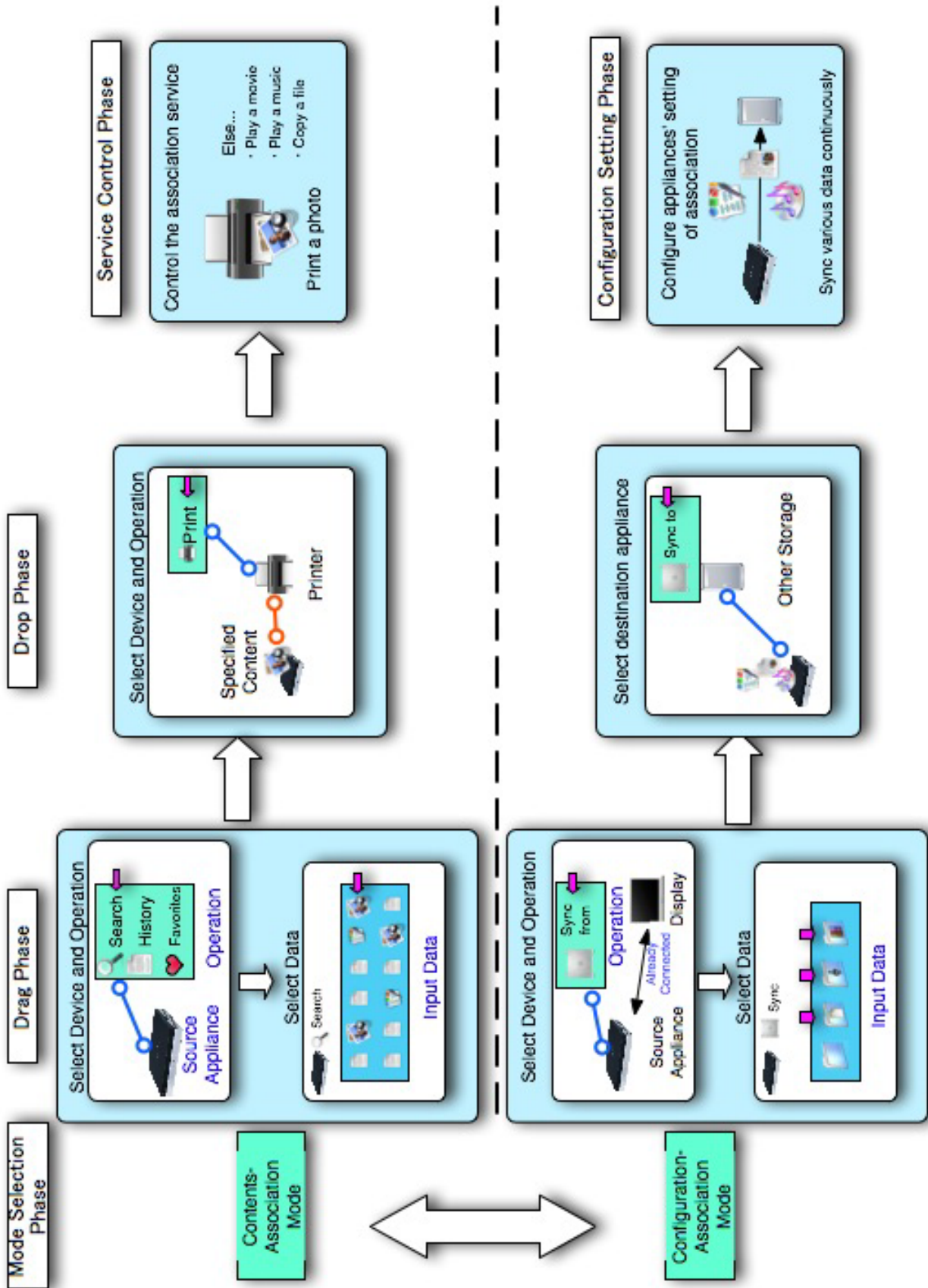


図 4-1: インタラクティブなシヨクシヨクモデル

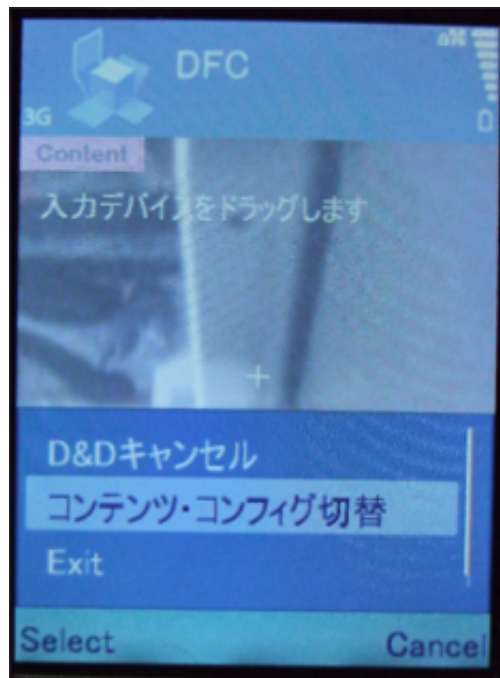


図 4-2: 2 種類のモードの切り替え



図 4-3: コンテンツ連携モードの表示



図 4-4: コンフィグレーション連携モードの表示

のリストを表示する(図4-8)。この中からいずれかのデータを選択すると、移動元デバイスを選択した旨のアイコンが画面に表示される(図4-9)。次に移動先デバイスであるディスプレイを選択し、機能を表示して連携を開始する(図4-10, 4-11)。連携時は早送り、巻き戻しなどを行えるコントロールUIを表示するが、ここでは簡易的にダイアログを表示している(図4-12)。



図 4-5: Sony Type X ビデオステーション



図 4-6: X ビデオステーションのアイコン 図 4-7: X ビデオステーションの機能を表示

コンフィグレーション連携モード 正常な時計のデータを時刻が狂ってしまう時計に定期的に同期するシナリオの場合、まず正常な時計デバイスを選択し、機能から“時刻を定期的に送る”を選択する(図4-13, 4-14)。このシナリオにおいて選択すべきデータは時刻情報しかないため、ダイアログが表示される(図4-15)。その後、時刻を修正したい時計を選択し、“時刻を修正する”を選択すると連携が開始する(図4-18)。連携時は、同期するタイミングを設定するUIを出す(図4-17)が、ここでは30秒後に設定した場合の画面を表示している(図4-17)。

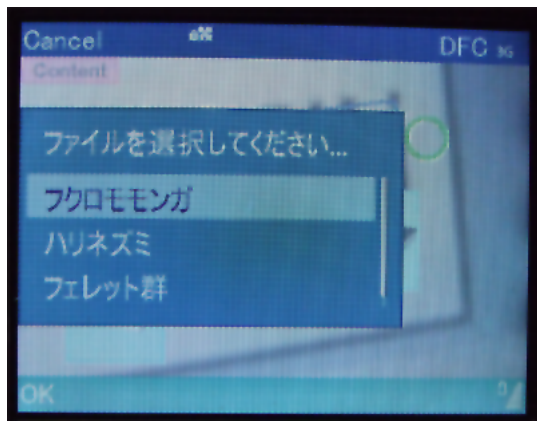


図 4-8: 再生可能な動画のリスト



図 4-9: 移動元デバイスを選択した状態

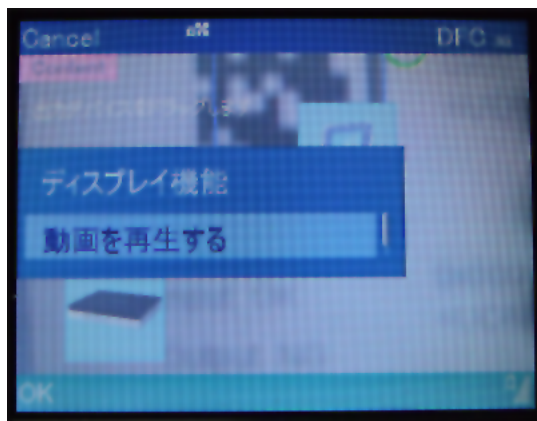


図 4-10: ディスプレイの機能

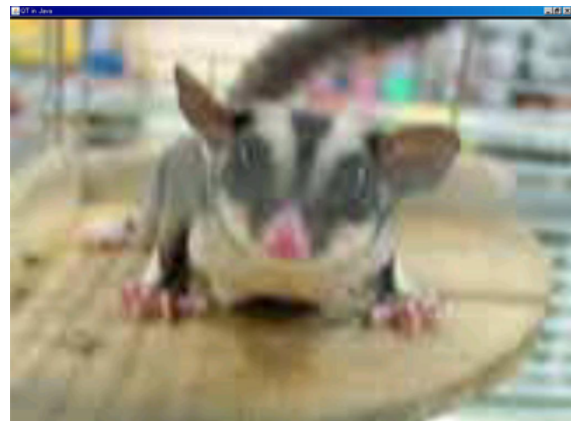


図 4-11: 動画再生中のディスプレイ

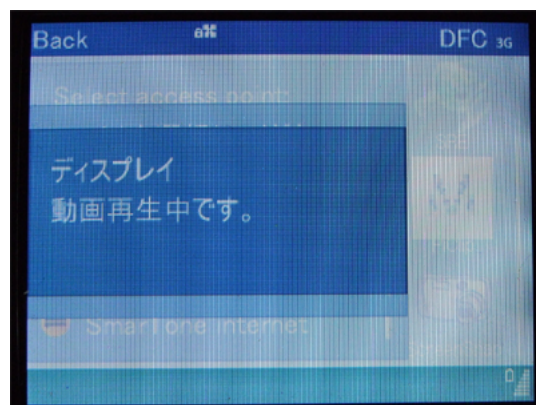


図 4-12: 動画再生中のコントロール UI



図 4-13: 時計デバイスのアイコン

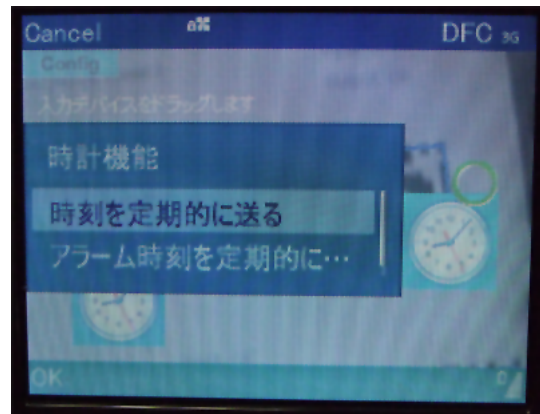


図 4-14: 時計デバイスの機能（移動元選択時）



図 4-15: 時計デバイスの時刻データダイアログ

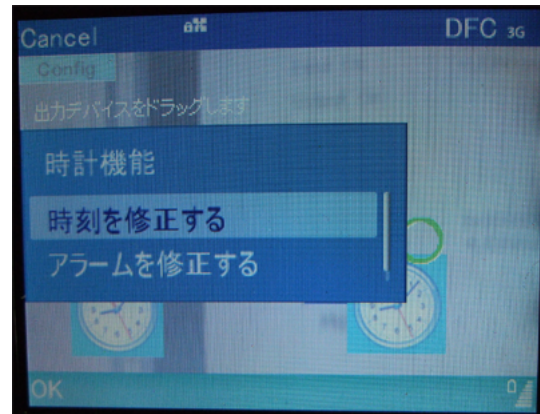


図 4-16: 時計デバイスの機能（移動先選択時）

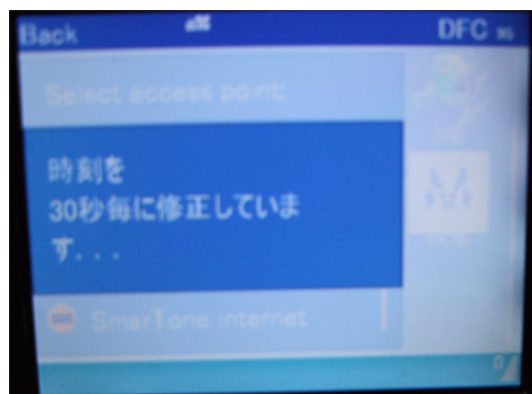


図 4-17: 時計の同期



図 4-18: 時計デバイス（右は時刻がずれている）

X ビデオステーションの機能やデータ，時計の機能やデータ，それぞれの連携部分は将来的には各家電メーカーが提供すべきアプリケーションであるため，一貫した実装を行えるようにしている（節 4.2.3）。

4.1.4 連携しない家電選択時の挙動

選択した家電と，移動先家電に連携の互換性がない場合もチェックを行っている。DataFlowController では，それぞれの家電に入力属性，出力属性を持たせ，移動元として選択するデバイスが入力属性を持たない場合，もしくは移動先として選択するデバイスが出力属性を持たない場合は，アイコンに×印がついて選択できないようになっている（図 4-19）。

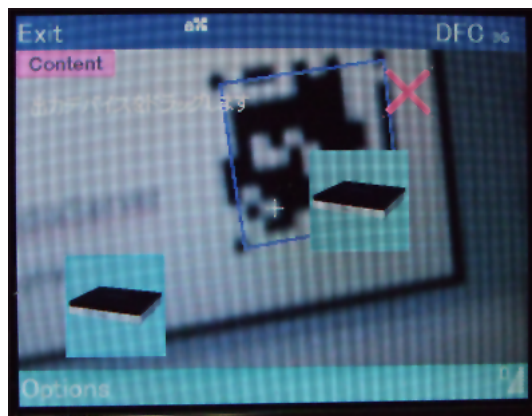


図 4-19: 連携できない場合

このように，移動元の情報家電の持つ多くの機能から 1 つを選び，その機能と移動するデータを，移動先の情報家電で対応するかどうか判別した上で，移動先の機能を選択できる仕組みを開発した。

4.2 実装

本システムは大きく分けて、携帯電話側の実装と、PC上の家電アプリの実装の二つに分けられる。携帯電話は海外製の Nokia N93 (プラットフォーム: S60 3rd) (図 4-20) を利用した。N93 はワイヤレス LAN やカメラを搭載しており、UPnP 通信を行えるため利用した。Symbian では C++ による開発が行え、カメラの利用や TCP/IP 接続などのネイティブレベルの API[†] も利用することが出来ることから、Symbian C++ での開発を行った。情報家電には、Sony Type X ビデオステーション (図 4-5)、ディスプレイ、時計を利用し、PC 上の家電アプリは Java の J2SE を用いて開発を行った。携帯電話と家電アプリで言語が異なるが、通信は UPnP を用いて SOAP 通信を行っているため、問題なく動作する。

本論文で実装したシステムの概略である。

携帯電話側

1. VisualCodes によるタグ解析システム
2. CyberLink の移植による UPnP 検出と操作
3. ドラッグ&ドロップ機構の構築
4. 情報家電独自のユーザインタフェースを携帯電話画面に表示するための UI エンジン

情報家電側

1. ディスプレイの再生機能
2. 時計の表示や時刻修正機能

プログラム行数は、携帯電話側は約 9000 行を新規作成した。家電アプリケーション側は約 2000 行を修正した。

4.2.1 利用技術

本節では、本研究に用いたライブラリや基盤技術について説明する。

[†]特定の権限などが必要



図 4-20: Nokia N93

VisualCodes

VisualCodes[7] は携帯電話のような浮動小数点プロセッサを持たない限られたリソースのデバイスでも、処理にかかる負荷を減らし、高速にタグを認識できるシステムである。画像を取得し ID を解析しながら、タグの回転、傾き、携帯電話とタグとの距離などを計算することが出来る。公開されているソースは、Symbian S60 2nd のため、カメラ API などに修正を加え 3rd 用に移植して利用した。

本システムでは、各情報家電に付与したビジュアルタグを解析し、その ID がどの情報家電が認識するために利用している。

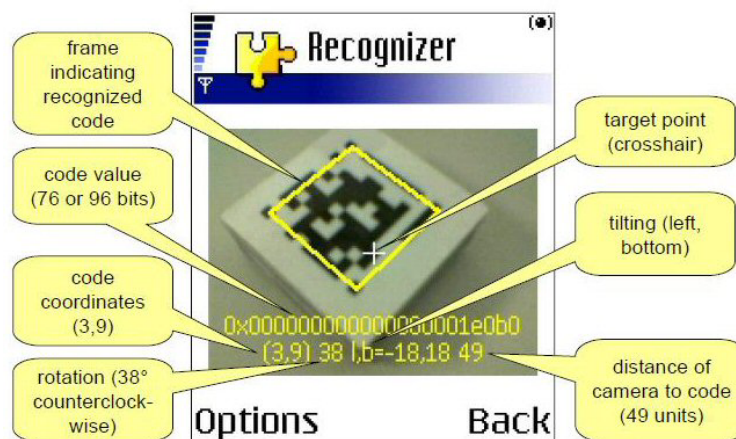


図 4-21: VisualCodes

CyberLink

CyberLink[‡] はネットワークに接続された家電やサービスを発見・制御するための技術仕様である UPnP のライブラリである。これを用いることにより、利用者は非常に簡単に UPnP サービスを作成、発見、制御できる。CyberLink は C++、Java、Perl、C など多様な言語で提供されているが、本研究では組み込み向けに開発された C 言語版を Symbian C++ に移植することで開発を行った[§]。移植したプログラム行数は 30,000 行を超える。移植の際、ソケット通信やスレッド、文字列型変数、XML パーサの違いに基づき移植を行った。

本システムでは、アプリケーション起動時に CyberLink により、LAN 内の情報家電を検出、リストに格納し、移動元デバイスの機能選択およびデータ選択、移動先デバイスの機能選択および連携後において、SOAP 通信を行っている。

QuickTime for Java

Sony Type X ビデオステーションでは、再生動画が MPEG-2 プログラムストリーム (MPEG-2 PS) 形式のため、ディスプレイアプリの再生機能にはコーデックが必要であった。したがって、MPEG-2 のコーデックを PC にインストールし、マルチメディアの再生などが可能な API である QuickTime for Java を用いてディスプレイアプリを実装した。

本システムでは、ハードディスクレコーダの動画をディスプレイに連携させて再生する際に利用した。

4.2.2 アーキテクチャ

アーキテクチャは図 4-22 のとおりになる。起動直後、CyberLink がマルチキャストを行い、ネットワークに存在する情報家電からレスポンスを受ける。それらの情報は保持され、カメラがビジュアルタグを認識した際に、ビジュアルタグ ID と情報家電のデバイス情報が関連付けられ ApplianceInfo クラスに格納される。ビジュアルタグは 16 進数の値を持っており、その値は情報家電のデバイス情報が記載された description.xml の UDN タグのハイフンを除いた下 10 桁と一致している (表 4-1)。

格納された家電情報は、選択時に D&D マネージャに渡され、D&D マネージャは各情報家電の UI コントローラクラスをインスタンス化する。このオブジェクト経由で UPnP メソッドを送り、機能選択、データ選択などの UI を表示する。移動元デバ

[‡]CyberLink: <http://www.cybergarage.org/net/index.html>

[§]2006 年度未踏ソフトウェア創造事業 (未踏コース) にて共同開発
<http://www.ipa.go.jp/jinzai/esp/2006youth/gaiyou/1-05.html>

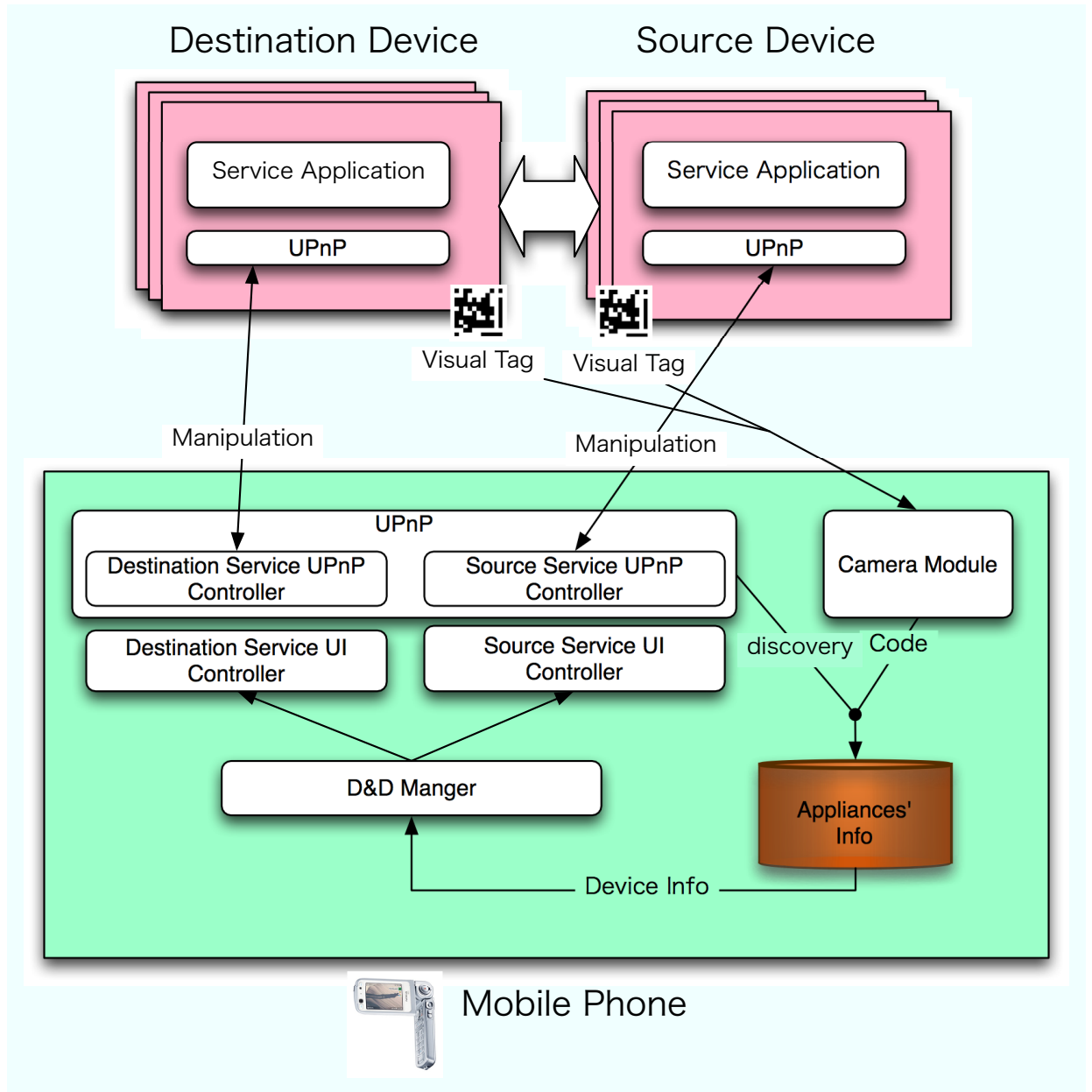


図 4-22: アーキテクチャ

VisualCode ID	0x000000000000013a9139749
UDN	<UDN>uuid:000000000000-1010-8000-0013a9-139749</UDN>

表 4-1: VisualCode ID と UDN の関連

イス選択後には、同様の手順を踏み、移動先デバイスの UI コントローラクラスをインスタンス化する。連携は移動先デバイスのメソッド内に記述する。

コンフィグレーション連携の命令は現状携帯電話から送られている。そのため、DataFlowController を終了するとコンフィグレーション連携は終了してしまう。このような実装の理由は、特定の UPnP 機能を持つコントロールポイントのサーバを立てると、家電連携コントローラが複数存在する場合、設定が競合してしまう可能性があるためである。例えば、あるユーザ A が室温に合わせて ±3 度にエアコンの温度設定を行っている時に、ユーザ B が同じエアコンを ±5 度で設定したら予測不可能な状態になってしまう。この問題は将来課題として後ほど述べる（6 章）。

4.2.3 サービスアプリ UI クラスの抽象化

様々な情報家電に対して、一貫したインタラクションモデルを提供するために、システムは情報家電の挙動を記述するクラスを抽象化し、常にそのクラスを継承してサービス UI コントローラを作成するよう実装した。システムは getName() を呼び出すことで各サービス UI コントローラに適した型にダイナミックキャストを行う。移動元デバイスの機能表示は displayDragSearchMenuContent() と displayDragSearchMenuConfig()、移動元デバイスのデータ表示は displayDragSelectMenuContent()、displayDragSelectMenuConfig() で行う。同様に移動先デバイスの機能表示は displayDropSearchMenuContent()、displayDropSearchMenuConfig() で行う。連携確立後には executeAssociation() をコールして、連携時に必要な UI を描画するようになっている。

ソースコード 4.1: ServiceAppUI.h

```
#ifndef SERVICEAPP_H_
#define SERVICEAPP_H_

#include <coecntrl.h>
#include "ApplianceInfo.h"

// CLASS DECLARATION
class CApplianceInfo;
```

```
/**
 * ServiceAppUI
 *
 */

class MServiceAppUI
{
public:

    // サービスアプリの名前を取得する
    virtual TDesC& getName(){};
    virtual void setName(){};

    // 移動元・移動先の機能名
    virtual TDesC& getDragFunctionName(){};
    virtual TDesC& getDropFunctionName(){};

    // 移動元の機能表示コードを記述するメソッド
    virtual TBool displayDragSearchMenuContent(){};
    virtual TBool displayDragSearchMenuConfig(){};

    // 移動元のデータ表示コードを記述するメソッド
    virtual TBool displayDragSelectMenuContent(){};
    virtual TBool displayDragSelectMenuConfig(){};

    // 移動先の機能表示コードを記述するメソッド
    virtual TBool displayDropSearchMenuContent(
        MServiceAppUI* dragServiceAppUI,
        TDes& AssocName){};
    virtual TBool displayDropSearchMenuConfig(
        MServiceAppUI* dragServiceAppUI,
        TDes& AssocName){};

    // 連携コードを記述するメソッド
```

```
virtual TBool executeAssociation(  
    CApplianceInfo* dragDeviceInfo ,  
    CApplianceInfo* dropDeviceInfo ,  
    TDesC& AssocName){};  
  
private :  
  
};  
  
#endif /*SERVICEAPP_H_*/
```

第5章

ユーザスタディ

本章では、本研究で行ったユーザスタディの概要や手順、評価方法について述べる。また、ユーザスタディの結果を元に考察を行い、今回の実験の反省点やシステムの改善点について述べる。

5.1 実験概要

本研究において、ユーザが情報家電の接続と連携を問題なく利用できるようになるか確かめるためのユーザスタディを行った。確認すべき点は、

- 移動元の家電から移動先の家電へのデータフローが分かりやすいものであったか
- 携帯電話で情報家電ネットワークを操作することについてどう感じたか
- ドラッグ&ドロップを拡張した本モデルは受け入れられるか

の3点である。システムを利用後被験者にインタビューを行い、それを元にシステムの評価を行った。

5.2 実験環境・被験者

実験は早稲田大学理工学研究科 中島研究室の一室で行った。利用した機器は Nokia N93 (図 4-20) を 1 台、ハードディスクレコーダーである Sony Type X ビデオステーション (図 4-5)、情報家電代わりに実装したアプリケーションを搭載したノート PC 1 台、ディスプレイ装置である。実験環境は図 5-1, 5-2 のような環境である。

被験者は 6 名で、5 名は情報工学を専攻する 20 代の学生である。残り 1 名は情報工学を専攻していない 20 代の学生である。



図 5-1: 実験環境その 1



図 5-2: 実験環境その 2

5.3 実験手順

まず最初に、被験者に実験の狙いを説明し、実験内で想定する以下の2種類のシナリオについて解説する。その後、被験者にシステムに関する簡単な説明を行う。本システムがドラッグ&ドロップを元にして家電を操作できることを述べた後に実験を開始する。

コンテンツ連携

ハードディスクレコーダーからテレビ番組を選択し、ディスプレイで再生する

コンフィグレーション連携

正常な時計のデータを送信し、時刻がずれる時計を定期的に同期させる

5.4 評価方法

被験者へのインタビュー項目は、本実験で行ってもらった作業を日常どれくらい行っているかという質問と、実験の感想に関する質問の二種類である。

日常の作業の頻度について

- PC 上などで普段ドラッグ&ドロップを行うか・いつどのような時に行うか
- 右ドラッグ&ドロップは行うか
- 日常生活の中で家電連携を行うことはあるか

- 携帯電話のカメラでどの程度ビジュアルタグを読み取るか

実験の感想について

- 第一印象はどのように感じたか
- データの流れは分かりやすかったか（コンテンツ連携・コンフィグレーション連携それぞれについて）
- 指示を受けて、何をすればいいのか、どのようなデータが流れたか理解できたか
- カメラ付き携帯電話を利用して家電連携を行うことをどのように感じたか・別の手法が望ましいか
- 提案したインタラクションモデルは理解できたか・ドラッグ&ドロップの拡張だと思えたか
- GUI上のドラッグ&ドロップと比較してどのように思うか

5.5 実験結果と考察

5.5.1 日常の作業の頻度について

まず、ほぼ全員の被験者がPC上でドラッグ&ドロップを頻繁に行うという結果が出た。行う作業はファイルの移動がほとんどで、2名がアプリケーションに対してドラッグ&ドロップを行うと答えた（表5-1）。ファイルのコピー（Alt+ドラッグ）を行うという被験者もあり、ドラッグ&ドロップを駆使するユーザも見られた。しかし、右クリックによるドラッグ&ドロップはショートカットの作成に見られるように少なく、この操作を知らない人が2名、知っているがめったに使うことがない人が4名いた。

操作内容	人数
フォルダ間などのファイル移動	6/6
アプリケーションへの連携	2/6
ファイルのコピー	1/6
ショートカットの作成	1/6

表 5-1: PC 上でドラッグ&ドロップをどのように利用するか

また、携帯電話でタグを読み取る頻度について、よく読み取る人が2名、たまに読み取る人が2名、あまり読まない人が2名と分かれる結果となった(表5-2)。タグをよく読み取る被験者2名は、アルバイトでタグを読み取るテスターを行った経験のあるエキスパートユーザである。

操作内容	人数
よく読み取る	2/6
たまに読み取る	2/6
ほとんど読まない	2/6

表 5-2: 携帯電話によるタグの読み取り頻度

5.5.2 実験の内容について

インタビューにより、6人中5人が本システムのインタラクションモデルに違和感を感じることなく操作できたと答えた。これはコンテンツ連携が比較的 GUI 上のドラッグ&ドロップと近いデータフローモデルであるためだと思われる。残る一名はコンフィグレーション連携に慣れなかったと答えた。GUI 上のドラッグ&ドロップは即時的にデータの流れが発生するのに対して、コンフィグレーション連携は継続的に発生するため、ドラッグ&ドロップというメタファがデータの道筋を作成することに結びつかなかったと考えられる。また、違和感を感じなかった一名の被験者からも、コンフィグレーション連携後の動作が分かりづらいという意見を頂いた。これはコンフィグレーション連携時の UI を何らかの形で連携前に見せることで解決できると思われる。画面やキーワードを表示したり、音声や連携できる情報家電をサジェストすることにより、これらの敷居は下げられると思われる。また、もう一つ改善案として、よく使う機能はあらかじめ別に登録しておきたいという意見も2名見られた。GUI 上では、シンプルなインタラクションである左ドラッグ&ドロップと、拡張された機能を持つ右ドラッグ&ドロップの二種類が存在するように、複数の手法を用意することも決してインタラクションを複雑にするわけではない。使い慣れたシナリオはデータフローを敢えて確認する必要がないとも言えるため、将来的には検討すべき事案である。

本実験の重要事項の一つである、何のデータがどこからどこへ移動するといった、データフローに関する質問も全員が特に問題なく答えることが出来た。これは本システムによるデータフローモデルが受け入れられやすいことを意味すると考えられる。

また全員が別のシナリオでも操作できそうだと答え、学習効果が見込める結果が

出た。ただし、実験開始後の説明で理解できなかった被験者も2,3名おり、使ってみるまでメンタルモデルを構築することは比較的難しく、改善の余地があるという課題も得られた。

本実験で生じた問題の一つとして、タグの選択に苦戦する人が大勢いたことが挙げられる。これには幾つか要因があるが、一つは VisualCodes と CyberLink の処理が想像以上に負荷がかかったため、カメラ解析のフィードバックが遅かったことが挙げられる。加えて、室内の光量が足らず、黒いタグの解析に必要な白の部分のコントラストを得ることが出来なかったこと、携帯電話の形状が90度捻って利用する形であることなども要因となり、ほとんど読み取ることが困難な状況も頻繁に発生した。カメラ ID 認識による直接操作手法は万人向けではないという意見もあり、このような結果から別の手法の検討も必要な結果となった。被験者の一人が、RFID のような非接触式 IC チップならば、携帯電話を閉じたままかざすと自動的に起動できるのではないかという意見があり、将来検討すべきだと考えられる。

また、入力家電と出力家電の選択時に、ボタンを押したままの被験者が2名いた。実験開始時にドラッグ&ドロップの要領で操作すると説明した際に、実際にボタンを押し続けるものと想像したためと思われる。メニュー選択などを行う必要性から、ボタンを離しても良いインタフェースにしたが、ボタンを押していないとデータを現在保持しているという感覚が薄れるという意見もあり、判断が分かれるところである。

全体を通して、フィードバックを更に検討する必要がある意見が見られた。現在何のデータを運んでいるか表示が欲しい、コンフィグレーション連携時に起こることを先に移動先デバイスを選択する際に見たい、タグが読み取れないときに何らかの表示を出して欲しいといった意見である。これらは比較的解決しやすい問題であり、早急に実装できればと考えている。

第6章

結論と将来課題

本論文では、情報家電の特性に合わせて一貫した接続、連携を行えるインタラク션을考案し、データフローの流れを認識しやすくする手法を提案した。その結果、情報家電の相互接続シナリオにおいてドラッグ&ドロップモデルのメタファは応用可能であり、学習効果も望めることがわかった。フィードバックの適切な表示をシステムに導入することで、データフローはよりユーザに認識されやすくなると考えられる。

しかしインタビューの結果によると、一般的にビジュアルタグは普及してはいるものの、カメラ ID 認識による直接操作手法はあまり万人向けなインタフェースではなく、RFID や他の代替案を含む検討が必要という回答も得られた。

今後の課題は、複数ユーザが同一の家電のコンフィグレーション連携を行うといった、設定の競合状態を解決することが挙げられる。本実装では全体の命令を把握する UPnP のコントロールポイントサーバの実装を見送ったが、ユーザに優先度を与えたり、時系列による判定を用いて実装すれば実現可能だと思われる。

また、GUI 上でドラッグ時に範囲選択を行うように、複数の情報家電をまとめて設定できるようなインタラクシオンの考案も要望があり検討の必要がある。

更に、本研究のインタラクシオンを考えるにあたって、データフローによる連携だけではなく、イベントをトリガとする家電連携のシナリオもあることが判明した。例えば、玄関の電気を消したら、家の家電の電源が切れるというシナリオは、やり取りするものはデータではなく、“電気を消した”というイベントである。今回はデータフローモデルに限定して議論を進めたが、イベントベースの家電連携も含めて同一のインタラクシオンに出来るかどうかは今後この分野の主眼になるとと思われる。

謝辞

本研究の機会を与えてくださり、丁寧な指導をしていただいた中島達夫教授に深く感謝いたします。国際学会の論文執筆に当たっては、木村浩章氏、Mr. Fahim Kawsar、Mr. Vili Lehdonvirta に多くの助言をいただきました。本システムの基盤部分の開発を行った太田英之氏には非常に苦勞をかけました。そして、中島研究室 Ambient Intelligence グループのメンバー、ならびに中島研究室の他の研究グループの先輩同輩の皆様にご心より感謝いたします。最後に忙しい中ユーザスタディにご協力していただいた方々、本当にありがとうございました。

参考文献

- [1] R.T. Azuma, et al. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments(1054-7460)*, Vol. 6, No. 4, pp. 355–385, 1997.
- [2] 首相官邸 IT 戦略本部. e-japan 重点計画-2004.
<http://www.kantei.go.jp/jp/singi/it2/kettei/ejapan2004/040615honbun.html>, 6 2004.
- [3] Y. Iwasaki, N. Kawaguchi, and Y. Inagaki. Touch-and-connect: a connection request framework for ad-hoc networks and the pervasive computing environment. *Pervasive Computing and Communications, 2003.(PerCom 2003). Proceedings of the First IEEE International Conference on*, pp. 20–29, 2003.
- [4] N. Kohtake, J. Rekimoto, and Y. Anzai. InfoStick: An Interaction Device for Inter-Appliance Computing. *Handheld and Ubiquitous Computing: First International Symposium, HUC'99, Karlsruhe, Germany, September 1999: Proceedings*, 1999.
- [5] J. Rekimoto. Pick-and-drop: a direct manipulation technique for multiple computer environments. *Proceedings of the 10th annual ACM symposium on User interface software and technology*, pp. 31–39, 1997.
- [6] J. Rekimoto, B. Ullmer, and H. Oba. DataTiles: a modular platform for mixed physical and graphical interactions. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 269–276, 2001.
- [7] M. Rohs and B. Gfeller. Using camera-equipped mobile phones for interacting with real-world objects. *Advances in Pervasive Computing*, pp. 265–271, 2004.
- [8] B. Shneiderman. 1.1 Direct manipulation: a step beyond programming languages. *Sparks of Innovation in Human-Computer Interaction*, 1993.
- [9] B. Ullmer, H. Ishii, and D. Glas. mediaBlocks: physical containers, transports, and controls for online media. *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pp. 379–386, 1998.
- [10] 情報家電の市場化戦略に関する研究会 Life 戦略研究会～. 基本戦略報告書「e-life イニシアティブ」. Technical report, 経済産業省 商務情報政策局 情報通信機器課, 4 2003.
- [11] デジタル情報家電のネットワーク化に関する調査研究会(羽鳥光俊ら). デジタル情報家電のネットワーク化に関する調査研究会 報告書概要. Technical report, 総務省 情報通信政策局通信規格課, 8 2004.

- [12] 綾塚祐二, 暦本純一. tranStick: 空間を越えて仮想的に繋がったメディア. *The 11th Workshop on Interactive Systems and Software*, 2003.
- [13] 石井威望ら. 情報家電ネットワーク化に関する検討会【中間取りまとめ】. Technical Report 1, 経済産業省 商務情報政策局 情報政策課, 7 2005.
- [14] 丹康雄, 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, 2004.