

筑波大学大学院博士課程

システム情報工学研究科特定課題研究報告書

携帯電話を活用する音声応答家電の開発
ー音声合成機能つき携帯電話への
UI の提示と対話操作の実現ー

徐 偉

(コンピュータサイエンス専攻)

指導教員 田中二郎

2012年 3月

概要

本プロジェクトは、筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻高度IT 人材育成のための実践的ソフトウェア開発専修プログラムにおける特定課題研究として、同大学院に所属する教員が提案した「携帯電話を活用する音声応答家電の開発」というテーマで、音声合成機能が搭載されている携帯電話を用いて、安価に目の不自由な方にも利用できる家電を製作し実動させた。

本プロジェクトでは、音声合成機能が搭載されている携帯電話を利用して家電と連携させることで、安価に目の不自由な方にも利用できる家電を製作し実動させる。使用する携帯電話としてNTT ドコモのらくらくホン6を採用し、対象とする家電としてIOC-125 おまかせミールさんを選定した。

仕様の設計を行う前、機器連携プロトコルを参考するため、調査を行った。調査において筆者は、プロトコルの一部と記述方式の調査を担当した。また、解析の方法を調査して、記述方式を選んだ。

仕様記述の設計では、初めから汎用のものを作るのではなく、事例をもとに仕様記述が満たすべき条件を整理し、汎用記述をあらためて定義する、という手順で行った。簡単な事例のプロトタイプを開発することで、簡潔かつ解析可能な記述方式を定めた。また、仕様記述からUIの提示とイベント処理の可能性を示した。

携帯用のソフトウェアの開発において筆者は、UI提示とコマンド作成機能を担当した。プロトタイプに基づいて、他のメンバーが開発したオリジナルコンポーネントを取り入れて、結合するため、仕様記述とソースを直した。また、新たな機能も追加した。

他のメンバーが担当した部分と連携し、音声でUI提示して、操作により家電を動かせることを実現した。音声付き携帯電話を用いて音声応答家電の開発の可能性を示した。

目次

第 1 章	はじめに	6
1.1	研究の背景	6
1.2	研究の目的	6
1.3	報告書の構成	6
1.4	用語の定義	7
第 2 章	機器連携に関する関連研究	8
2.1	ECHONET	8
2.2	Jini	9
2.3	UPnP	9
2.4	URC (Universal Remote Console)	10
2.5	HAVi (Home Audio Video Interoperability)	11
第 3 章	システム構成及び開発計画	12
3.1	機能要件	12
3.2	非機能要件	14
3.3	機材の選定	15
3.3.1	端末の選定	15
3.3.2	通信方式の選定	16
3.3.3	家電用マイコンの選定	16
3.3.4	実証用家電の選定	17
3.4	全体設計	18
3.5	開発体制	19
3.6	スケジュール	20
第 4 章	家電のリバースエンジニアリング	22
4.1	基本機能	22
4.2	制約条件	22
4.3	機能一覧	23
4.4	GUI プロトタイプ作成	24
第 5 章	仕様記述形式に関する調査	25
5.1	仕様記述形式の調査	25
5.1.1	Versit 形式	25
5.1.2	XML 形式	25
5.1.3	JSON 形式	26
5.2	解析方式の調査	26
5.3	仕様記述方式の検討	27
第 6 章	仕様記述の検討	28
6.1	簡単な事例の試作	28
6.2	UI 仕様の記述と解析の検証	29
6.3	制御仕様の記述と解析の検証	29
6.4	赤外線エンコード部分の記述方式と解析の検証	30

第 7 章	携帯電話用端末ソフトウェアの開発	32
7.1	携帯電話用ソフトウェアの構成	32
7.2	仕事分担	33
7.3	担当部分のメソッド構成	34
7.4	仕様記述の構成と解析、実現方式	35
7.4.1	コンポーネント表示	36
7.4.2	画面遷移	40
7.4.3	イベント処理	41
7.4.4	グループ処理	44
7.4.5	変数処理	49
7.4.6	CIR コマンドのフォーマット処理	49
7.4.7	if 文処理	49
7.4.8	CIR 通信処理	50
7.4.9	音声読み上げ処理	50
第 8 章	プロジェクトの分析	51
8.1	開発体制	51
8.2	開発環境	52
8.3	開発計画と実績	53
8.4	成果物	54
第 9 章	結論	55
謝辞		56
参考文献		57

図目次

図 2-1	ECHONET クラスグループ例	8
図 3-1	システム概要	12
図 3-2	ユースケース図	13
図 3-3	システム構成図	18
図 3-4	仕事分担	19
図 3-5	携帯側開発スケジュール	20
図 3-6	家電側開発スケジュール	21
図 4-1	GUI プロトタイプ	24
図 5-1	Versit 形式の例	25
図 5-2	XML 形式の例	26
図 5-3	JSON 形式の例	26
図 6-1	ラジオカセットレコーダーの実行画面	28
図 6-2	複雑な制約条件の実行画面	30
図 6-3	コマンド作成部分の試作画面	31
図 7-1	全体クラス図	33
図 7-2	仕様記述の構成図	35
図 7-3	コンポーネント表示の記述例	36
図 7-4	UI 表示部分の処理フロー	38
図 7-5	UI 表示部分の試作画面	39
図 7-6	画面遷移の試作画面	40
図 7-7	イベント処理の記述例	41
図 7-8	イベント処理の主処理フロー	42
図 7-9	イベント処理の条件式の処理フロー	43
図 7-10	グループ処理部分の記述例	44
図 7-11	グループの構造例	45
図 7-12	グループ作成時の処理フロー	46
図 7-13	グループを処理する時のフロー	47
図 7-14	グループ制御の試作画面	48
図 7-15	変数部分の記述例	49
図 7-16	if 文の記述例	49
図 8-1	ミーティング実績	51
図 8-2	携帯側開発予定と実績	53

表目次

表 1-1	用語集	7
表 2-1	URC の定義ファイルの種類	11
表 2-2	HAVi で取り扱われる情報とその概要	11
表 3-1	DoJa-5.1LE プロファイルに対応した機種	15
表 3-2	通信方式一覧	16
表 3-3	赤外線通信方式	16
表 3-4	検討家電一覧	17
表 3-5	役割分担一覧	19
表 3-6	携帯側開発スケジュール	20
表 3-7	家電側開発スケジュール	21
表 4-1	基本機能と制約事項一覧	23
表 7-1	uicomponent パッケージの構成	32
表 7-2	remotecontrol パッケージの構成	32
表 7-3	UICreation クラスのメソッド一覧	34
表 7-4	仕様記述の属性名と概要一覧	37
表 7-5	コンポーネントごとの属性の利用状況一覧	37
表 8-1	開発環境	52
表 8-2	プログラムの実績	54

第1章 はじめに

1.1 研究の背景

平成 18 年 7 月 1 日現在、全国の身体障害者数（在宅）は、3,483,000 人（このうち、63.5% が 65 歳以上）となっている。このうち視覚障害者は 31 万人と推計されている[1]。また、平成 15 年 9 月 15 日現在における全国の 65 歳以上人口（推計）は 2,431 万人で、総人口の 19.0% を占め、総人口のおよそ 5 人に 1 人の割合となっている。65 歳以上人口の割合は今後も上昇を続け、2015 年には総人口の 26.0%（3,277 万人）と、およそ 4 人に 1 人が 65 歳以上になると見込まれている[2]。

現在、家電の高機能化が進んでおり、視覚障害者には使いにくくなっている問題がある。例えば、電子レンジの場合、昔はボタン数個でコントロールしたが、新商品はほとんど数十種類以上のメニューを持っている。視覚障害者は音声のついてない家電を使用するのは困難である。視覚障害者は家電に音声機能を付ける要望に対して、音声付き家電はほとんど普及していない。白物家電にはコストの制約が厳しく、発声機能を付けるのは困難である。この課題を解決するソリューションが求められる。

1.2 研究の目的

個々の家電に音声をつけるようになると、高性能プロセッサ、アンプ、スピーカーが要る。値段が高くなる。ところで、視覚障害者はメールやウェブブラウジングのために音声合成機能のついた携帯を常用している、この携帯と家電を連携させて低コストで家電の音声応答を実現する。

1.3 報告書の構成

本報告書は全 8 章から構成される。

第 2 章では、外部から家電機器を操作する方法を調査するため、行った関連研究について述べる。

第 3 章では、システムの機能要求、非機能要求から機材を選定した後、システム設計と開発計画、開発体制について述べる。

第 4 章では、実証機器であるおまかせミールさんの機能解析について述べる。

第 5 章では、仕様記述を決める前、仕様記述方式と解析方式を調査してから、記述方式を決めることについて述べる。

第 6 章では、簡単な事例を試作しながら、記述方式を検討することについて述べる。

第 7 章では、携帯ソフトウェアの全体構成と自分担当する部分の構成、実現方法について述べる。

第 8 章では、プロジェクトの進行上における問題について述べる。

第 9 章では、本プロジェクトの成果から出した結論と今後の展望について述べる。

1.4 用語の定義

表 1-1 は本報告書の理解のために必要な用語とその定義を示す。

表 1-1 用語集

用語	定義
おまかせミールさん	正式商品名は Iwatani IOC-125 モーニングセットで、愛称はおまかせミールさんである。自動朝食調理器で、卵焼き、パン焼き、コーヒーを同時に作ることができる
らくらくホン	NTT ドコモの音声合成機能つき携帯電話
i アプリ	NTT ドコモの携帯電話で実行出来る Java を使用する Java アプリケーション及びサービスである
DoJa	NTT ドコモグループの携帯電話である mova シリーズ及びFOMA シリーズに搭載される Java 実行環境の仕様である。
CIR	単方向赤外線通信の規格
IrDA	双方向赤外線通信の規格

第2章 機器連携に関する関連研究

本特定研究課題で、開発する「家電音声化システム」は、音声合成機能つき携帯電話と家電を連携させて、音声ガイドによる家電の外部制御を実現するものである。家電音声化システムを設計する上では、他の機器連携システムの動作原理が参考になるはずである。そのため、これまでに知られている機器連携システムに関する調査を行った。

2.1 ECHONET

ECHONET (Energy Conservation and Homecare Network) [3]は、国内家電メーカーにより設立された ECHONET コンソーシアムが提唱している家電の外部制御の規格であり、外出先からのエアコンの制御や携帯電話による施錠確認などがサービスシナリオとして紹介されている。

このような制御を実現するために、ECHONET 規格では、オブジェクト指向により家電機器をモデル化して、相互接続実現に必要な共通の・基本的機能をサービスミドルウェアとして規格化している。規格の中で、家電機器を制御するためのオブジェクト定義は制御方法を細かく定義している。それは機器オブジェクトと呼ばれる[4]。機器の制御部分はクラス内に定義する。

機器オブジェクトはいくつのクラスグループに分かれており、この下にクラスがある。制御内容又は機器状態は各クラスのプロパティとして定義されている。クラスグループコード、クラスコード及びプロパティにより、どんな装置であるか、どのような機能を持っているかが分かる。例えば、電気ポットは、図 2-1 のように調理、家事関連機器クラスグループに分類され、蓋開閉状態、湯切れ警告などのプロパティを持っている。

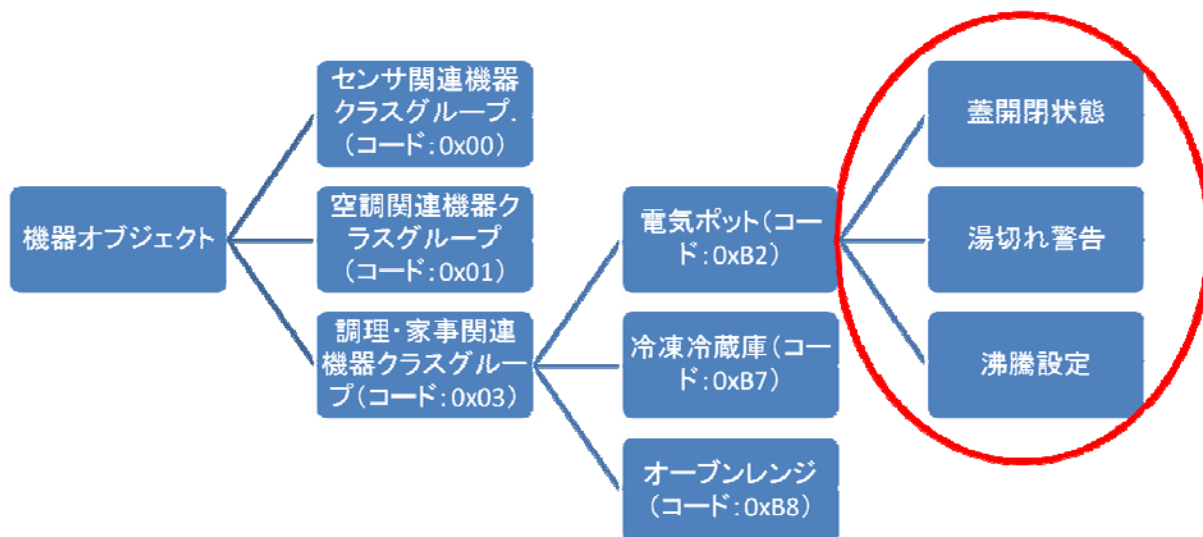


図 2-1 ECHONET クラスグループ例

機器を制御する時、目的機器の ECHONET アドレス、オブジェクト ID (クラスグループコード)、プロパティ ID (クラスコード)、及びサービス内容 (SET、GET など) を指定したコマンドを送信することで制御が行われる [4]。例えば、電気ポットを用いて、沸騰させようと思う時、機器の ECHONET アドレス (0x03)、オブジェクト ID (0xB2)、沸騰設定 (0x41) を送信することで、水を沸騰させる。

しかし、プロパティ内容が事前に決められているので、拡張性が弱く、今回の開発に適していない。

2.2 Jini

Jini (Java Intelligent Network Infrastructure) [5] とは、デジタル機器をネットワークに接続するだけで、機器間での協調動作を実現するための Java 分散オブジェクト応用技術である。Jini のポイントは、ハードウェアもソフトウェアも区別せず、全部オブジェクトとして扱う点と、互いの情報を持たないオブジェクト同士でも通信できる点である。

Jini はサービスの提供側である「サービスプロバイダ」、サービスの利用側である「クライアント」、ネットワーク内のすべてのサービスを管理する「Lookup サービス」から構成される [6]。Discovery、Join、Lookup の 3 つ過程でプラグアンドプレイが実現される。

まずサービスプロバイダは Lookup サービスを見つける (Discovery)。そして、サービスプロバイダは自サービスのプロキシオブジェクトを Lookup サービスに登録する (Join)。クライアントはサービスプロバイダを使用しようと思う時、Lookup サービスから探して、そのプロキシをダウンロードする。それで、クライアントとサービスプロバイダはプロキシを介して通信することで実現できる。

例えば、プリンタとコンピュータを繋ぐ時、プリンタが Lookup サービスを発見して、自サービスのプロキシオブジェクトを Lookup サービスに登録する。そして、コンピュータは Lookup サービスからプリンタを発見して、そのプロキシをダウンロードする。それで、コンピュータはプロキシを介してプリンタと通信し、プリンタを利用する [7]。

Jini では RMI [8] にて家電製品を連携させるということで、家電同士のやり取りはオブジェクトを使って行われる。しかし、携帯電話で RMI が使用できないので、今回の開発に適していない。

2.3 UPnP

UPnP とは、家庭内のパソコンや周辺機器、AV 機器、電話、家電製品などの機器をネットワークと通じて接続し、相互に機能を提供しあうための技術仕様である。

に UPnP の仕様や接続手順について詳しく記述している。

接続の手順は以下のようになる。

1. アドレッシング

アドレスがないと転送が始まらない。

各機器は DHCP または Auto-IP で IP アドレスが振られているものとする。

2. ディスカバリー

ネットワーク上のデバイスの検出を行なう。サーバーからでも、クライアントからでもできる。

クライアントの場合、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) の送出を行ない、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。

サーバーの場合、コントロールポイント側からマルチキャストパケット (M_SEARCH 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。

3. デスクリプション

接続されたデバイスが提供できる情報を記述した XML ファイルである。

デバイス自身が持っているサービスなどが書いているデバイスデスクリプションとサービスが持っている Action などが書いているサービスデスクリプションの 2 種類がある。仕様を作成するにはこの部分を良く理解する事が重要である。

4. コントロール

コントロールには 2 種類ある。

サービスが持っている機能呼び出す Action と、デバイスの状態変数を問合せるクエリーである。

これらのメッセージは、XML によって記述された SOAP (ソフトウェア同士がメッセージ (オブジェクト) を交換するためのプロトコル) が使われる。

5. イベント通知

接続されたデバイスに対して、特定の状態変数を指定してイベント購読を要求すると、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。

6. 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。

[9]に UPnP のデバイスが提供できる情報の例について詳しく記述している。携帯から家電をコントロールするために、まずはその家電は何者か、どんな機能を持っているか、どのように使うかについて知る必要がある。それを家電の方からダウンロードしてから、家電を使うことになる。

2.4 URC (Universal Remote Console)

URC[10]は INCITS/V2 技術委員会によって開発された規格である。URC は携帯電話や PDA 等にソフトを組み込んで、様々な機器を操作可能にする端末の名称でもある。機器との通信には無線 LAN や Bluetooth を使い、制御情報は XML でやり取りする。

URC では、以下の 4 つのファイルを用いて制御情報をやり取りしている。

表 2-1 URC の定義ファイルの種類

ファイルの種類	概要
ユーザーインタフェースソケット ディスクリプション	操作する機器の機能や状態を表す
プレゼンテーションテンプレート	UI を構築するために必要な構造やヒントを提供する
ターゲットディスクリプション	機器に接続するために URC に必要な情報を提供する
リソースディスクリプション	ラベル、ヘルプ、アクセスキーとキーワードを含む UI を構築するために必要なコンポーネントのリソースを 提供する

2.5 HAVi (Home Audio Video Interoperability)

HAVi[11]とは、グルンディッヒ、日立、松下、フィリップス、シャープ、ソニー、トムソン、東芝の 8 社によって策定された情報家電のためのミドルウェアである。HAVi は、IEEE1394 を使って、AV 機器をメーカーや機種にとらわれることなく、相互に接続するための共通基盤である。

HAVi では SDD(Self Describing Device)データと呼ばれる HAVi 機器としてのプロフィールや制御情報を含んだデータが Configuration ROM に置かれる。SDD データに含まれる情報は以下の 4 つである。

表 2-2 HAVi で取り扱われる情報とその概要

情報	概要
HAVi Device Profile	メーカーなどの機器全体に関する情報と、機器に含まれる機能に関する情報を定義
HAVi User Preferred Name	HAVi に準拠する機器ではユーザーが任意に機器に名前を付けることができる。この名前に関して定義
HAVi DCM, HAVi DCM Profile, HAVi DCM Reference	制御プログラムとその属性情報, URLなどを定義
HAVi Device Icon Bitmap	HAVi 機器を表すアイコンのデータ

第3章 システム構成及び開発計画

本プロジェクトは2段階がある。まず、様々な家電に汎用に使えるものを作り、次にそれを使って実際の家電の音声応答を実証する。

図 3-1 はシステムの概要を示す。端末は仕様を取得して解析することで、UI を作成する。そして、家電と通信することで家電の制御を実現する。

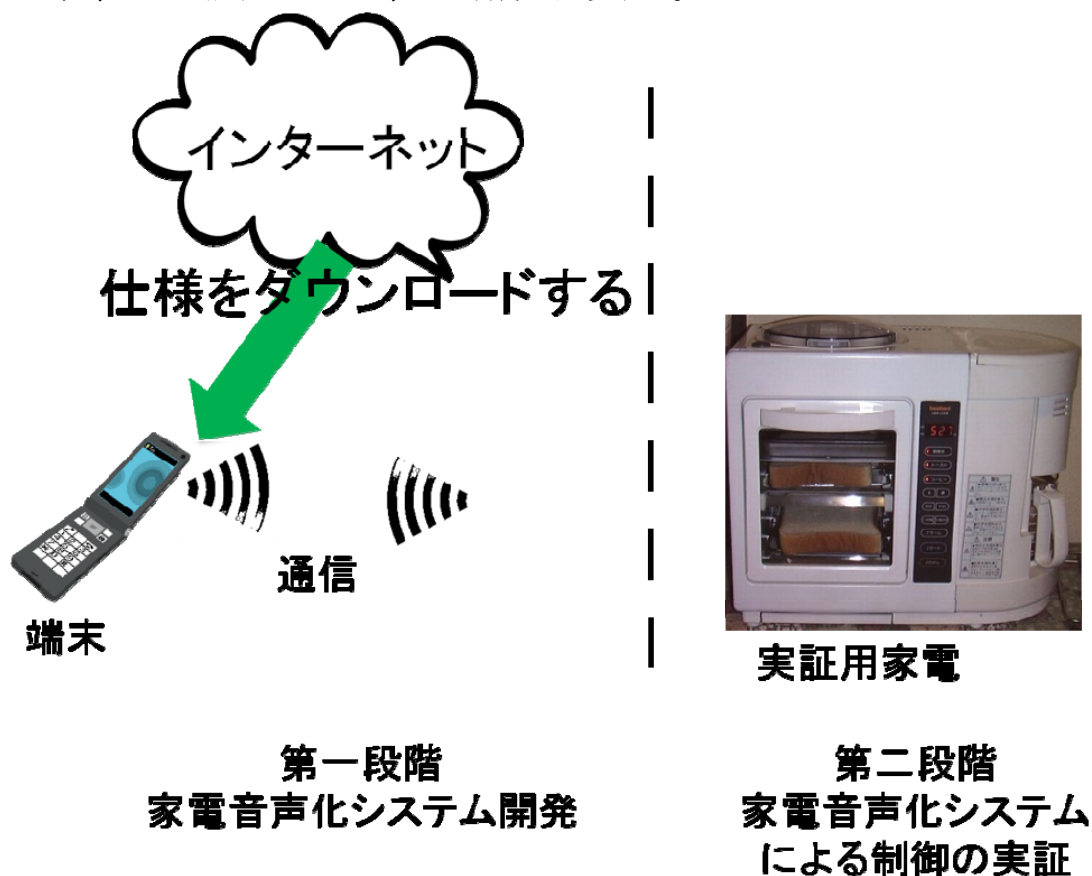


図 3-1 システム概要

3.1 機能要件

図 3-2 は家電と通信用端末に搭載する機能を示す。

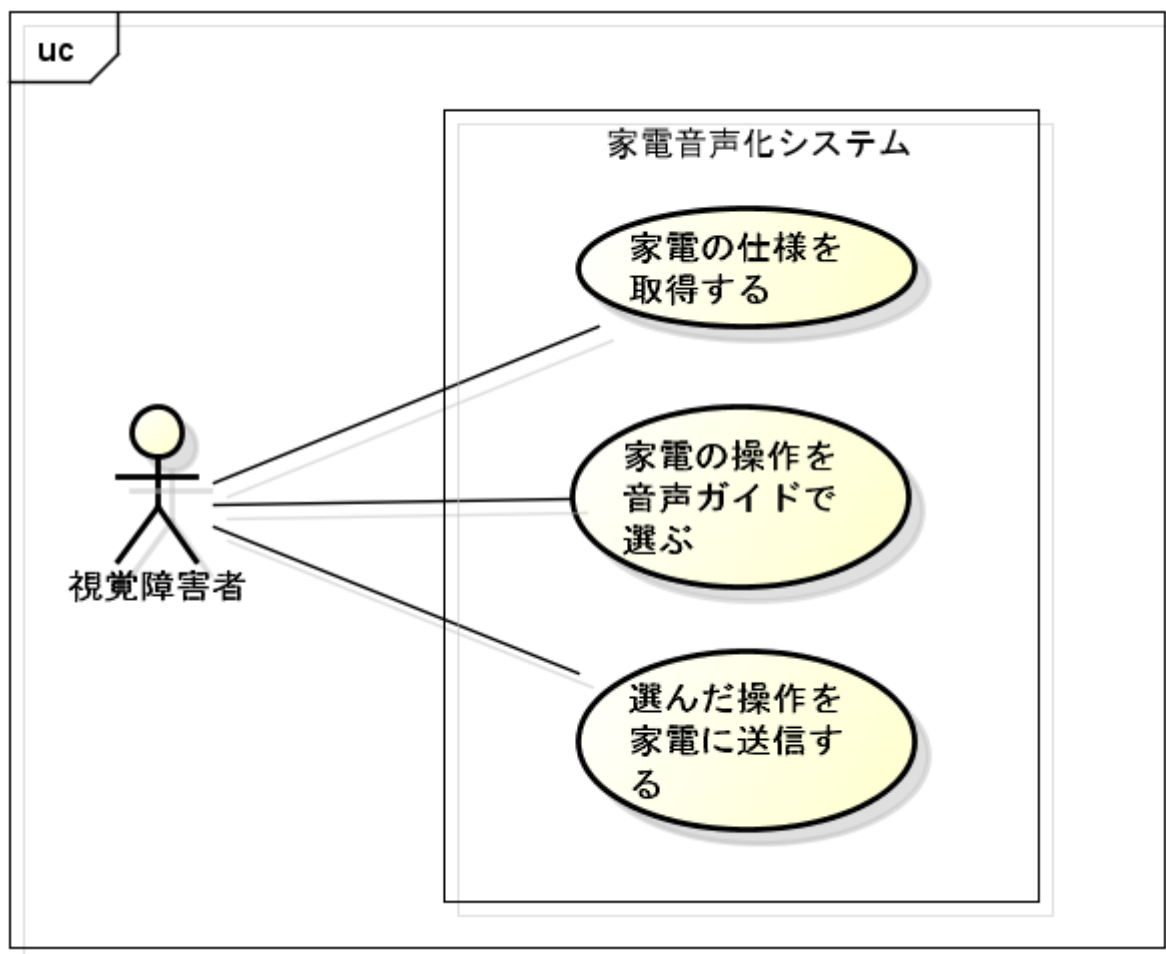


図 3-2 ユースケース図

■ 家電の仕様を取得する

視覚障害者は家電と通信用端末を用いて家電の仕様をダウンロードする。家電と通信用端末は家電の情報を取得してインターネットから仕様を選んでダウンロードする。以下の 2 つの機能から構成される。

➤ 仕様取得機能

インターネットからファイルの一覧を取得して表示する。視覚障害者は選んだ仕様をダウンロードする。

➤ 仕様保存機能

取得した仕様を端末に保存する。

■ 家電の操作を音声ガイドで選ぶ

視覚障害者は携帯を利用して家電を操作する、以下の 2 つの機能から構成される。

➤ UI 生成機能

ダウンロードした仕様を読み込んで、解析して、画面を作成する。画面の初期表示コンポーネント間の制約も実現する。

- 音声読み上げ機能
画面の項目内容を視覚障害者に理解できるように読み上げる。
- 選んだ操作を家電に通信する
ユーザが選択した操作をコマンドに変換して家電側に送信する、以下の 2 つの機能から構成される。
 - コマンド作成機能
情報家電に送信するコマンドを作成する。画面の入力値を仕様からとったコマンドフォーマットに従って変換する。
 - 家電に送信する
作成したコマンドを家電に送信する。

3.2 非機能要件

- 機能性
家電の操作は全て携帯上で実現できる。
- 移植性
いろいろな家電にも対応できる移植性を持たせる。家電と通信用端末側では、特定の家電に専用アプリを作るのではなく、解釈プログラムを開発する。家電の仕様記述を取得し UI とコマンドのフォーマットを解釈できるようにする。仕様記述を変えれば、他の家電にも簡単に対応できる。
家電側の通信モジュールに関しては、使用リソースを抑え、様々な家電のマイコンに移植できるようにする。
- 信頼性
家電が誤動作しないように、通信プロトコルを十分考えて設計する。通信内容の欠落を克服し、かつ、克服不能な欠落が生じても誤動作しないように設計する。
- 操作性
視覚障害者にもわかりやすい音声ガイドがある。
また、家電上における面倒な操作を軽減できる。例えば、時間を入力しようと思う時、多くの家電は入力キーボードを持っていなくて、ボタンを 1 つずつ押すことで、時か分を 1 つずつ変えている。設定したい時間に辿り着くのは、同じボタンを何十回押さないといけないので、かなり面倒なことになってしまう。本システムでは、入力画面に数字などの情報を直接入力することで、操作は便利になる。

3.3 機材の選定

3.3.1 端末の選定

下記の2つのメリットがあるので、らくらくホンを選定した。

- 視覚障害者の間でシェアが高い、目の不自由の方は簡単に入手できる。92%の視覚障害者は携帯電話を使っている、NTT DoCoMo らくらくホンシリーズのシェアは、79% [12]で、他の端末を圧倒している。
- 音声読み上げ機能がついていて、ユーザープログラムから発声可能である。auにも簡単ケータイを販売しているが、ユーザープログラムから発声することができない。

NTT ドコモの携帯電話上で動作するユーザアプリケーションは i アプリ (i-appli) [13] と呼ばれる。i アプリは DoJa [14] というプロファイルに書かれた Java アプリケーションである。

DoJa-5.1 プロファイル向け i アプリ開発ツール (i-appli Development Kit for DoJa-5.1) は、DoJa-5.1 プロファイル向け i アプリの開発をサポートする。i アプリのビルドから PC 上でのエミュレートまでをサポートする。Eclipse と連動させて、i アプリの作成、実行、デバッグが簡単にできる。

i アプリで音声合成機能を実現するには SpeechSynthesizer クラスを用いる。これは DoJa-5.1 からサポートされている [15]。従って、本研究では、らくらくホンのうち以下の5機種を対象とする。表 3-1 は現在対応できる機種である。

表 3-1 DoJa-5.1LE プロファイルに対応した機種

製品名	型式番号	Java プロファイル
らくらくホンプレミアム	F884i	DoJa 5.1LE
らくらくホンベーシックⅢ	F-08C	DoJa 5.1LE
らくらくホンV	F884i-ES	DoJa 5.1LE
らくらくホン6	F-10A	DoJa 5.1LE
らくらくホン7	F-09B	DoJa 5.1

3.3.2 通信方式の選定

表 3-2 は一般的な通信方式とらくらくホンの対応状況である。

表 3-2 通信方式一覧

通信方式	端末対応	価格
IEEE802.11	未対応	
Bluetooth	未対応	
パケット通信	対応	高価
赤外線通信	対応	安価

らくらくホンで利用できる通信手段はパケット通信[16]と赤外線通信[17]である。パケット通信により家電へアクセスする場合はインターネット経由の通信となるため、家電側に TCP/IP スタックとグローバル IP が必要となっており、設備も高価である。そのため、今回の開発に適していない。それで、赤外線通信方式を選定する。

らくらくホンがサポートしている赤外線の通信方式は表 3-3 に示す。

表 3-3 赤外線通信方式

赤外線通信方式	通信	速度	価格
CIR (Consumer IR)	単方向	低速	安価
IrDA	双方向	高速	やや高価

ここで、制御対象の家電が 2 種類あることに注意する必要がある。オーブンレンジ、洗濯機など多くの家電はメニューの選択結果を転送してスタートされればよいので、単方向通信で間に合う。しかし、HDD レコーダーなどは双方向通信が必要である。

白物家電はコスト制約が大きいので、CIR を採用する。情報家電等、双方向通信は必要なものは IrDA[18]を採用する。例えば、HDD レコーダーを操作する場合、録画済み番組リストを表示してから再生対象を選択する。HDD レコーダーにコマンドを送信するだけではなく、HDD レコーダーは番組リストを携帯に送信しなければならない。

おまかせミールさんは白物家電なので、CIR を採用する。

3.3.3 家電用マイコンの選定

家電用マイコンとは、携帯と通信するための通信ボードのマイコンである。下記の理由で Microchip 社の PIC24FJ64GA002（本番の環境のマイコン、28 ピン）と PIC24FJ128GA010（試作環境のマイコン、100 ピン）を選定した。

- ① 赤外線通信方式である CIR と IrDA 両方利用できる。
- ② 高レベルのプロトコル IrOBEX で通信できるライブラリを利用できる。
- ③ 安価である。

3.3.4 実証用家電の選定

実証用家電を検討する時、下記の 2 つのポイントを満たさないといけない。

- 音声案内するため、ある程度複雑なメニューを持つ必要がある。家電の操作機能は単純すぎると、音声案内を追加してもあんまり意味がない（なくても使える）。
- 集積度が適切、改造可能である

表 3-4 は検討した家電を示す。

表 3-4 検討家電一覧

実証用家電	利点	問題点
マイコン内蔵オープンレンジ	多数の調理メニューを持つ	基板の集積度が高すぎ、改造が不能
洗濯機	洗濯メニューを持つ	基板の防水処理（ポッティングという）により加工が困難
照明	改造は簡単	単純すぎ
扇風機	改造は簡単	単純すぎ

それで、イワタニ「おまかせミールさん」は複雑な機能を持ち、集積度も低いので、「おまかせミールさん」を選定した。

家電音声化システムの構築後、これを用いて携帯から実際の家電が制御できることを実証した。

3.4 全体設計

家電音声化システムの構成を、以下の図 3-3 に示す。

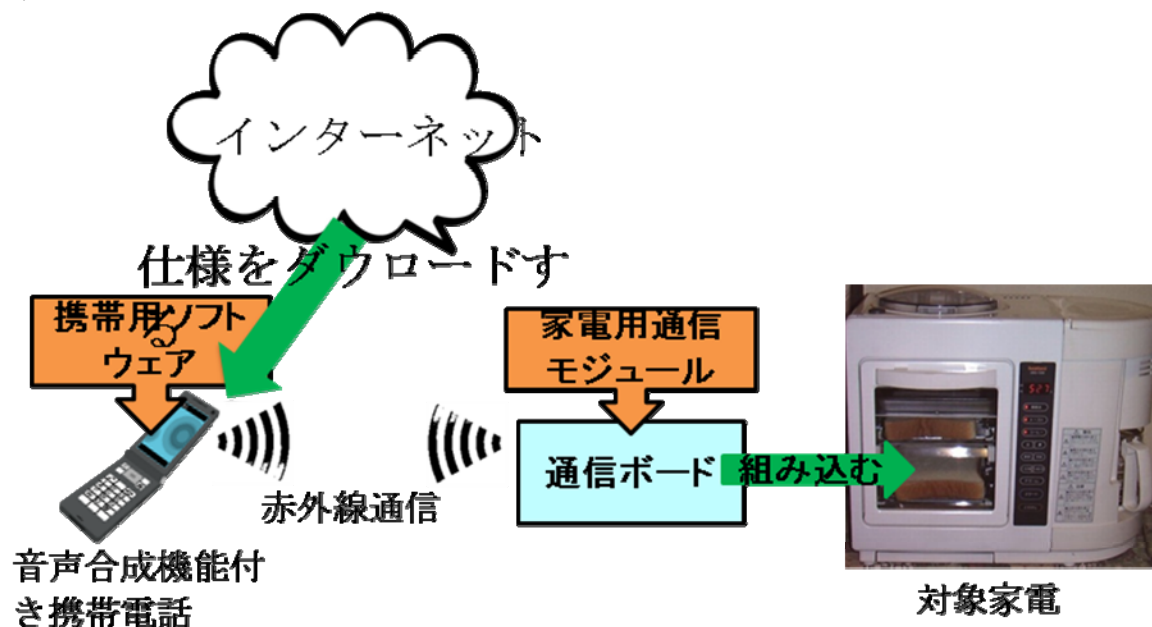


図 3-3 システム構成図

家電音声化システムは、携帯側に搭載する携帯用ソフトウェアと家電側に搭載する家電用通信モジュールで構成する。

まず、携帯はインターネットから記述仕様をダウンロードする。この仕様は UI 仕様と制御仕様からなる。UI 仕様の解釈結果に基づいて、画面描画機能と音声読み上げ機能を利用することで、音声付き操作画面を作成させる。ユーザの入力を制御仕様に従ってエンコードして通信ボードに送信する。

家電側では、通信ボードとおまかせミールさんの LED/SW Board、Mother Board を繋がる。通信ボードを受け取った操作のボタン入力をエミュレーションして、家電を制御する。

3.5 開発体制

表 3-5 は全体の役割分担を示す。

表 3-5 役割分担一覧

名前	役割
持田 辰弥	リーダー
徐 偉	サブリーダー
章 程	ドキュメント管理
劉 俊鵬	進捗管理

リーダーの持田は、プロジェクト全体の進め方を把握し、プロジェクトの目的を達成する。筆者は持田をサポートするほか、連絡事項の周知、会議の招集も担当していた。成果物（文書）の管理、議事録は、章が担当している。劉は進捗管理を担当している。週ごとに進捗を把握して、遅延があった場合には、遅れた情報を本人に通知して、改善を促進する。

図 3-4 は全体の仕事分担を示す。

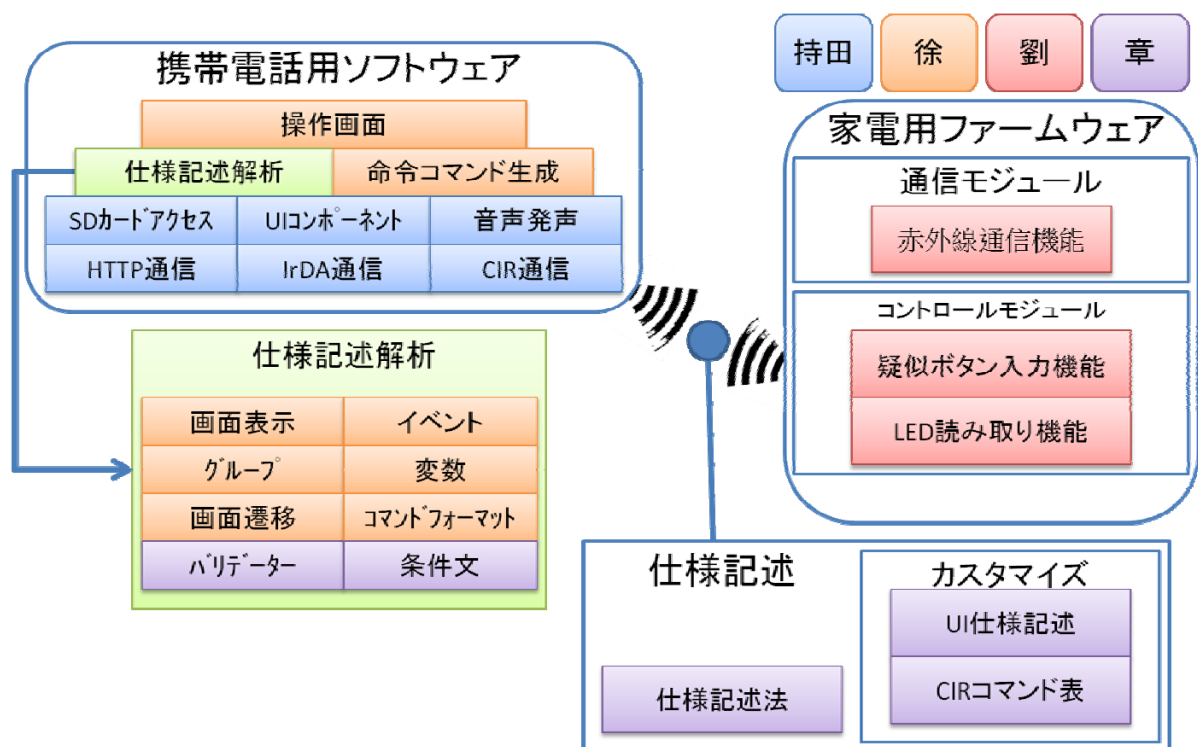


図 3-4 仕事分担

3.6 スケジュール

開発スケジュールについて、携帯側と家電側が違う。図 3-5 と表 3-6 は携帯側開発スケジュールを示す。図 3-6 と表 3-7 は家電側開発スケジュールを示す。

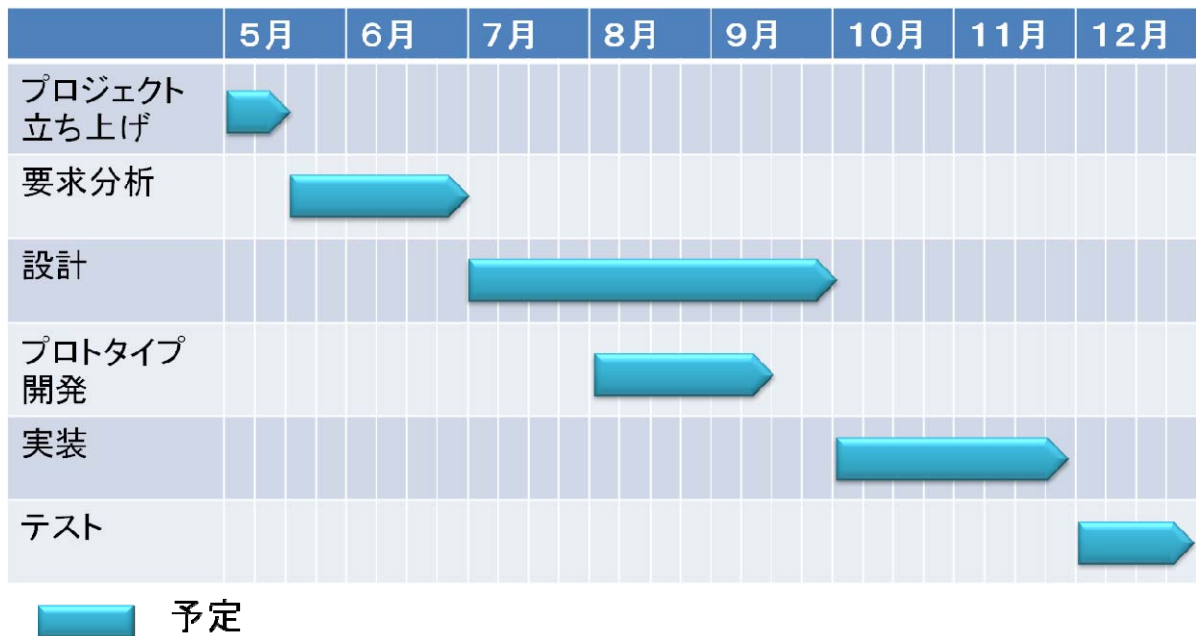


図 3-5 携帯側開発スケジュール

表 3-6 携帯側開発スケジュール

工程	予定期間
プロジェクト立ち上げ	2011 年 5 月 1 日～2011 年 5 月 15 日
要求分析	2011 年 5 月 16 日～2011 年 6 月 30 日
設計	2011 年 7 月 1 日～2011 年 9 月 30 日
プロトタイプ開発	2011 年 8 月 1 日～2011 年 9 月 15 日
実装	2011 年 10 月 1 日～2011 年 11 月 30 日
テスト	2011 年 12 月 1 日～2011 年 12 月 31 日

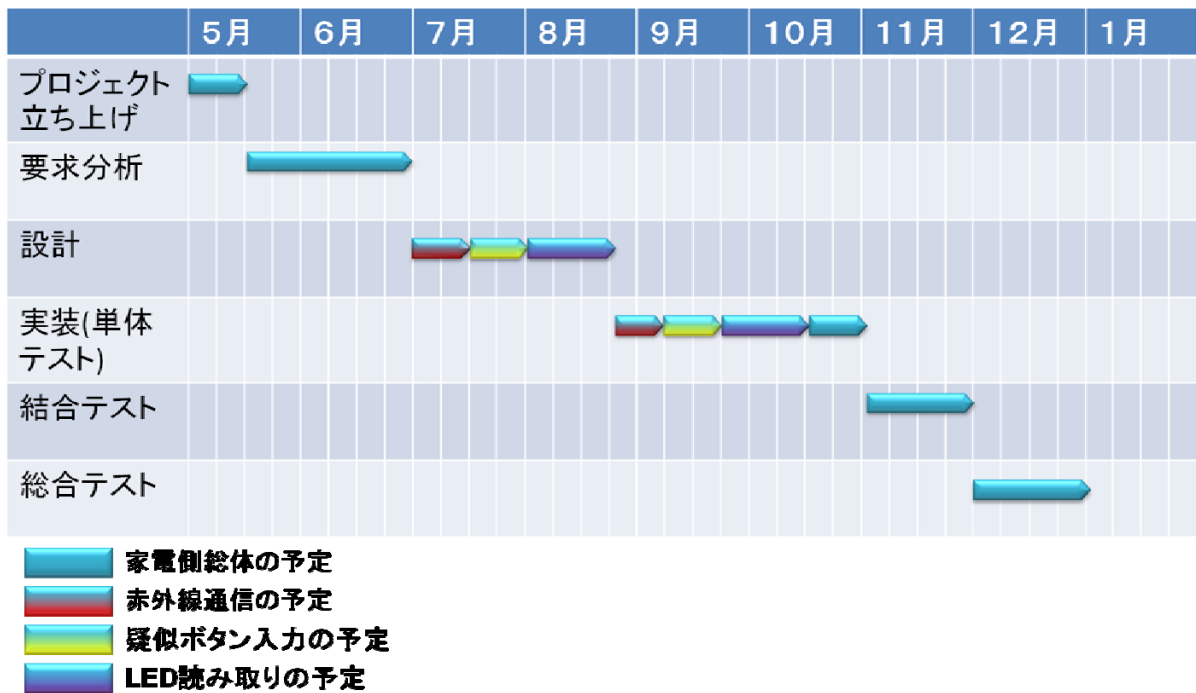


図 3-6 家電側開発スケジュール

表 3-7 家電側開発スケジュール

工程	予定期間
プロジェクト立ち上げ	2011 年 5 月 1 日～2011 年 5 月 15 日
要求分析	2011 年 5 月 16 日～2011 年 6 月 30 日
設計	2011 年 7 月 1 日～2011 年 8 月 21 日
実装（単体テスト）	2011 年 8 月 22 日～2011 年 10 月 31 日
結合テスト	2011 年 11 月 1 日～2011 年 11 月 30 日
総合テスト	2011 年 12 月 1 日～2011 年 12 月 31 日

第4章 家電のリバースエンジニアリング

家電音声化携帯に家電の仕様で読み込ませることで、様々な家電に対応できるシステムにしたい。このためには、携帯が取得する家電の仕様にどのような形式で書くべきか、どのような情報を含めるべきかえ、決めなければならない。しかし、最初から汎用的な設計をするのは困難であり、おまかせミールさんを事例として、汎用の記述を決定した。

以下、4章ではおまかせミールさんの機能解析である。

5章では記述形式の調査と検討である。

6章では簡単な事例を元に仕様記述方法を検討する。

4.1 基本機能

卵焼き、トースト、コーヒーを作る機能があり、設定時刻に同時に仕上がる。卵焼きとトーストは最大2個（枚）までセットでき、焼き加減の濃淡を選択できるほか、調理時間を10秒単位で指定することもできる。

複数のメニューを選択して調理することもできる。

アラーム機能と予約機能もある。

4.2 制約条件

卵焼きとトーストの濃淡は個別に設定できない。

時計の設定がされていない時は予約、アラームが設定できない。

調理時間の設定と予約は同時に選べない。

複数のメニューを選択して調理する場合、調理時間の設定はできない。

4.3 機能一覧

基本機能と制約事項をまとめて表 4-1 に示す。
左側には選択したメニュー、右側には選択されたメニューにおける機能。

表 4-1 基本機能と制約事項一覧

卵焼き	トースト	コーヒー	数	濃淡	予約	調理時間
●			○	○		
●					○	
●						○
	●		○	○		
	●				○	
	●					○
		●			○	
●		●	○	○		
●		●			○	
	●	●	○	○		
	●	●			○	
●	●	●	○	○		
●	●	●			○	

4.4 GUI プロトタイプ作成

図 4-1 に基づき、GUI プロトタイプを作成した。GUI プロトタイプを利用することで、アプリケーションを開発する時、制約事項と作成したコマンドの正確性を簡単に調べることができる。

設定区分は調理、時刻設定、アラーム設定から選択させる。

依存関係に関して、トースト、卵焼き、コーヒー、予約のチェックボックスは調理を選択した時のみ有効とする。予約のチェックボックスがチェックされていない時のみ調理時間を選択可能とする。

予約とアラームの値の範囲は、時は 1～12 である、分は 0～59 である。

調理時間の値の範囲は、分は 1～10 である、秒は 00～50、10 秒単位とする。

すべての選択が終わったら、下のコード表示ボタンを押すと、別途定めた CIR (Consumer InfraRed) コード表に従い、コマンド列を生成する。

今回プロトタイプを作成したことで、仕様記述に制約条件とグループ化を含めることがわかった。

Form1

設定区分 調理

トースト ☒ 数量 一枚

卵焼き ☒ 数量 一個

濃淡 普通

コーヒー ☒

予約 ☒

時間調整 ☐

調理時間調整

分 秒

時間設定

AM 7 時 19 分

コード表示

00100001 10000111 1110011 01100001 10000111 1110011 10111111 10000111 1110011

図 4-1 GUI プロトタイプ

第5章 仕様記述形式に関する調査

仕様記述を決める前、どのような記述形式が良いかを調べるため、仕様記述方式の調査を行った。

5.1 仕様記述形式の調査

おまかせミールさんの機能解析の結果により、仕様は実証用家電の基本機能と制約条件を含める。仕様の記述方式定めることを目的として、仕様記述形式の調査を行った。

5.1.1 Versit 形式

Versit 形式[19]はアップル、AT&T Technologies、IBM、シーメンスによって設立された Versit コンソーシアムによって提唱された電子名刺の標準フォーマットである。個人情報（氏名、電話番号、メールアドレスなど）をデータ化してアプリケーション間でやりとりすることができる。

Versit の記述は、「BEGIN: VCARD」から「END: VCARD」に囲まれた範囲が 1 つのデータを表す。各行は「(項目名):(値)」の形になっている。ただし、各項目に補足がある場合には、「(項目名);(オプション):(値 1):(値 2):(値 3)...」のような形になっている。

例えば、図 5-1 の例で、「ORG:Tsukuba University」とあれば、アイテム名が「ORG」で値が「Tsukuba University」である。そして、「EMAIL;PREF;INTERNET:joy@tsukuba.ac.jp」ならば、「EMAIL」が名前を表すアイテム名で続く「INTERNET」がオプションで、「joy@tsukuba.ac.jp」が値の 1 つ目となる。

```
BEGIN: VCARD
VERSION:2.1
ORG:Tsukuba University
NAME:Joy
TEL;WORK;VOICE:(111) 1111-111
TEL;HOME;VOICE:(222) 2222-222
EMAIL;PREF;INTERNET:joy@tsukuba.ac.jp
... ..
END: VCARD
```

図 5-1 Versit 形式の例

5.1.2 XML 形式

XML[20]とは、拡張可能なマーク付け言語 (eXtensible Markup Language)。XML により統一的な記法を用いて独自の意味や構造を持ったマークアップ言語を作成することができるため、異なる情報システム間のデータ交換を容易になる。

XML 文書は、内容を要素名(element)の中に記述する。開始タグは「<要素名>」、終了タグは「</要素名>」で記述する。また、要素は内部に子要素を含むことができる。

例えば、図 5-2 の例で、要素 students の中に子要素がある。子要素の開始タグと終了タグ

の間に内容を記述する。子要素の情報を組み合わせて1つの学生情報(student)になる、学生情報を全部集めると、全学生(students)情報になる。

```
<students>
  <student>
    <stuId>0000001</stuId>
    <major>システム情報</major>
    <name>筑波太郎</name>
  </student>
  ...
</students>
```

図 5-2 XML 形式の例

5.1.3 JSON 形式

JSON[21]とはJavaScript Object Notationの略である。JavaScriptにおけるオブジェクトの表記法をベースとした軽量のデータ交換フォーマットである。

オブジェクトは{}で全体を囲み、キーと値のペアをコロン(:)で区切って記述する。カンマ(,)で複数のキーと値を記述することも可能である。

JSONはJavaScriptのオブジェクト表記構文のサブセットとなっており、XMLと比べると簡潔に構造化されたデータを記述することができるため、記述が容易かつ理解しやすいメリットを持っている。

XMLには閉じタグを使用するが、JSONの場合カッコに対応する。JSONの記述方式は簡潔で、可読性は高い。図 5-3 の例は記述方式を示す。

```
{
  {
    "stuId" : " 0000001" ,
    "major" : " システム情報" ,
    "name" : " 筑波太郎" ,
  },
  ...
}
```

図 5-3 JSON 形式の例

5.2 解析方式の調査

正規表現[22]とトークナイザを利用して解析しようと思うので、正規表現とトークナイザの調査を行った。

ネイティブな正規表現パッケージとして `java.util.regex` をサポートしている[23]が、Doja 側には `regex` を含めていないので、利用できない。

次は「字句解析」のための文字列操作に関して、Java では、`java.util` パッケージに `StringTokenizer` という便利なクラスがあるので、これを使うと簡単にソースコードを字句へ分解できる。しかし、このメソッドにも Doja 側に含めていないので、利用できない。

結果として、既存のメソッドは利用できないので、字句解析は自分のプログラムで実装するしかない。

5.3 仕様記述方式の検討

字句解析は自分のプログラムで実装するため、結構たいへんになる。調査した記述方式を Java オブジェクトに変換することが出来れば、容易になる。そして、各記述方式は Java オブジェクトに変換の可能性を調査した。

まず、JSON を Java オブジェクトに変換するライブラリについて、すでに多数存在するが、代表的なライブラリは `JSONIC`[24]、`Json-lib`[25]、`google-gson`[26]である。いずれも Doja に対応していない。

XML で記述されたファイルを解釈するには `DOM`[27] (Document Object Model) や `SAX`[10] (Simple API for XML Parsing) を用いる。DOM は XML 文書中のデータを XML パーサが読み込むと、データをツリー構造としてオブジェクトに保存する。そして、このツリー構造のオブジェクトに対し、様々な操作が簡単にできる。一方、XML 文書を先頭から順番に一行ずつ読んでいき、要素が出現するたびに処理を実行していく。SAX は自由に構造をたどれない為に、XML 文書の構造が複雑になると、プログラムが複雑になってしまう。いろいろな家電の記述仕様（特に複雑な記述仕様）を対応できるため、DOM には今回の開発に適していると考えているが、調べた結果、Doja 側にも使えない。

他の XML を Java オブジェクトに変換するライブラリにも調べた。たくさん存在している (`JAXB`[28]とか `Commons Digester`[29]等) が、Doja に対応できるものはない。それで、より解析しやすい記述方式を選ぶしかない。JSON と XML は記述方式が多少違うが、実質は一緒である。XML、JSON とともに Java のライブラリがあり、使えるものがあれば、それを採用すればよいが、Doja 側にライブラリがない。Versit は携帯同士の電話帳やスケジュールの転送に使われており、項目名と値を容易に分割できることで、解析しやすいと考えるので、Versit 形式を選定した。

第6章 仕様記述の検討

最初から汎用的な設計をするのは困難であり、おまかせミールさんを対象にして、汎用の仕様記述を決定した。プロトタイプを開発しながら、汎用の仕様記述を検討する。

家電と携帯を連携させるためにUI仕様と制御仕様の形式を定めなければならない。

仕様記述の実現可能性を検証するため、検証用プログラムを作成した。また、仕様記述を定める時、設計者の視点だけから設計すると、実装困難な又は効率よくない状況も起こりうる。それで、仕様記述を定めながら検証プログラムを開発する方法を採用した。

6.1 簡単な事例の試作

おまかせミールさんの画面項目と制約条件は複雑なので、より簡単な事例（ラジオとラジオカセットレコーダー）を試作してみた。

実行した時の画面は図 6-1 に示す。仕様記述を解析して UI 部分を作成することだけではなく、コンポーネント間の制約も正しく動いている。例えば、ラジオは AM、FM どちらも受信できる、カセットモードは再生、停止、録音が行える、再生、停止、録音は同時に選べない、AM または FM 受信の時、再生はできない。

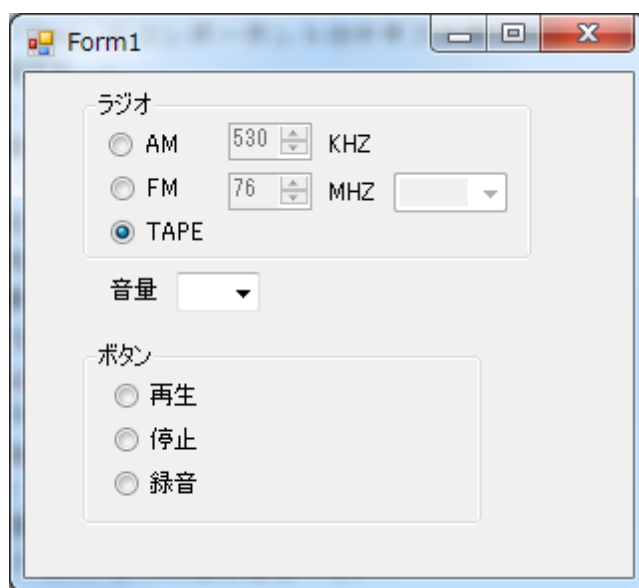


図 6-1 ラジオカセットレコーダーの実行画面

6.2 UI 仕様の記述と解析の検証

i アプリの Panel クラスを参考して、サポートすべきコンポーネントとしては下記を用意する。

- ラベル
- ボタン
- テキストボックス
- ドロップダウンリスト
- チェックボックス
- ラジオボタン

各コンポーネントには下記の属性を持たせる。

- コンポーネントの種類
- 座標
- 名称
- サイズ
- テキスト
- 最大値と最小値
- 初期値
- グループ
- アクション

6.3 制御仕様の記述と解析の検証

制御仕様では、画面各項目間の制約条件を Action に記述する。制約条件の記述位置は 3 箇所だと考えられる。

- イベントを発生するコンポーネントの中
- イベントの影響を受けるコンポーネントの中
- 仕様記述の最後にまとめて記述

イベントを受けるコンポーネントの中に記述方式と、最後にまとめて記述する方式について、イベントを発生した時、記述する順番によりイベント処理の順序が異なる。そのため、イベントを発生するコンポーネントの中に記述する必要がある。また、イベントを正しく発生させるため、条件式はその条件式の左辺に現れるコンポーネントすべての Action 属性に書いておかなければならない。

ラジオカセットレコーダーの制約条件は単純すぎて、他の機能が複雑な家電に適用できないかもしれないので、実際に使う可能性がある条件式にも考慮して実装した。

複数の条件式 (AND 又は OR 関係) から複数のコンポーネントを制御することや $A \rightarrow B$ 、 $B \rightarrow A$ の矛盾する式にも対応しなければならない。 $A \rightarrow B$ 、 $B \rightarrow A$ の矛盾式は、A の変化を B に影響を与えると同時に、B の変化にも A に影響を与えて、無限ループに入ったことである。

実行した画面は図 6-2 に示す。左側は OR 関係を表す。2 つのチェックボックスを 1 つチェックされたら、下の複数のテキストボックスが使えるようになる。

右側は AND 関係を表す。2 つのチェックボックスを両方チェックされないと、下のテキストボックスが使えない。

一番下は $M \rightarrow N$ 、 $N \rightarrow M$ 矛盾する式を表す。どちらのチェックボックスがチェックされると、他のチェックボックスもチェックされる。

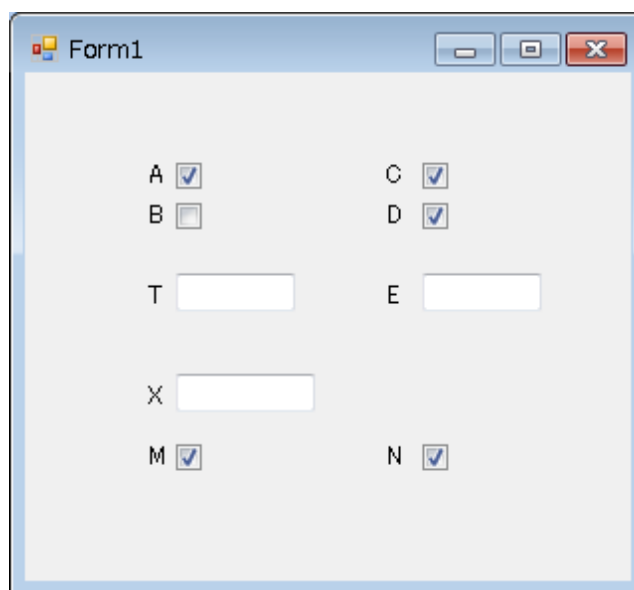


図 6-2 複雑な制約条件の実行画面

6.4 赤外線エンコード部分の記述方式と解析の検証

下記の例のようなフォーマットを選定した。固定されるコードの部分には直接に記述する。“#”で他の部分と繋がることを表す。変数の部分には“{”と“}”で囲んで表す。この中に2つの部分から構成される。前の部分は変数であって、後ろの部分はこの変数が占める桁数である。桁数は足りないと前に「0」を埋める。計算した結果は2進数で表示する。また、変数の部分は画面のコンポーネント名を記述する。

例：10#{hour:4}#00

将来おまかせミールさんの時刻設定画面を対応出来るため、試作画面は図 6-3 のように作った。時と分2を入力して、コード表示ボタンを押すと、作成したコードは下のテキストに表示される。

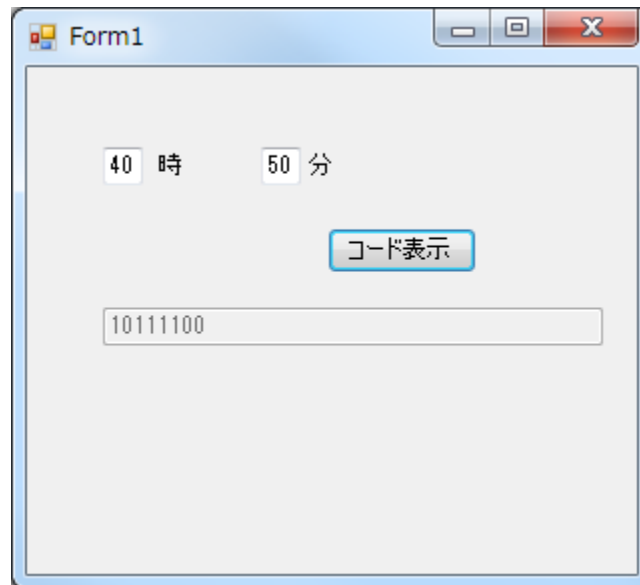


図 6-3 コマンド作成部分の試作画面

おまかせミールさんのコマンドフォーマットは簡単だが、他の家電は複雑なフォーマットを使う可能性もあるかもしれない。例えば、下記の例で、普通の計算式と括弧いくつを組み合わせる形式、更に括弧のネストもある。

例：{((A+B) / (C-4))*3 : 2}

このような形式を処理するため、逆ポーランド記法[24]で計算する部分を実装してみた。仕様記述を書く時は普通のような計算式を書いたが、プログラムは逆ポーランドの表記方式に変換して、計算を行う。

例：1#[(hour - minute / 2) * 2:6]#0

図 5-4 は上記のフォーマットのような計算を行って、「30」の計算結果を得た。

第7章 携帯電話用端末ソフトウェアの開発

4-6章の検討結果をもとに、定めた仕様記述を解釈し、UIを提示するソフトウェアを開発する。

7.1 携帯電話用ソフトウェアの構成

携帯用ソフトウェアの機能を2つのパッケージにわけ、uicomponent パッケージは UI コンポーネントを描画する機能で、その他の機能をまとめて remotecontrol パッケージに入れる。

表 7-1 は uicomponent パッケージの構成を示す。

表 7-1 uicomponent パッケージの構成

クラス名	概要
OComponent	各コンポーネントの親クラス
OGroup	グループ処理用
OLabel	ラベルの項目を描画
OButton	ボタンの項目を描画
OTextBox	テキストボックスの項目を描画
ODropDownList	ドロップダウンリストの項目を描画
OCheckBox	チェックボックスの項目を描画
ORadioButton	ラジオボタンの項目を描画

表 7-2 は remotecontrol パッケージの構成を示す。

表 7-2 remotecontrol パッケージの構成

クラス名	概要
MultiplepurposeRemoteControl	メインクラス
RemoteControlChanger	描画のためのクラス
UICreation	仕様解析と UI 作成、コマンド作成
SDMemoryCardAccess	SD カードアクセス用
HTTPConnection	HTTP 通信用
IrDAConnection	IrDA 通信用
TextToSpeech	読み上げ機能用
CIRConnection	CIR 通信用

図 7-1 は携帯電話用ソフトウェアのクラス図を示す。

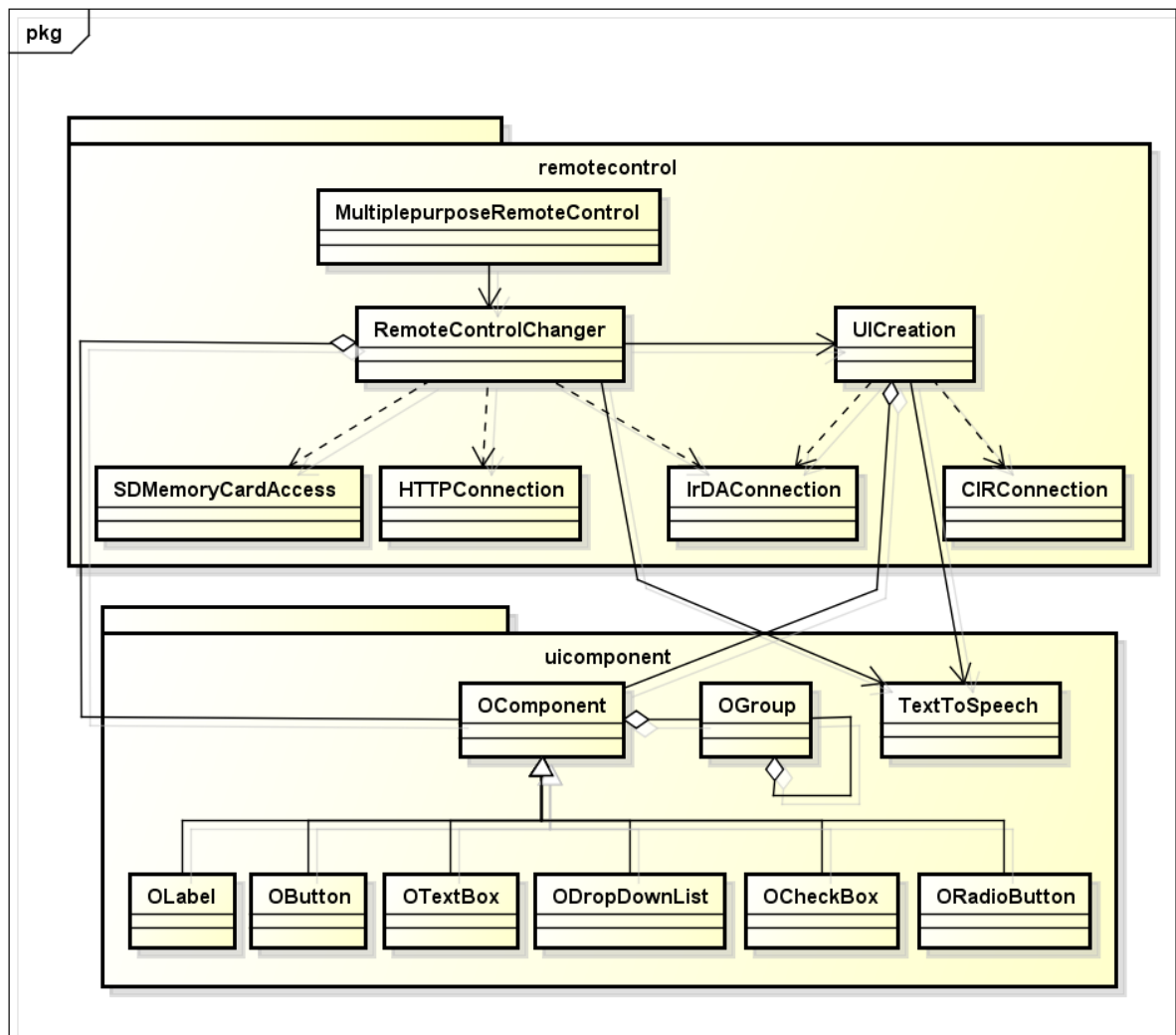


図 7-1 全体クラス図

7.2 仕事分担

画面描画するため、Doja は「Panel」と「Canvas」を用意している。Panel では、フォーカス移動する時、ユーザ側プログラムが取得できないから、音声の読み上げ処理をつけることができない。「Canvas」を利用する。「Canvas」は低レベル API で、コンポーネントが提供されていないので、描画からイベント処理まで作らないといけない。

筆者は仕様解析と UI 作成機能、コマンド作成機能の部分を担当している。持田はコンポーネント描画機能、仕様ダウンロード保存機能、家電と通信機能、読み上げ機能を担当してい

る。章は主に仕様の設計を担当していて、後半からは実装に加わった。入力内容のチェックと if 文の解析部分の実装を担当している。

7.3 担当部分のメソッド構成

表 7-3 は UICreation クラスのメソッド構成を示す。

表 7-3 UICreation クラスのメソッド一覧

メソッド	概要
init	仕様を解析してから画面表示のコンポーネントを作成する
getText	仕様記述のフォーマットに従って、表示用テキストを作成する
getNext	同じグループの次の項目を取得する
groupAdd	グループにオブジェクトを追加する
paint	「Canvas」クラスのメソッド、「Graphics」クラスで定義した画像を表示。全部コンポーネントを描画する。
processEvent	「Canvas」クラスのメソッド、低レベルイベント発生時に呼び出し。携帯電話機の「キー選択」に関するイベント。
getGroupName	グループ名を取得する
processEnt	アクション部分の処理
commandSend	2進数 String 型のコマンドを変換して送信する
chgBytes	Int から byte へ変換する
codeMake	家電送信用コマンド作成
CalcString	逆ポーランド記法で計算する
setParameter	変数の値を設定する
getStatus	コンポーネントの状態を取得する
groupSet	グループを作成する
statusSet	コンポーネントの状態を変更する
processIMEEvent	「Canvas」クラスのメソッド、文字入力後の処理
ifben	if 分解析用
checkIf	if 文の条件式の true、false をチェックする
nextIf	子 if 文の処理
split	文字列を分割する
splitCheck	分割時の境界チェック
isNumeric	数字チェック
CompSpeak	音声読み上げ機能を呼び出す
GetObjectById	コンポーネントの名前でオブジェクトを取得する

7.4 仕様記述の構成と解析、実現方式

仕様記述の構成は図 7-2 を示す。仕様記述は主に UI 記述と制御記述から構成される。

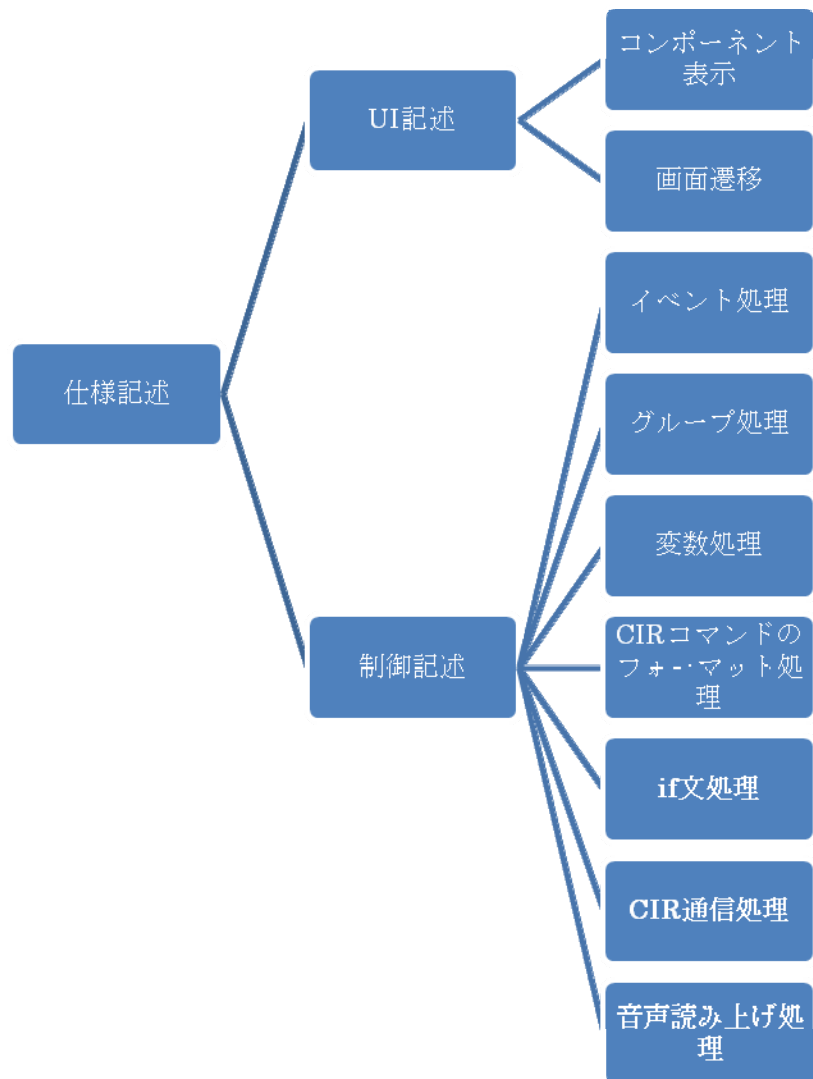


図 7-2 仕様記述の構成図

各機能の構成と解析方式を 7.4.1 から説明する。

7.4.1 コンポーネント表示

Versit 形式を採用したので、下記のように記述した。

コンポーネントは「BEGIN: コンポーネント名」から「END: コンポーネント名」に囲まれた範囲が 1 つのコンポーネントを表す。この中に、各行で、「(項目名):(値)」の形で記述する。そして、値が多い時、「(項目名):(値 1);(値 2);(値 3)…」のように「;」で区切って記述する。図 7-3 は例を示す。

```
BEGIN:PAGE
ID:1
TITLE:メインメニュー
SOUND:料理を作るか、時間を設定するか
BEGIN:BUTTON
  ID:make
  NAME:1. 料理を作る
  ENABLED:true
  SOUND:料理を作る
  ACTION:make.clicked->page=2
END:BUTTON
BEGIN:CHECKBOX
  ID:setTime
  ITEMS:料理を作る;時間を設定する
  ENABLED:true
  SOUND:時間を設定する
END:CHECKBOX
... ..

END:PAGE
```

図 7-3 コンポーネント表示の記述例

各属性名と概要を表 7-4 に示す。

表 7-4 仕様記述の属性名と概要一覧

属性名	概要
TITLE	ページのタイトル
ID	コンポーネントの識別子
NAME	コンポーネントの表示テキスト
ENABLED	初期値（使用可能／不可）
XPOSITION	X 座標
YPOSITION	Y 座標
SIZE	サイズ
ACTION	イベント
SOUND	読み上げる内容
ITEMS	リスト

表 7-5 はコンポーネントごとの属性の利用状況（可能か不可）を示す。

表 7-5 コンポーネントごとの属性の利用状況一覧

	Label	Button	TextBox	CheckBox	RadioButton	DropDownList
ID	○	○	○	○	○	○
NAME	●	●	—	●	●	—
ENABLED	○	○	○	○	○	○
XPOSITION	●	●	●	●	●	●
YPOSITION	●	●	●	●	●	●
SIZE	—	—	●	—	—	●
ACTION	—	○	○	○	○	○
SOUND	○	○	○	○	○	○
ITEMS	—	—	—	—	—	●

※ ●は必須、○は使用可能、—は使用不可

画面の各コンポーネント、ページなどを解析する時、処理フローは多少違うが、図 7-4 は体系的な処理フローを示す。

コンポーネントごとに仕様を読み込むため、2 つのループを設けた。外のループは全部の記述を終わりまで読む用、中のループは 1 つのコンポーネント属性を終わりまで読む用。まずは外のループから初めて、コンポーネントの始める部分と読んだら、もう 1 つのループに入れて、このコンポーネントの属性を一通り終わりまで読んで、属性を変数に一時保存する。1 つのコンポーネントの記述が読んだ後コンポーネントを作成する。

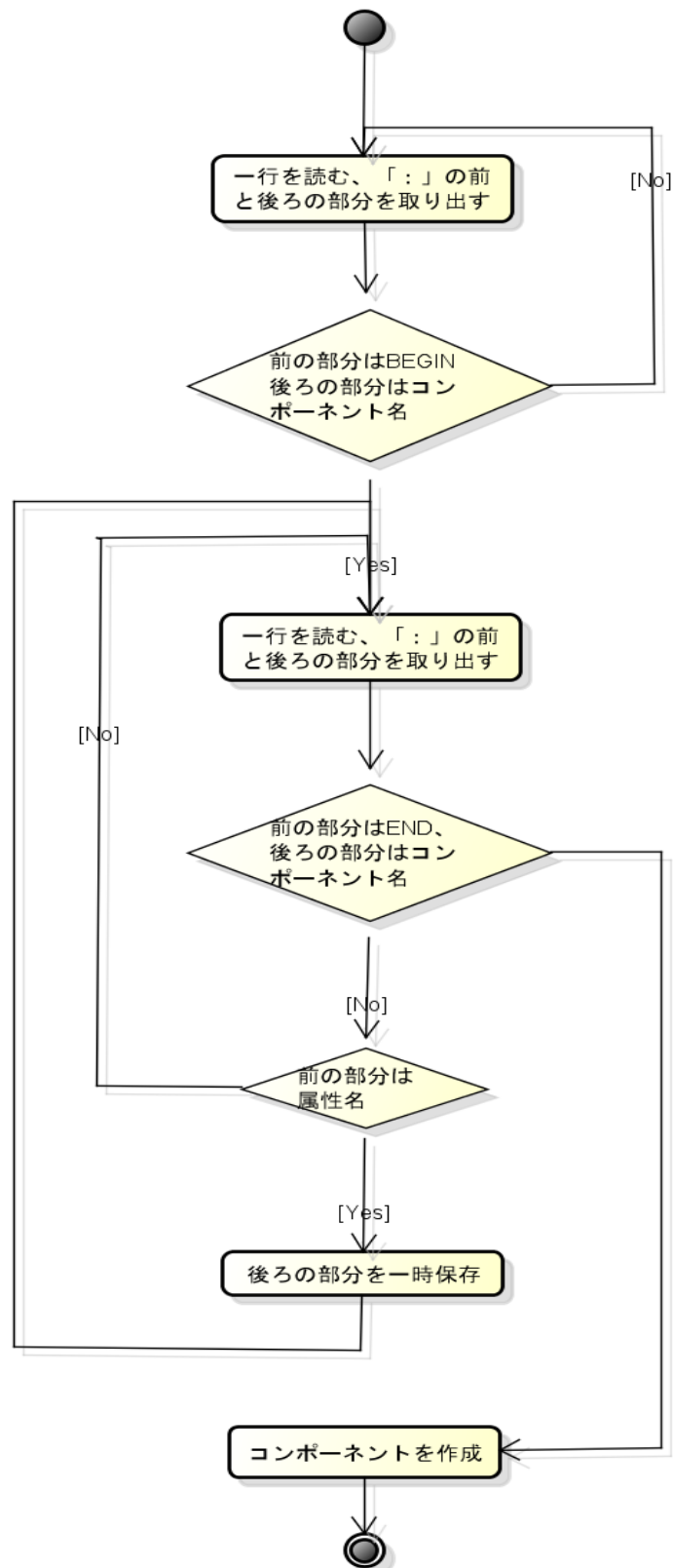


図 7-4 UI 表示部分の処理フロー

7-5 は仕様記述を解析してからコンポーネントを作成した例を示す。

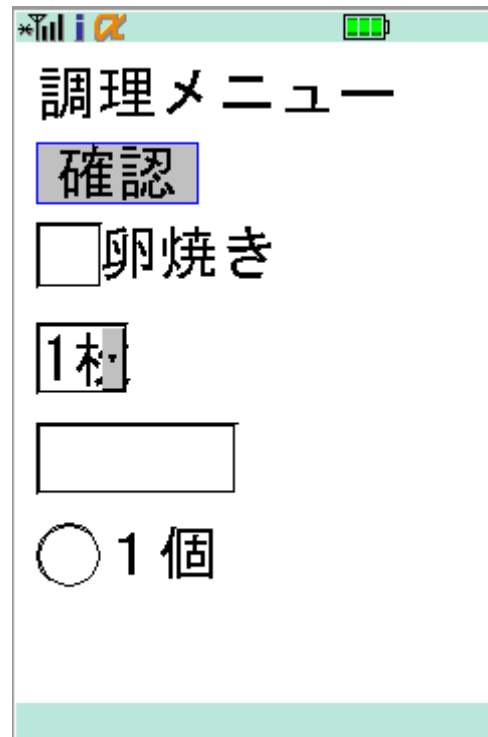


図 7-5 UI 表示部分の試作画面

面上表示されるコンポーネントはボタン、チェックボックス、ドロップダウンリスト、テキストボックス、ラジオボタンである。

7.4.2 画面遷移

図 7-6 は画面遷移部分の試作画面である。ボタンは3つある。「料理を作る」ボタンをクリックすると、調理メニュー画面に遷移する。「戻る」ボタンを押すと、メインメニュー画面に戻る。

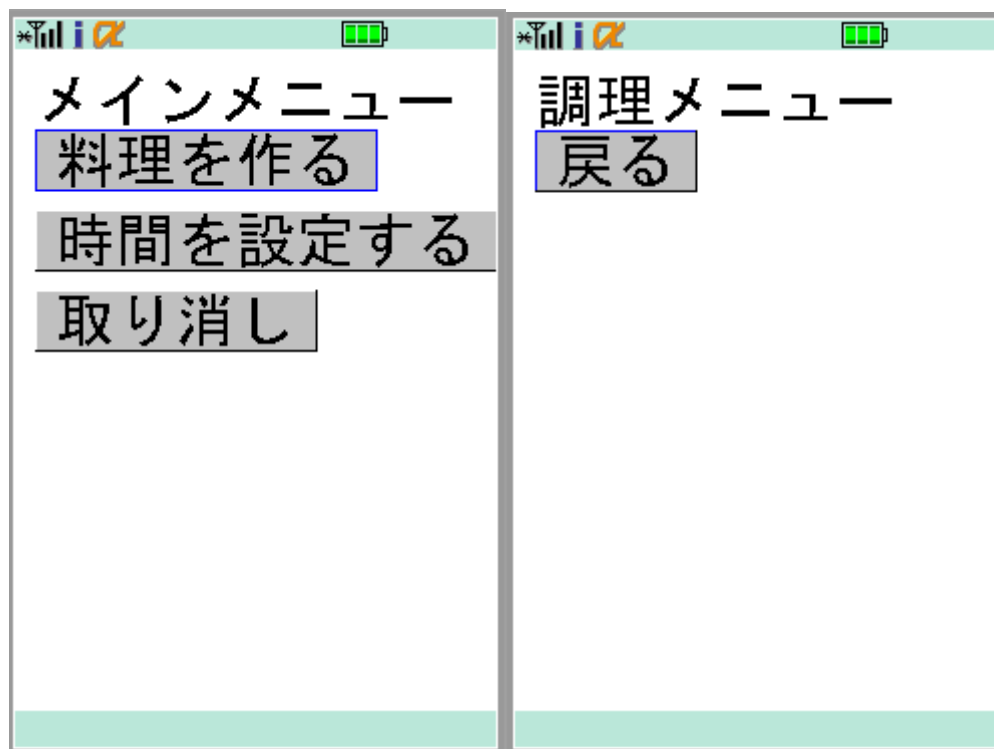


図 7-6 画面遷移の試作画面

毎回は「PAGE」の ID により、合う「PAGE」部分しか作成しない。前の変数部分を一緒に考えると、画面 ID (move) も変数の 1 つとして保存する。それで、初期画面を設定しようと思う時、も move にセットすることで実現できる。ボタンをクリックしたら、遷移先の画面 ID を move にセットして、再描画を行う。

まず「PAGE」を解析する部分に ID により判断する。仕様記述を一行ずつ解析する時、PAGE 単位で画面 ID が違う仕様記述飛ばす。次はボタンを押下する時、イベント処理部分に画面 ID を変数にセットして、画面の再描画を行う。ただし、再描画を行う前、変数の設定を行わないため、変数セット部分には実行しない。

7.4.3 イベント処理

イベント部分の内容もコンポーネントの1つ属性として、各コンポーネントにある、最初読む時他の属性と同じように set して、イベントが発生する時内容を get して解析する。

制約条件の記述方式は「条件→結果」の形で記述される。複数の条件式を連結する時、「&&」は AND 関係を表す、「||」は OR 関係を表す。例えば、図 7-7 の例で、A の Action には「A あるいは B をチェックされた時 T のコンポーネントに影響を与える」という意味を表す。C の Action には「C と D を両方チェックされないと、E が使えない」という意味を表す。

```
A.checked || B.checked->T.enabled  
C.checked&&D.checked->E.enabled
```

図 7-7 イベント処理の記述例

図 7-8 はイベント処理の主処理フローを示す。まず **action** に書いたイベントを字句単位で分割する。そして、条件と結果の部分进行分離させる。

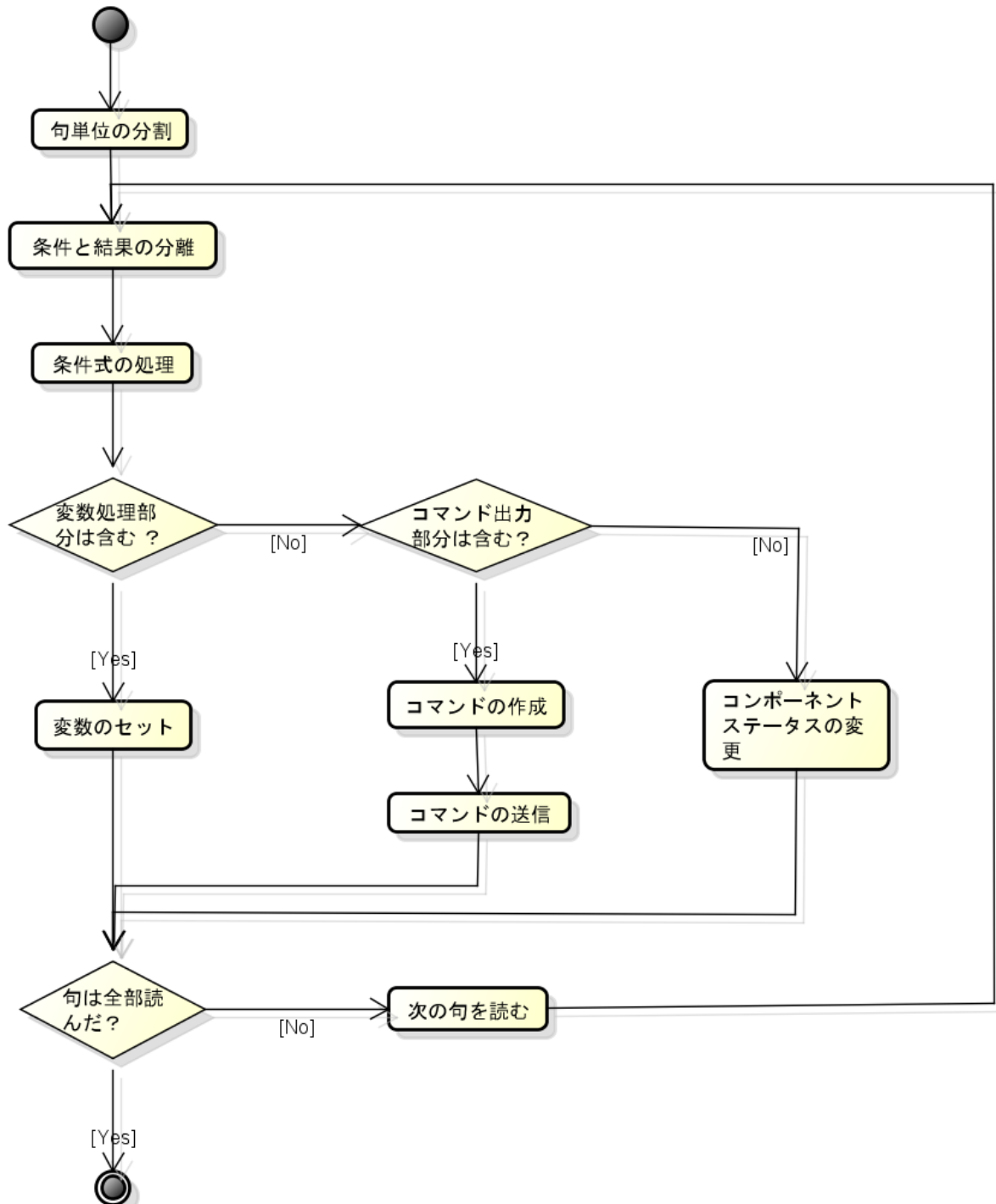


図 7-8 イベント処理の主処理フロー

イベントは試作プログラムのようなコンポーネントステータス変更以外に、変数部分の処理とコマンド出力部分の処理もしないといけない。この2つの部分の具体的な処理は後ろの章を参照する。

図 7-9 は条件式の処理部分のフローを示す。or で繋ぐ状況と and で繋ぐ状況、1 つだけの状況に分けた。オブジェクトを取得してからステータスを取得する。それで、ステータスによりフラグをセットする。

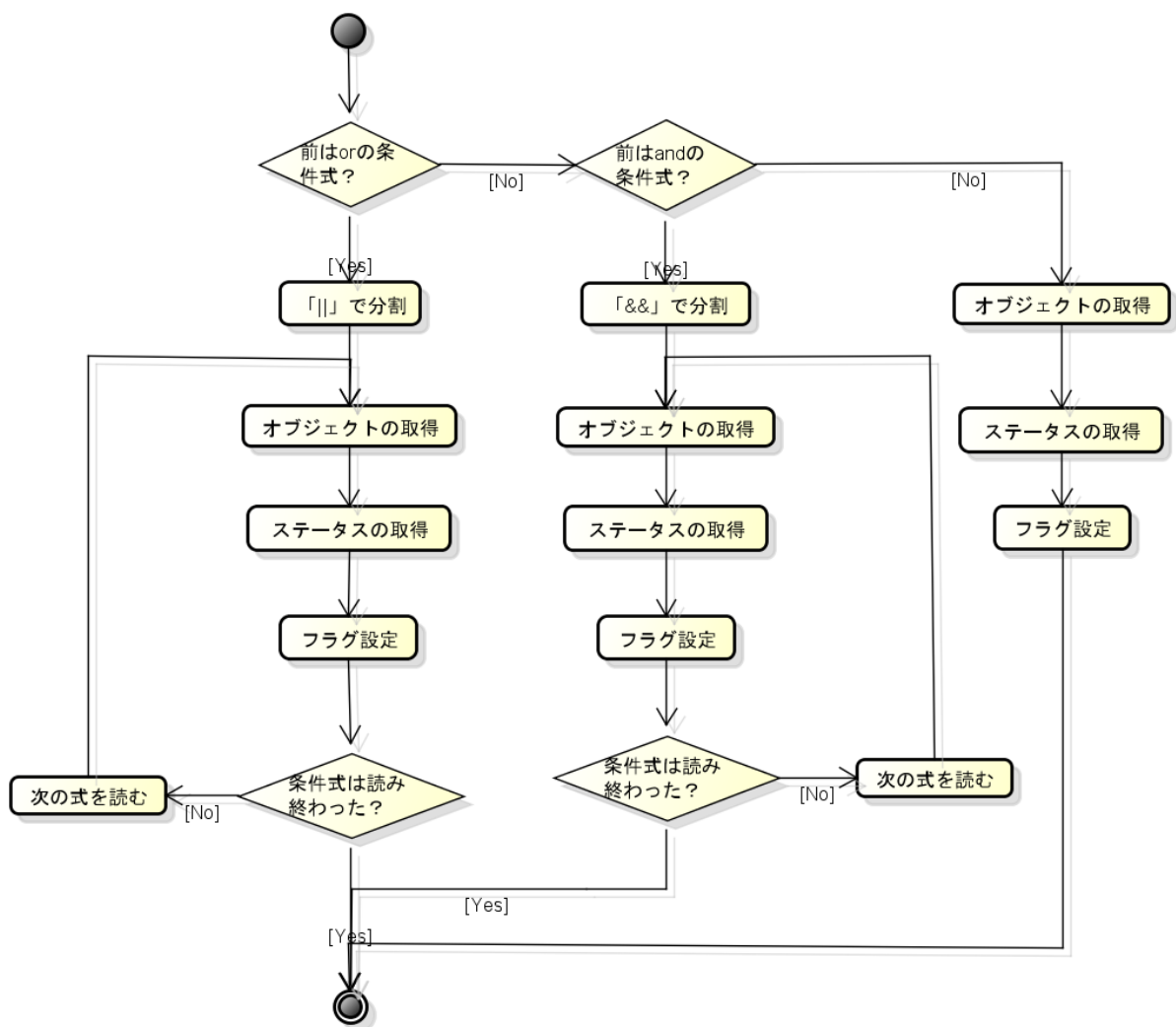


図 7-9 イベント処理の条件式の処理フロー

7.4.4 グループ処理

グループ化とは、コンポーネントを1つのグループにして、グループ単位で処理を可能とする。例えば、グループは **enabled**、**disabled** と設定されたら、グループに含まれたコンポーネントを全部 **enabled** 化 **disabled** にする。また、既にできたグループと別のグループ又はコンポーネントと一緒に新たなグループを作ることもできる。

画面のコンポーネントを制御する時、グループ単位でコントロールするのは楽だと考える、また、仕様記述を書く手間も軽減できる。

group の書き方には2つの方式がある。group の所属情報をコンポーネントごとに書いて、最終に group のイベント情報も書く。それとも、前には書かなくて、最終にまとめて記述する。後ろにまとめて書くと、階層構造は一目瞭然で、書くと読む時は楽。

例えば、図 7-10 の例は現在のトーストの部分の記述例である。

```
BEGIN:GROUP
NAME:breadGroup
MEMBER:breadMinute;breadSecond
BEGIN:GROUP
NAME:amount
MEMBER:onePiece;twoPiece
END:GROUP
BEGIN:GROUP
NAME:cooked
MEMBER:dark;thin
END:GROUP
END:GROUP
... ..
```

図 7-10 グループ処理部分の記述例

グループの中に更にグループの定義を記述する。親グループと子グループの記述方式は他のコンポーネントと一緒に、「BEGIN」と「END」の間に記述する。「NAME」はグループ名を表す。「MEMBER」は含めるコンポーネントを記述して、「;」で区切る。

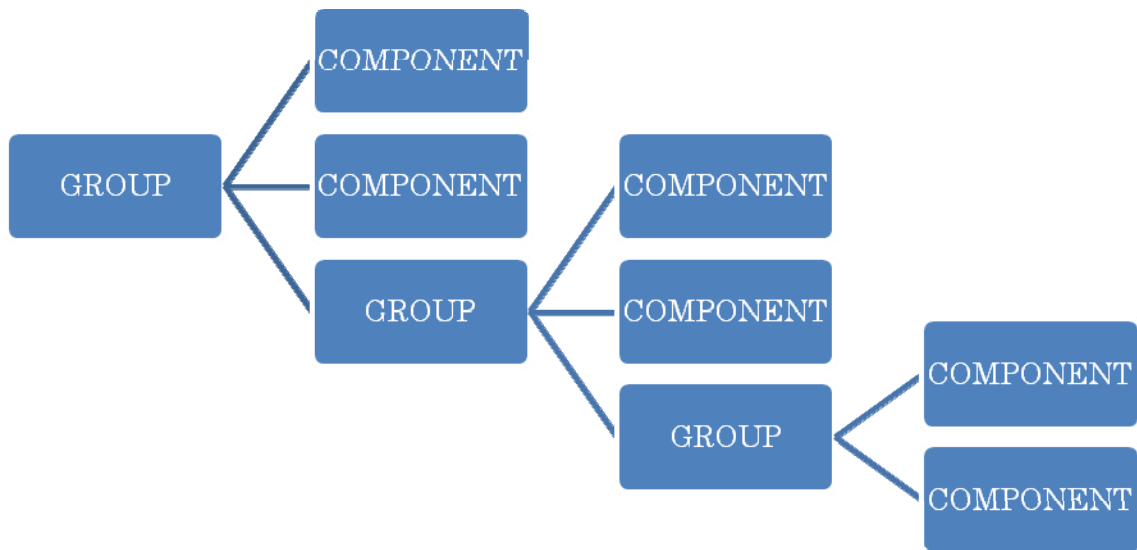


図 7-11 グループの構造例

図 7-11 はグループの構造を示す。グループの中に子グループかコンポーネントを含むので、再帰でこのような構造を解析する。

再帰を実現するため、まず保存する変数をつくる。コンポーネントとグループ両方とも入れるため、**Group** というクラスを作って、モノリストは **Vector** と定義する。全部グループ情報は **Group** クラスという形で **Vector** に保存する。

それで、コンポーネントの場合、直接もののリストに入れる。グループの場合、新たなグループを作って引数として渡して自分を呼び出す。図 7-12 は処理フローを示す。

また、グループの開始と終了標識は同じなので、グループはいつ終了したかを知る必要もある。つまり、ネストの深さを図るため、変数 **nest** を追加した。初期値は「0」。グループ開始の記述 (**BEGIN:GROUP**) を読んだ時、1 を増やして、終了の記述 (**END:GROUP**) を読んだ時、1 を減らす。「0」に戻った時、一つのグループ記述は終わった。

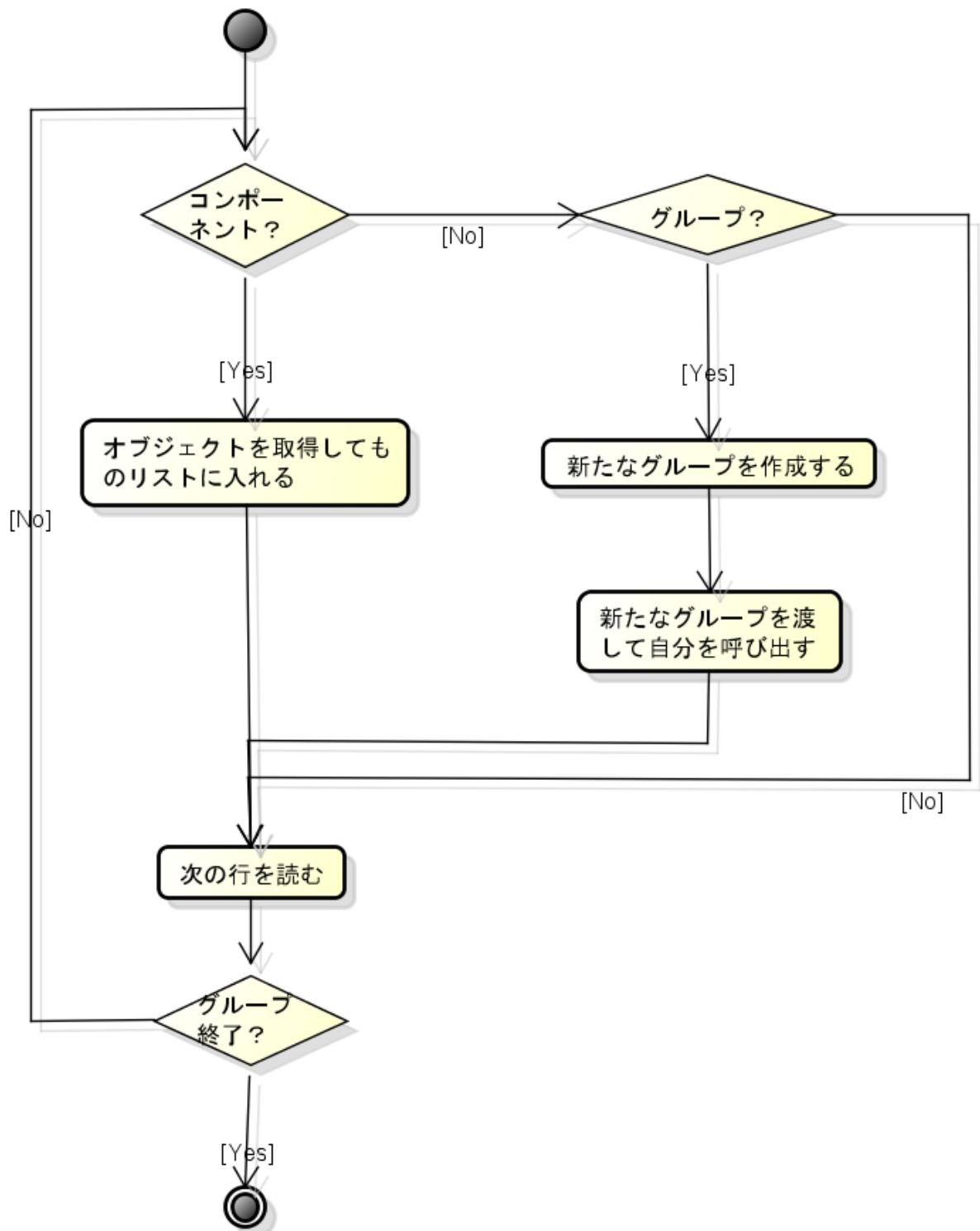


図 7-12 グループ作成時の処理フロー

イベントの中にグループの状態を変更する時、再帰で実現する。また、グループの中に 2 種類のオブジェクト（コンポーネントと子グループ）があるので、これを分けて処理する必要がある。

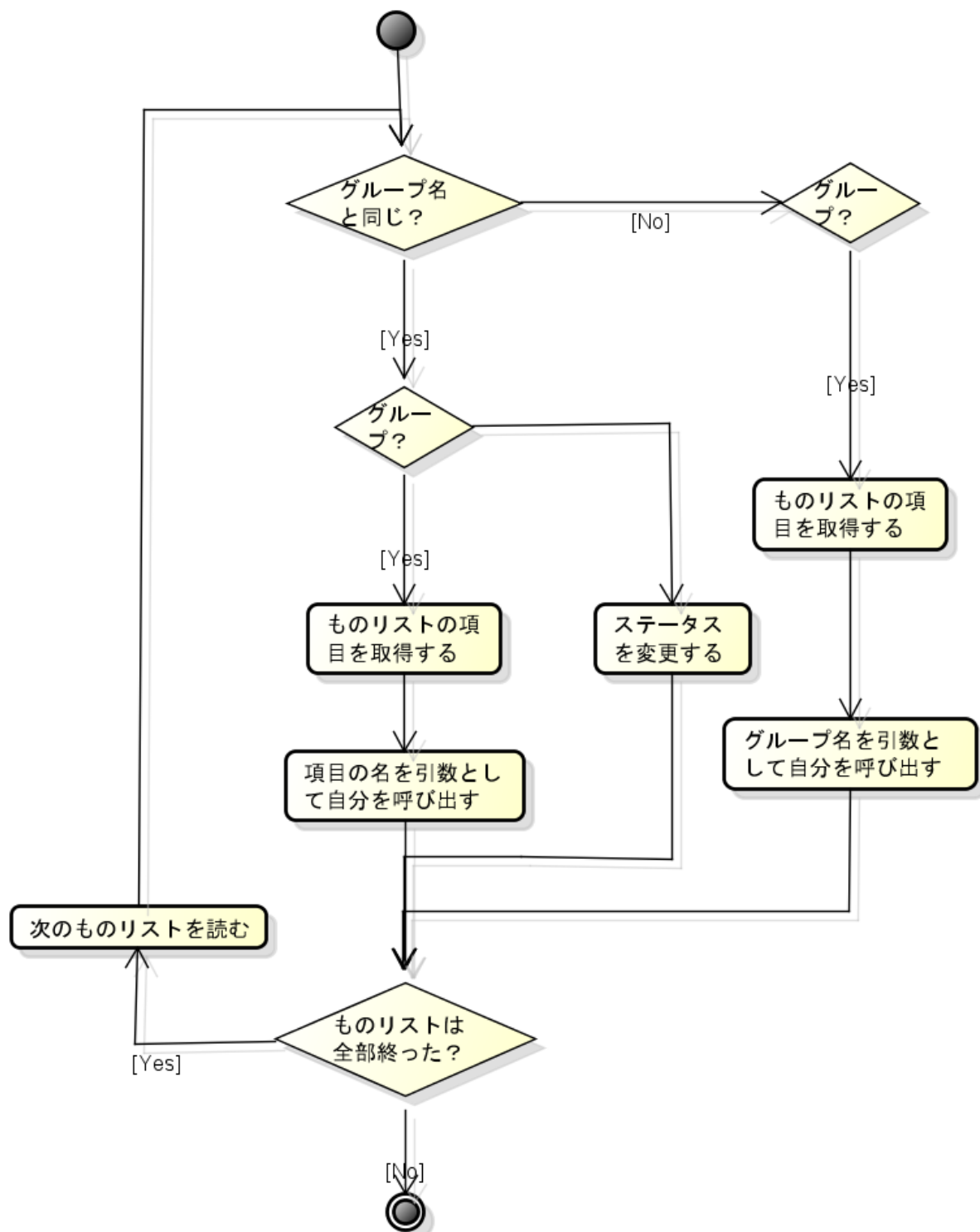


図 7-13 グループを処理する時のフロー

図 7-13 は処理フローを示す。普通の再帰関数と違って、ループも入れた。なぜなら、グループの中にもいくつかのオブジェクトがあって、見つけない限り、全部チェックしないといけ

ない。

指定したグループ名とあった場合、コンポーネント名指定の場合、直接ステータスを変更する。グループ名指定の場合、親の名前を引数として渡して自分呼び出す。指定したグループ名と合っていない場合、自分呼び出すが、渡す変数名が違う。指定したグループ名を引数として渡す。

図 7-14 は実行した画面を示す。トーストの下に数量、焼き色、調理時間を 1 つのグループに指定しておいて、トーストのチェックにより下のグループ全体の状態は変わっていく。



図 7-14 グループ制御の試作画面

7.4.5 変数処理

変数は主に画面遷移の時、値を一時保存用。変数の部分は BEGIN:PARAMETER と END:PARAMETER の間に入れる。初期値を指定することも可能である。簡単に利用するため、変数名はキーとして、初期値は値として連想配列に保存する。図 7-15 は変数部分の記述例を示す。

```
BEGIN:PARAMETER
    move=1
    eggPiece=0
    cookMinuteMIN
    ... ..
END:PARAMETER
```

図 7-15 変数部分の記述例

しかし、変数を初期化する場合もある。例えば、現在の送信方式では、変数によりコマンドを決める、送信した後、変数を初期化しないと、新たなコマンドの作成に影響を与える。それで、変数を保存する連想配列を 2 つ用意して、普通は一つの変数を利用する、もう 1 つは変数の初期値を保存する用。初期化しようと思う時、保存用変数の値を普通用変数にセットする。

7.4.6 CIR コマンドのフォーマット処理

コードを作成する時、フォーマットも仕様から解析して利用する。フォーマットの形式と解析方式は 5.4.4 と同じ。ただし、イベントに書くので、他のイベントと区別するため、前に「output=」を付ける。「output」を読み込んだ時、コマンドを出力することがわかる。

例：

```
output=00+{isEgg:1}+{isToast:1}+{isCoffee:1}+{color:2}+{isNow:1};
```

7.4.7 if 文処理

コマンドを作成する時、入力により違うので、分岐は必要。if 文の形式を処理すると考える。if 文の形式は下記の例を示す。他の記述方式と同じように、BEGIN と END の間で記述する。IF の後ろに条件式を書いて、DO の後ろに実行文を書く。図 7-16 は実際の記述例を示す。

```
BEGIN: IF
    IF: isEgg==1&&isCookMinuteSet==0
    DO: output=010110#{eggPiece:2};commandSend=1
END: IF
```

図 7-16 if 文の記述例

if 文の解析関数は章を担当していて、筆者は条件式と実行文の処理を担当する。実行文の

処理はイベント処理と一緒になので、同じイベント処理関数（processEnt）で処理する。

7.4.8 CIR 通信処理

CIR 通信しようと思う時、cirSend クラスの send メソッドにコードを渡せばできるはずだが、send メソッドは byte[] 型の引数を使用する。前のソース作ったコードは String 型の 2 進数、byte[] へ変換しないといけない。それで、作成したコードはまず Integer に変換して、byte に変換して、1 つずつ byte[] にセットする。

7.4.9 音声読み上げ処理

読み上げ機能は Speech クラスの speak メソッドを利用する。ただし、これだけでは足りない。例えば、いつまでも前の内容を読むのではなく、フォーカスを切り替えたら、前の内容を読むのを止めるべき。なので、cancel メソッドも必要。

読みあげたい内容を仕様記述の SOUND 属性に書く。仕様から取った音声内容をコンポーネントの属性の 1 つとして保存する。

後は読み上げる時期を決めないといけない。普通はコンポーネントがフォーカスされる時、内容を読み上げるので、フォーカスを取得するところに音声読み上げる機能呼び出す。

また、最初の画面を立ち上げる時、画面のタイトルを読み上げる必要もあるので、画面のタイトルもラベルとして作成するから、最初のところにコンポーネントを読み上げる機能呼び出す。

しかし、listBox と普通のコンポーネントと違って、入力画面もあるから、特別の処理は必要。入力画面で選択する時、読み上げ機能も必要。そして、listBox は 1 つの音声ではなく、選択肢ごとにもっている。選択肢の音声読み上げも考慮しないといけない。追加したが、スクロールする時だけを読む、最初の選択肢は読めなくなった。それで、画面を開いた時、最初の選択肢を読み上げる機能も追加した。

TextBox の処理も違って、音声は仕様により決められたではなく、入力値を読み上げる。なので、入力値を画面に設定すると音声の部分（SOUND）のセット処理も一緒に行う。

第8章 プロジェクトの分析

本章では、筆者が開発したシステムの体制と計画と実績、開発環境、成果物について述べる。

8.1 開発体制

本プロジェクトは、開発プロセスとしてウォーターフォール型のプロセスを採用したが、設計の段階には、プロトタイプの開発も並行した。理由として、設計の段階では、技術面にたくさんの問題が残されていて、実現可能性を検証する必要がある。また、今回の仕様は抽象的な特徴があって、机上の検討は実際の家電に適用できない又は不十分な場合も想定される。

メンバー全員は課題担当教員と相談して、システムにおける作業をまとめ、各々が担当する部分を決定した。

毎週課題担当教員とミーティングするかたちでプロジェクトを推進する。その場で、各々が自分担当する部分の進捗、作業内容、問題点などを報告して、課題担当教員と他のメンバーから意見を貰う。また、毎回ミーティングの日の前日に報告したい内容をまとめて報告書を作って課題担当と他のメンバーに送信する。

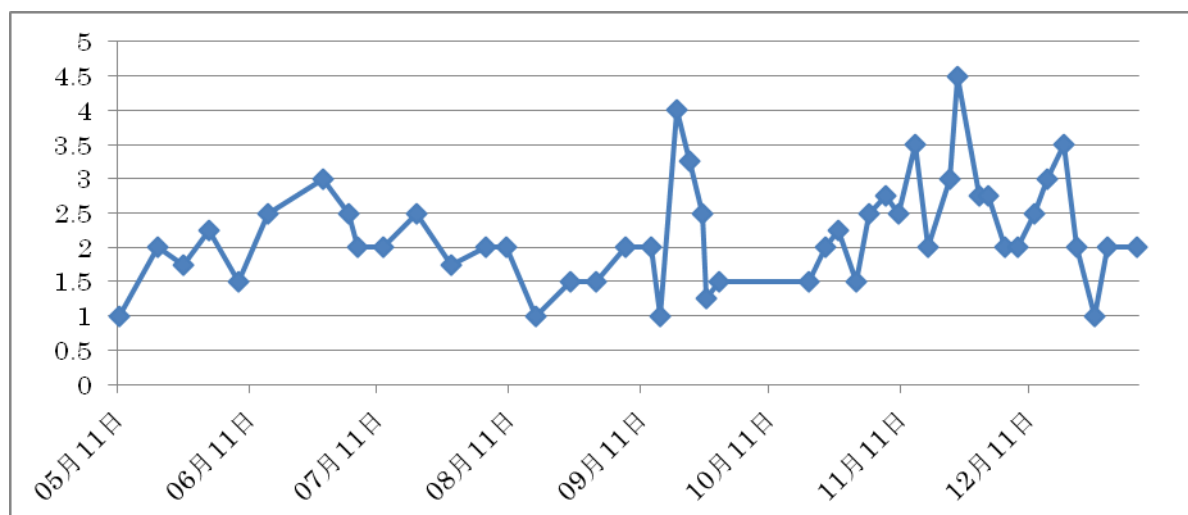


図 8-1 ミーティング実績

図 8-1 は毎回のミーティング実績を示す。縦軸は毎回ミーティング時間である。最初は週一回程度だが、途中進捗が遅れたので、9月の中旬から週二回まで増やした。

8.2 開発環境

本プロジェクトにおける開発環境を表 8-1 に示す。

表 8-1 開発環境

項目	内容
開発言語	Java
統合開発環境	Eclipse 3.5
開発ツール	iappliTool for DoJa-5.1(FOMA)
バージョン管理	DropBox (ドキュメント) CVS (ソース)
OS	Windows Vista、Windows 7

i アプリを開発するため、開発言語は Java になった。

チームで開発を進む際、特に筆者と持田の部分に関連性が高いので、ソースプログラムの構成管理が重要である。バージョン管理ツールは数多く存在しているが、Eclipse の CVS[30] クライアント機能を利用するため、CVS サーバを構築した。フリーソフトである「CVSNT」[31]を利用してサーバを立ち上げ、Eclipse の CVS レポジトリを利用して接続する。また、ドキュメントは DropBox で管理する。

8.3 開発計画と実績

本プロジェクト携帯側における開発計画の予定と実績を図 8-2 に示す。

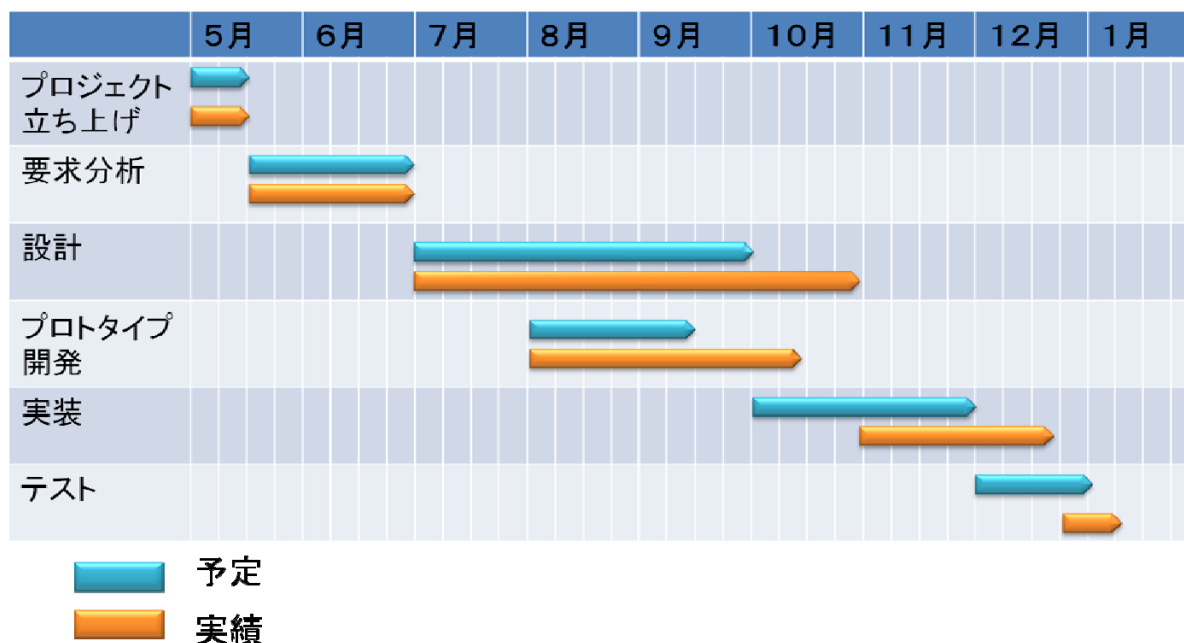


図 8-2 携帯側開発予定と実績

プロジェクトを立ち上げた時点、課題担当教員と相談して今後の仕事を洗い出して、スケジュールを完成した。技術面に不確定又は不明な点は多いので、設計とプロトタイプが開発が並行する方式を採用した。また、設計は全体スケジュールの中に一番長い時期（3ヶ月）を設けた。

プロジェクト立ち上げと要求分析の段階には予定通り進行した。設計から一ヶ月ぐらいの遅延が発生して、後工程に影響し、スケジュールが遅延した。遅延原因について、下記の4つだと考える。

- ① 見積もりは甘かった。プロトタイプの開発中にいろいろと問題が抽出され、開発内容がふくらんだと考えられる。また、開発途中で、提供するコンポーネントはが使えず、オリジナルのコンポーネントを開発しなければならないことが判明した。オリジナルコンポーネントの開発は実装工程のベースなので、オリジナルコンポーネントの開発は終わらない限り、実装工程が始められないはずである。
- ② 目標は明確していなかった。プロトタイプの開発中にいろいろと問題が抽出された。これらに対応するため、かなり時間がかかっても、実装する時ぜんぜん使わない機能もあった。すべての機能に対応するアプリを作るのは理想的だが、時間が限られたので、全部機能を実装するのは困難である。
- ③ 互いの周知は不十分である。毎週のミーティングは一回だけで、もし何の問題があつて、先生又はみんなに聞きたい時、何日を待たないといけない。メールでもできるが、話し合わないとうまく説明できない問題はけっこうある。

- ④ 役割分担に問題があった。仕様の設計は他人を担当した。設計者は機能しか考えなくて、実装が全く考慮しないので、実装は困難またはできない問題点がかなり多くで、ほとんど利用できない。毎回筆者は実装する前、設計から直さないといけないので、けっこう時間がかかった。つまり、設計にはほとんど役に立たないので、実装に負担をかけた。設計者自身経験は不足の原因もあるが、実装をやらないとどのような設計が良いかわからないのは主因である。

これらの問題を解決するため、下記の措置をどんどん取り込んだ。

- ① オリジナルコンポーネントの開発を待たずに、未完成品を利用して実装する。
- ② 機能に優先度を付く、重要度から順番に実装する。
- ③ ミーティング時間を週二回まで増やしたと、コアタイムを設けた。コアタイムは週二回で、毎回5時間である。ミーティングと合わせると、毎週ほぼ毎日全員会えて意見を交換することができる。
- ④ 全員は実装に関わる。

上記の措置を取り込んで、途中遅延が発生してもある程度完成した製品を開発した。ただし、別の問題が発生し、予定通り進んでいない状況もある。①番は未完成品を使うので、画面のコンポーネントがフォカスできない、選択できないなどの問題が発生し、これらの問題を直すのは時間を費やした。また、②番と④番を取り込んだ時期(12月から)は遅かったなので、効果は薄かったである。

8.4 成果物

筆者はプロトタイプの開発と実装の部分を担当している。各プログラムにおける実績を表 8-2 に示す。

表 8-2 プログラムの実績

成果物		数量
ソースコード	GUI プロトタイプのアプリ	630 ステップ(コメント、空行除く)
	解析プロトタイプのアプリ	750 ステップ(コメント、空行除く)
	UI とコマンド作成のアプリ	1568 ステップ(コメント、空行除く)

第9章 結論

本プロジェクトでは、萩川先生からの委託により、音声応答家電の開発を行った。要件に従い、音声応答機能付き NTT ドコモらくらくホンと全機能調理機「おまかせミールさん」を選定した。プロトタイプ開発と設計を並行し、実装とテストを行った。開発したシステムを用いて「おまかせミールさん」を制御できたことにより、音声応答携帯電話を用いて音声応答家電の開発の可能性を示した。

比較的複雑な機能を有するおまかせミールさんが制御できたことにより、オープンレンジや洗濯機なども同様に制御できると考えられる。それで、汎用的なものを開発するため、設計の段階でハードウェアにもソフトウェアにも工夫した。

筆者が担当した携帯電話上で家電を制御する部分で、ある家電（例えば、おまかせミールさん）に専用したアプリを開発するのは簡単だが、他の家電に対応するため、まだ一からアプリを開発しないといけない。アプリの開発はかなり面倒なので、汎用性はない。そして、専用アプリの開発を代わりに、解釈系のアプリを開発した。仕様記述を読み込んで UI を音声でユーザに提示して、入力と仕様によりコマンド作成して家電に送信することで家電を制御するという方法を選んだ。それで、将来のシステム開発者は記述規則に従い、仕様を書くだけで、新たな家電に対応することができ、効率も高いし、バグも少ないというメリットがある。

通信方式の選択にも汎用性を考慮した。おまかせミールさんは CIR だけで十分かつ容易だが、他の家電は IrDA を利用する可能性があるので、CIR と IrDA を両方利用して実装することになった。最終 IrDA のハードウェア部分の開発にいろいろな問題が起こってしまい、間に合わなかったが、現在ハードウェア部分の技術問題は解決した。

設計の段階で、仕様記述を定める時、初めから汎用のものを作るのではなく、事例をもとに仕様記述が満たしているべき条件を整理し、汎用記述をあらためて定義する方法を採用した。このやり方で、仕様記述は頭の中だけで考えるものではなく、実世界に近いものは開発できたことで、リスクは軽減できる。

今後、メーカー様は本システムの開発成果を利用して、音声応答機能を家電に組み込むことで、目の不自由の方の生活に役立つことを期待する。

謝辞

本報告書を作成するにあたり、研究開発プロジェクトの委託元教員として数々のご助言とご指導を頂いた萩川友宏准教授に深く感謝いたします。お忙しい中にも丁寧に対応して頂き、ありがとうございました。

指導教員の田中二郎教授には、二年間丁寧かつ熱心なご指導を賜りました。心から感謝申し上げます。

本報告書執筆にあたり、副査としてご助言を戴いた山戸昭三教授に深く感謝致します。

また、本プロジェクトのチームメンバーである持田辰也君、劉俊鵬君、章程君にもお世話になりました。開発上にもプロジェクトの進行上にもいろいろなアドバイスを頂いたことで、自分自身を成長させることができました。

最後に、様々な面で支援いただいた家族や友人、諸先輩方、大学生活でお世話になったすべての方々に心より感謝致します。

参考文献

- [1] 厚生労働省, 平成 18 年身体障害児・者実態調査結果, 2008/3/24
- [2] 総務省 統計局, 高齢者人口の現状と将来, 2008
- [3] ECHONET CONSORTIUM, APPENDIX ECHONET 機器オブジェクト詳細規定 Ver. 3.21 Release b, 2005/10/13
- [4] 多鹿陽介・門間信行・一色正男, ネットワーク家電の標準化技術—ECHONET™ 技術とその応用, 日本ロボット学会誌 Vol. 21 No. 6, pp. 626~631, 2003
- [5] Sun Microsystems, Jini Architecture Specification Version 1.2, 2001/12
- [6] 門脇恒平・早川裕志・小板隆浩・佐藤健哉, インタフェースレス志向による Jini システム構築, FIT2006 (第 5 回情報科学技術フォーラム)
- [7] RBBTODAY, Jini (ジニー。Java Intelligent Network Infrastructure) ,
(<http://dictionary.rbbtoday.com/Details/term1570.html>), 2000
- [8] William Grosso, Java RMI, オライリー・ジャパン(2002/06)
- [9] Device Control Protocols, UPnP Forum, 2012
- [10] URC コンソーシアム : <http://myurc.org/>
- [11] 樋口正生, 森岡隆一郎, 稲垣勝利, 戸崎明宏 : 家庭内 AV ネットワーク HAVi の概要, PIONEER R&D VOL. 11 NO. 2, pp. 39-49
- [12] 宮城愛美・渡辺哲也・南谷和範・長岡英司, 視覚障害者のパソコン・インターネット・携帯電話の利用状況に関する調査, 2007
- [13] 杉之原亮・田中成典, Eclipse ではじめる「i アプリ」開発, 工学社, pp. 2-41, 2008/4/20
- [14] NTT DOCOMO, INC, アプリコンテンツ開発ガイド for DoJa-5.x/5.x LE 詳細編 第 3.00 版, pp. 10-109, 2008
- [15] NTT ドコモ株式会社 : i アプリコンテンツ開発ガイド for DoJa-5.x/5.x LE ~ i アプリオプション・i アプリ拡張編 ~ 第 3.10, 2007/10/24
- [16] 電子通信学会, パケット交換技術とその応用, 社団法人電子通信学会, pp. 19-25, 1980/8/20
- [17] 日本アイ・ビー・エム PC カード開発, 赤外線通信プログラミングガイド, ソフトバンク株式会社, pp. 2-9, 1995/11/19
- [18] 坂田史郎・嶋本薫, 無線通信技術大全, 株式会社リックテレコム, pp. 437-438, 2007/2/28
- [19] International Business Machines Corp., Lucent Technologies, Inc., and Siemens , A versit Consortium Specification, 1996/09/18
- [20] Erik T. Ray, 入門 XML 第 2 版, 株式会社オライリー・ジャパン, pp. 1-111, 2008/11/26
- [21] D. Crockford, JSON.org, Network Working Group, Request for Comments: 4627, The application/json Media Type for JavaScript Object Notation (JSON), 2006/7
- [22] Jeffrey E.F. Friedl, 詳説正規表現 第 3 版, 株式会社オライリー・ジャパン, pp. 359-397, 2008/4/25
- [23] Sun Microsystems, Inc, Java™ 2 Platform Standard Ed. 5.0, 2004
- [24] Hidekatsu Izuno, simple json encoder/decoder for java Version 1.2, 2011

- [25]Json.org, JSON-lib, 2010/12/14
- [26]Google Project Hosting, google-gson , 2011
- [27]Elliottte Rusty Harold、W.Scott Means, XML クイックリファレンス, 株式会社オライリー・ジャパン, pp.363-384、 pp.385-402, 2002/12/25
- [28]Terms of Use、Privacy Policy, JAXB User' s guide, 2008
- [29]The Apache Software Foundation, The Digester Component, 2011/7/11
- [30]Jennifer Vesperman, 実用 CVS, 株式会社オライリー・ジャパン, pp.1-207, 2003/12/19
- [31]March Hare Software , Windows Installation Guide for CVS Suite 2009R2, 2011