

## メニュー操作を考慮した機種非依存の家電統合インタフェース

小川耕司<sup>†</sup> 山下浩司<sup>†</sup> 小暮悟<sup>††</sup> 小西達裕<sup>††</sup> 伊東幸宏<sup>†††</sup>

<sup>†</sup>静岡学大学院情報学研究所 <sup>††</sup>静岡大学情報学部 <sup>†††</sup>静岡大学創造科学技術大学院  
〒432-8011 静岡県浜松市中区城北 3-5-1

E-mail: <sup>†</sup>{gs06014, gs07050}@s.inf.shizuoka.ac.jp, <sup>††</sup>{kogure, konishi}@inf.shizuoka.ac.jp,  
<sup>†††</sup>itoh@inf.shizuoka.ac.jp,

あらまし 近年家電製品が高機能化し、その操作方法も複雑化し、操作を覚えることが困難になってきている。さらに、買い替え等によって使用する機器が変わった場合には、改めてその複雑な操作を覚えなおさなければならないという問題がある。われわれはこれまでに、この問題に対処するために、ある家電機器における様々な機種をすべて同じ感覚で操作可能とするインタフェースをすでに構築している。本稿では、そのインタフェースの機能を拡張し、メニュー操作を伴う操作も利用可能にした。実際に、HDDレコーダを例にとり、メニューを含む操作が可能なインタフェースが動作することを確かめた。

### Household appliance unification interface of the model independence that considered menu operation

Koji OGAWA<sup>†</sup> Koji YAMASHITA<sup>†</sup> Satoru KOGURE<sup>††</sup> Tatsuhiko KONISHI<sup>††</sup> Yukihiro ITOH<sup>†††</sup>

<sup>†</sup>Graduate school of information, Shizuoka University

<sup>††</sup>Faculty of information, Shizuoka University

<sup>†††</sup>Graduate school of science and technology, Shizuoka University

3-5-1 Johoku, Hamamatsu, Shizuoka, 432-8011 Japan

E-mail: <sup>†</sup>{gs06014, gs07050}@s.inf.shizuoka.ac.jp, <sup>††</sup>{kogure, konishi}@inf.shizuoka.ac.jp,  
<sup>†††</sup>itoh@inf.shizuoka.ac.jp,

**Abstract** Recently, the performance of household electrical appliances is very high, and the operation method becomes complicated, so it becomes difficult to learn the operation method. Furthermore, there is a problem to have to learn the complicated operation method again when the equipment used changes. In order to solve this problem, we have already constructed the interface to do an interface by which we can operate various models in certain a household appliance. In this study, we expand the function of the interface and make the operation with a menu operation available. We took the digital video recorder for an example and confirmed that the proposed interface can treat the operation included the menu instruction.

#### 1. はじめに

近年、家電製品はますます高機能化しており、その操作方法も複雑化している。そのため、ユーザが操作を覚えるのに時間がかかる。また、同じ機能を持つ別機種を購入すると、操作方法を覚えなおさなければならない。このような家電機器の操作の複雑化という問題を解消する必要がある。この問題に対処するためには、「操作の簡略化」「機種ごとに違う操作を統一化する」という点を解決しなければならない。例えば、リモコン操作で行う機器操作を PDA 等の端末で統一的に操作可能なインタフェースに関する研究がある[1]。また、複数の情報家電が置かれた部屋において、個々の機器を意識せず、同じ感覚で利用可能なインタフェースの構築に関する研究も行われている[2]。また我々は、過去にカーナビゲーションで、入力方法を統一し、複数機種で統一感のある操作が可能となるインタフェースの研究を行ってきた[3]。さらに、リモコンで複数回のボタン入力が必要な操作を 1 回のコマンド入力で操作可能にし(「操作の簡略化」)、同一家電機器の複数機種を同

感覚で操作可能なインタフェース(「機種ごとに違う操作を統一化する」)を構築し、実際に、ビデオデッキ 4 機種の実機を同感覚で操作可能なインタフェースを実現した[4]。しかし、これまでの研究では、機器の単純な基本操作のみが対象となっており、近年の家電製品に多く見られる、メニューを利用した操作に対応できないという欠点があった。[1][2]の関連研究においても、単純な動作しか対応できていないという同様の欠点がある。現在機器が高機能化するにつれ、メニューを利用した操作がより複雑になっている。したがって、本研究では、これらのメニュー操作への対応を考慮した、あらゆる機種を同じ感覚で操作可能となるインタフェースの構築をめざす。

また、本研究では主に音声インタフェースを扱っている。音声には、手がふさがっている状態でも使える、訓練の必要がないなどの利点があり、音声インタフェースの研究が盛んである[5][6]。また、家電で音声インタフェースを利用する研究も行われており[7][8]、本研究でも音声インタフェースを利用する。

## 2. 機種非依存の家電統合インタフェース [4]

### 2.1 研究の狙い

本研究の狙いは「機種ごとの違いを吸収すること」と「リモコンだと2回以上の操作が必要な操作を1回のコマンド入力で操作可能にする」ことである。前者については、例えばビデオデッキのスロー再生は、ある機種ではスロー再生ボタンを押すが、ある機種では、再生中に一時停止を押しその後で再生ボタンを長押しする必要があるような状況であり、後者については、例えば HDD レコーダで保存した番組を削除する場合に、番組を選択してメニュー内をカーソル移動して削除の項目を選択する必要があるような状況のことである。

機種非依存のインタフェースを実装するために、我々は「標準モデル」「統一文体」という2種類のアプローチを組み合わせて利用するインタフェースを提案する。

### 2.2 機種非依存の実装

#### 2.2.1 標準モデル

標準モデルとは、家電の種類ごとに標準的な機能を定義し、その標準的な機能に対するコマンドを受け付けることで、同じ種類の機器であればどのメーカーの機器でも同じコマンドで操作できるようになるというものである。通常のリモコンを利用した操作と、標準モデルを利用した操作のイメージは図1のようになる。リモコンを使った操作では、機種ごとにボタン配置が違い、ボタンの使い方も違うため、機種ごとに操作方法を覚えなければならない。しかし、標準モデルを用いる事で、ある機器に対して機種ごとの操作方法を習得する必要がなくなり、その機器の標準モデルの操作を習得するだけで、すべての機種を操作可能となる。

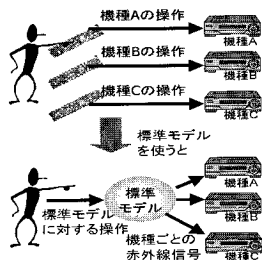


図1 標準モデルを利用した操作イメージ。

標準モデルは以下の数式で表すことができる。

$$\text{標準モデル: } M = (S, P, O)$$

$$\text{状態: } S = (s_0 \cdots s_n)$$

$$\text{操作: } O = (o_0 \cdots o_m)$$

$$\text{状態遷移: } P = S \times O \rightarrow S$$

標準モデルは、状態、操作、状態遷移で表すことができる。状態とは機種ごとに定義した状態の集合である。操

作は、その機器で利用可能な標準的操作の集合である。ある状態から、ある操作が行われることで、次状態を決定することができ、その状態へ遷移する。

ここで言う状態とは、状態が物理的に変化する場合を差している。機器の内部状態、内部パラメータが変更される操作は状態遷移図とは別に管理する。標準モデルに登録する基準を以下に示す。

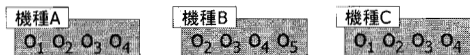


図2 標準モデルに含む基準。

例えば、機種A~Cにおいて、図2のような操作が利用可能であるとする。このような場合、操作 $o_2 \sim o_4$ によって起動される機能はどの機器にも備わっているので、標準モデルに含める。一方操作 $o_1$ によって起動される機能は、Bでは使われていないが、機種ACでは使われている。一部の機種では使われていなくても、一般的であると判断できれば、標準モデルに含める。操作 $o_5$ は、機器固有の操作と思われるため、この段階では標準モデルには含めない。しかし、ある時期では機器固有の操作であるとしても、将来的に標準的な操作となった場合、その段階で標準モデルに登録するということが可能である。

#### 2.2.2 統一文体

ユーザは、標準モデルで定義した操作Oを実際にシステムに伝える際に、その操作を音声言語でどのように表現するかを決める必要がある。例えば、全く制限のない自然言語での入力を希望するかもしれないし、単語のみで入力する方法も考えられる。前者はユーザにとっては有効な入力手法であるが、音声認識や言語理解を考慮した場合、自然言語での入力は非現実的である。また、後者は、例えばカーナビゲーションではすでに実現されており、実現可能性は非常に高いが、多様な機能を持つ家電に対する操作を1単語ですべて表現することは難しい。そこで、本研究では、以下に示すような統一文体[3]を音声で入力するインタフェースを採用している。

**対象 を 操作**

**属性 は 値**

使用例： エアコンをつける 温度は28度

目的地を設定する 場所は静岡県

文体を制限することにより、ユーザはこの2文体の使い方を理解するだけで、すべての機種を同じ感覚で操作することが可能となる。また、ユーザは入力できる文体をイメージしやすくなるという利点もある。

現象を表すような表現は、すべて**属性は属性値**という形に同義変形できることがわかっている[9]。ただし、例えば、「状態は再生」のような表現は、不自然なため、「対象を操作」という形式(例えば、「テープを再生」)も発話可能にしている[3]

## 2.3 システムの構成

### 2.3.1 全体構成

「標準モデル」「統一文体」の2種類のアプローチを元にシステムの構築を行う。システムの全体構成は図3のようになっている。

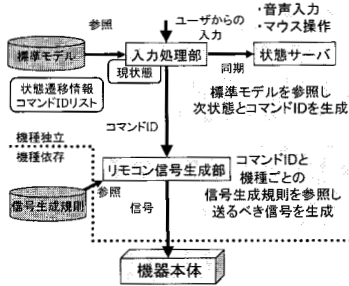


図3 システムの構成図。

### 2.3.2 入力処理部

入力処理部はユーザの入力が機器に対するどの操作を意味しているか、判断を行う部分となる。ユーザの入力は、音声入力の他にマウス等 PC 上のインタフェースを用いた、GUI による操作も受け付ける事ができる。

音声認識にはオープンソースの音声認識システムである Julian を利用する[10]。Julian をサーバとして起動しておき、入力処理部をクライアントとして通信を行い、認識結果が出力されたときに結果を解析して目的の操作を決定する。音声認識を行う場合は、構文解析等を行い、出力結果を理解しなければならないが、統一文体を利用する事によって、入力される形式は決められているため、複雑な構文解析等が必要とならず、曖昧性なく操作を決定することが可能となる。

ユーザからの入力が入力処理部に送られてくると、入力処理部は、現状を確認し、標準モデルを参照する。入力された操作が、現状で受理可能な操作かどうかを判断し、遷移すべき次状態を決定する。

次状態が決定されると、機器の状態を変化させるための操作の ID (コマンド ID) を生成し、リモコン信号生成部に送信する。コマンド ID を生成する理由は、入力処理部とリモコン信号生成部で、サーバ、クライアントとして分けられているためである。入力処理部とリモコン信号生成部で、お互いに共通の ID リストを保持することにより、送信される操作がわかりやすくなっている。

システムからユーザへ、行われた操作に対して、次の操作の指定、動作の確認など、応答文を返すことができる。これは操作に複数ターンの対話が必要となる操作に利用することができる。一般的に複数ターンの対話が必要となる操作は、操作を完了するために、必要数の

情報をユーザから取得しなければならない場合に起こり得る。例えば、予約録画をする際には、ユーザから「録画日」「開始・終了時間」「チャンネル」の情報を取得しなければならない。このような場合にそれぞれシステムから聞き、ユーザがそれに答えるという形で操作を完了することができる。

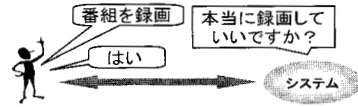


図4 システムとの対話例。

また、図4のように、誤認識が起こった場合に危険な操作に対して、ユーザから確認をとるといことも考えられる。以上のような場合にシステムと対話することによって、操作を簡単化することができる。

機器の状態が変化する操作については状態遷移図で対応しているが、状態は変化しないが内部状態が変更するような操作についても対応をする必要がある(例:テレビのチャンネル, エアコンの設定温度)。これらについては、内部状態を変化させる操作にどのようなものがあるかということ調査し、その値の一覧を表としてまとめ、コマンド ID+設定値という形でリモコン信号生成部に送信することにより対応する。

### 2.3.3 リモコン信号生成部

リモコン信号生成部は、入力処理部から受け取ったコマンド ID を実際に機器に送信する信号の形に変換し、送信をする部分である。

リモコン信号生成部ではコマンド ID リストとコード ID リストを保持している。コマンド ID リストは入力処理部と同じものを持っている。コード ID リストはコマンド ID と実際の信号を付け合せるリストである。

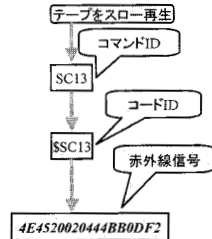


図5 信号生成の流れ。

実際の信号生成の流れは図5のようになっている。入力処理部から送られてきたコマンド ID をコード ID の形に変換し、赤外線信号に変換する。

コマンド ID をコード ID の形に一度変換している理由は、機種依存の操作方法を吸収するためである。これは、コマンド ID が同じでも送信しなければいけないボ

タンの信号は違う場合があるからであり、例としては、2.1節で挙げたスロー再生のような場合である。

生成されたリモコン信号は赤外線デバイスを通して実際に機器に送られる。

### 2.3.4 ビデオデッキへの対応

実際にこれらの方針にしたがってシステムを具体的に構築するに当たり、対象機器としてまずビデオデッキを選択した。ビデオデッキを選択した理由としては次の理由からである。

1. どこ家庭にでもあると思われる
2. ある程度複雑な機能を含んでいる

標準モデルに登録された各要素は、状態が 17、操作は 20 となった。この章で述べた実装方法をもとに、システムの構築を行い、動作することを確認した[3]。しかし、1章でも述べたように、近年の家電製品はメニュー操作を含むものが多く、この段階のシステムではまだメニューに対応することができないという欠点があった。次章からメニューを含んだ操作への対応方法について述べる。

## 3. メニュー操作への対応

### 3.1 家電の操作体系

リモコンを利用した家電製品には、オーディオ、エアコン、デッキなど様々なものがあるが、リモコンで指示可能な操作は以下のように整理することができる。

- (A) リモコンシングル操作でしか操作不可能
- (B) リモコン複数ボタン操作でしか操作不可能
- (C) メニューでしか実行不可能
- (D) メニューでもそれ以外でも実行可能

先行研究では、(A) (B)のように、リモコンボタンの組み合わせで操作可能となるパターンのみしか対応をすることができなかった。あらゆる家電製品を操作するためには、(C) (D)への対応が不可欠であり、本研究では (C) (D)を使った操作に対応可能となる。複数の家電製品を調査し、家電におけるメニューとはどのような物があるか調査を行った。

### 3.2 メニューへの対応方法

#### 3.2.1 メニューの形状

メニューの形状には以下のような種類が存在する。

- ・リング型メニュー
- ・ $n \times m$  型メニュー
- ・変則型メニュー
- ・クロスメディアパー形式

このように複数の形式が存在するが、これらすべては、 $n \times m$  形のメニューの組み合わせで実現可能である。

#### 3.2.2 メニューの選択と実行

メニュー項目の一覧画面でできることは以下のように分類できる。

- (1) メニューを選択するためカーソルを移動する
- (2) 項目を決定するため決定ボタンを押す
- (3) それ以外

それぞれ詳細に説明する。

- (1) メニューを選択するためカーソルを移動する
  - (1-1) 上下左右に移動する (端以外)
  - (1-2) 端で上下左右移動する
    - (1-2-1) 同じ行・列でループする
    - (1-2-2) カーソルは移動しない
    - (1-2-3) 次の行・列、前の行・列に移動する
    - (1-2-4) 別のメニューの、特定の項目に移動する (前に移動してきたときのカーソル位置に依存しない)
  - (1-2-4') 別のメニューの、特定の項目に移動する (前に移動してきたときのカーソル位置に依存する)

(1-1) において、カーソル移動に使えるキーは、上下左右の4つが基本である。メニュー一覧に対して順移動のみ可能 (移動に使えるキーが1つ) の場合もあるが、上下左右での移動を定義するときに、順方法以外の3つを何もしないと定義すれば実現可能である。

- (2) 項目を決定するために決定ボタンを押す
  - (2-1) 動作を伴うもの
    - (2-1-1) 動作が行われ、その状態に移行する (再生状態など)
    - (2-1-2) 動作が行われ、メニューを開く前の状態に戻る (番組の削除など)
    - (2-1-3) 動作が行われ、メニューに復帰する (メニューの変化なし)
    - (2-1-4) 動作が行われ、メニューに復帰する (メニューに変化あり)   
 ゴミ箱に移動・番組一覧の表示形式の変更
  - (2-2) 動作に必要なパラメータを変更するもの
    - (2-2-1) パラメータを相対変化させる (カーソル上下による数値変更やモード切り替え)
    - (2-2-2) パラメータを直接入力させる (数字の直接入力)
- (3) それ以外
  - (3-1) メニューを移動する (フォルダ移動、一覧の前後移動、)
  - (3-2) 戻る (前の表示に戻る、メニュー階層を1段上に戻るなど)

### 3.2.3 メニュー操作への対応

3.2.2節の定義を元に、メニューに対する操作を実現する。そのため、まずシステムはメニューの構造を把握しなければならない。それぞれのメニュー画面を開いた

ときに表示されるメニュー項目をまとめたファイルを作成し、外部ファイルとして読み込む。以下の左に示すようなメニューを実現するための外部ファイルの例を図の右に示す。

<pre> メインメニュー ├DVD 管理 │├初期化 │└消去 ├設定 │├ネットワーク ││├DHCP ││├DNS ││└WLAN └本体 └システム情報         </pre>	<pre> &lt;?xml version="1.0" encoding="Shift_JIS"?&gt; &lt;RD-X6&gt;   &lt;tree name="メインメニュー" id="m01" open="RCxxx"&gt;     &lt;menu name="DVD 管理" sp="1" id="01" up="03" down="02" call="m01"&gt;       &lt;menu name="初期化" sp="0" id="0101" up="0102" down="0102" call="01" /&gt;       &lt;menu name="消去" sp="0" id="0102" up="0101" down="0101" /&gt;     &lt;/menu&gt;     &lt;menu name="設定" sp="1" id="02" down="03" up="01"&gt;       &lt;menu name="ネットワーク" sp="1" id="0201" up="0202" down="0202" call="02"&gt;         &lt;menu name="DHCP" sp="0" id="020101" up="020103" down="020102" call="0201" /&gt;         &lt;menu name="DNS" sp="0" id="020102" up="020101" down="020103" cmd="MCxx"/&gt;         &lt;menu name="WLAN" sp="0" id="020103" up="020102" down="020101" /&gt;       &lt;/menu&gt;       &lt;menu name="本体" sp="0" id="0202" up="0201" down="0201" /&gt;     &lt;/menu&gt;     &lt;menu name="システム情報" sp="0" id="03" up="02" down="01"&gt;&lt;/menu&gt;   &lt;/tree&gt; &lt;/RD-X6&gt;         </pre>
---	--

sp が 1 の場合、新しいメニューを展開するかどうかを管理している。id はそれぞれの項目をシステムが判断するために割り当てられており、up, down で移動先の id を記述している。cmd は実際に入力処理部から送られてくるコマンド ID が割り当てられている。call は同レベルのノードの中で、1 つの項目しかもっていない情報で、その親ノードから移動してきた際の初期位置になる。

実際に動作をする順番を以下に記述する。

- ① システムがメニューコマンドを受け取る
- ② メニュー構造の記述された XML ファイル全体を走査し、指定されたメニューコマンドを持つメニュー項目を見つける
- ③ ② で発見した項目の兄弟ノードを走査し、call 属性を持つノードを見つける。
- ④ ③ で見つけたノードから② で見つけたノードへのカーソル移動するためのパスを調べ、記録する。
- ⑤ ③ で見つけたノードを呼び出すためのノード (call に記述されたノード) を探し、これがルートノードであれば終了、そうでなければ、③ の処理に戻り、ルートへ戻るまで繰り返す。

以上の動作で作成されたパスをリモコン信号で送信するとメニュー構造を辿る動作が実現できる。システムが目的の操作までのパスを知ることができれば、入力された操作に対して自動的にパスを作成し、目的の操作まで複数回の操作が必要になる場合においても、システムが肩代わりすることが可能となる。

### 3.2.3 標準モデルの拡張

メニュー操作を扱うために、標準モデルにメニューコマンドを集めたファイルと、リモコンコマンドを集めたファイルを新たに追加する。前者は、例えば「削除」のように、メニューに対する操作をすべて集めたファイルとなる。後者はリモコンのボタンそのままに割り当てられたもので、例えばカーソルの上などをリスト化する。

## 4. HDD レコーダへの対応

メニュー操作に対応させるため、機器の例として、HDD レコーダを選択し、システムの実装を行った。HDD レコーダを選択した理由としては、ビデオデッキと操作体系は似ているが、より複雑な操作を含んでいるという点。また、一般に普及している家電製品の中で、ある程度複雑なメニュー操作を含んでいるという点が挙げられる。

### 4.1 標準モデルの作成

複数メーカーのサイトからマニュアルをダウンロードし、調査を行った。メーカーごとに代表的な機種を選択し、どのような状態をもっているか調査を行い、最終的に 37 の状態を標準モデルとして定義した。具体的には「電源 OFF」「HDD 再生」「地上デジタル番組表」などである。さらに内部状態を変更するコマンドについても 3 種類定義した (チャンネル変更, 音声切換, 録画モード切換)。

標準モデル、内部状態変更コマンドリストともに外部ファイルとしてシステム本体とは別に作成し、利用する機器に応じて外部ファイルを作り変えるだけで、様々な機器に対応できるようシステムの構築を行った。

### 4.2 認識辞書の作成

標準モデルから、操作に必要な語彙を選択し、認識辞書の作成を行った。文法は基本的には標準モデルの形に当てはめることとなる。ただし、対象が自明である場合は、その省略形も許すこととしている。例えば、DVD の再生中に「早送り」という発話が認識されたとする、その対象は DVD であると予測がつくため、「早送り」を「DVD を早送り」というように解釈する。

### 4.3 リモコン信号ファイルの作成

実際に東芝の HDD レコーダ RD-X6 の信号の調査を行った。信号の調査には、自作の USB 接続赤外線信号受信モジュールを利用し、リモコンから信号を受信しリスト化する。ID と信号を対応させるリストを、入力処理部と同様に外部ファイルとして作成を行った。複数機器に対応させる場合は、利用する機器ごとにこの外部ファイルを差し替えることによって、機種ごとに対応した信号を送信することができる。

### 4.5 メニュー操作への対応

HDD レコーダではメニューを原則として  $n \times m$  の長方形型の選択肢を画面に出して、カーソルによる相対移動で選択をさせる。絶対指定の方式については、一般的なメ

ニューには存在するが、HDD レコーダではこの方式のメニューはないため、カーソル移動のみ対応を行う。

実際に以下のメニューの実装を行った。

- ・番組表
- ・録画番組一覧 (ライブラリ)
- ・予約一覧

以下の XML はライブラリのメニューの一部である。

```
<?xml version="1.0" encoding="Shift-JIS"?>
<RD-XG>
  <tree name="クイックメニュー (見るナビ)" id="m02" open="RC6">
    <menu name="タイトル情報" sp="1" id="01" up="11" down="02"
call="m02"></menu>
    <menu name="タイトルサムネイル設定" sp="1" id="02" up="01"
down="03"></menu>
    <menu name="ダビング" sp="1" id="03" up="02" down="04"></menu>
    <menu name="タイトル名変更" sp="1" id="04" up="03"
down="05"></menu>
    <menu name="編集機能" sp="1" id="05" up="04" down="06">
  <menu name="チャプター自動生成" sp="0" id="0501" up="0504"
down="0502" call="05"></menu>
  <menu name="同一料金予約プレリスト化" sp="0" id="0502" up="0501"
down="0503"></menu>
  <menu name="同一毎週予約プレリスト化" sp="0" id="0503" up="0502"
down="0504"></menu>
  <menu name="メニュー背景登録" sp="0" id="0504" up="0503"
down="0501"></menu>
  </menu>
  <menu name="こみ箱に送る" sp="1" id="10" up="09" down="11">
  <menu name="はい" sp="0" id="1001" up="" down=""></menu>
  <menu name="いいえ" sp="0" id="1002" up="" down=""></menu>
  </menu>
  <menu name="タイトル削除" sp="1" id="11" up="10" down="01"
cmd="MC1">
  <menu name="はい" sp="0" id="1101" up="" down=""></menu>
  <menu name="いいえ" sp="0" id="1102" up="" down=""></menu>
  </menu>
</tree>
```

ライブラリ状態で番組を削除しようとしたときに、リモコン操作では、サブメニューを開いて一覧の中から削除の項目をカーソルキーで選択してから決定という操作が必要になる。XML 形式で記述された上記のファイルにより、システムはタイトル削除までの操作手順を知ることができる。したがって、「削除」という発話に対して、タイトル削除までのパスを辿ることにより、ユーザは 1 回の操作ですべての操作を達成することが可能となる。

## 5. まとめと今後の課題

ある機器の様々なメーカーの機種を同じ感覚で使用可能とするという目的に対し、我々は「標準モデル」「統一文体」という 2 つのアプローチで操作できるインタフェースの提案した。この方式でインタフェースの設計を行えば、外部ファイルを機種ごとに用意するだけで、様々な機種に対して同じ感覚で操作が可能となると思われる。

また、メニュー構造を持った操作についても、メニューの構造を一般化し機種ごとに XML 形式で記述したファイルを走査することにより、目的の操作を達成する。

今後は、作成したシステムの評価を行う必要がある。リモコンでの操作をするグループと、本システムを利用するグループに分け、しばらくある家電を使ってもらい、その後、別の機種に変更し、同様に操作してもらったときの移行の困難の度合いを判断してもらうという方法を考

ている。

また、本システムは、ある程度対象の家電の操作を知っている人間を対象としているが、初心者を対象とした、チュートリアルシステムの構築も、同時に進められている。チュートリアルシステムでは、2.3.2 節「システムとの対話」の項での方法を基本とし、システムと対話しながら統一文体の使い方を学習できる仕組みとなっている。また、現在いる状態から操作可能な操作をリスト表示し、今何が出来るのかをユーザに示すことができるようになる予定である。

## 参考文献

- [1] Jeffrey Nichols, Brad A. Myers, Michael Higgins, Joseph Hughes, Thomas K. Harris, Roni Rosenfeld, Mathilde Pignol, "Generating Remote Control Interfaces for Complex Appliances", Proc. of the 15<sup>th</sup> Annual ACM Symposium on User Interface Software and Technology UIST'2002, ACM Press (2002).
- [2] 橋本政朋, 泉田大宗, 森彰, "リビングから始めるユビキタス - 情報家電インタフェース -", 情報処理学会誌, Vol. 48, No. 1, pp. 53-58 (2007).
- [3] 森田浩康, 桂川桂子, 林真弘, 伊藤敏彦, 甲斐充彦, 小西達裕, 伊東幸宏, "車載器との音声言語インタフェースにおける文体統一の効果", 情報学ワークショップ 2003(WiNF2003) 論文集, pp. 185-189 (2003).
- [4] 小川耕司, 小暮悟, 小西達裕, 伊東幸宏, "異機種を統一的に利用可能な家電統合インタフェースの構築", 情報学ワークショップ 2007(WiNF2007) 論文集, pp. 99-102 (2007).
- [5] 甲斐充彦, 盛浩和, 中野崇弘, 中川聖一, "フォーム型 Web 情報検索サービスのための音声ユーザインタフェースシステムと操作性の評価", 情報処理学会論文誌, Vol. 46, No. 5, pp. 1318-1329 (2005).
- [6] 原直, 白勢彩子, 宮島千代美, 伊藤克互, 武田一哉, "音声対話による楽曲検索システム", 情報処理学会研究報告 SLP-53-6, pp. 31-35 (2004).
- [7] 渡辺裕太, 関口芳廣, 鈴木良弥, "ビデオ装置を例とした家電品の音声対話機能について", 情報処理学会論文誌, Vol. 44, No. 11, pp. 2690-2698 (2003).
- [8] 吉田諒, 吉田恵, 安村通晃, "音声とテンキーを用いた視覚障害者向けリモコンの試作と評価", ヒューマンインタフェース学会論文誌, Vol. 9, No. 2, pp. 87-92 (2007).
- [9] 池ヶ谷 有希, 野口靖浩, 小暮悟, 伊藤 敏彦, 小西達裕, 近藤真, 麻生英樹, 高木朗, 伊東幸宏, "対話文脈を利用した構文意味解析", 人工知能学会論文誌, Vol. 22, No. 3, pp. 291-310 (2007).
- [10] 大語彙連続音声認識エンジン Julius <http://julius.sourceforge.jp/>