

NAIST-IS-MT0651040

## 修士論文

# 家電状態を利用したホームネットワークシステムの ための音声操作インタフェースの提案

榊原 弘記

2008年2月7日

奈良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

本論文は奈良先端科学技術大学院大学情報科学研究科に  
修士(工学) 授与の要件として提出した修士論文である。

榊原 弘記

審査委員：

松本 健一 教授 (主指導教員)  
木戸出 正繼 教授 (副指導教員)  
門田 暁人 准教授 (副指導教員)  
中村 匡秀 准教授 (神戸大学)

# 家電状態を利用したホームネットワークシステムのための音声操作インタフェースの提案\*

榊原 弘記

## 内容梗概

ホームネットワークシステム (HNS) には, 多様なネット家電やセンサ機器が偏在的に接続される. したがって, 従来家電で用いられてきたリモコンに代わり, 新しい操作インタフェースの登場が期待されている. 本論文では特に, HNS における音声操作インタフェースの適用可能性を考察する.

従来から, 様々な機器を音声によって操作するアプリケーションが提案されている. しかしながら, それらの多くは, ユーザの発話を操作コマンドに単純にマッピングする手法に基づいている. そのため, ユーザが事前に音声コマンドを覚えておく必要があり, 機器の種類, 数が多くなりがちな HNS において学習のコストが非常に高くなる. また, システムが正しくコマンドを認識するかどうかは, 音声認識エンジンの性能や環境状態に大きく左右される. コマンドの誤認識による不具合や事故を最小限に抑えるためにも, 認識精度のさらなる向上に対する取り組みが必要である.

これらの問題を改善するために, 本論文では HNS のための新しい対話型コマンドインタフェースを提案する. 提案手法では, システムがユーザとの対話を通して, 機器名, 操作名, パラメータの順番で, ユーザが段階的にコマンドを組み立てる段階的コマンド構築機構を導入している. さらに, 操作対象の家電が現状で実行可能な操作のみをシステムが提示・解釈する, 家電状態を利用したコマ

---

\* 奈良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 修士論文, NAIST-IS-MT0651040, 2008年2月7日.

ンド提示・解釈機構を採用している。これら2つのキーアイデアにより、ユーザの学習コストを下げ、かつ、システムの音声誤認識を大幅に改善することができる。

また、提案手法を実装し、現在運用中のHNSに組み込んで評価実験を行った。実験の結果、提案手法では従来法に比べて、コマンド誤認識によるシステムの誤動作が平均75%削減された。また、SUS(System Usability Scale)を用いたユーザビリティ評価では、提案手法が10%ほど高い数値を示し、提案システムの使用性の高さが示された。

## キーワード

ホームネットワークシステム、音声認識、音声リモコン、対話型インタフェース、家電機器状態管理

# Proposal of Voice Command Interface to Operate Home Network System Using Appliance States\*

Hiroki Sakakibara

## Abstract

Various network appliances and sensor devices are connected to Home Network System(HNS) ubiquitously. Thus, the novel user interface is expected to appear for HNS control instead of remote controllers which are used for the conventional appliances. In this thesis, we especially consider about the application possibility of voice control interface for HNS.

There are many proposals of applications to operate various equipments by voice control conventionally. However, these proposals are based on the method by mapping users' utterances to the operation commands simply. Therefore, users must remember the operation commands beforehand to operate HNS. The type and number of appliances tend to be many in HNS, so the learning cost should be very high to operate. Whether the system recognize the command correctly depends on the performance of voice recognition engine and environmental condition. It is necessary to tackle with improvement of the voice recognition accuracy to minimize accidents due to wrong recognition.

In this thesis, we propose the novel interactive command interface for HNS. In this proposed method, we introduce **gradual command construction structure** which allows users gradually construct the command through the interaction

---

\* Master's Thesis, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT0651040, February 7, 2008.

with the user and the system. We also adopt **command giving / interpretation structure using appliances' state**. System can give and interpret only the executable method depends on current state of target appliance. It is able to reduce the learning cost of users and to improve the wrong recognition by the system greatly by applying these two key ideas.

We implemented the proposal method and did evaluation experiment by applying to HNS currently operating. In this experiment, we found that the wrong operation of proposal method due to wrong recognition was reduced by 75% in average compared to the conventional method. The usability experiment using SUS(System Usability Scale) showed that the proposal method gain 10% higher evaluation compared to the conventional method. This result showed that the proposal method has better usability than the conventional method.

**Keywords:**

Home Network System, voice recognition, voice remote controller, interactive voice interface, home appliance state management

# 目次

1. はじめに	1
2. 準備	3
2.1 ホームネットワークシステム (HNS)	3
2.2 家電機器操作インタフェース	6
2.3 既存の操作インタフェースの限界	7
2.4 音声による機器操作	8
2.5 音声操作インタフェースの HNS への適用可能性	8
3. 提案する HNS 操作インタフェース	11
3.1 システム要求	11
3.2 キーアイデア	11
3.3 対話による音声コマンドの段階的構築支援	13
3.4 家電の状態に応じたコマンド提示・解釈	20
4. 実装	24
4.1 システム構成	24
4.1.1 音声認識エンジン (Voice Recognition Engine)	24
4.1.2 システム管理モジュール (System Manager)	25
4.1.3 機器操作実行モジュール (Appliance Invoker)	26
4.1.4 コアモジュール群 (Core Module 群)	26
4.2 提案システムでの操作例	27
4.3 提供する操作	28
4.3.1 機器毎の個別操作	28
4.3.2 連携サービスによる操作	28
5. 評価実験	30
5.1 実験概要	30
5.2 実験環境	30

5.2.1	タスク	31
5.3	実験手順	32
5.4	結果と評価	33
5.5	考察	38
5.5.1	M1:発話文字数	38
5.5.2	M2:ユーザ発話失敗回数	38
5.5.3	M3:音声認識失敗回数	39
5.5.4	M4:機器動作間違い回数	39
5.5.5	M5:音声コマンド確認回数	39
5.5.6	M6:SUS スコア	40
5.5.7	要求の満足度	40
6.	まとめ	42
	謝辞	43
	参考文献	44



## 図目次

1	ホームネットワークシステム (HNS) . . . . .	5
2	フルコマンド法操作フロー図 . . . . .	16
3	提案システム操作フロー図 . . . . .	17
4	メソッドフィードバック・パラメータフィードバックフロー図 . . . . .	18
5	確認フィードバックフロー図 . . . . .	19
6	カーテンの状態遷移図 . . . . .	21
7	ライトの状態遷移図 . . . . .	22
8	提案システム構成図 . . . . .	24
9	実験結果表 . . . . .	36
10	System Usability Scale(SUS) シート . . . . .	37

## 表目次

1	フルコマンド法で用いるコマンド例 . . . . .	10
2	状態を用いたカーテンの音声操作例 . . . . .	21
3	状態を用いたライトの音声操作例 . . . . .	23
4	タスクリスト . . . . .	32

## 1. はじめに

ユビキタス技術の進歩に伴い、近い将来、宅内のあらゆるモノがネットワークに接続されることが予想されている [27]。ホームネットワークシステム (HNS) は、宅内の様々な家電やセンサをネットワークに接続し便利で快適なサービスを実現する次世代のユビキタスアプリケーションとして注目されている。家電をネットワークに接続することで、宅内外からの遠隔監視や遠隔操作、さらには複数機器の連携や外部ネットワークリソースとの融合など、様々な付加価値サービスの登場が期待されている [10]。現在、いくつかの大手家電ベンダは HNS をすでに商品化しており、今後ますます一般家庭への普及が期待されている [24][25][26]。

HNS における重要な研究課題のひとつとして、操作インタフェースが挙げられる。HNS はその構成機器の多様性・偏在性から、従来家電で用いられてきたリモコン主体の操作方法では限界があると考えられる。そのため、新しい操作インタフェースの期待が高まっている。そこで、本論文ではユーザが音声を用いて HNS を操作する操作インタフェースについての考察を行う。

従来から、様々な機器を音声を用いて操作するアプリケーションが提案されている。例えば、カーナビゲーションシステム [11] や情報案内システム [3] などが存在する。しかしながら、それらの従来アプリケーションの多くは、ユーザの発話を操作コマンドに単純にマッピングする手法 (フルコマンド法と呼ぶ) に基づいている。フルコマンド法では、ユーザが事前に音声コマンドを覚えておく必要があるため、機器の種類、数が多くなりがちな HNS において学習のコストが非常に高くなる。また、システムが正しくコマンドを認識するかどうかは、音声認識エンジンの性能や環境状態に大きく左右される。コマンドの誤認識による不具合や事故を最小限に抑えるためにも、認識精度のさらなる向上に対する取り組みが必要である。

これらの問題を改善するために、本論文では HNS のための新しい対話型コマンドインタフェースを提案する。提案手法は、以下の 2 つのキーアイデアに基づいている。まず一つ目は、段階的コマンド構築機構を提案する。提案法では、従来法のように連続した発話で一度に完全なコマンドを指定しない。システムはユーザとの対話を通して、機器名、操作名、パラメータの順番で、ユーザが段階的か

つ正確にコマンドを組み立てるための支援を行う．これによりユーザは完全なコマンドを覚えなくても HNS を効率的に利用できる．

二つ目は，家電状態を利用したコマンド提示・解釈機構である．HNS 内に配置されるネット家電はそれぞれ状態を持ち，状態に応じた振る舞いを提供する．HNS ではすべての家電の状態をネットワーク越しに取得することができる．したがって，対象の家電が現状態で実行可能な操作のみを，システムが提示・解釈することにより，言い間違いや雑音に伴う音声コマンドの誤認識や意図しない動作を大幅に削減することが可能となる．

また，本論文では，音声認識エンジン Julius for SAPI [16] を用いて，提案手法の実装を行った．さらに，開発したシステムを現在運用中の HNS(NAIST-HNS) [10] に組み込み，評価実験を行った．実験では，12 名の被験者に対して従来のフルコマンド法に基づいた音声操作と提案手法による音声操作タスクを行ってもらい，性能比較を行った．評価実験の結果，提案手法ではフルコマンド法に比べて，コマンド誤認識によるシステムの誤動作が，平均 75%削減された．また，SUS(System Usability Scale)[6] を用いたユーザビリティ評価では，提案手法が 10%ほど高い数値を示し，提案システムの使用性の高さが示された．

本論文の以降の構成は以下のとおりである．2 章では，HNS および操作インタフェースについての現状と課題を概観する．3 章では，課題解決のためのシステム要求を述べるとともに，提案手法について述べる．4 章ではシステムの実装について述べ，5 章で評価実験を述べる．最後に，6 章で結論とともに今後の課題を述べる．

## 2. 準備

### 2.1 ホームネットワークシステム (HNS)

ホームネットワークシステム (HNS) は一般に、宅内のホームネットワークに接続された複数のネットワーク家電とそれらを制御するホームサーバから構成される。また各ネットワーク家電は、ホームネットワークに接続する能力のほかに機器動作状態の獲得や機器の振る舞いを制御することが可能な Application Program Interface(API) を備えている。一般的に、この API を実行するためには、操作する機器名、実行したい操作名、実行に必要なパラメータの 3 つの情報が必要である。

図 1 に HNS の構成例を示す。図に示すとおり、テレビ、DVD プレイヤー/レコーダ、エアコン、照明、カーテン、換気扇といったホームネットワークに接続された多様な家電機器とその API を利用することで、HNS は付加価値の高いサービスをユーザに提供することが可能である。代表的な付加価値サービスとして、以下の二つが実際に開発・提案されている。

- 連携サービス

連携サービスとは、複数の家電を同時もしくは連続的に制御することで、ユーザの目的を最小限の操作で実現するサービスである。例えばユーザが DVD を視聴したい、という目的を持っているとする。この目的を達成する際、通常、ユーザは TV、DVD プレイヤー、音響装置等の電源を投入し、それらの入出力を適切に設定した後に DVD プレイヤーを再生する、場合によっては照明を落とす、という操作を行う必要がある。しかし、上記動作を「DVD シアターサービス」という連携サービスとして予め構築しておくことで、ユーザは DVD を視聴する際、DVD シアターサービスを呼び出すだけで簡単に実現することができる。[18] では、連携可能な機器の組み合わせこそ制限されるものの、上記のような DVD プレイヤーとテレビ・スピーカ間の連携サービスを実現している。

- 遠隔サービス

遠隔サービスとは、家庭内の機器を宅外から監視/制御するサービスである。

家庭内ネットワークはインターネットに接続されており，インターネットからホームサーバ等のゲートウェイを経由して制御することが可能である．例えばユーザが帰宅前にエアコンを稼働させておきたいと考えた場合，ユーザは携帯電話などのインタフェースを通してエアコンの電源を投入し，適切な運転モードや温度を設定する．ホームゲートウェイはエアコンの API を利用し，ユーザが設定した通りにエアコンを制御する [19]．他のアプリケーションとして，[20] では，宅外から宅内機器の動作状態を確認する機能を提供している．このアプリケーションでは，携帯電話から専用のサーバを介して宅内情報コントロール端末に接続する．宅内機器はこの端末にネットワークを介して接続されているので，洗濯機や冷蔵庫，電子レンジといった家電の状態を取得することが可能である．

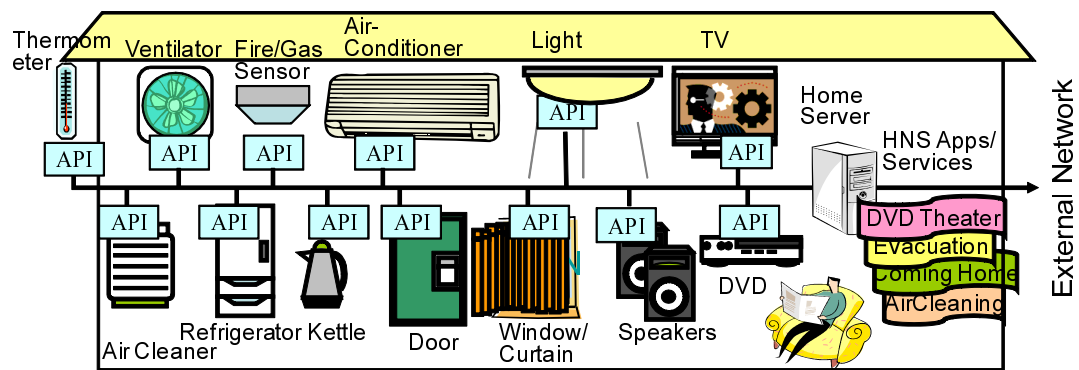


図 1 ホームネットワークシステム (HNS)

## 2.2 家電機器操作インタフェース

2.1 で述べた機器の操作インタフェースとしては，コントローラデバイスにユーザが手を触れて操作するタイプのものが一般的である．代表的なものとして各機器に備え付けの専用リモコンがある．その他，複数の機器の操作を学習できる汎用学習リモコン，ビルトイン型の操作パネル，携帯電話などが存在する．

### 専用リモコン

個別機器専用のリモコンは HNS 環境だけでなく，従来の様々な家電に対して最もよく用いられてきたインタフェースである．機器に備わった機能のほぼすべてを利用できるが，操作対象によってはボタンの数が増え，操作が煩雑になりやすい．

HNS 環境では，連携サービスなど HNS 特有のサービスを利用するための専用リモコンも存在する．この場合の専用リモコンは，いくつかの HNS 特有のサービスなどを呼び出すための統合的なインタフェースである場合が多い．これらは予め用意されたサービスを利用するには有用であるが，新たな機器やサービスが導入された場合に簡単に対応できず，利点である統合性を保つことが難しい．

### 汎用学習リモコン

専用リモコンは機器毎に必要となるため，機器の数が増えたとリモコンの数も増えてしまう．汎用学習リモコンは専用リモコンが出力する信号を学習してボタンに割り当てることで，1 台で専用リモコンの代替とすることができる．使用するリモコンの数が減らせるという利点がある一方，学習した信号の数によって汎用学習リモコン自体の操作が煩雑になる可能性がある．

### ビルトイン型操作パネル

ビルトイン型操作パネルは主に照明や浴室制御，エアコンなどに用いられている．壁面に固定されているため，リモコンのように紛失するおそれが無く，液晶

パネルなどとの併用により操作が平易であるという利点があるが、(1) 操作のためにユーザが移動しなければならない、(2) 機器の変更や追加に対応する際の柔軟性に欠ける、といった問題点がある。

## 携帯電話

携帯電話は主に遠隔サービスを利用する際に用いられる。現在普及している携帯電話の多くはインターネットに接続する機能を持っており、ブラウザや独自アプリなどを利用して家庭の機器の状態確認/制御をするインタフェースが提供されている [22]。

## 2.3 既存の操作インタフェースの限界

これらのいわゆるリモコンやパネルタイプのインタフェースの場合、ユーザは実現したい機器の振る舞いを提供するリモコンとそのボタン配置を学習しなければならない。そのため、HNS に配備される機器の種類や数が増加すると、ユーザの学習コストが増加し、操作が煩雑になる傾向がある。特に高齢者や障害者については、多くのユーザがリモコン操作に不便を感じており、特に下記のような不便さが調査によって明らかになっている [9]。

### リモコンのボタンや液晶表示の視認が困難

エアコンやビデオのリモコンのようにボタンの数が多いものは、弱視や視覚障害を持つユーザにとってはラベルの文字が小さく視認が困難である。そのため、ボタンの配置や対応する操作を記憶しなければならず、複雑な操作を実行しづらい。結果として電源の ON/OFF 程度の操作しか実行できない。

### 機能やボタンの数が多すぎる

高齢者のユーザに多い意見で、機能が多くの機器やリモコンのボタンの数が多い機器は操作が難しく使い方がわからないため、すべての機能の半分から四分の一程度の機能しか利用できない。



また，高齢者や障害者だけでなく健常者が利用する際でも，室内が暗い場合リモコンの位置が分かりにくい，料理中で手がふさがっている場合は操作ができない，といった不便さがある．以上のような問題点から，リモコンやパネルといった操作のための特別なデバイスを利用する必要がないインタフェースが現在期待されている．

## 2.4 音声による機器操作

操作のための特別なデバイスを必要としない次世代の操作インタフェースとして，音声操作インタフェースが提案されている．

音声操作インタフェースとは，音声認識エンジン [1][2] を利用してユーザの発話内容を認識し，認識結果と特定の機器の振る舞いを対応付けることが可能な操作インタフェースである．実際の商品として，カーナビゲーションシステム [11] や対話型音声情報案内システム [3] などが実用化されており，[12] は家電機器のインタフェースとして利用されている．

内田ら [13] が提案する音声を利用した家電操作インタフェースでは，ユーザの自然な発話による家電操作を提案している．例えば，「エアコンの設定温度は，今何度．」とユーザが発話すると，提案システムは「18度に設定されています．」と返答を返す．同様にして，「もっと暖かくして」という発話が行われると，システムは「20度に設定しました．」と返答すると同時に，エアコンの温度設定を変更する．

他にも，テレビ番組をビデオで録画する操作を音声で行うもの [4] や，頭の向きと音声を併用した四肢まひ者向けの家電機器操作インタフェース [5]，駅の情報案内システムへの適用 [3] などが提案されている．

## 2.5 音声操作インタフェースのHNSへの適用可能性

音声操作インタフェースをHNSに適用した事例は，我々の知る限り未だに存在しない．しかしながら，既存の音声インタフェース (例えば [13][14][12]) を用いて，発話された内容をコマンドと解釈し家電APIを実行することで，HNSでの

音声操作は可能である。例えば、ユーザが「テレビ、オン」や「テレビを4チャンネルにして」といった発話内容をシステムが音声認識し、それぞれ `TV.on()`、`TV.setChannel(4)` という API を実行することで、音声を利用した HNS の操作が実現可能である。

このように、HNS において従来技術を使用して、ユーザの発話を操作コマンドと認識し対応する家電 API を呼び出す手法を、本稿ではフルコマンド法と呼ぶことにし、これをもとに提案手法との比較を行う。

フルコマンド法では、ユーザの一度の発話が API 実行に必要な情報全て含んだ完全なコマンド (=フルコマンド) をあらわしていることに注意されたい。表1に、フルコマンド法を実現するための音声コマンド表の例を挙げる。

フルコマンド法は、既存技術を用いて容易に実装可能である反面、以下のような問題点が想定される。

P1: 操作対象機器が増えると操作コマンドの数も増える HNS に配備される機器の種類・数が増えると、それらの機器を制御するための操作コマンドの種類・数も増える。フルコマンド法では、ユーザは操作コマンドを事前に知っている必要があるため、操作コマンドの数の増加はユーザのコマンド学習コストの増大に容易に繋がる。結果として、機器をユーザの思い通りに操作することが困難になったり、操作コマンドを忘れてしまって操作できないといった状況が発生する可能性がある。

P2: 音声認識を失敗すると、制御される機器の振る舞いがユーザの意図しないものになる可能性がある HNS 環境における音声インタフェースは、主に音声認識エンジンの精度やユーザの発話環境の問題で、従来のリモコン操作などと比べるとユーザの発話した操作コマンドを正しく認識する確率が低い(例えば、[21]の音声認識精度は88.3%である)。特に HNS 環境では、家電機器の振る舞いがユーザの日常生活と密に影響するため、このような誤認識によってユーザの意図したとおり機器が制御されないという状況が危険やユーザのストレスを引き起こす要因となることがある。

機器名	コマンド	操作の内容
テレビ	テレビオン	テレビの電源を入れる
	テレビオフ	テレビの電源を切る
	テレビ音量上げて	テレビの音量を上げる
	テレビ音量下げて	テレビの音量を下げる
	テレビチャンネル進めて	テレビのチャンネルを次の局にする
	テレビチャンネル戻して	テレビのチャンネルを前の局にする
	テレビ入力切替	テレビの入力を DVD 地上波で切り替える
	テレビ消音	テレビを消音にする
DVD	DVD オン	DVD の電源を入れる
	DVD オフ	DVD の電源を切る
	DVD 再生	DVD を再生する
	DVD 停止	DVD を停止する
	DVD チャプター進めて	DVD のチャプターを進める
	DVD チャプター戻して	DVD のチャプターを戻す
	DVD 音量上げて	DVD の音量を上げる
	DVD 音量下げて	DVD の音量を下げる
	DVD 消音	DVD を消音にする
カーテン	カーテン開けて	カーテンを開ける
	カーテン閉めて	カーテンを閉める
空気清浄機	空気清浄機オン	空気清浄機の電源を入れる
	空気清浄機オフ	空気清浄機の電源を切る
	空気清浄機一発強	空気清浄機を一発強モードにする
	空気清浄機イオン	空気清浄機をイオン発生モードにする
扇風機	扇風機オン	扇風機の電源を入れる
	扇風機オフ	扇風機の電源を切る
	扇風機風量強	扇風機の風量を強風にする
	扇風機風量中	扇風機の風量を中風にする
	扇風機風量弱	扇風機の風量を弱風にする
	扇風機ルーパー	扇風機のルーパー回転を入/切する
ライト	ライトオン	ライトをつける
	ライトオフ	ライトを消す
	ライト全灯	ライトを全灯（最大照度）にする
	ライト豆球	ライトを豆球（最小照度）にする
	ライトタイマー	ライトのタイマーを設定する

表 1 フルコマンド法で用いるコマンド例

## 3. 提案する HNS 操作インタフェース

### 3.1 システム要求

2.5 で述べたフルコマンド法の問題点にもとづき，本稿で想定する HNS 環境における音声インタフェースが満たすべき要求を以下に述べる．

R1: 機器数が増加した場合でも，ユーザの操作コストと音声コマンド学習コストを増やさない 機器が増大するとそのための操作方法が多様になり，ユーザの操作コストと学習コストが増大するという問題は，これまでの HNS 操作インタフェース全般にとっての問題点でもある．特に音声インタフェースでは，音声コマンドに含まれるキーワード全てを記憶する必要があるため，操作・学習コストがリモコン操作などよりも高くなる可能性がある．このような状況を避けるためには，機器の数が増えてもユーザが記憶・学習しなければならない音声コマンドの量が変化しないような仕組みが必要である．

R2: 音声コマンドの誤認識発生時のフェイルセーフ機構と誤認識発生を削減するための仕組みを実現する 音声コマンドの誤認識は避けられないものであるため，ユーザが意図していない動作をしないような機構が必要である．ここで，ユーザ発話認識後の機器制御前に確認を行う機構は最もシンプルな解決策のひとつである．また，フェイルセーフ機構だけでなく，HNS から得られる情報にもとづいて誤認識発生を削減するための仕組みを実現することは，家電機器操作時のユーザストレスの削減といった観点からも重要であると考えられる．

### 3.2 キーアイデア

K1:対話による音声コマンドの段階的構築支援 ユーザに対して一度に全てのコマンドの発話を求めずに，システムとの対話を通して段階的にコマンドを構築可能なインタフェースを作成する．我々が提案するインタフェースでは，ユーザは必ずしも一度に機器の API 実行に必要な全ての情報を発話する必要は無い．ユーザが機器名を発話すると，その機器に応じた操作が提示される．その後ユーザの

選択（発話）した操作名に応じて、操作が受け入れ可能なパラメータを順にシステムがガイドする．このように段階的にフルコマンドの構築を支援することで、ユーザが事前に知っていなければならない音声コマンドに関する知識を大幅に減らすことができ、操作に対するコストも低減することができる．

**K2:家電の状態に応じたコマンド提示・解釈** HNS 内に配置されるネット家電はそれぞれ状態を持ち、状態に応じた振る舞いを提供する．HNS ではすべての家電の状態をネットワーク越しに取得することができる．したがって、対象の家電が現状態で実行可能な操作のみを、システムが提示・解釈することにより、言い間違いや雑音に伴う音声コマンドの誤認識や意図しない動作を大幅に削減することが可能となる．

### 3.3 対話による音声コマンドの段階的構築支援

ユーザが機器を制御するための音声コマンドには、操作する機器名、実行したい操作名と、必要に応じて実行に必要なパラメータの2つないし3つの情報が含まれている。表1の扇風機を例に挙げると、扇風機の風量を強にするためには、ユーザは「扇風機、風量、強」のように機器名、操作名、パラメータの順に正しく音声コマンドを発話しなければならない。また、カーテンの場合は、開閉のために「カーテン、開けて」または「カーテン、閉めて」と発話する必要があるが、こちらは機器名と操作名だけで動作し、パラメータに相当する情報を必要としない。

図2は、従来のフルコマンド法におけるユーザ発話から機器API実行に至るまでのフローチャートである。このとき音声インタフェースは、受け付けた音声コマンドに機器制御に必要な全ての情報が含まれているか、すなわちフルコマンドであるかどうかをまず判断する。インタフェースは音声コマンドがフルコマンドとして正しければ、その内容を確認後に機器APIを実行する。必要な情報が含まれていない場合や間違っている場合等の理由で正しくないと判断されれば、コマンド入力のやりなおしを受け付ける。

一方、我々が提案する段階的な音声コマンド構築のための対話型インタフェースでは、ユーザの発話する音声コマンドが必ずしも最初から完全である必要はない。提案インタフェースは、事前に登録された表1に相当する音声コマンドのDB(音声コマンドDB)を利用し、不完全な音声コマンドが入力された場合に何が不完全であるかをユーザにフィードバックする。ユーザはそのフィードバック内容に対応し、不完全部分の音声コマンドを追加で発話することで完全な音声コマンドを構築し、対応する機器APIを呼び出すことができる。

図3、図4、図5は提案インタフェースのフローチャートを表している。以下にフローチャートに対応した提案インタフェースの処理プロセスを示す。

STEP1: ユーザの発話した内容(発話A)を音声コマンドとして解釈し、その発話Aで指定されたコマンドが音声コマンドDBに含まれているかどうかの確認を行う。ここで含まれているかどうかの判断は、発話Aが(1)機器名・操作名・パラメータ、(2)機器名・操作名、(3)機器名、のどれかであるという想定において、その発話内容が音声コマンドDB内の任意のコマンドの

一部分であるかどうかを検証することで行われる。含まれていない場合は再度ユーザに発話を促し、含まれている場合はSTEP2に移行する。

STEP2: 有効性の確認後、音声コマンドの完全性(フルコマンドになっているかどうか)を評価する。発話Aで指定されたコマンドがフルコマンドであると判断できた場合、STEP10に移行する。フルコマンドでなければ、STEP3に移行する。

STEP3: このSTEPでは、発話AがSTEP1の(2)機器名・操作名、(3)機器名、のどちらであるかを確認する。(2)、すなわちパラメータが不足している場合はSTEP8に移行し、(3)、すなわち操作名とパラメータが不足している場合はSTEP4に移行する。

STEP4: 発話Aに含まれる機器のもつ操作名の集合を音声コマンドDBより抽出した後に、その操作名を音声でユーザにフィードバックし、STEP5へ移行する。

STEP5: STEP5では、STEP4でフィードバックした操作名にもとづいてユーザが追加の音声コマンド(発話B)を発話するのを待つ。ここで発話される内容は、(4)操作名、(5)操作名・パラメータ、のどちらかであることを想定している。(4)(5)どちらの場合においても、操作名が発話Aに含まれる機器が保持するものであることを音声コマンドDBを用いて確認した後に、STEP6に移行する。

STEP6: ここでは発話Aあるいは発話AとBの組み合わせによって指定された機器名・操作名の組がパラメータを必要とするかどうかを音声コマンドDBにもとづいて確認する。パラメータを必要としない組み合わせの場合はSTEP10に移行し、パラメータを必要とする場合はSTEP7に移行する。

STEP7: 発話Bにパラメータが含まれているかどうかを判断する。含まれていた場合、STEP10に移行し、含まれて居なかった場合はSTEP8に移行する。

STEP8: このSTEPに到達した時点で、機器名と操作名が指定されている。そ

のため，ここでは対象の機器名・操作名の持つパラメータの集合をユーザにフィードバックし，STEP9に移行する．

STEP9: STEP8でフィードバックされた内容にもとづいてユーザが追加の音声コマンド（発話C）を発話するのを待つ．このSTEPでは，発話Cが発話Aもしくは発話A，Bによって指定された機器名・操作名に対応するパラメータであることを確認した後に，STEP10へ移行する．

STEP10: 発話A，B，Cの内容を組み合わせた後に，その音声コマンドの内容をユーザにフィードバックする．その後ユーザからの返答が「はい」であれば，その音声コマンドに対応する機器APIの呼び出しを行い，「いいえ」であればSTEP1に戻る．

以上のプロセスが示すように，ユーザは機器名を記憶しておきさえすれば，提案インタフェースによって提示されるメソッド名，パラメータ名の集合のフィードバックにもとづき，音声コマンドを段階的に構築することができる．一方で，既に記憶している音声コマンド（フルコマンド）については，従来のインタフェースと同じプロセスで機器を制御することが可能であるため，不必要なプロセスをユーザに強いることは無い．



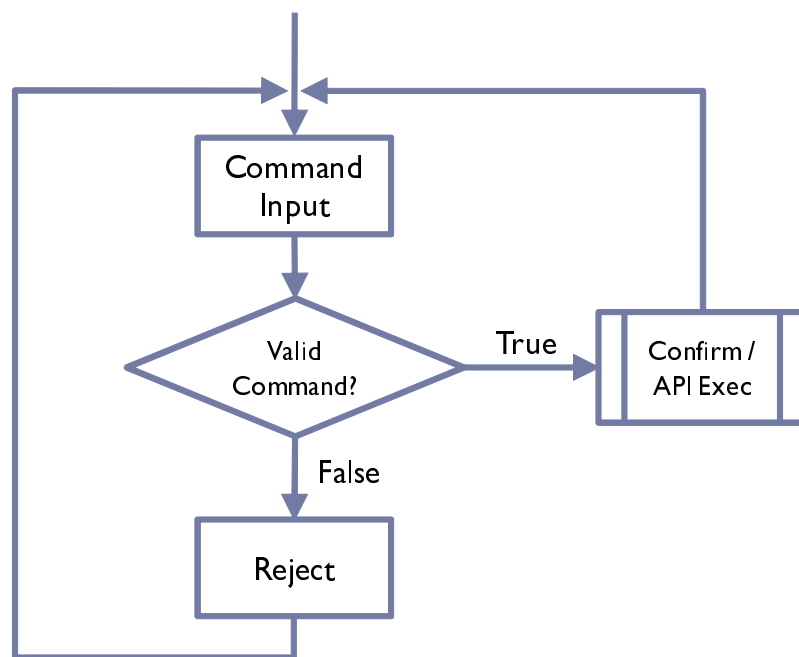


図 2 フルコマンド法操作フロー図

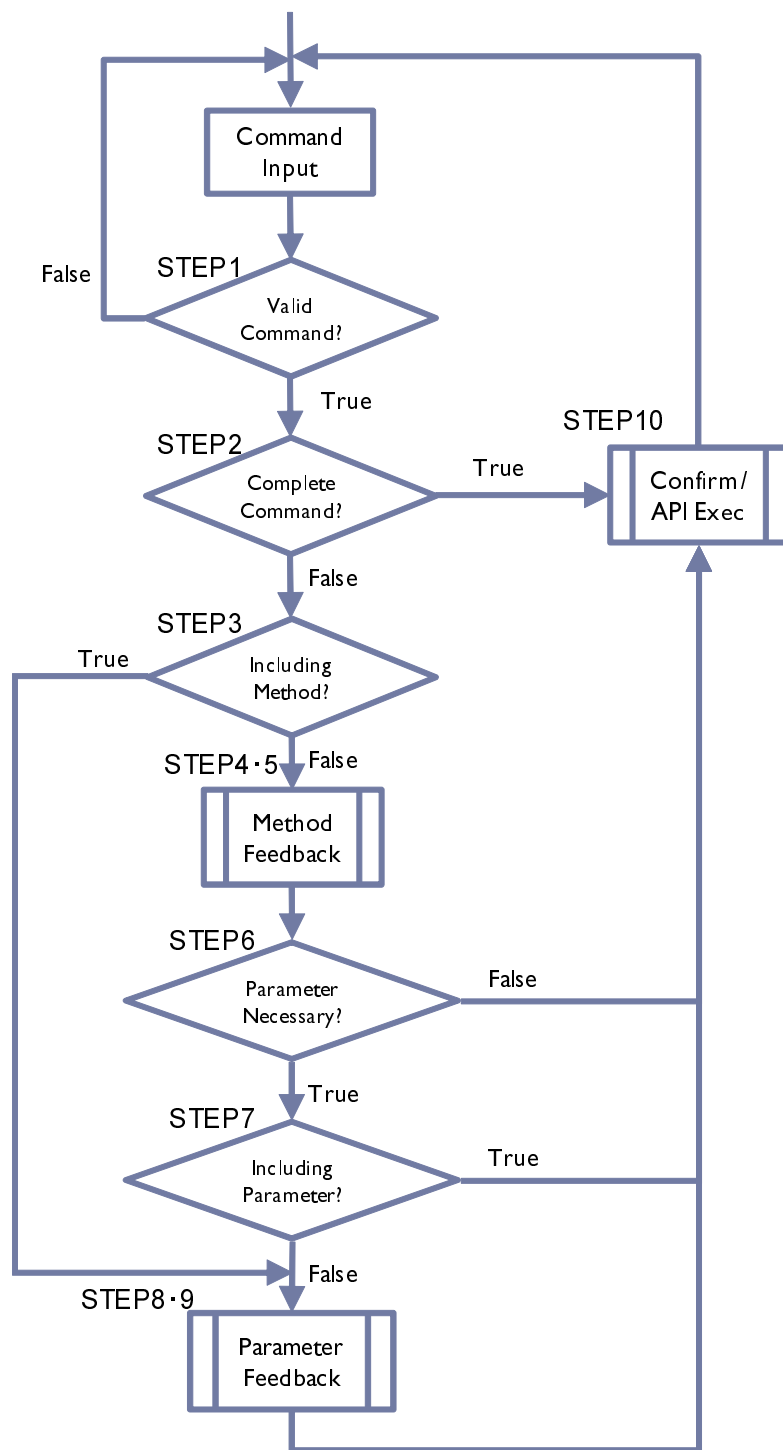


図 3 提案システム操作フロー図

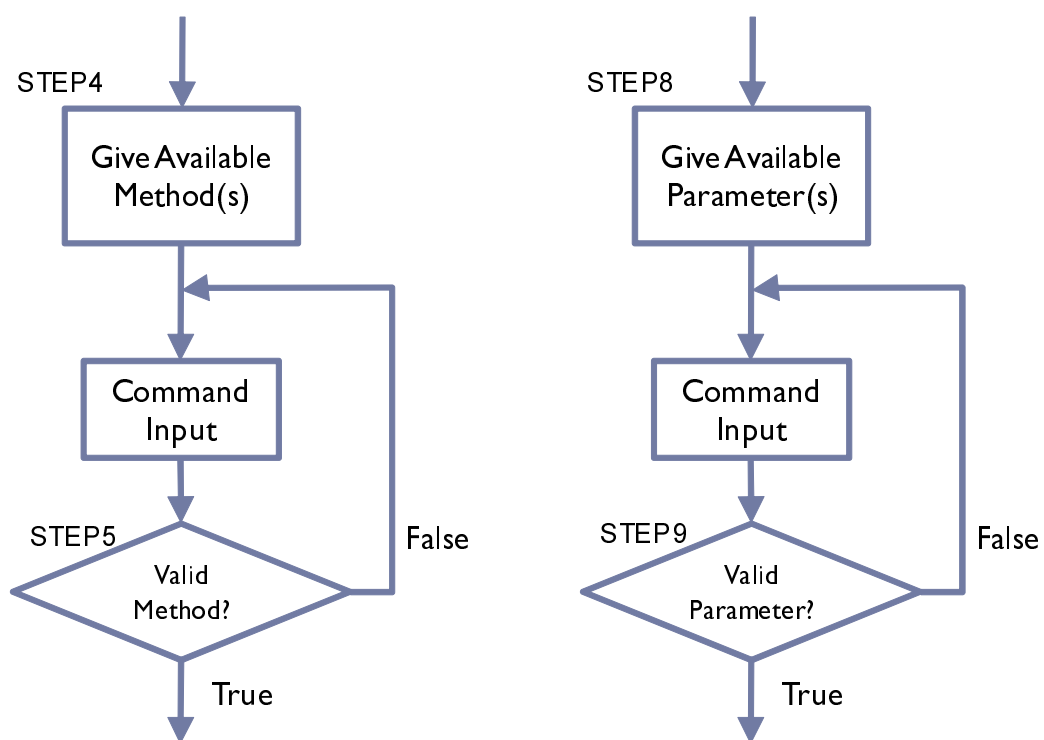


図 4 メソッドフィードバック・パラメータフィードバックフロー

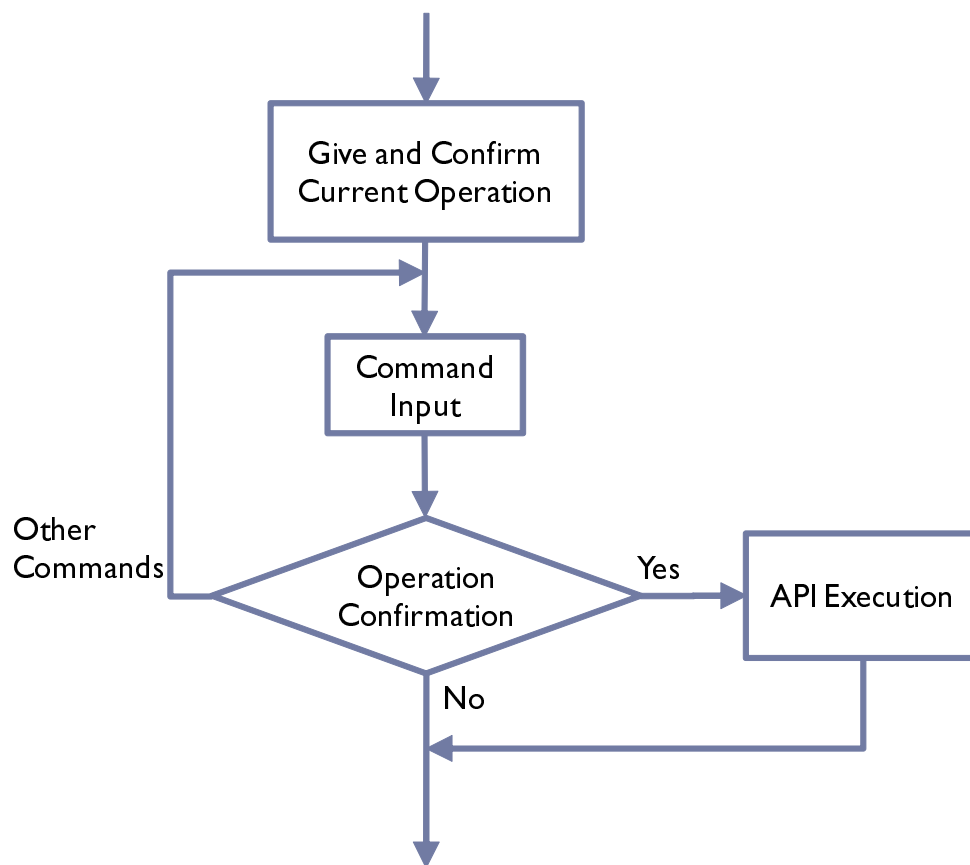


図 5 確認フィードバックフロー図

### 3.4 家電の状態に応じたコマンド提示・解釈

HNS に配備される家電はそれぞれ内部状態を持っており、各状態ごとに実行可能な操作、およびパラメータが決まっている。例えば、テレビは電源の状態を持っている。電源がオフの状態では、音量を上げたりチャンネルを選択する操作は実行できない。唯一、電源をオンにする操作だけが実行可能である。また、一般的な自動カーテンであれば、「開」「閉」という二つの状態を持っている。「カーテンを開ける」という操作は、カーテンの状態が「閉」の時のみ実行可能である。同様に、「カーテンを閉める」は状態が「開」の時のみ有効である。

このように、各機器が元来持っている操作は複数存在するが、機器の現在状態に着目すれば、そのうち実行可能なものだけに絞り込むことができる。このことを利用すれば、ユーザからの音声コマンドをより高い精度で解釈・実行することが可能となる。さらに、状態に応じて実行可能な操作パラメータのみを、3.3 節で述べたフィードバック機構で提示すれば、ユーザの機能選択を効率的に支援できる。また、実行可能な操作がひとつしかない場合には、操作の選択プロセスを省略することも可能である。なお、HNS における各家電は内部状態を取得するための API を備えており、家電の状態はホームサーバ経由で取得することができる。

以上のことを踏まえ、家電の状態に応じたコマンド提示・解釈を行う家電状態モニタ (Appliance Status Monitor) を実現する。家電状態モニタは、HNS に配備される各機器を状態遷移機械 (Finite State Machine, FSM) でモデル化して保持する。家電状態モニタは、操作する家電機器名を与えるとホームサーバに当該機器の現在状態を問い合わせる。次に、取得した現在状態とその機器の FSM モデルを照合し、現状態で実行可能な操作の集合 ( $E$  とする) を取得する。ユーザからの音声コマンドに対応する操作が  $E$  の中に存在しなければ、コマンドを棄却する。

また、音声コマンドが不完全な場合、 $E$  を操作可能な候補としてユーザに提示する。また、 $|E| = 1$ 、すなわち、実行可能な操作が 1 つしか無い場合には選択の必要が無いので、ユーザには実行するかしないかの決定を「はい | いいえ」で決定させる確認プロセスに進む。これにより、ユーザのコマンド組み立ての手間を効率化できる。

例として、自動カーテンの制御シナリオを用いて説明する。図 6 にこのカーテ

ンの状態遷移図を示す．今，カーテンの状態が Open(=開いている)と仮定するとき，ユーザとシステムの対話は表 2 のようになる．まず，1 行目の音声コマンド「カーテン開けて」では，カーテンは既に開いているため，システムはコマンドを棄却する．3 行目の「カーテン」は，コマンドとして不完全である．したがって，システムは操作候補をフィードバックすべきであるが，現状態で実行可能な操作は close() のみである．よって，操作選択プロセスを省略し，close() を実行すべきかどうかの確認プロセスに進んでいる．ユーザは，改めて「カーテン閉めて」を与える必要はなく，「はい」「いいえ」で答えるだけでよい．

表 2 状態を用いたカーテンの音声操作例

No.	ユーザ	システム	カーテン状態
1	「カーテン開けて」		Open
2		「カーテンは既に開いています」	
3	「カーテン」		
4		「カーテンを閉めますか？」	
5	「はい」		
6		「カーテンを閉めました」	Close
7	「カーテン開けて」		
8		「カーテンを開けますか？」	
9	「いいえ」		
10		「操作を中止します」	

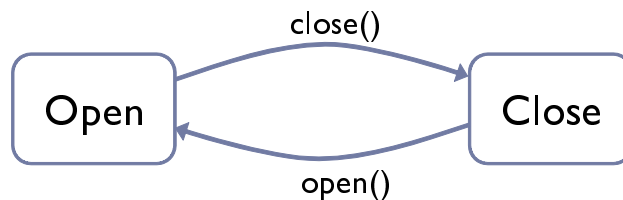


図 6 カーテンの状態遷移図

もう少し複雑な例を図 7 のライトを用いて説明する．このライトは，OFF(消

灯), Full(全点灯), LED(豆球)の3つの状態を持っている。はじめにこのライトがOFF状態にあると仮定する。表3に音声操作例を示す。この例では, ユーザがライトを全点灯した後豆球にする際, システムは状態 Full で実行可能な2つの操作をフィードバックしている。その後, ユーザは豆球にする操作を選択している。ライト自体は, 4種類の操作を持っているが, 状態を考慮することで受理すべきコマンドの候補を絞り込むことができる。

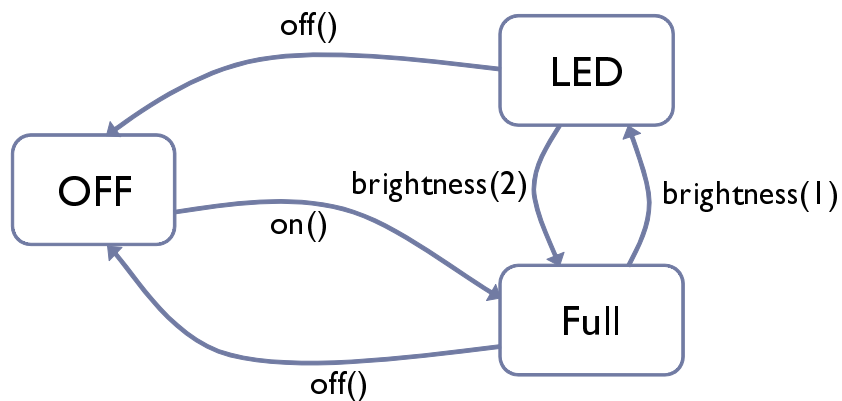


図7 ライトの状態遷移図

音声を用いたインタフェースでは, その性質上区別しにくい音声コマンドを誤認識することが多い。例えば「オン」と「オフ」, 「中」と「9」など, 発音が紛らわしいコマンドに関しては, システムが非常に高い確率で誤認識してしまう。この認識率は音声認識エンジンの性能に大きく依存し, 従来のフルコマンド法ではそれ以上改善を期待できない。

一方, HNS に適用する目的においては, 本節で提案した状態を用いたコマンド認識手法を導入することで, 音声認識エンジンの性能に頼らない方法で, 認識率を向上することができる。結果として, 要求 R2 を効果的に達成することができる。

表 3 状態を用いたライトの音声操作例

No.	ユーザ	システム	ライト状態
1	「ライト」		OFF
2		「ライトをつけますか？」	
3	「はい」		
4		「ライトをつけました」	Full
5	「ライト」		
6		「ライトを消しますか？それとも豆球にしますか？」	
7	「豆球」		
8		「豆球にしました」	LED



## 4. 実装

### 4.1 システム構成

3章で提案した2つのキーアイデアに基づいて、HNS 機器を操作するための音声インタフェースシステムを実装した。提案システムの構成図を図8に示す。提案システムは、次に挙げるモジュールによって構成されている。

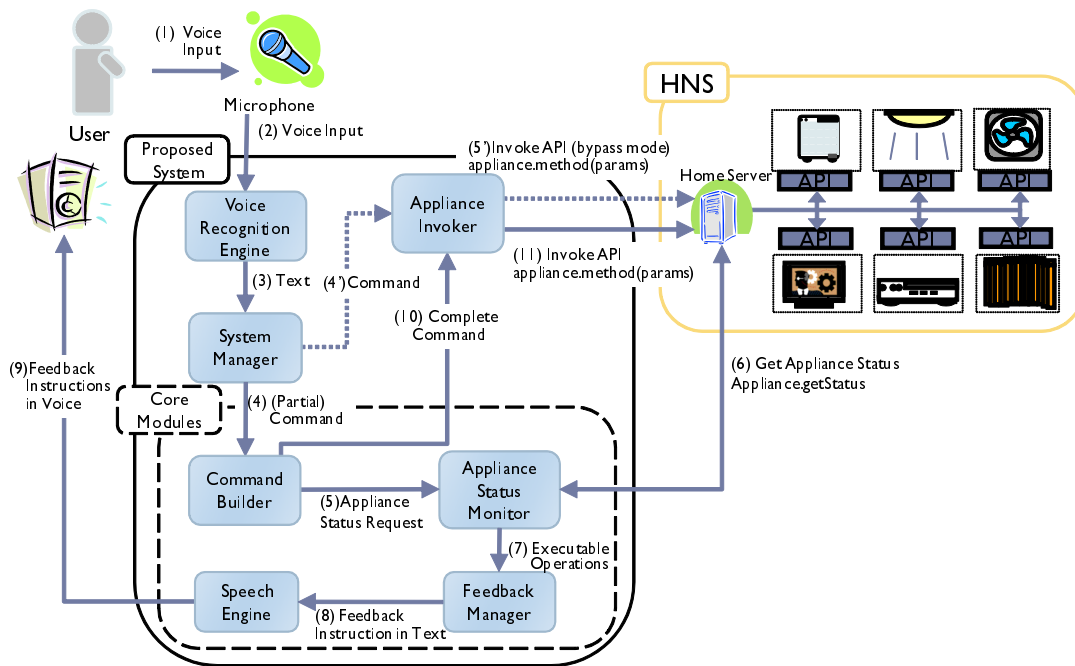


図8 提案システム構成図

#### 4.1.1 音声認識エンジン (Voice Recognition Engine)

マイクやヘッドセットから入力されたユーザの発話を、音声認識によりテキストに変換する。提案システムでは、オープンソースの音声認識エンジン、Julius for SAPI [16] を用いた。Julius は、日本語の音声認識エンジンとしては非常に評価が高く、特に以下の特長を持っているため採用した。

- (a) 適用するアプリケーションに応じて、認識すべき語彙や文法を自由にカスタマイズできる。
- (b) プログラムに組み込むための API が用意されている。

特長 (a) を用いて、まず HNS のための音声認識モデルを作成した。具体的には、HNS 操作のための 52 種類の語彙と 29 種類の文法を Julius に登録した。これにより HNS 操作に必要な語のみを定義したモデルを構築できる。登録されていない語を発話した場合は、「認識失敗」として棄却することができる。

また、(b) については、Microsoft 社が提供する標準的な音声 API、Microsoft SAPI Speech SDK 5.1 [15] から Julius を利用できるように設定した。これにより、システムから汎用的な手続きで Julius を利用できる。

Julius への音声入力装置は、Plantronics 社製ヘッドセット Voyager 510 を用いた。この機器は Bluetooth での無線接続が可能で、耳にかけることでハンズフリーの操作が可能である。有線接続のマイクを使用した場合、手で持つかテーブルなどに固定しておく必要があるため、2.3 節で述べた専用リモコンの問題点を十分に解消しているとは言えないので、ハンズフリーのマイクを用いることとした。

#### 4.1.2 システム管理モジュール (System Manager)

音声認識エンジンから渡されるテキストを解析し、有効な音声コマンドかどうかを判断する。ユーザの発話において、HNS への音声コマンドとそれ以外の一般的な発話（会話や雑談）とを区別するために、ユーザが「コンピュータ」という発話を契機にコマンド受付モードに切り替える実装にした。

ユーザは「コンピュータ」と発話するとシステムはコマンド受付モードに遷移し、音声コマンドを待つ。また、提案手法における段階的コマンド構築の段階管理を行い、入力された音声コマンドが、機器名、操作名、パラメータのどの段階に該当するかを分類、コマンドが不足しているかどうかを判断する。

また、従来法（フルコマンド法）との比較を行うため、任意の音声コマンドをフルコマンドとして認識し、直接 API を起動するモード（フルコマンドモード）も実装している（図 8 のフロー（4'）参照）。

#### 4.1.3 機器操作実行モジュール (Appliance Invoker)

構成されたコマンドに従って，HNS のホームサーバに機器 API 実行をリクエストする．本研究で利用した NAIST-HNS では，機器 API が SOAP/RPC を用いた Web サービスとして公開されていた．これらを提案システムから利用するために，SOAP プロトコルの C++ 実装である gSOAP[28] を用いて実装した．

#### 4.1.4 コアモジュール群 (Core Module 群)

図 8 下部の破線で囲まれた部分は，提案手法の核となる 2 つのキーアイデアを実装する．以下の 4 つのモジュールで構成される．

- (1) コマンド構成モジュール (Command Builder) 提案する段階的コマンド構築において (3.3 節)，コマンドの構成過程を管理する．コマンドが不完全な場合，コマンドを一時的に記憶し機器状態モニタに操作対象機器の状態を取得するように要求する．一方，必要なコマンドがすべて入力されたら操作実行モジュールに完全なコマンドを渡す．
- (2) 家電状態モニタ (Appliance Status Monitor) 3.4 節で述べた家電状態モニタの実装である．コマンド構成モジュールから対象機器の状態リクエストを受けると，HNS のホームサーバにアクセスし，当該機器の現在状態を取得する．また，その機器の FSM モデルを照会して，現状態で実行可能な操作・必要なパラメータを判断し，フィードバック管理モジュールに送る．
- (3) フィードバック管理モジュール (Feedback Manager)  
家電状態モニタによって渡される実行可能な操作の集合に基づき，ユーザへ最適な提示操作フィードバックをテキストとして生成する．
- (4) スピーチエンジン (Speech Engine)  
テキストで与えられる操作指示を音声に変換し，音声によるフィードバックを行う．

ユーザは音声フィードバックを受け、必要な操作名およびパラメータを再度音声コマンドで指定する。コマンド構成モジュールに完全なコマンドが構築されるまで、繰り返しユーザからの音声コマンドを受け付ける。

実装したシステムの諸元は以下のとおりである。まず、開発言語はC++で、規模は2,746行(音声認識エンジンは除く)である。Microsoft社のVisual Studio 2005を用いて開発した。プラットフォームは、Windows XP Service Pack 2、マシンの仕様は、Intel CoreDuo CPU 1.66GHz, RAM 1.5GByteである。

## 4.2 提案システムでの操作例

図8を用いて提案システムでの操作シナリオを説明する。ここでは、テレビの電源がオンのときに電源をオフにする操作を示す。

各処理の番号は図中の処理を示す矢印の番号と対応している。

- (1) ユーザが「テレビ」と発話する。
- (2) 音声認識エンジンに発話された音声を入力する。
- (3) 入力音声をテキストに変換して、システム管理モジュールに渡す。
- (4) 入力テキストをコマンドとして解析する。入力が機器名のみでコマンド構成モジュールにコマンドを渡し一時的に記憶しておく。
- (5) 機器状態モニタに、コマンドに対応する機器の状態を取得するよう要求する。
- (6) 機器状態モニタがHNSのホームサーバにアクセスし、操作対象機器の状態を取得する。
- (7) 取得した機器の状態から現在実行可能な操作を列挙し、フィードバック管理モジュールに渡す。
- (8) 実行可能な操作に対応したフィードバックを生成し、スピーチエンジンにテキストとして渡す

- (9) テキストを音声に変換しユーザにフィードバックを行い，ユーザの次の発話を待つ．
- (10) (9) までの操作を繰り返し行い，必要な操作とパラメータが入力されコマンドが完成したら機器操作実行モジュールにコマンドを渡す．
- (11) コマンドに従ってHNSのホームサーバに機器の操作をリクエストする．

### 4.3 提供する操作

提案システムでは，HNSに接続された家電機器の操作として，各機器毎に操作する個別操作と複数の機器を連携して動作させる連携サービスを提供する．以下に本提案システムで提供する操作の詳細を述べる．

#### 4.3.1 機器毎の個別操作

機器毎の個別操作を行う場合，ユーザはまず機器名を発話することによって操作対象の機器を指定する．するとシステムが実行可能な操作を提示するので，ユーザはそれに返答することによって操作を完了することができる．ユーザの返答の方法は提示される操作によって異なる．提示される操作が「カーテンを開けますか？」のように一種類の操作のみの場合は，ユーザは「はい」または「いいえ」と発話することによって操作の完了または取り消しを行うことができる．また，提示される操作が「扇風機の風量を変えますか？電源を消しますか？」のように二種類以上ある場合は，「風量を変えて」または「消して」のように発話することで行いたい操作を選択することができる．

#### 4.3.2 連携サービスによる操作

複数の機器を連携させて動作させる連携サービスによる操作を行う場合は，あらかじめDVDを視聴する際の環境を整える「DVDシアターサービス」や外出時に機器の電源をすべてオフにする「お出かけサービス」などのサービスを定義し

ておくことで、ユーザはサービス名を発話することでサービスを起動することができる。

## 5. 評価実験

### 5.1 実験概要

提案手法の有効性を評価するために，被験者に HNS に接続された家電機器を操作してもらう実験を行った．本実験では，提案手法である対話式のインタフェースと，2.5 で述べたフルコマンド法の二種類のインタフェースを用いて被験者にタスクを行ってもらった．家電機器を音声で操作するインタフェースはまだ普及しているとは言えず，一般に音声インタフェースを使用した経験が少ない，あるいはまったくない人が多数を占めると考えられる．そこで，本稿では音声インタフェースの使用経験がない，もしくは日常的に使用していない 12 名を被験者として実験を行った．

### 5.2 実験環境

本実験では，NAIST-HNS を操作対象の HNS とし，NAIST-HNS に接続されている 6 種類のネットワーク家電を操作する実験を行った．実験で操作対象とした機器を以下に示す．

- テレビ：NEC PX-50XM2
- DVD プレーヤ：パイオニア HTZ-535DV
- ライト：松下電工 HHFZ5310
- 電動カーテン：ナビオ パワートラック
- 空気清浄機：日立 EP-V12
- 扇風機：森田電工 MCF-257NR

音声の入力装置としては，Plantronics 社製ヘッドセット Voyager 510 を用いた．この機器は Bluetooth での無線接続が可能で，耳にかけることでハンズフリーの操作が可能である．有線接続のマイクを使用した場合，手で持つかテーブルなど

に固定しておく必要があるため，2.3 で述べた専用リモコンの問題点を十分に解消しているとは言えないので，今回の実験ではハンズフリーのマイクを用いることとした．

提案システムの実装は以下のとおりである．

- 音声認識エンジン 音声認識エンジンには，Windows の Speech API(SAPI) である Microsoft SAPI[15] 用の音声認識エンジン Julius for SAPI[16] を用いた．使用したバージョンは，Microsoft SAPI は Speech SDK 5.1 for Windows を，Julius for SAPI はバージョン 2.3 を用いた．Julius for SAPI では，認識すべき語を XML 形式の文法ファイルで記述できるため，アプリケーションで必要な語のみを定義することで，認識結果を限定することができる．これにより，定義されていない語を発話した場合は「認識失敗」として判別することができる．
- システム構成モジュール 提案システムを構成するモジュールを実装したアプリケーションは，Microsoft Visual C++を用いて作成した．
- スピーチエンジン 音声フィードバックを出力するスピーチエンジンとして，PENTAX 社製の VoiceText Engine[17] を用いた．

### 5.2.1 タスク

提案手法及びフルコマンド法を用いて行ってもらうタスクのリストを表 4 に示す．被験者はタスク番号順にタスクを遂行する．フルコマンド法を用いるタスクでは，表 1 に示すコマンドのリストを被験者に渡し，3 分間時間を与え実行可能な操作とそれに対応するコマンドを記憶してもらう．提案手法を用いるタスクに関しては，機器名を記憶するだけで操作ができるが，タスクリストに機器名が明記されているため，コマンドを記憶する作業は発生しない．



タスク番号	タスク内容
1.	カーテンを開ける
2.	空気清浄機をつける
3.	扇風機をつける
4.	扇風機の風量を一番強くする
5.	ライトを豆球にする
6.	ライトを明るくする
7.	テレビの電源をつける
8.	テレビの入力を DVD にする
9.	DVD を再生する
10.	DVD の音量を下げる
11.	DVD を停止する

表 4 タスクリスト

### 5.3 実験手順

本実験では，提案手法を先に行う被験者とフルコマンド法を先に行う被験者を半分ずつの各 6 名として実験を行った．先に行う手法で操作を行うことで音声インタフェースによる操作に慣れてしまい，後に行う手法との間で音声操作に対する慣れの度合いが変わってしまう可能性がある．こういった，学習効果による影響を少なくするため，各手法を行う順番を変えて実験を行うこととした．

以下に提案手法を先に行う場合の実験手順を示す．

1. 被験者に実験でどの HNS 機器が操作の対象となるか説明し，使用するマイクについても使い方の説明を行う．
2. 提案手法とフルコマンド法の各手法についての説明を行う．説明項目は，提案手法での操作方法，フルコマンド法での操作方法，どちらの手法を先に行うかという実験の順番，の 3 項目である．

3. 実験タスクのリストを被験者に渡し、順番通りにタスクを行ってもらおうよう説明する。
4. 提案手法を用いてタスクを行ってもらおう。
5. 従来法で用いるコマンドの一覧を渡し、3分間で記憶してもらおう。
6. 従来法を用いてタスクを行ってもらおう。

従来法を先に行う場合の実験手順は、提案手法を先に行う場合の順番を変え、

1. 2. 3. 5. 6. 4. の順番で実験を行ってもらおう。

## 5.4 結果と評価

図9は実験結果の一覧を表にしたものである。フルコマンド法と提案手法それぞれについて、以下の全被験者の6つのメトリクスを示している。

**M1:発話文字数** タスク実行時にユーザが発話した総文字数を示している。例えば「カーテン開けて」とユーザが発話した場合は7とカウントされる。この数値が大きいほどユーザの発話量が多く、学習コストが高くなる可能性があると考えられる。

**M2:ユーザ発話失敗回数** ユーザが発話した音声コマンドがタスクを実現するための音声コマンドと異なっていた場合にカウントされる。ただし提案手法の場合、3.3節で述べたように、ユーザが発話した音声コマンドがタスク実現のための音声コマンドに含まれていれば、失敗とはカウントされない。例えば、5.2.1のタスク番号2「空気清浄機をつける」というタスク実行時に、ユーザが「扇風機つけて」といった間違った音声コマンドを発話するようなケースでカウントを行う。この数値が大きいほど、ユーザが自身の意図したとおりの正しい音声コマンドを発話するのが困難であったことを示すと考えられる。

M3:音声認識失敗回数 このメトリクスは，ユーザがタスク実行のための正しい音声コマンドを発話したにも関わらず，音声認識エンジンが間違っ了解釈をした場合にカウントされる．この数値は音声認識エンジンの精度をあらわしている．

M4:機器動作間違い回数 今回の実験ではフルコマンド法，提案手法両方の音声インタフェースにおいて，機器制御を実行する前にユーザに制御内容を確認する．機器動作間違い回数はこの確認の際にインタフェースによって提示される制御内容がタスクと異なっていた場合にカウントされる．このときインタフェースは，ユーザの発話した音声コマンドを解釈した結果が実行可能である場合にのみ制御内容の確認を行い，実行不可能な音声コマンドの入力に対しては反応しない．故に機器動作間違い回数は，ユーザが発話を間違えた，もしくは音声認識エンジンが誤認識をした際に，その音声コマンドがたまたま他の機器の音声コマンドと一致した場合にのみカウントされる．結果として，この数値が大きければ大きいほど，ユーザの意図しない機器制御が実行されてしまうリスクをあらわしている．

M5:音声コマンド確認回数 音声コマンド確認回数は，インタフェースに与えるべき音声コマンドの内容をユーザが確認した回数をあらわしている．このメトリクスの数値が大きいくほど，ユーザにとって音声コマンドの学習コストが大きいくと考えられる．

M6:SUS スコア システムやインタフェースの効率性や満足度などのユーザビリティを評価するための手法に質問紙による方法があり，代表的なものに QUIS[29, 30]，SUMI[31, 32]，SUS[6] が挙げられる．QUIS，SUMI は多くの質問項目を有し，回答に多くの時間を有するものはユーザへの負担が大きいく．本評価では質問紙への回答そのものの負担が質問の回答に与える影響を低減させるため，SUS を用いた．SUS は SUMI との相関も指摘されており，比較的短時間で回答が可能でユーザへの負担が少ない．SUS は 10 の質問項目を用いて有効性，効率性，満足度を総合した SUS スコアを算出する．SUS の原著は英語であるが，今回の被験者は全員日本人であり，原著のまま使用することで質問の意味を読み違える可能性が高くなり，結果的に SUS を用いた評価の信頼性が低下する可能性があるため，

事前に日本語に訳した SUS 質問紙を用意した。原著の意味を崩さずに日本人が解釈できるように、原著の文脈から質問内容を熟考した結果、意識となった部分が存在する。今回作成した日本語版 SUS を図 10 に示す。これによって得られた値から、フルコマンド法と提案手法とでユーザビリティ比較を行う。

被験者No.	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	平均	中央値
年代	20代	30代	20代	20代	20代	20代	20代	20代	20代	20代	20代	30代		
性別	男	男	男	男	男	男	男	女	男	男	女	女		
実験の順	従来先	従来先	従来先	従来先	従来先	従来先	提案先	提案先	提案先	提案先	提案先	提案先		
ユーザ発話失敗回数	7	6	0	5	15	1	10	5	3	4	1	3	5.0	4.5
音声認識失敗回数	1	16	0	0	6	12	6	8	3	6	1	6	5.4	6
機器動作間違い回数	5	12	0	2	4	6	4	5	0	0	1	5	3.7	4
音声コマンド確認回数	2	0	0	1	2	1	0	1	0	0	0	2	0.8	0.5
発話文字数	215	378	128	159	239	267	269	276	172	217	146	206	222.7	216
SUSスコア	55	35	82.5	47.5	37.5	70	85	60	92.5	70	87.5	65	65.6	67.5
ユーザ発話失敗回数	3	6	2	5	6	3	15	4	2	10	0	16	6.0	4.5
音声認識失敗回数	4	9	4	7	1	18	3	8	2	10	1	7	6.2	5.5
機器動作間違い回数	0	0	0	1	0	2	3	1	0	2	1	1	0.9	1
音声コマンド確認回数	1	0	0	0	0	0	1	0	0	1	0	1	0.3	0
発話文字数	165	175	147	186	151	243	291	216	148	234	155	224	194.6	180.5
SUSスコア	50	65	87.5	52.5	80	70	95	72.5	82.5	65	100	62.5	73.5	71.25

図 6 提案提案データ

## 評価シート

※「まったくそう思わない」を1、「非常にそう思う」を5として、1～5の間で評価してください

	方法1	方法2
1) このシステムをしばしば使いたいと思う	<input type="text"/>	<input type="text"/>
2) このシステムは不必要なほど複雑であると感じた	<input type="text"/>	<input type="text"/>
3) このシステムは容易に使えると思った	<input type="text"/>	<input type="text"/>
4) このシステムを使うのに技術専門家のサポートを必要とするかもしれない	<input type="text"/>	<input type="text"/>
5) このシステムでは、さまざまな機能がよくまとまっていると感じた	<input type="text"/>	<input type="text"/>
6) このシステムでは、一貫性の無いところが多くあったと感じた	<input type="text"/>	<input type="text"/>
7) たいいていのユーザは、このシステムの使い方を素早く学べるだろうと思った	<input type="text"/>	<input type="text"/>
8) このシステムはとても扱いにくいと思った	<input type="text"/>	<input type="text"/>
9) このシステムを使うのに、自分は自信があると感じた	<input type="text"/>	<input type="text"/>
10) このシステムを使い始める前に多くのことを学ぶ必要があると感じた	<input type="text"/>	<input type="text"/>

図 10 System Usability Scale(SUS) シート

## 5.5 考察

評価実験で得られた結果を考察し，各メトリクスを測定することによって得られた知見を示す．また，提案手法が3.1で挙げた要求を満たしているかどうかを結果から示す．

### 5.5.1 M1:発話文字数

従来法で大きな外れ値があるほかは，両手法とも同じくらいの分布を示しているが，従来手法より提案手法の方が平均値と中央値が低い．

従来法より提案手法の方が発話文字数が多い被験者は全体の半数である6名だが，この場合両手法の間でほとんど差が見られない．一方，提案手法より従来法の方が発話文字数が多い被験者は残り半数の6名だが，被験者によっては従来法での発話文字数が提案手法の2倍以上になる場合があるため，従来法の学習コストが提案手法と比べ著しく高くなるユーザが存在することが示された．

発話失敗回数と認識失敗回数が提案法と従来法で差がない場合でも，提案手法の方が発話文字数が少ない場合が多い．これは，従来法では操作を失敗するとコマンドをすべて言い直さなければならないが，提案手法では3.3で示したキーアイデア K1 により発話が段階的に行えるため，メソッド名やパラメータ名を言い直すだけでやりなおすことができるからである．被験者によっては従来法で提案手法の倍以上の文字数を発話しているが，これも従来法はコマンドをすべて言い直さなければならないからである．

### 5.5.2 M2:ユーザ発話失敗回数

両手法とも被験者によっては回数が多いものもあるが，どちらもほとんどの被験者が6~7回以下であり，回数が0の被験者もいるため，中央値・平均値はさほど高くない．

従来法でユーザの発話が失敗する場合として，コマンドリストに定義されていないコマンドを発話してしまうことがあった．ユーザはコマンドリストを記憶しリストを見ずにタスクを行うため，リストに定義されていないコマンドを発話し

てしまうことがあり，それが発話失敗回数として測定された．提案手法でユーザの発話が失敗する場合として，システムが提示した操作名やパラメータの候補と異なった発話をしてしまうことがあった．例えば，「テレビの音量を上げますか？下げますか？」という提示に対して「大きくして」などのシステムが予期していない発話があった場合操作が実行されないため，それが発話失敗回数として測定された．

#### 5.5.3 M3:音声認識失敗回数

提案手法の方が平均回数は多いが，とびぬけて回数が多い外れ値があるため，中央値は提案手法の方が少ない．従来法では失敗していないのに提案手法で失敗している被験者もいる．

#### 5.5.4 M4:機器動作間違い回数

提案手法は全被験者が3以下にかたまっており，平均値・中央値ともに低いが，従来法ではとびぬけて高い値があり，そのほかの値も0~6と分散が広いため平均値・中央値は提案手法の4倍の値である．このメトリクスの結果に対して有意水準5%としてt検定を行ったところ，p値は $p=0.0016 < 0.05$ であったため，提案手法の平均値は従来法の平均値に対して有意な差があると言える．

#### 5.5.5 M5:音声コマンド確認回数

全体的に数が少なく，多くても2回なので大きな差は出なかった．従来法では，コマンドリストに定義されたコマンドを忘れてしまった場合に確認を求めている．また，提案手法で，ユーザの発話失敗回数が少ないのにコマンド確認をしている場合，誤認識した発話をコマンドを間違えたと考え確認した場合も考えられる．正しい発話であるにもかかわらずコマンドを誤認識した場合，その発話が正しい発話であったことをユーザに通知できればこの問題は解決するが，提案システムでは実現が難しい．



### 5.5.6 M6:SUS スコア

提案手法の分布の方が高い値に分布しており，従来法と比べて全体的に評価が高い．平均や中央値に関しても提案手法の方がより高い値を得ている．従来法のスコアの方が提案手法のスコアより高い被験者は4名いるが，いずれの場合も従来法と提案手法の間に大きな差は見られず同程度の満足度と見ることができる．一方，提案手法のスコアの方が従来法のスコアより高い被験者は7名いるが，そのうちスコアの差が大きい被験者が2名おり，従来法と比べ提案手法に対する満足度が十分高い例も存在すると言える．

### 5.5.7 要求の満足度

実験結果から得られた知見に基づき，提案手法が要求を満足しているかどうかについて，各メトリクスと対応付けて述べる．

メトリクス M1 では，計測された数値が高いほど手法の学習コストが高いということが言えるため，M1 は要求 R1 に対応するメトリクスであると言える．M1 では，提案手法の平均値と中央値が従来法より低く，従来法に比べ提案手法での発話文字数が半分以下である例もあるため，提案手法の方が学習コストが低いと考えられるので，提案手法は要求 R1 を満たしていると考えられる．

メトリクス M2 では，従来法で操作を行う場合，測定された数値がコマンドを記憶できなかつたり間違っして記憶してしまうことによる発話の失敗回数を示しているため，数値が高いほど学習コストが高いと言える．一方，提案手法で操作を行う場合は，提示された操作名やパラメータと違う発話をしてしまったことによる発話の失敗回数を示しているため，数値と学習コストには特に関係がない．

メトリクス M3 の測定値は音声認識エンジンの性能に関する値であり提案手法の有用性に関するものではないので，要求の満足度に関しては関係のないものとする．

メトリクス M4 では，計測された数値は大きいほどユーザの意図しない操作が行われたという意味であるため，M4 は要求 R2 に対応するメトリクスであると言える．M4 においては，提案手法で操作した際の誤動作数が従来手法で操作し

た場合の 25 %程度になり， $t$  検定でも有意な差があることが確認されたので，提案手法は要求 R2 を満たしていると考えられる．

メトリクス M5 では，計測された数値が高いほど手法の学習コストが高いと言えるため，M5 は要求 R1 に対応するメトリクスであると言える．提案手法の方が平均値や中央値が低く，確認を求めた被験者の数も少ないため，提案手法の方が学習コストが低いと考えられるので，提案手法は要求 R1 を満たしていると考えられる．

## 6. まとめ

本論文では、2.5において、現状の音声による HNS 操作インターフェースの問題点として、操作対象機器が増えると操作コマンドの数も増えることと、音声認識を失敗すると制御される機器の振る舞いがユーザの意図しないものになる可能性があることの2つを挙げた。そして、3.1でこれらの問題点を解決するために要求される、R1とR2の2つの要求を明確にした。これら2つの要求を満たすため3.2においてK1,K2の2つのキーアイデアを提案し、4でキーアイデアを実現する、段階的な音声コマンド構築のための対話型インターフェースを実装した。また、5では提案するインターフェースの評価するための実験として、従来技術を利用した HNS 音声操作インターフェースであるフルコマンド法と提案手法を比較する実験を行い、得られた結果の考察を行った。

今後の研究課題として、さらなる誤動作の低減やより自然な会話に近い対話インターフェースの構築、雑音環境下におけるロバスト性などが挙げられる。家電機器を操作する音声インターフェースは未だ一般的なものとなっていないが、これらの課題を解決することで広く一般に利用されるシステムにより近づくことができると考えられる。

## 謝辞

本研究を進めるにあたり多くの方々に、御指導、御助力を頂きました。心より深く御礼申し上げます。

奈良先端科学技術大学院大学 情報科学研究科 松本健一 教授には、常日頃より御指導、御助力を頂きました。研究方針に関するだけでなく、研究活動に対する姿勢や心構えについて、多くの事柄を御教示頂きました。心より御礼申し上げます。

奈良先端科学技術大学院大学 情報科学研究科 木戸出正繼 教授には、研究発表の際、貴重な御意見、御指摘を頂きました。心より御礼申し上げます。

奈良先端科学技術大学院大学 情報科学研究科 門田暁人 准教授には、多くの丁寧な御指導、御指摘を頂きました。研究の進め方や論文の書き方などを詳細に渡って御指導頂きました。心より御礼申し上げます。

神戸大学 工学研究科 中村匡秀 准教授には、私の研究生生活の全ての過程において熱心な御指導を頂きました。研究の進め方や資料作成、発表技術から論文の執筆に至るまで研究生生活のあらゆる過程で御指導、御指摘を賜りました。特に論文執筆の際には多大なる御支援と御指導を頂きました。心より御礼申し上げます。

神戸大学 工学研究科 井垣宏 特任助教には、研究に関する御指導、実験を行う際の御指摘など、様々な状況で御助力を頂きました。また、論文執筆の際には長時間にわたり御指導頂きました。心より御礼申し上げます。

奈良先端科学技術大学院大学 情報科学研究科 大平雅雄 助教には、研究方針に関し、貴重な御指導、御指摘を頂きました。心より御礼申し上げます。

奈良先端科学技術大学院大学 栗山進 氏には、論文、資料の作成を行う際に御指導、御助力を頂きました。心より御礼申し上げます。

奈良先端科学技術大学院大学 情報科学研究科 ソフトウェア工学講座の皆様には、研究活動、日々の学生生活に関して、多くの御指導、御指摘、御助力を頂きました。心より御礼申し上げます。

## 参考文献

- [1] NUANCE, “IBM ViaVoice”, <http://www.nuance.com/viavoice/>
- [2] NUANCE, “ドラゴンスピーチ”,  
<http://japan.nuance.com/naturallyspeaking/>
- [3] 鹿野 清宏, 川波 弘道, 西村 竜一, 李 晃伸, “音声情報案内システム「たけまるくん」および「キタちゃん」の開発,” 情報処理学会研究報告. SLP, 音声言語情報処理, Vol.2006, No.107, pp.33-38, 2006.
- [4] 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.
- [5] 伊藤 英一, “頭の向きと音声により操作する四肢まひ者向け家電コントローラ,” ヒューマンインタフェース学会論文誌, Vol.4, No.1, p.25-31, 2002.
- [6] J. Brooke, “SUS - a quick and dirty usability scale,” In P. W. Jordan, B. Thomas, B. A. Weerdmeester, A. L. McClelland (eds.) Usability Evaluation in Industry, London: Taylor and Francis, 1996.
- [7] ISO 9241-11, “Ergonomic requirements for office work with visual display terminals. Part 11: Guidance on Usability,” 1998.
- [8] 北山広治, 後藤真孝, 伊藤克亘, 小林哲則, “音声スポット: 人間同士の会話中に音声認識が利用できる新たな音声インタフェース,” WISS2003, 近代科学社, pp.9-18, 2003.
- [9] 共用品推進機構, “不便さ調査データベース,”  
[http://www.kyoyohin.org/02\\_syougai/index.php](http://www.kyoyohin.org/02_syougai/index.php)
- [10] Masahide Nakamura, Akihiro Tanaka, Hiroshi Igaki, Haruaki Tamada, and Ken-ichi Matsumoto, “Constructing Home Network Systems and Integrated Services Using Legacy Home Appliances and Web Services,” International Journal of Web Services Research, Vol.5, No.1, pp.82-98, January 2008.

- [11] Pioneer, “CYBER NAVI”,  
<http://pioneer.jp/carrozzeria/cybernavi/function/design/sound.html>
- [12] Innotec Systems Inc. “Accenda,”  
<http://www.innotechsystems.com/accenda/index.html>
- [13] 内田尚和, 常盤大樹, 高木朗, 麻生英樹, 森彰, 橋本政朋, 伊東幸宏, 小林一郎, 中島秀之, 八名和夫, “意味の位置づけを可能にする意味表現を用いた情報家電操作のための対話的インタフェース,” 第19回人工知能学会全国大会予稿集, 2005.
- [14] 松政宏典, 田中克幸, 滝口哲也, 有木康雄, 李義昭, 中林稔堯, “情報家電操作における脳性麻痺構音障害者の音声認識評価,” 電子情報通信学会技術研究報告. WIT, 福祉情報工学, Vol.107, No.61, pp. 33-38, 2007.
- [15] Microsoft SAPI,  
<http://www.microsoft.com/japan/msdn/accessibility/speech/default.aspx>
- [16] 河原達也, 李晃伸, “連続音声認識ソフトウェア Julius,” 人工知能学会誌, Vol.20, No.1, pp.41-49, 2005.
- [17] PENTAX, “VoiceText Engine,” <http://voice.pentax.co.jp/>
- [18] Panasonic, “VIERA Link,” <http://panasonic.jp/viera/network/>
- [19] 三菱電機, “みまもりサーバー,”  
[http://www.mitsubishielectric.co.jp/home/kirigamine/it\\_b.html](http://www.mitsubishielectric.co.jp/home/kirigamine/it_b.html)
- [20] 庄田明, 柏本隆, 浦出正和, “くらしネット家電ソフトウェアプラットフォーム,” Matsushita Technical Journal, Vol.50, No.3, pp.31-35, 2004.
- [21] 河原崎徳之, 安齋良恵, 清水佑歌, 吉留忠史, 西原主計, “音声認識による家電機器のリモコン制御,” 福祉工学シンポジウム講演論文集, Vol.2004, pp.197-200, 2004.

- [22] NTT ネオメイト, “ユーコンセントサービス,”  
<http://www.ntt-neo.com/news/2007/070419.html>
- [23] Jacob Nielsen, “ウェブ・ユーザビリティ,” エムディエヌコーポレーション,  
2000.
- [24] 三洋電機, “統合型ホームネットワークシステム ELiFES,”  
<http://www.sanyo.co.jp/koho/hypertext4/0410news-j/1004-1.html>
- [25] 松下電器産業株式会社, “くらしネット,”  
<http://national.jp/appliance/product/kurashi-net/>
- [26] 東芝, “東芝ネットワーク家電 Feminity,”  
<http://www3.toshiba.co.jp/feminity/about/index.html>
- [27] 森川博之, “ワイヤレスが開くユビキタスネットワーク,” 電子情報通信学会  
誌 Vol.87, No.5, pp.356-361, 2004.
- [28] gSOAP : C/C++ Web Services and Clients,  
<http://www.cs.fsu.edu/engelen/soap.html>
- [29] Questionnaire for User Interaction Satisfaction (QUIS),  
<http://www.lap.umd.edu/QUIS/>
- [30] John P. Chin , Virginia A. Diehl , Kent L. Norman, “Development of an  
instrument measuring user satisfaction of the human-computer interface,”  
*Proceedings of the SIGCHI conference on Human factors in computing  
systems*, pp.213-218, May 15-19, 1988, Washington, D.C., United States.
- [31] Software Usability Measurement Inventory (SUMI),  
<http://www.ucc.ie/hfrg/questionnaires/sumi/>
- [32] Kirakowski, J. and Corbett, M. “SUMI: The software usability  
measurement inventory,” *British Journal of Educational Technology*, 24(3),  
pp.210-212, 1993.