

NAIST-IS-MT0651051

修士論文

家電連携サービス作成支援システム  
“BAMBEE”の実装と評価

関本 純一

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。

関本 純一

審査委員：

松本 健一 教授 (主指導教員)

伊藤 実 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)

# 家電連携サービス作成支援システム

## “BAMBEE”の実装と評価\*

関本 純一

### 内容梗概

ホームネットワークシステムの代表的なアプリケーションとして、複数のネットワーク家電を連携制御し、付加価値の高いサービスを実現する家電連携サービスがある。従来、家電連携サービスはソフトウェアアプリケーションとしてサービスプロバイダによって開発され、非専門家であるエンドユーザが自由に作成することはできなかった。そこで本研究では、プログラミングの知識を持たないエンドユーザが家電連携サービスを作成できるようになることを目的とし、サービス作成を支援するシステム（BAMBEE）を開発した。BAMBEEではタッチパネルに表示されたGUI上で簡単な操作を行うことにより、家電連携サービスの作成・編集・削除・動作テストを行うことができる。また、開発したシステムを22歳から72歳までの幅広い年齢層の被験者28名に使ってもらい、ユーザビリティテストに基づく実験的評価を行った。実験の結果、全ての被験者が家電連携サービスの作成・編集・削除タスクを完了でき、BAMBEEの有効性が示せた。

### キーワード

ホームネットワークシステム, 連携サービス, Web サービス, GUI, ビジュアルプログラミング

---

\* 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-MT0651051, 2008年2月7日.

# BAMBEE: Supporting End-Users for Creating Integrated Services in Home Network System\*

Jun-ichi Sekimoto

## Abstract

The integrated services are applications of the home network system, which provide value-added features with integrating multiple networked home appliances. Currently, the integrated services are developed by service providers as proprietary applications. Thus, it is quite difficult for end-users to create such services without special expertise in software and hardware.

This paper presents a service creation support system, called BAMBEE, in order to help such non-expert end-users create their own integrated services. With BAMBEE, the end-user can create, edit, test and delete integrated services by using intuitive GUI displayed on a touch panel. We have conducted an experimental evaluation of BAMBEE, through usability testings for 28 subjects in 22- to 72-year-old demographics. It was shown in the experiment that all subjects were able to complete tasks of creation, editing, and deletion of integrated services.

## Keywords:

Home Network System, Integrated Service, Web Service, GUI, Visual Programming

---

\* Master's Thesis, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT0651051, February 7, 2008.

# 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 NAIST-HNS	3
2.3 連携サービス	5
2.4 連携サービスの作成方法と問題点	6
2.4.1 プログラムによる連携サービス作成法	6
2.4.2 学習リモコンによる連携サービス作成法	7
2.4.3 既存の連携サービス作成法の問題点	8
2.5 エンドユーザによる連携サービス作成の必要性	9
2.6 本研究の目的	10
3. 提案システム	11
3.1 要件定義とアプローチ	11
3.2 システム概要	13
3.3 ハードウェア選定	15
3.4 GUI	16
3.4.1 GUI 概要	16
3.4.2 GUI の画面遷移と音声案内	17
3.4.3 操作フロー	19
3.5 システムの主機能	20
3.5.1 連携サービスの作成機能	20
3.5.2 連携サービスの編集機能	22
3.5.3 連携サービスの削除機能	24
3.5.4 連携サービスの動作テスト機能	26
3.6 システムの補助機能	29
3.6.1 操作フロー編集機能	29

3.6.2	自動挿入機能	31
3.7	実装	33
<b>4.</b>	<b>評価実験</b>	<b>35</b>
4.1	実験概要	35
4.2	被験者	35
4.3	使用した家電機器	36
4.4	実験手順	37
4.5	タスク	38
4.5.1	連携サービスの作成	38
4.5.2	連携サービスの編集	39
4.5.3	連携サービスの削除	40
4.6	評価指標	41
4.6.1	有効性	41
4.6.2	効率性	43
4.6.3	満足度	44
<b>5.</b>	<b>実験結果</b>	<b>46</b>
5.1	各タスクの実験結果	46
5.2	連携サービスの作成に関する結果	47
5.2.1	有効性	47
5.2.2	効率性	48
5.3	連携サービスの編集に関する結果	50
5.3.1	有効性	50
5.3.2	効率性	52
5.4	連携サービスの削除に関する結果	53
5.4.1	有効性	53
5.4.2	効率性	55
5.5	満足度	57

<b>6. 考察</b>	<b>60</b>
6.1 連携サービスの作成に関する考察 . . . . .	60
6.2 連携サービスの編集に関する考察 . . . . .	61
6.3 連携サービスの削除に関する考察 . . . . .	62
6.4 満足度に関する考察 . . . . .	62
<b>7. まとめ</b>	<b>63</b>
謝辞	64
参考文献	66
付録	68
A. BAMBEE 操作マニュアル	68

## 目 次

1	Perl による連携サービスの作成 (DVD シアターサービス) . . . . .	6
2	学習リモコン . . . . .	7
3	学習リモコンによる連携サービスの作成 . . . . .	8
4	予備実験のアンケート結果 . . . . .	10
5	システム構成 . . . . .	14
6	index ファイルの内容 . . . . .	14
7	使用したタッチパネル . . . . .	15
8	GUI 概要 . . . . .	16
9	GUI の画面遷移 . . . . .	18
10	操作フロー . . . . .	19
11	連携サービスの作成手順 . . . . .	21
12	連携サービスの作成の流れ . . . . .	22
13	連携サービスの編集手順 . . . . .	23
14	連携サービスの編集の流れ . . . . .	24
15	連携サービスの削除手順 . . . . .	25
16	連携サービスの削除の流れ . . . . .	26
17	連携サービスの動作テスト手順 . . . . .	27
18	連携サービスの動作テストの流れ . . . . .	28
19	操作フロー編集ボタン . . . . .	29
20	操作フロー編集機能 . . . . .	30
21	表示される選択肢 . . . . .	32
22	生成された連携サービスファイルの内容 . . . . .	34
23	実験の様子 . . . . .	37
24	SUS 質問紙の質問事項 . . . . .	45
25	タスク達成率 (連携サービスの作成) . . . . .	47
26	操作エラー率 (連携サービスの作成) . . . . .	48
27	タスク達成時間 (連携サービスの作成) . . . . .	49
28	総操作数 (連携サービスの作成) . . . . .	49



29	タスク達成率（連携サービスの編集）	51
30	操作エラー率（連携サービスの編集）	51
31	タスク達成時間（連携サービスの編集）	52
32	総操作数（連携サービスの編集）	53
33	タスク達成率（連携サービスの削除）	54
34	操作エラー率（連携サービスの削除）	55
35	タスク達成時間（連携サービスの削除）	56
36	総操作数（連携サービスの削除）	56
37	SUS スコア	57

## 表 目 次

1	依存する家電の操作	32
2	被験者区分	35
3	タスク達成率の定義（連携サービスの作成）	42
4	タスク達成率の定義（連携サービスの編集）	42
5	タスク達成率の定義（連携サービスの削除）	43
6	各タスクの実験結果	46

## 1. はじめに

近年，ネットワークを介して家庭内の家電を操作するホームネットワークシステム（HNS）が，人々の生活をより便利で快適にすることのできる基盤技術として注目されており，家電メーカーではネットワーク対応家電が開発されている [1] [2]．そのような中，我々の研究グループは独自のHNS（NAIST-HNS）を構築し，HNS内の家電を逐次操作させる家電連携サービス（以降，連携サービスと呼ぶ）という技術に関する研究を行っている．その理由としては，連携サービスはネットワーク家電の付加価値を高める技術だと考えているからである．

現在，連携サービスはベンダが作成し，エンドユーザは与えられたサービスを実行するという利用スタイルになっている．しかし，エンドユーザの嗜好や生活スタイルは，家庭環境や個人ごとに異なる．また，将来的に家電の種類が増えた場合は，自らの生活スタイルや嗜好に合わせた連携サービスを，エンドユーザ自らの手で作成したいという要求が生まれることは自然である．ここで，もしエンドユーザに連携サービスを作成したいという意思が無ければ，現在の連携サービスの利用スタイルを変える必要はない．しかし，50名のエンドユーザに連携サービスを作成させるという予備実験を行った結果，88%のエンドユーザが「連携サービスを自分で作れたら役に立つ」とアンケートで答えた．よって，エンドユーザには連携サービスを作成したいというニーズがあることがわかった．

現在，エンドユーザが連携サービスを作成する方法としては，2つの方法が考えられる．一つ目はプログラムによる方法，二つ目は学習リモコンを用いる方法である．しかし，これらの方法ではプログラミングに関する知識や煩雑な作業が必要であるため，エンドユーザにとっては困難だという問題点がある．

本論文では，GUIを用いたビジュアルプログラミングにより，エンドユーザが連携サービスを作成することのできるシステム（BAMBEE）を提案する．具体的には，エンドユーザが連携サービスの作成・編集・削除を行えるようにするという目的を果たすため，提案システムに必要な5つの要件を定義し，その要件を満たすように実装を行った．なお，5つの要件とは，「プログラミングの知識が無くても連携サービスの作成ができる」「PCや機械の熟練度に関係なく操作できる」「連携サービスの作成・編集状況が視覚的に把握できる」「文字による操作方法の

説明が少なくても操作できる」「作成・編集した連携サービスの動作確認ができる」「ユーザの入力忘れを防止できる」ということである、

本研究の目的が果たされているかどうかを確認するため、提案システムのユーザビリティテストを実施した。ユーザビリティテストでは、エンドユーザ 24 名と提案システムの熟練者 4 名に対し、連携サービスの作成・編集・削除を行うという 3 つのタスクを与えた。そして、実験結果の評価には、ユーザビリティに関する国際規格 ISO9241-11[16] の評価指標を用い、提案システムのユーザビリティを測定した。その結果、25 名の被験者が全タスクを達成することができたが、3 名の被験者が連携サービスの編集を行うタスクに失敗した。しかし、連携サービスの編集を行うタスクでは、編集を行った後に動作テストを行うという作業があり、タスクに失敗した 3 名はその作業に失敗していたので、実質、連携サービスの編集自体は達成できていた。よって、全被験者が連携サービスの作成・編集・削除を行えたといえる。

本論文の章構成は以下の通りである。第 2 章では、まず、ホームネットワークシステム (HNS) について説明する。その後、連携サービスについて説明し、その作成方法と問題点について述べる。そして最後に、本研究の目的を述べる。第 3 章では、提案システムの要件定義とそれに対するアプローチを述べた後、提案システムの機能について述べる。第 4 章では提案システムの評価を行うユーザビリティテストの詳細を述べる。第 5 章にはユーザビリティテストの結果を示す。第 6 章ではユーザビリティテストの結果から得られた考察を述べる。第 7 章では本研究のまとめを述べる。

## 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、家庭内のネットワークに接続された複数のネットワーク家電から構成される。ネットワーク家電は、ユーザがネットワークを通して操作できるように、家電操作 API を持っている。そして、この家電操作 API の呼び出しを実行するため、ネットワーク家電はプロセッサとストレージを持つことが一般的である。

ネットワーク家電間の通信は、専用の家電プロトコルに基づいて行われ、現在、情報家電用の DLNA[3] や白物家電用の ECHONET[4] といった家電プロトコルが普及しつつある [5]。しかし、これらのプロトコルは主に家電間のネットワーク層 (アドレス設定、メッセージフォーマット等) を規定するものであり、アプリケーション層における相互接続性を保証するものではない。よって、現在実用化されている HNS の多く [1] [2] は、家電のベンダを統一する必要がある。また、HNS アプリケーションもベンダが提供するものに限られるという問題点もある。

### 2.2 NAIST-HNS

NAIST-HNS とは、中村、井垣らのフレームワーク [6] [7] を基にして、我々の研究グループが構築した HNS であり、2.1 節で述べた HNS の問題点を解決している。NAIST-HNS ではシステムに組み込む家電のベンダを統一する必要が無く、マルチベンダの家電をユーザが任意に組み合わせることができる。また、ネットワークに対応していない従来家電でも、システムに組み込むことができる。

その方法は、まず、赤外線リモコンで操作可能な家電を対象に、外部の PC から家電を操作するための API (家電操作 API) を、ホームサーバのような HNS を管理する PC に用意する。そして、家電操作 API を保持している PC には、学習リモコン [8] を接続しておき、家電操作 API が呼び出された場合に、学習リモコンから家電を操作する赤外線信号を送信するようにする。学習リモコンとは、家電用リモコンの赤外線信号を登録できるリモコンである。このようにして、NAIST-HNS

では家電のベンダを統一する必要が無く、従来の家電もシステムに組み込めるようにしている。

HNS において家電を連携操作させることを想定した場合、家電操作 API をそのまま使用することは実用的ではない。そこで、複数の家電操作 API 呼び出しを家電の論理機能単位でまとめ、家電の種類やベンダに非依存な Web サービスとしてネットワークに公開する。公開された Web サービスは、自己完結したコンポーネントとして、ネットワークを介して様々なアプリケーションから利用可能となる。

Web サービスとは、ソフトウェアをサービスという単位でまとめ、Web サーバ上でその機能を提供するためのフレームワークである [9]。Web サービスの最大の特徴は、クライアント側からのアクセスに Web ブラウザを必要とせず、データを直接やり取りできる機構が提供されていることである。サービスの利用は、クライアントプログラムから XML-RPC[10] や SOAP[11] といったリモートプロシージャコール (RPC) と呼ばれるメッセージ交換プロトコルで行われる。また、サービスは Web サービスのインタフェース記述言語である WSDL[12] で定義され、WSDL を参照することでサービスの利用方法を取得する。このように、Web サービスで標準化された手法を用いることで、アプリケーション開発者は送受信するメッセージの書式やプロトコルを意識することなく、通常の方法で呼び出しとほぼ同じ方法で、Web サーバ上のサービスをアプリケーションに組み込むことができる。アプリケーションを作成するための開発環境も、RPC に対応してさえいけば、他は何の制限も無く、また、RPC の一つである SOAP は、多くのプログラミング言語で専用の API を導入することで利用できる。

## 2.3 連携サービス

HNS の機能として連携サービスが挙げられる。連携サービスとは、複数の家電を逐次操作することで、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、DVD を鑑賞したいという目的をユーザが持っているとする。この目的を果たす場合、従来の操作では、テレビ、DVD プレーヤ、照明といった家電を全て操作する必要がある。しかし、これらの操作を「DVD シアターサービス」という連携サービスとして登録しておけば、その連携サービスを実行するだけで、全ての家電の操作を行ってくれる。NAIST-HNS では以下のような連携サービスが実装されている。

- DVD シアターサービス

DVD シアターサービスとは、DVD の観賞ができるように家電の状態を変更させる連携サービスである。サービスを開始すると、まず、電動カーテンが閉まり、天井照明が消える、次に、テレビの入力が DVD プレーヤに切り替わり、音量が調整される。そして最後に、DVD が再生される。

- 空気清浄サービス

空気清浄サービスとは、部屋の中の空気を綺麗にする連携サービスである。サービスを開始すると、まず、空気清浄機の電源が ON になり、次にサーキュレータの電源が ON になって、風量が「強風」に設定される。

- おでかけサービス

おでかけサービスとは、複数の家電の電源を全て OFF にし、ユーザが直ちに外出できる状態にする連携サービスである。サービスを開始すると、テレビ、空気清浄機、サーキュレータの電源が OFF になる。そして、天井照明が消え、電動カーテンが閉まる。

- おかえりサービス

おかえりサービスとは、複数の家電の電源を全て ON にし、ユーザが帰宅した際に直ちに部屋を利用できるようにするための連携サービスである。サービスを開始すると、天井照明が点き、電動カーテンが閉まる。そして、空気清浄機、サーキュレータ、テレビの電源が ON になる。

## 2.4 連携サービスの作成方法と問題点

現在，連携サービスを作成する方法としては，プログラムによる方法と学習リモコンを用いた方法が考えられる．本論文では，2.4.1 節にプログラムによる作成方法を，2.4.2 節に学習リモコンによる作成方法を示す．そして，2.4.3 節ではそれら 2 つの作成法の問題点を述べる，

### 2.4.1 プログラムによる連携サービス作成法

Perl や JAVA などのプログラミング言語により，連携サービスを作成することができる．NAIST-HNS では連携サービスを Perl で記述しており，例えば，2.3 節で述べた DVD シアターサービスは，図 1 のようなプログラムである．このプログラムでは，2 行目で電動カーテンの操作，3～4 行目で天井照明の操作，5～6 行目でテレビの操作，7～8 行目で DVD プレーヤの操作を行っている．

```
1 sub theater_play{
2     $service_curtain -> close(); #電動カーテンを閉める
3     $service_light -> brightnessControl(0); #照明 1 を消す
4     $service_light2 -> brightnessControl(0); #照明 2 を消す
5     $service_pdp -> on(); #テレビの電源を ON にする
6     $service_pdp -> changeInput(1); #テレビの入力を DVD プレーヤに
切り替える
7     $service_dvd -> on(); #DVD プレーヤを ON にする
8     $service_dvd -> play(); #DVD を再生する
9 }
```

図 1 Perl による連携サービスの作成 (DVD シアターサービス)

#### 2.4.2 学習リモコンによる連携サービス作成法

図2のような学習リモコン [8] (スギヤマエレクトロン クロッサム 2+USB) を用いることにより、プログラミングを行わずに連携サービスを作成することができる。学習リモコンのボタンには、家電用リモコンの赤外線信号を複数登録することができ、ボタンを押すことで登録した赤外線信号が自動で逐次送信され、これによって連携サービスを実現する。例えば、[8] を用いて連携サービスを作成するには、図3に示す手順で行う。



図 2 学習リモコン



1. 学習リモコンを「学習モード」に設定する
2. 登録先にする学習リモコンのボタンを押す
3. 家電用リモコンと学習リモコンを向かい合わせにし、家電用リモコンの登録させたいボタンを押す
4. 1～3を繰り返し、学習リモコンに複数の信号を登録する
5. 学習リモコンを「特殊機能モード」に設定する
6. 学習リモコンを「連続操作学習モード」に設定する
7. 登録先にする学習リモコンのボタンを押す
8. 1～4で登録した操作のうちで、連携サービスに加えたい操作を登録したボタンを押す
9. 8を繰り返す

図 3 学習リモコンによる連携サービスの作成

#### 2.4.3 既存の連携サービス作成法の問題点

2.4.1 節で述べたプログラムによる連携サービス作成法の問題点は、プログラミングに関する知識を持っていないエンドユーザは、連携サービスを作成することが不可能だということである。

2.4.2 節で述べた学習リモコンによる連携サービス作成法の問題点は、赤外線信号の登録作業が煩雑なため、手間のかかる作業や機械の操作が苦手なエンドユーザが敬遠するということである。また、図3の手順8～9では、連携サービスの作成状況が視覚的にわからないため、どこまで作業を進めたのかが分かりづらいという問題点もある。

HNS を利用するエンドユーザには、子供、主婦、高齢者など、PC や機械に関

する知識を持たない人たちも多い．よって、既存の連携サービス作成法では、エンドユーザが連携サービスを作成することは困難だといえる．

なお、これ以降「エンドユーザ」とはプログラミングに関する知識を持たないエンドユーザを指すものとする．

## 2.5 エンドユーザによる連携サービス作成の必要性

2.4.3 節において、エンドユーザが連携サービスを作成することは困難だという問題点を挙げたが、この節ではエンドユーザが連携サービスを作成したいと考えているのかどうかということについて述べる．

一般的な HNS では、エンドユーザはベンダが提供した連携サービスしか利用することができない．しかし、その連携サービスがエンドユーザの所望するものと異なる動作をする場合、エンドユーザは連携サービスをカスタマイズ、もしくは新規に作成したいと考えるはずである．それを確認するため、本論文で提案するシステムのプロトタイプを実装し、50 名のエンドユーザに対し、NAIST-HNS 内で連携サービスを作成させるという予備実験を行った．予備実験終了後、被験者には「自分で作成した連携サービスが使えたらどう思うか？」というアンケートに 5 段階評価（非常に役に立つ・役に立つ・どちらでもない・役に立たない・非常に役に立たない）で答えてもらった．アンケート結果は図 4 のようになり、非常に役に立つ・役に立つという肯定的な意見は 44 名（88% ）、どちらでもないという意見は 6 名（10% ）、役に立たない・非常に役に立たないという否定的な意見は 0 名（0% ）となった．このことから、エンドユーザの連携サービス作成のニーズは十分にあると考えた．

(問) 自分で作成した連携サービスが使えるらどう思うか？

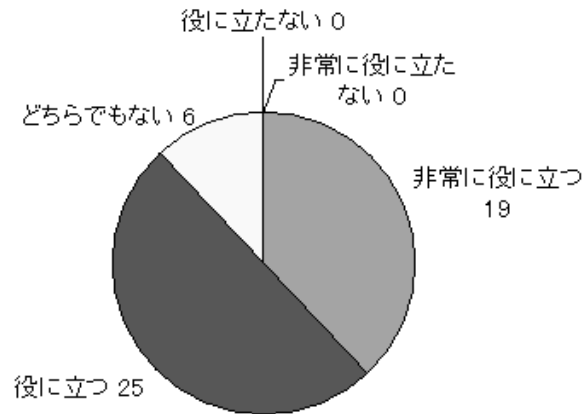


図 4 予備実験のアンケート結果

## 2.6 本研究の目的

2.4.3 節で述べた既存の連携サービス作成法の問題点を解決するため、本研究では、プログラミングに関する知識を持たないエンドユーザが、連携サービスの作成を行えるようになることを目的とする。そして、そのためのシステムを提案する。また、連携サービスの作成だけではなく、編集と削除も行えるようにすることで、エンドユーザが作成した連携サービスを取り扱えるようにする。

## 3. 提案システム

### 3.1 要件定義とアプローチ

2.4.3 節で述べた既存の連携サービス作成法の問題点を解決するため、本研究では、エンドユーザが連携サービスを作成・編集・削除できることを目的としたシステム（BAMBEE）を構築する。このシステムでは、GUIを用いたビジュアルプログラミングにより、エンドユーザが連携サービスの作成・編集・削除を行うことができるようにする。

提案システムに要求される要件を以下のように定義した。また、その要件を満たすためのアプローチの概要も共に示す。要件 R1～R3 は 2.4.3 節で述べた既存の連携サービス作成法の問題点を解決するものであり、要件 R4～R6 は提案システムの利便性を向上させるためのものである。

- 要件 R1：プログラミングの知識が無くても連携サービスの作成ができる
- 要件 R2：PC や機械の熟練度に関係なく操作できる
- 要件 R3：連携サービスの作成・編集状況が視覚的に把握できる
- 要件 R4：文字による操作方法の説明が少なくても操作できる
- 要件 R5：作成・編集した連携サービスの動作確認ができる
- 要件 R6：ユーザの入力忘れを防止できる

要件 R1 はプログラムによる連携サービス作成法の問題点を解決する要件である。R1 が解決する問題点は、プログラミングの知識を持っていないエンドユーザは、連携サービスを作成することが不可能だということである。R1 を満たすアプローチとしては、エンドユーザが BAMBEE の GUI を操作することにより、家電操作 API を呼び出す SOAP クライアントプログラムの生成を行うようにすることである。BAMBEE が生成する SOAP クライアントプログラムに関しては、3.7 節で詳しく述べる。

要件 R2 は学習リモコンによる連携サービス作成法の問題点を解決する要件である。R2 が解決する問題点とは、機械の操作が苦手なエンドユーザが、連携サービス作成の作業を敬遠するということである。R2 を満たすアプローチとしては、指一本で操作できるシステムを構築することである。具体的には、以下の2つの方法を R2 を満たすアプローチとした。1つ目は、BAMBEE の GUI を表示する装置にタッチパネルを用いることである。これに関しては、3.3 節で詳しく述べる。2つ目は、GUI をボタンを並べただけのシンプルなものにすることである。これに関しては、3.4.1 節で詳しく述べる。

要件 R3 は学習リモコンによる連携サービス作成法の問題点を解決する要件である。R3 が解決する問題点は、連携サービスの作成・編集状況が把握しづらいということである。エンドユーザが連携サービスの作成・編集状況を把握しやすくするためには、その状況を視覚化する必要があると考えた。R3 を満たすアプローチとしては、作成・編集中の連携サービスの内容（操作する家電とその操作）を、操作フローとして表示させるようにすることである。操作フローには連携サービスに組み込まれる家電とその操作がアイコンになって、時系列順に表示されるようにする。詳しくは 3.4.3 節に示す。

要件 R4 はシステムの利便性を向上させる要件である。R4 を定義した理由としては、操作方法の説明を行うために GUI に文字を多用した場合、画面内の情報量が増え、逆に操作方法が理解しづらくなると考えたためである。そこで、文字を読んで理解する GUI ではなく、見て聞いて理解できる GUI を目指した。R4 を満たすアプローチとしては、以下の2つの方法を用いた。1つ目は、GUI にアイコンを多用することである。これに関しては、3.4.1 節で詳しく述べる。2つ目は、音声でも操作の案内を行うようにすることである。これに関しては、3.4.2 節で詳しく述べる ..

要件 R5 はシステムの利便性を向上させる要件である。R5 を定義した理由としては、エンドユーザは作成・編集した連携サービスが所望の動作をしているかどうかを確認したいはずだと考えたためである。R5 を満たすアプローチとしては、連携サービスの動作テストを行う機能を実装することである。詳しくは 3.5.4 節に示す。

要件 R6 はシステムの利便性を向上させる要件である。R6 を定義した理由としては、2.5 節で述べた BAMBEE のプロトタイプを用いて連携サービスをエンドユーザに作成させるという予備実験において、必要な操作を入れ忘れるエンドユーザが多かったためである。例えば、DVD プレーヤの電源を ON にせず音量設定を行おうとすることなどがそれに該当する。R6 を満たすアプローチとしては、連携サービスの作成時、依存する家電の操作を自動で挿入する機能を実装することである。詳しくは 3.6.2 節に示す。

## 3.2 システム概要

図 5 に示すように、BAMBEE はエンドユーザが操作する GUI と、ファイル操作を行うバックグラウンドプログラム（以降、BG-PG と呼ぶ）で構成される。

連携サービスファイルは連携サービスを実行させるプログラムであり、BAMBEE によって作成・編集・削除・動作テストが行われる。詳しくは 3.7 節で述べる。

index ファイルは既存の連携サービスファイルの情報を登録しておくもので、BG-PG は index ファイルの情報を GUI に表示させ、エンドユーザに操作を行わせる。ここで、index ファイルの内容を図 6 に示す。index ファイルでは、行の先頭に連携サービスファイルの番号が記述されており、その後に家電操作命令が記述されている。連携サービスファイルに番号を付けているのは、GUI で編集・削除・動作テストを行う連携サービスを決める時に使用するためである。

エンドユーザが GUI を操作すると、その操作内容が BG-PG に送られ、BG-PG が連携サービスファイルに対して作成・編集・削除・動作テストなどの処理を行う。そして、BG-PG が連携サービスファイルに対して何らかの処理を行うと、その結果を index ファイルに反映させ、連携サービスファイルと同期をとるようにしている。以上のような動作が BAMBEE の動作の概要である。

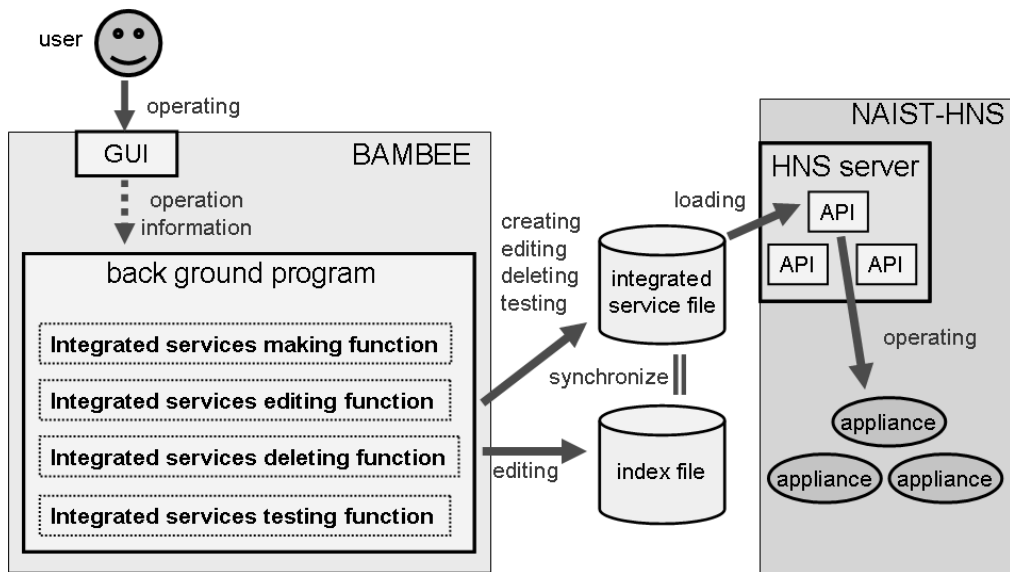


図 5 システム構成

```

1,$service_ac->on(),$service_curtain->open()
2,$service_cir->on(),$service_cir->airfrowControl(2)
3,$service_cir->on(),$service_cir->airfrowControl(2),$service_ac->on()
4,$service_pdp->off(),$service_dvd->off(),$service_hdr->off()

```

図 6 index ファイルの内容

### 3.3 ハードウェア選定

要件 R2 を満たすためのアプローチの 1 つとして、BAMBEE の GUI を表示する装置にタッチパネルを用いることにした。タッチパネルは銀行の ATM などでも利用されており、最近では携帯ゲーム機にも採用され、扱いやすいヒューマンインタフェースデバイスとして注目されている。

タッチパネルでは、クリック、ダブルクリック、ドラッグ&ドロップなどのマウスによる操作を指一本で行うことができるが、キーボードの役目を果たすものは無い。よって、タッチパネルに表示させる GUI は、マウスのみで操作できるものにしておく必要がある。しかし、その条件さえ満たしていれば、PC の熟練度に関係なく、エンドユーザは指一本で BAMBEE を操作することができる。

本研究で利用したタッチパネルは、三菱電機エンジニアリングの TSD-T124-M[13] である。その概観を図 7 に示す。



図 7 使用したタッチパネル



## 3.4 GUI

本節ではタッチパネルに表示させる BAMBEE の GUI について述べる。3.4.1 節では GUI の画面例を示し、それを基に GUI の概要を述べる。3.4.2 節では GUI の画面遷移について述べ、また、音声で操作方法を案内する機能についても述べる。3.4.3 節では連携サービスの内容を表示する操作フローについて述べる。

### 3.4.1 GUI 概要

BAMBEE の GUI は、要件 R2 を満たすために、指一本で操作できるボタンを並べただけのシンプルなものになっている。GUI の画面の例として、連携サービスの作成画面を図 8 に示す。図 8 では、右上の家電リストに「テレビ」「ライト」などのボタンがあり、右下には「入れ替え」「削除」などのボタンがある。エンドユーザはこれらのボタンを押すだけで、連携サービスを作成することができる。

また、要件 R4 を満たすためのアプローチの 1 つとして、GUI にアイコンを多用するようにしている。例えば、図 8 では家電リスト内のボタンを家電のアイコン付きにしている。

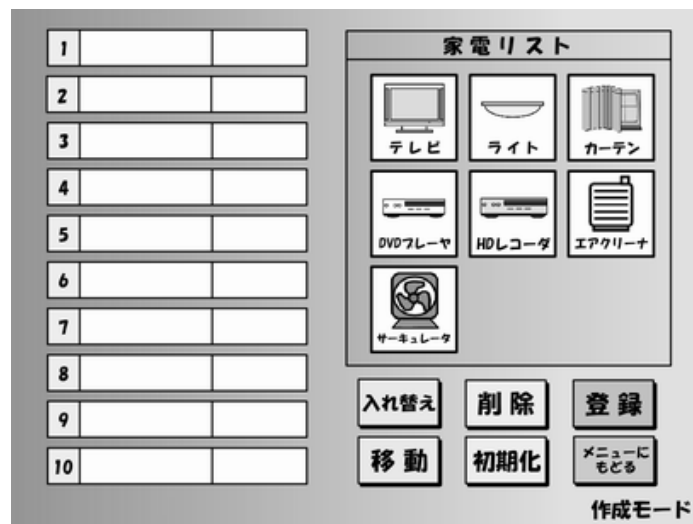


図 8 GUI 概要

### 3.4.2 GUIの画面遷移と音声案内

図9にGUIの画面遷移を示す。BAMBEEのGUIでは、まず始めにメニュー画面から「作成モード」「編集モード」「削除モード」「テストモード」を選択できるようになっている。そして、いずれかのモードを選択すると画面が移り変わり、そのモードの操作方法が音声で流れる。この音声案内は、要件R4を満たすアプローチの1つであり、これにより文字による操作方法の案内を減らすようにしている。例えば、「削除モード」では「右側のサービスリストから連携サービスを選び、削除ボタンを押してください。」という音声案内が流れるようになっている。

メニュー画面で「作成モード」を選択すると、連携サービスの作成画面に移り変わり、連携サービスの作成方法に関する音声案内が流れる。作成画面では、まず右上の家電リストから連携サービスに組み込みたい家電を選択する。そうすると、その家電の操作を選択する画面に切り替わり、エンドユーザは必要な操作を選択する。これを繰り返すことで、連携サービスを作成することができる。

メニュー画面で「編集モード」を選ぶと、編集を行う連携サービスを選択する画面に移り変わり、連携サービスの編集方法に関する音声案内が流れる。次に、右上のサービスリストから編集したい連携サービスの番号を押し、右下の「選択ボタン」を押すと、連携サービスの編集画面に移り変わる。ここで、エンドユーザは連携サービスを編集することができる。

メニュー画面で「削除モード」を選ぶと、削除する連携サービスを選択する画面に移り変わり、連携サービスの削除方法に関する音声案内が流れる。ここで、エンドユーザが右上のサービスリストから削除したい連携サービスの番号を選ぶことで、その連携サービスを削除することができる。

メニュー画面で「テストモード」を選ぶと、動作テストを行う連携サービスを選択する画面に移り変わり、連携サービスの動作テストの方法に関する音声案内が流れる。ここで、エンドユーザは右上のサービスリストから動作テストを行いたい連携サービスの番号を選ぶことで、その連携サービスの動作テストを行うことができる。

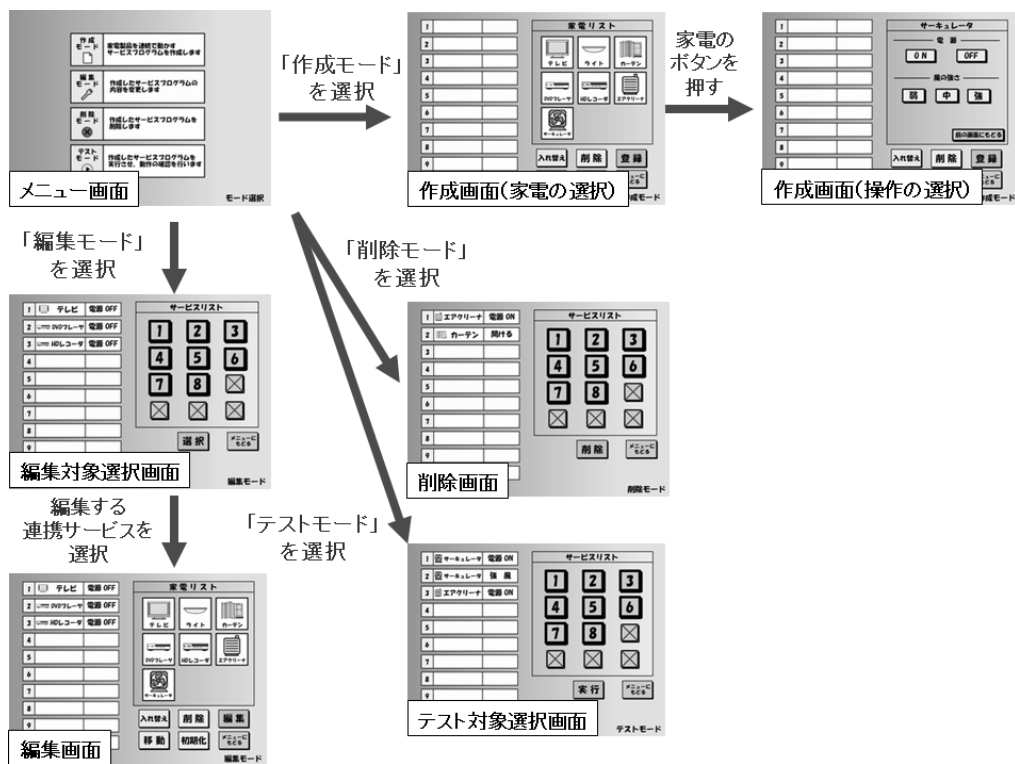


図 9 GUIの画面遷移

### 3.4.3 操作フロー

要件 R3 を満たすアプローチとして、作成・編集中の連携サービスの内容（操作する家電とその操作）を、操作フローとして表示させるようにした。操作フローは図 10 のようになり、連携サービスに組み込まれる家電とその操作がアイコンになって、時系列順に表示されるようになっていて、例えば、図 10 の操作フローでは、まず始めにエアクリーナの電源が ON になり、次にサーキュレータの電源が ON になり、最後にサーキュレータの風量が強風になるようになっている。

また、編集・削除・動作テストを行う連携サービスを選択する時に、操作フローに現在選択中の連携サービスの操作内容を表示させることにより、エンドユーザは視覚的に連携サービスの操作内容を理解することができるようになっている。



図 10 操作フロー

## 3.5 システムの主機能

連携サービスの作成・編集・削除・動作テストがBAMBEEの主な機能である。本節ではそれらの機能について述べる。

### 3.5.1 連携サービスの作成機能

連携サービスの作成機能は、連携サービスを新規作成する機能であり、エンドユーザが所望の連携サービスを作成するために必要な機能である。

図11に連携サービスの作成手順を示す。まず、メニュー画面で「作成モード」を選択する。次に、右側の家電リストから操作させたい家電を選択し、その操作も選択する。そうすると、左側の操作フローに先ほど選択した操作が追加される。このようにして必要な操作を操作フローに追加していき、最後に「登録ボタン」を押すことで連携サービスが作成される。

図12に連携サービスの作成の流れを示す。まず、エンドユーザがGUIで連携サービスを作成する。そうすると、GUIで作成された連携サービスの情報（どのような家電のどのような操作を、どのような順番で行うかという情報）がBG-PGに送信される。BG-PGはその連携サービスの情報を基にして、連携サービスファイルを作成する。また、それと同時に作成した連携サービスの情報をindexファイルに書き込み、現存する連携サービスファイルとの同期をとる。

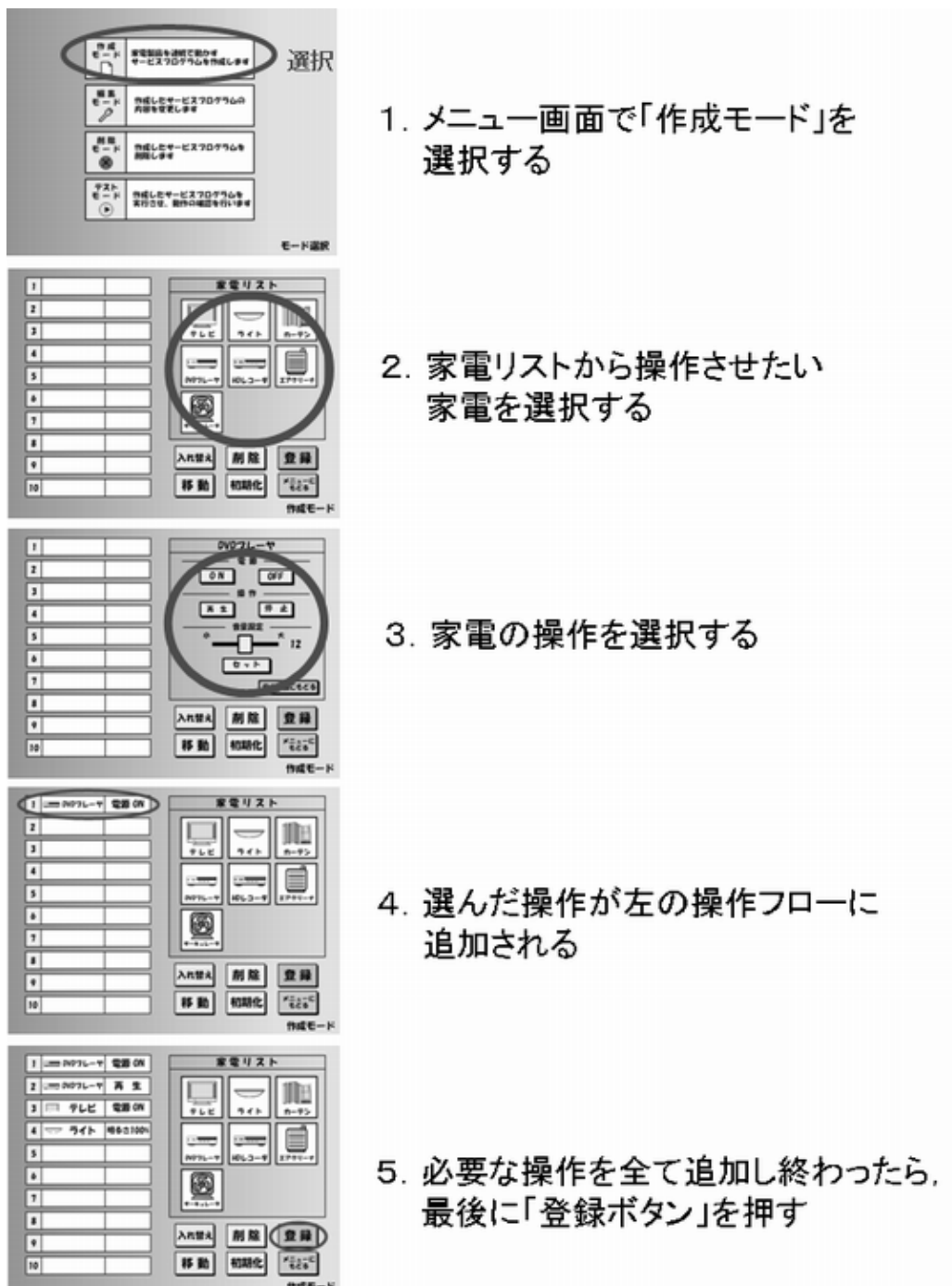


図 11 連携サービスの作成手順

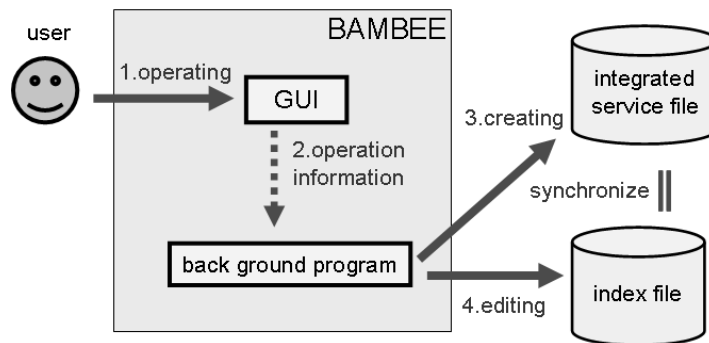


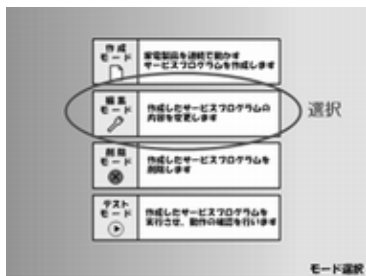
図 12 連携サービスの作成の流れ

### 3.5.2 連携サービスの編集機能

連携サービスの編集機能は、既存の連携サービスを編集する機能であり、作成した連携サービスを作り変えたいというエンドユーザのニーズに答えるための機能である。

図 13 に連携サービスの編集手順を示す。まず、メニュー画面で「編集モード」を選択する。次に、右側のサービスリストから編集したい連携サービスを選択する。ここで、サービスリスト内の数字ボタンを押すことにより、その番号の連携サービスの操作内容が、左側の操作フローに表示される。数字ボタンを押していき、編集したい連携サービスが見つかったら、「選択ボタン」を押すことで編集画面に切り替わる。そして、ここで連携サービスの編集を行い、最後に「編集ボタン」を押すことによって、連携サービスの編集が完了する。

図 14 に連携サービスの編集の流れを示す。まず、エンドユーザが GUI の「編集モード」を選択すると、BG-PG が index ファイルから既存の連携サービスファイルの情報を取得し、GUI に表示させる。エンドユーザは GUI で編集したい連携サービスを選び、編集を行う。そうすると、編集を行った連携サービスの情報が BG-PG に送られ、その情報を基に BG-PG が連携サービスファイルの内容を編集する。また、それと同時に index ファイルの連携サービス情報も編集し、現存する連携サービスファイルとの同期をとる。



1. メニュー画面で「編集モード」を選択する



2. サービスリストから編集したい連携サービスを選択する



3. 選択した連携サービスの操作内容が操作フローに表示される



4. 編集する連携サービスが決まったら「選択ボタン」を押す



5. 編集を行い、終わったら最後に「編集ボタン」を押す

図 13 連携サービスの編集手順



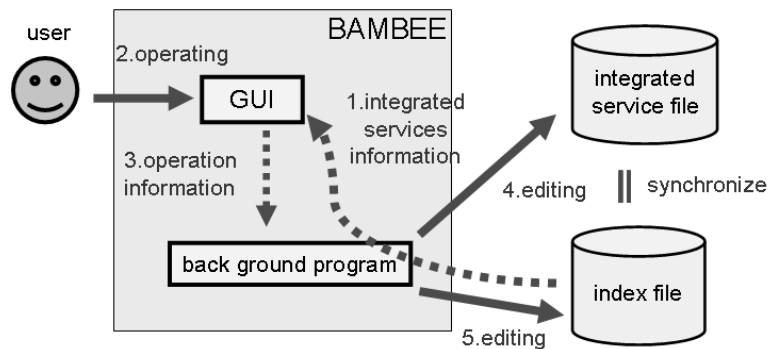


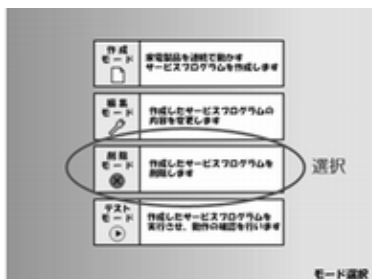
図 14 連携サービスの編集の流れ

### 3.5.3 連携サービスの削除機能

連携サービスの削除機能は、不要になった連携サービスを削除する機能である。BAMBEE では 12 個までしか連携サービスを扱うことができないので、不要になった連携サービスは削除する必要があり、そのためにこの機能が必要である。

図 15 に連携サービスの削除手順を示す。まず、メニュー画面で「削除モード」を選択する。次に、右側のサービスリストから削除したい連携サービスを選択する。ここで、サービスリスト内の数字ボタンを押すことにより、その番号の連携サービスの操作内容が、左側の操作フローに表示される。数字ボタンを押していき、削除したい連携サービスが見つかったら、「削除ボタン」を押すことで連携サービスが削除される。

図 16 に連携サービスの削除の流れを示す。まず、エンドユーザが GUI の「削除モード」を選択すると、BG-PG が index ファイルから既存の連携サービスファイルの情報を取得し、GUI に表示させる。そして、エンドユーザは GUI で削除したい連携サービスを選び、削除を行う。そうすると、削除した連携サービスの情報が BG-PG に送られ、その情報を基に BG-PG が連携サービスファイルを削除する。また、それと同時に index ファイルの連携サービス情報を削除し、現存する連携サービスファイルとの同期をとる。



1. メニュー画面で「削除モード」を選択する



2. サービスリストから削除したい連携サービスを選択する



3. 選択した連携サービスの操作内容が操作フローに表示される



4. 削除する連携サービスが決まったら「削除ボタン」を押す

図 15 連携サービスの削除手順

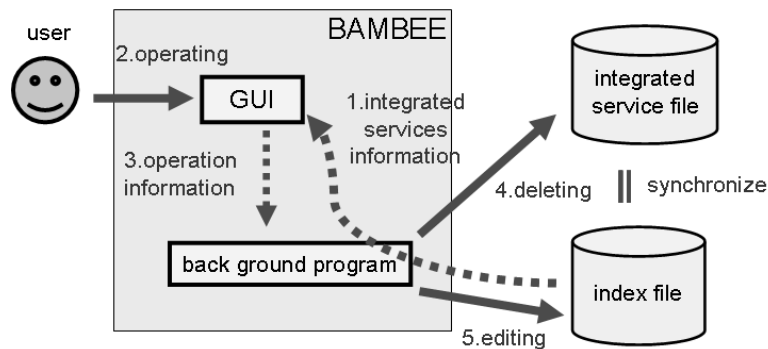


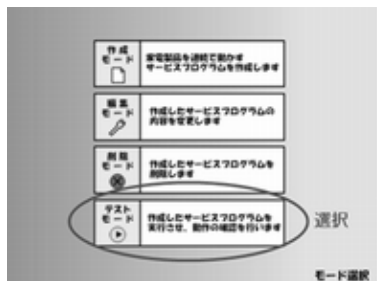
図 16 連携サービスの削除の流れ

### 3.5.4 連携サービスの動作テスト機能

連携サービスの動作テスト機能は、作成・編集した連携サービスの動作テストを行う機能であり、要件 R5 を満たすための機能である。

図 17 に連携サービスの動作テスト手順を示す。まず、メニュー画面で「テストモード」を選択する。次に、右側のサービスリストから動作をテストしたい連携サービスを選択する。ここで、サービスリスト内の数字ボタンを押すことにより、その番号の連携サービスの操作内容が、左側の操作フローに表示される。数字ボタンを押していき、動作をテストしたい連携サービスが見つかったら、「実行ボタン」を押すことで連携サービスが実行される。

図 18 に連携サービスの動作テストの流れを示す。まず、エンドユーザが GUI の「テストモード」を選択すると、BG-PG が index ファイルから既存の連携サービスファイルの情報を取得し、GUI に表示させる。そして、エンドユーザは GUI で動作をテストしたい連携サービスを選ぶ。そうすると、指定された連携サービスファイルが実行され、HNS サーバの家電操作 API を呼び出し、家電に操作命令を送る。



1. メニュー画面で「テストモード」を選択する



2. サービスリストから動作テストさせたい連携サービスを選択する



3. 選択した連携サービスの操作内容が操作フローに表示される



4. 動作テストさせる連携サービスが決まったら「実行ボタン」を押す

図 17 連携サービスの動作テスト手順

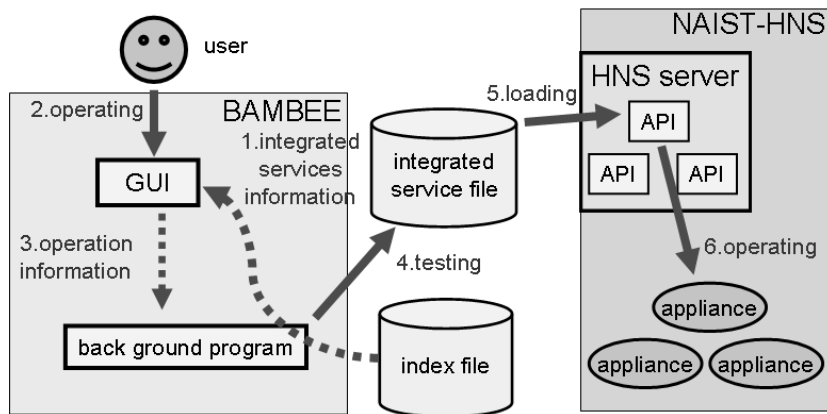


図 18 連携サービスの動作テストの流れ

### 3.6 システムの補助機能

操作フローの編集機能と自動挿入機能がBAMBEEの補助機能である。本節ではそれらの機能について述べる。

#### 3.6.1 操作フロー編集機能

操作フロー編集機能は、連携サービスの作成・編集時に操作フローの編集を行う機能である。操作フロー編集機能としては、図19に示すように、「削除」「初期化」「入れ替え」「移動」がある。それぞれの機能についての説明は、図20と以下に示す。

- 削除：操作フローに最後に追加した操作を削除する
- 初期化：操作フローの状態を初期化する
- 入れ替え：2つの操作の順番を入れ替える
- 移動：操作の順番を移動させる

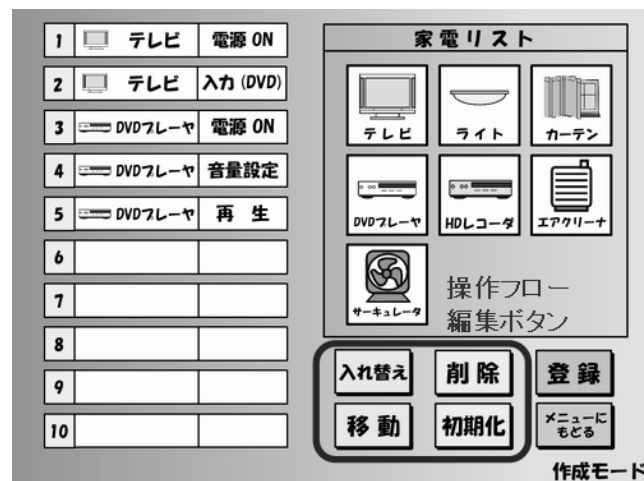


図19 操作フロー編集ボタン

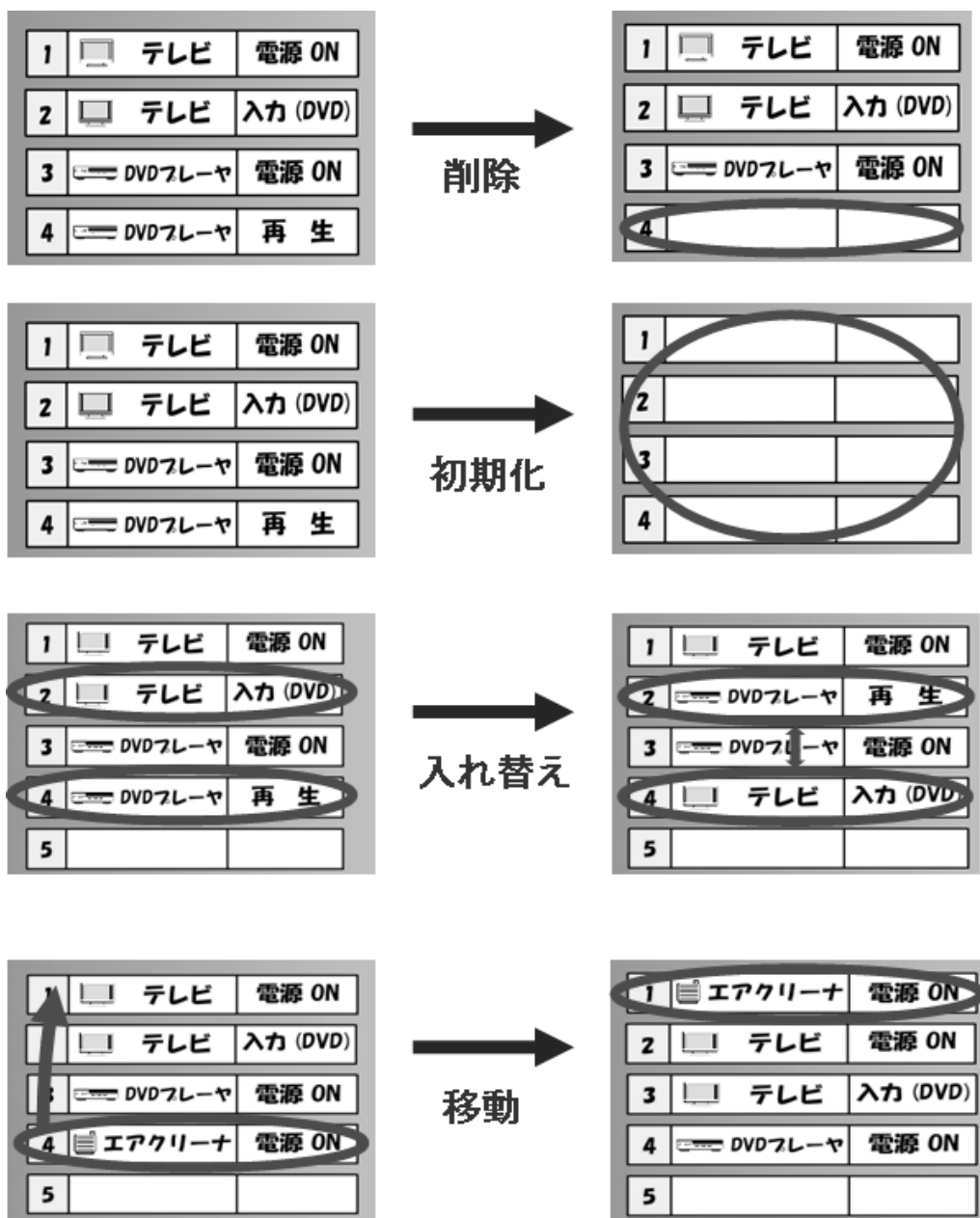


図 20 操作フロー編集機能

### 3.6.2 自動挿入機能

自動挿入機能は、連携サービスの作成・編集時に、依存する家電の操作を自動で挿入する機能であり、要件 R6 を満たすためのアプローチである。エンドユーザが家電の操作を入力した時に、その操作に依存する操作が事前に入力されていなかった場合、自動挿入機能が作動する。自動挿入機能は2つのパターンがあり、それらを以下に示す。

1つ目のパターンは、必要な操作が強制的に挿入されるパターンである。例えば、サーキュレータは電源を ON にした後でしか風量を変更することはできない。よって、「サーキュレータの電源を ON にする」という操作を入力せずに、「サーキュレータの風量を強風にする」という操作を入力しようとした場合、「サーキュレータの電源を ON にする」という操作が強制的に挿入される。

2つ目のパターンは、図 21 のような選択肢が出現し、関連する操作を挿入するかどうかをエンドユーザが選択できるパターンである。DVD プレーヤを再生させる図 21 の例を基に、このパターンを説明する。DVD プレーヤを再生する場合に必ず必要な操作は、「DVD プレーヤの電源を ON にする」という操作である。よって、図 21 の選択肢のどちらを選んでも、この操作は強制的に挿入される。しかし、「テレビの電源を ON にする」と「テレビの入力を DVD プレーヤに切り替える」という操作は、必ずしも必要な操作ではない。なぜならば、DVD プレーヤでは CD を再生することもできるので、DVD を見るのではなく CD を聞きたい場合は、テレビの操作は必要無いのである。よって、図 21 の選択肢で「はい」を選ぶと、「テレビの電源を ON にする」と「テレビの入力を DVD に切り替える」という操作が挿入され、「いいえ」を選択するとそれらは挿入されない。

依存する家電の操作は、表 1 のようになっており、2つ目のパターンが設定されているのは、DVD プレーヤと HD レコーダの操作のみである。



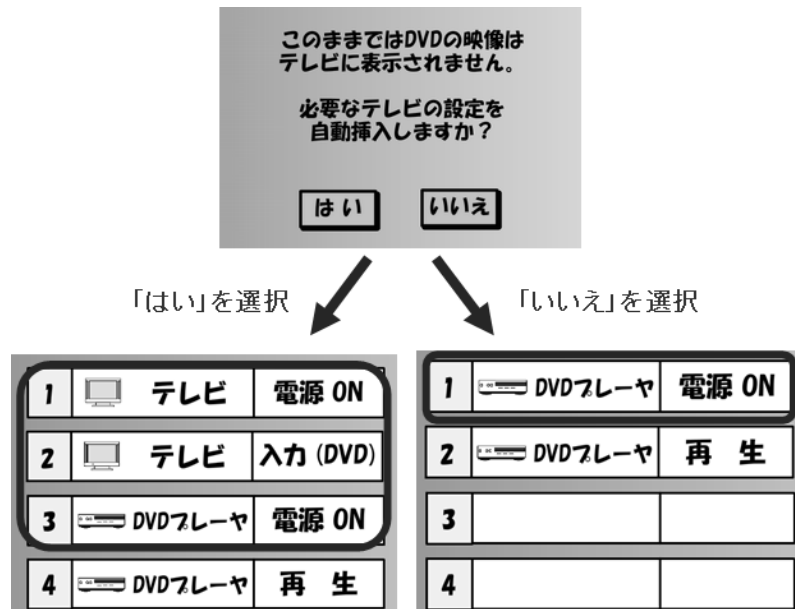


図 21 表示される選択肢

表 1 依存する家電の操作

入力する操作	強制的に挿入される操作	挿入するかどうかを選択できる操作
テレビ (音量設定)	テレビ (電源 ON)	
テレビ (入力切替 [テレビ])	テレビ (電源 ON)	
テレビ (入力切替 [DVD プレーヤ])	テレビ (電源 ON)	
テレビ (入力切替 [PC])	テレビ (電源 ON)	
DVD プレーヤ (再生)	DVD プレーヤ (電源 ON)	テレビ (電源 ON) テレビ (入力切替 [DVD プレーヤ])
DVD プレーヤ (音量設定)	DVD プレーヤ (電源 ON)	
HD レコーダ (再生)	HD レコーダ (電源 ON)	テレビ (電源 ON) テレビ (入力切替 [テレビ])
サーキュレータ (風量設定 [弱])	サーキュレータ (電源 ON)	
サーキュレータ (風量設定 [中])	サーキュレータ (電源 ON)	
サーキュレータ (風量設定 [強])	サーキュレータ (電源 ON)	

### 3.7 実装

1台のノートPCの中にBAMBEEを実装し、3.3節で述べたタッチパネルディスプレイに接続した。BAMBEEの実装環境を以下に示す。

- PC : DELL Latitude D600 ( PentiumM 1.60GHz, 1024MB, 40GB, WinXP Pro )
- GUI : Macromedia FLASH8
- BG-PG : Active Perl

BAMBEEにより生成される連携サービスファイルは、Perlで記述されており、図22のようになっている。要件R1を満たすため、連携サービスファイルはSOAPにより家電操作APIを呼び出すSOAPクライアントプログラムの役割を果たしており、SOAPの実装にはSOAP::Liteモジュール[14]を用いた。

図22に示した連携サービスファイルの例では、まず、3行目でSOAP::Liteモジュールを使用することを宣言している。6行目では家電操作APIが保存されているPC (NAIST-HNSのホムサーバ)のアドレスを指定している。9~16行目ではホームサーバに保持されているWSDLの場所を指定している。19~26行目ではSOAP::Liteモジュールを使用するために、インスタンス化を行っている。最後に、29行目でエアクリーナをONにするという命令、30行目でカーテンを開けるという命令を行っている。

なお、1~28行目のコードはテンプレートファイルとして予め準備している。そして、連携サービスの作成時は、まず始めにテンプレートファイルの内容を呼び出し、次にGUIに入力された家電の操作命令を29行目以降に追加することで、連携サービスファイルを作成するようにしている。

```

1 #!/usr/bin/perl
2
3 use SOAP::Lite;
4
5 #サーバ名
6 $server = 'http://minipc01.naist.jp:8080';
7
8 #サービスエンドポイント (WSDL を指定)
9 $endpoint_pdp = "$server/crossamobject/services/PDP?wsdl";
10 $endpoint_dvd = "$server/crossamobject/services/DVD?wsdl";
11 $endpoint_curtain = "$server/crossamobject/services/CURTAIN?wsdl";
12 $endpoint_light = "$server/crossamobject/services/LIGHT?wsdl";
13 $endpoint_light2 = "$server/crossamobject/services/LIGHT2?wsdl";
14 $endpoint_cir = "$server/crossamobject/services/CIRCULATOR?wsdl";
15 $endpoint_ac = "$server/crossamobject/services/AIR_CLEANER?wsdl";
16 $endpoint_hdr = "$server/crossamobject/services/HDR?wsdl";
17
18 #サービスインスタンス化
19 $service_pdp = SOAP::Lite -> service("$endpoint_pdp");
20 $service_dvd = SOAP::Lite -> service("$endpoint_dvd");
21 $service_curtain = SOAP::Lite -> service("$endpoint_curtain");
22 $service_light = SOAP::Lite -> service("$endpoint_light");
23 $service_light2 = SOAP::Lite -> service("$endpoint_light2");
24 $service_cir = SOAP::Lite -> service("$endpoint_cir");
25 $service_ac = SOAP::Lite -> service("$endpoint_ac");
26 $service_hdr = SOAP::Lite -> service("$endpoint_hdr");
27
28 # ***** MAIN CODE ***** #
29 $service_ac->on();
30 $service_curtain->open();

```

図 22 生成された連携サービスファイルの内容

## 4. 評価実験

### 4.1 実験概要

BAMBEE を利用することにより，エンドユーザが連携サービスの作成・編集・削除を行うことができるかどうかを確認するため，ユーザビリティテストを実施した．この実験では，BAMBEE を利用して連携サービスを作成・編集・削除するという3つのタスクを，24名のエンドユーザと4名のBAMBEEの熟練者に課した．実験結果の評価には，ユーザビリティに関する国際規格ISO9241-11[16]の評価指標を用い，システムの有効性，効率性，満足度を測定した．

### 4.2 被験者

プログラミングに関する知識を持たないエンドユーザ24名を被験者にした．また，参考のためにBAMBEEを使い慣れた熟練者4名も同じ実験を行った．エンドユーザの区分には，マーケティングで用いられる集計区分[15]を用い，表2のように6グループに区分した．M1は20～34歳の男性，M2は35～49歳の男性，M3は50歳以上の男性，F1は20～34歳の女性，F2は35～49歳の女性，F3は50歳以上の女性で構成される．なお，1グループの人数は4人である．

表 2 被験者区分

group	sex	age	the number of people
M1	Male	20 - 34	4
M2	Male	35 - 49	4
M3	Male	50 -	4
F1	Female	20 - 34	4
F2	Female	35 - 49	4
F3	Female	50 -	4

### 4.3 使用した家電機器

本実験では以下の家電機器を使用した。

- テレビ：NEC PX-50XM2
- HDレコーダ：東芝 RF-XS46
- DVDプレーヤ：パイオニア HTZ-535DV
- 天井照明：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
- サーキュレータ：森田電工 MCF-257NR

#### 4.4 実験手順

以下の手順で実験を行った．実験の様子を図 23 に示す．

1. 実験部屋の説明を行う
2. 連携サービスの説明を行う
3. 10 分間，システムの説明書を見ながらシステムを使ってもらう
4. 3つのタスクを行ってもらう（タスクの順番は被験者ごとに変える）
5. SUS 質問紙と自由記述アンケートに回答してもらう

タスク内容に関しては 4.5 節に示す．SUS 質問紙と自由記述アンケートに関しては 4.6.3 節に示す．



図 23 実験の様子

## 4.5 タスク

本節では被験者に行ってもらった以下の3つのタスクについて示す。

- 連携サービスの作成
- 連携サービスの編集
- 連携サービスの削除

### 4.5.1 連携サービスの作成

このタスクは「DVDシアターサービス」という連携サービスを作成するタスクである。「DVDシアターサービス」とは、DVDの観賞ができるように家電の状態を変更させる連携サービスである。

#### タスク手順

1. 以下の操作を実行する「DVDシアターサービス」を作成する。

- 天井照明を消灯する
- カーテンを閉じる
- テレビの電源をONにした後、入力をDVDプレーヤに切り替える
- DVDプレーヤの電源をONにした後、再生する

2. 作成した「DVDシアターサービス」の動作テストを行う
3. 正しく作成することができた場合はタスクを終了し、間違っていた場合は作り直す。

#### 4.5.2 連携サービスの編集

このタスクは既に作成されている「空気清浄サービス」という連携サービスを編集するタスクである。「空気清浄サービス」とは、部屋の中の空気を綺麗にする連携サービスであり、以下のような操作を行う。

1. サーキュレータの電源を ON にする
2. サーキュレータの風量を強風にする
3. 空気清浄機の電源を ON にする

#### タスク手順

1. 既に作成されている「空気清浄サービス」を以下のように変更する。なお、操作の順番は以下の順番を守るものとする。

1. カーテンを開ける
2. サーキュレータの電源を ON にする
3. サーキュレータの風量を弱風にする
4. 空気清浄機の電源を ON にする

2. 編集した「空気清浄サービス」の動作テストを行う
3. 正しく編集することができた場合はタスクを終了し、間違って編集した場合は編集し直す。



### 4.5.3 連携サービスの削除

このタスクは既に作成されている「おでかけサービス」という連携サービスを削除するタスクである。「おでかけサービス」とは、複数の家電の電源を全てOFFにし、ユーザが直ちに外出できる状態にする連携サービスであり、以下のような操作を行う。

1. 天井照明を消灯する
2. カーテンを閉める
3. テレビの電源を OFF にする
4. DVD プレーヤの電源を OFF にする
5. HD レコーダの電源を OFF にする
6. 空気清浄機の電源を OFF にする
7. サーキュレータの電源を OFF にする

#### タスク手順

1. 既に作成されている「おでかけサービス」を削除する。
2. 正しく削除ができた場合も、間違っって別の連携サービスを削除してしまった場合もタスクを終了する。

## 4.6 評価指標

本実験での評価指標として、ユーザビリティに関する国際規格 ISO9241-11[16] の評価指標を用いた。ISO9241 は視覚表示装置 (VDTs) を用いたオフィス作業に対する人間工学的要求事項を取り扱っており、全 17 章で構成される、ユーザビリティの定義は第 11 章で行われている。

ISO9241-11 では、ユーザビリティは「ある製品が、指定された利用者によって、指定された利用の状況下で、指定された目標を達成するために用いられる際の有効さ、効率及び満足度の度合い」として定義されている。そこで、本実験では BAMBEE の有効性・効率性・満足度を測定した。

### 4.6.1 有効性

有効性とは「ユーザが指定された目標を達成する上での正確さ、完全性」を示す。有効性を示す値 [17] として、本実験では以下の値を測定した。

- 操作エラー率
- タスク達成率

操作エラー率は総操作数に対するエラー操作数の割合であり、以下の式で表される。

$$\text{操作エラー率} = \frac{\text{エラー操作数}}{\text{総操作数}}$$

総操作数はタスクを開始してから終了するまでに、被験者がタッチパネルに触れた回数である。エラー操作数はタスクに関係のない操作の回数と、触れても反応のない場所を触れた回数の合計である。

タスク達成率は与えられたタスクの達成の度合いを表すものである。タスクを達成できた場合は、タスク達成率を 100% とし、失敗した場合は 0% とする。また、タスクを達成できた場合でも、事前に設定しておいた減点条件に該当していた場合は、1 つの条件につきタスク達成率から 25% 減算するものとする。3 つのタスクの達成条件、失敗条件、減点条件はそれぞれ異なり、連携サービス作成に

関するものは表3に、連携サービスの編集に関するものは表4に、連携サービスの削除に関するものは表5に示す。

表3 タスク達成率の定義（連携サービスの作成）

達成条件（タスク達成率 100%）	指定したとおりに連携サービスを作成できた
失敗条件（タスク達成率 0%）	指定したものと異なる内容に作成して終了した 諦めて作業を終えた
減点条件（タスク達成率 -25%）	タスクに関する質問をした
	タスク実施中にマニュアルを読んだ
	操作エラー率が20%以上になった タスク達成時間が5分以上になった

表4 タスク達成率の定義（連携サービスの編集）

達成条件（タスク達成率 100%）	指定したとおりに連携サービスを編集できた
失敗条件（タスク達成率 0%）	指定したものと異なる内容に編集して終了した 諦めて作業を終えた
減点条件（タスク達成率 -25%）	質問をした
	タスク実施中にマニュアルを読んだ
	操作エラー率が20%以上になった タスク達成時間が5分以上になった

表 5 タスク達成率の定義（連携サービスの削除）

達成条件（タスク達成率 100%）	指定した連携サービスを削除できた
失敗条件（タスク達成率 0%）	指定したものと異なる連携サービスを削除した
	諦めて作業を終えた
減点条件（タスク達成率 -25%）	質問をした
	タスク実施中にマニュアルを読んだ
	操作エラー率が 20% 以上になった
	タスク達成時間が 2 分以上になった

#### 4.6.2 効率性

効率性とは「ユーザが目標を達成する際に、正確さと完全性に費やした資源」を示す。効率性を示す値 [17] として、本実験では以下の値を測定した。

- タスク達成時間
- 総操作数

タスク達成時間はタスク開始の合図から被験者がタスクの終了を宣言するまでの時間である。総操作数はタスクの開始から終了までに、被験者がタッチパネルに触れた回数である。

### 4.6.3 満足度

満足度とは「製品を使用する際の不快感の無さ，および肯定的な態度」を示す．満足度を示す値 [17] として，本実験では以下のアンケートを被験者に回答してもらった．

- SUS 質問紙
- 自由記述アンケート

SUS ( The System Usability Scale ) 質問紙とは，図 24 に示す 10 個の質問を被験者に 5 段階で評価させることにより，システムの主観的評価を行うものである [18] ．結果は 0 から 100 までの得点として表される．

自由記述アンケートでは，以下の項目について自由に回答してもらった．

- システムの良かった点について
- システムの悪かった点について
- その他，意見や感想など

1. このシステムをしばしば使いたいと思う
2. このシステムは不必要なほど複雑であると感じた
3. このシステムは容易に使えると思った
4. このシステムを使うためには技術専門家のサポートを必要とするかもしれない
5. このシステムは様々な機能がよくまとまっていると感じた
6. このシステムは一貫性の無い所が多くあったと感じた
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
8. このシステムはとても扱いにくいと思った
9. このシステムを使うことに、自分は自信があると感じた
10. このシステムは使い始める前に多くのことを学ぶ必要があると感じた

図 24 SUS 質問紙の質問事項

## 5. 実験結果

### 5.1 各タスクの実験結果

各タスクの実験結果を表6に示す．この表ではタスクを達成した人数と失敗した人数を，タスクごとに示している．連携サービスを作成するというタスクは，全被験者が達成することができた．このタスクの実験結果の詳細は，5.2節に示す．連携サービスを編集するというタスクは，3名の被験者が失敗した．このタスクの実験結果の詳細は，5.3節に示す．連携サービスを削除するというタスクは，全被験者が達成することができた．このタスクの実験結果の詳細は，5.4節に示す．

表6 各タスクの実験結果

	creating of integrated service		editing of integrated service		deleting of integrated service	
	successful	failed	successful	failed	successful	failed
M1	4	0	4	0	4	0
M2	4	0	4	0	4	0
M3	4	0	4	0	4	0
F1	4	0	3	1	4	0
F2	4	0	4	0	4	0
F3	4	0	2	2	4	0
expert	4	0	4	0	4	0
total	28	0	25	3	28	0

## 5.2 連携サービスの作成に関する結果

連携サービスを作成するというタスクは、全被験者が達成することができた。このタスクによって得られた有効性と効率性に関する実験結果を以下に示す。

### 5.2.1 有効性

タスク達成率の結果を図 25 に示す。縦軸はタスク達成率、横軸は被験者のグループを示している。タスク達成率の最低値は50%、最高値は100%であった。各グループのタスク達成率の平均は、M1が100%、M2が100%、M3が100%、F1が100%、F2が約94%、F3が約81%、熟練者が100%であった。全被験者がタスクを達成することはできたが、減点条件に該当した被験者がF2に1名、F3に2名いたため、F2とF3のタスク達成率の平均は、他のグループに比べて低くなった。

操作エラー率の結果を図 26 に示す。縦軸は操作エラー率、横軸は被験者のグループを示している。操作エラー率の最低値は0%、最高値は15%であった。各グループの操作エラー率の平均は、M1が0%、M2が0%、M3が約2%、F1が約1%、F2が約2%、F3が約10%、熟練者が0%であった。

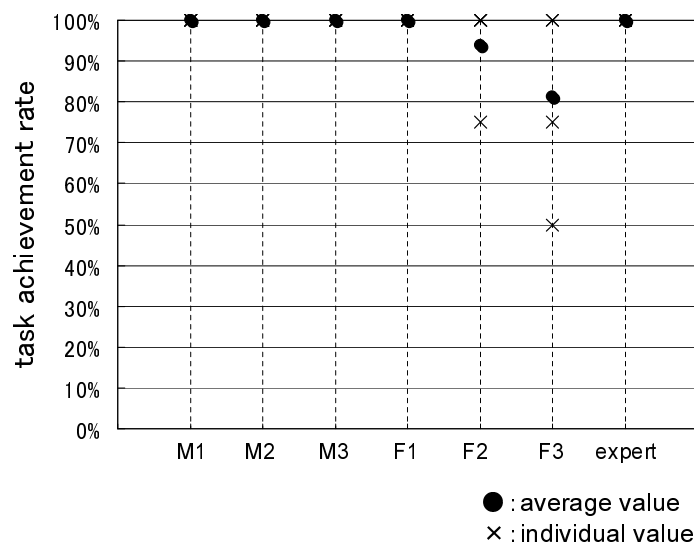


図 25 タスク達成率（連携サービスの作成）



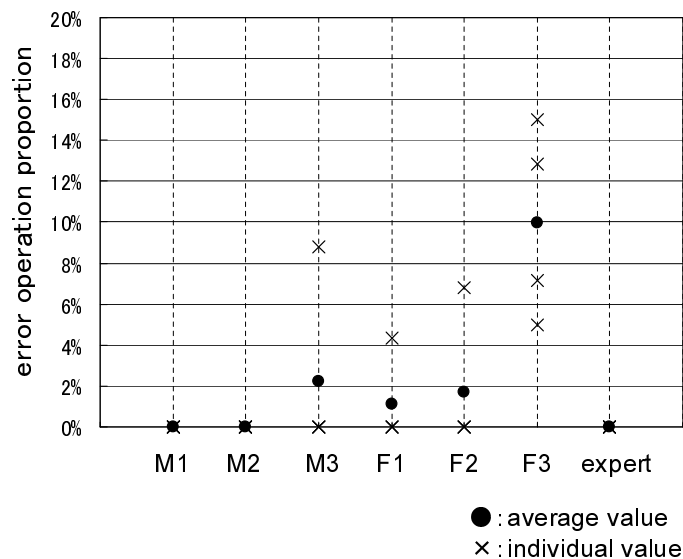


図 26 操作エラー率（連携サービスの作成）

### 5.2.2 効率性

タスク達成時間の結果を図 27 に示す。縦軸はタスク達成時間、横軸は被験者のグループを示している。タスク達成時間の最短値は 34 秒、最長値は 398 秒であった。各グループのタスク達成時間の平均は、M1 が約 87 秒、M2 が約 63 秒、M3 が約 116 秒、F1 が 78 秒、F2 が 166 秒、F3 が約 237 秒、熟練者が約 49 秒であった。よって、F2、F3 は他のグループに比べると、タスクの達成までに時間がかかっていることがわかる。

総操作数の結果を図 28 に示す。縦軸は総操作数、横軸は被験者のグループを示している。総操作数の最低値は 12 回、最高値は 44 回であった。各グループの総操作数の平均は、M1 が約 16 回、M2 が約 16 回、M3 が 21 回、F1 が約 17 回、F2 が約 27 回、F3 が約 28 回、熟練者が約 15 回であった。

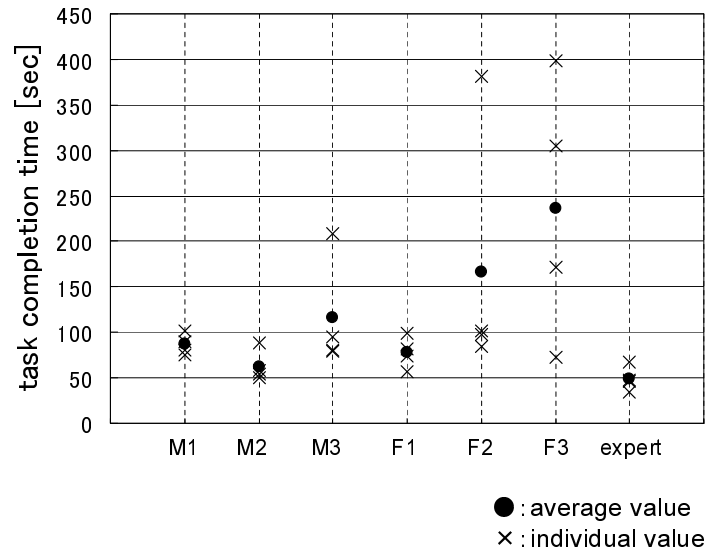


図 27 タスク達成時間（連携サービスの作成）

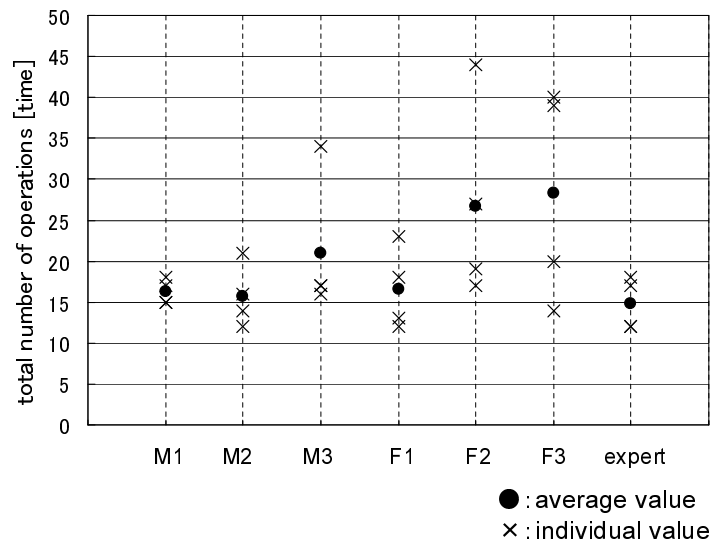


図 28 総操作数（連携サービスの作成）

## 5.3 連携サービスの編集に関する結果

既存の連携サービスを編集するというタスクは、全被験者が達成することはできなかった。タスクを失敗した被験者が F1 に 1 名、F3 に 2 名いたため、操作エラー率、タスク達成時間、総操作数の平均値の算出には、タスクを達成させた被験者の結果のみを用いた。このタスクによって得られた有効性と効率性に関する実験結果を以下に示す。

### 5.3.1 有効性

タスク達成率の結果を図 29 に示す。縦軸はタスク達成率、横軸は被験者のグループを示している。タスク達成率の最低値は 0%、最高値は 100%であった。各グループのタスク達成率の平均は、M1 が約 88%、M2 が 100%、M3 が約 94%、F1 が 75%、F2 が 100%、F3 が約 31%、熟練者が 100%であった。タスクを失敗した被験者が F1 に 1 名、F3 に 2 名いたため、F1 と F2 のタスク達成率の平均は、他のグループに比べると低くなった。その 3 名以外はタスクを達成することができたが、減点条件に該当する被験者が、M1 と F3 に 1 名ずついた。

操作エラー率の結果を図 30 に示す。縦軸は操作エラー率、横軸は被験者のグループを示している。この結果はタスクを達成できた被験者の結果のみを示している。操作エラー率の最低値は 0%、最高値は約 24%であった。各グループの操作エラー率の平均は、M1 が約 10%、M2 が約 6%、M3 が約 9%、F1 が約 12%、F2 が約 7%、F3 が約 18%、熟練者が約 1%であった。他のタスクの操作エラー率に比べると、このタスクの操作エラー率は最も悪い結果となった。

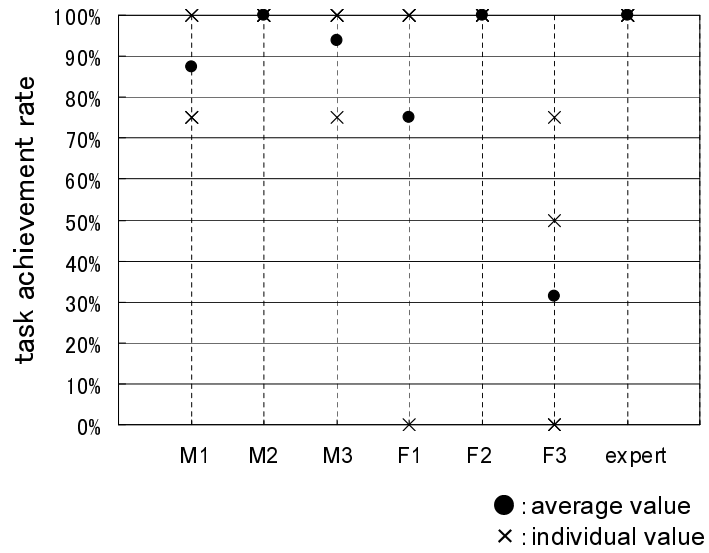


図 29 タスク達成率（連携サービスの編集）

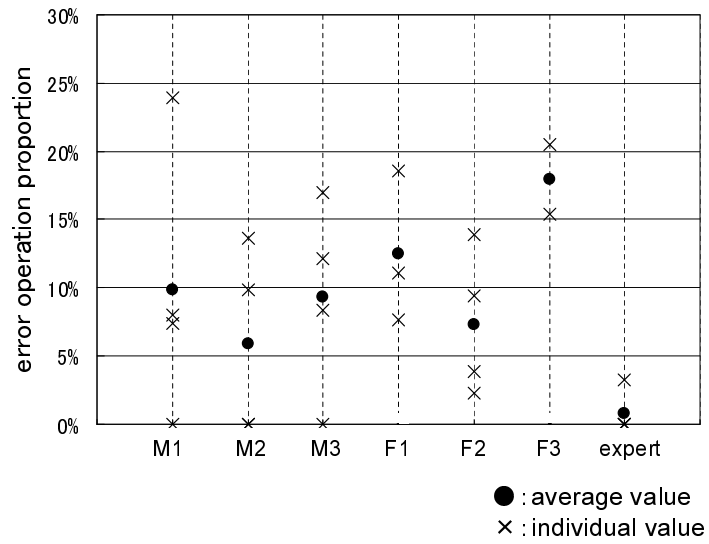


図 30 操作エラー率（連携サービスの編集）

### 5.3.2 効率性

タスク達成時間の結果を図 31 に示す。縦軸はタスク達成時間、横軸は被験者のグループを示している。この結果はタスクを達成できた被験者の結果のみを示している。タスク達成時間の最短値は 46 秒、最長値は 326 秒となった。各グループのタスク達成時間の平均は、M1 が約 155 秒、M2 が約 125 秒、M3 が 192 秒、F1 が約 102 秒、F2 が約 168 秒、F3 が約 267 秒、熟練者が約 64 秒であった。よって、F3 は他のグループに比べると、タスクの達成までに時間がかかっていることがわかる。

総操作数の結果を図 32 に示す。縦軸は総操作数、横軸は被験者のグループを示している。この結果はタスクを達成できた被験者の結果のみを示している。総操作数の最低値は 22 回、最大値は 71 回となった。各グループの総操作数の平均は、M1 が約 30 回、M2 が約 50 回、M3 が 38 回、F1 が 32 回、F2 が約 35 回、F3 が約 46 回、熟練者が約 25 回であった。

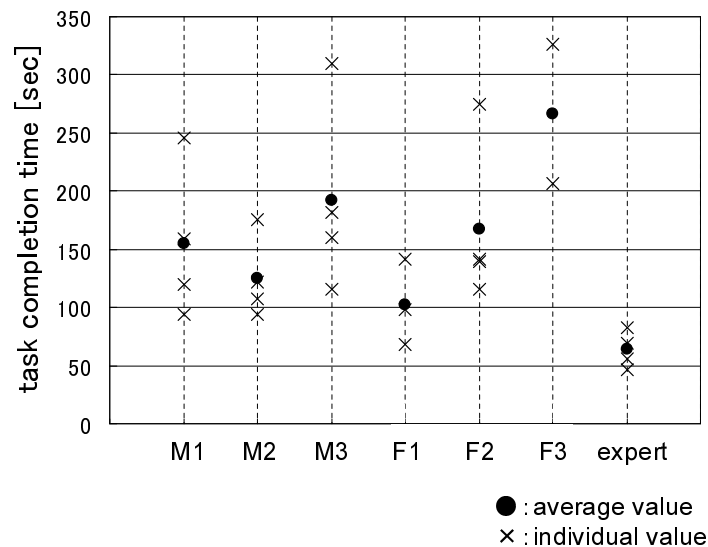


図 31 タスク達成時間（連携サービスの編集）

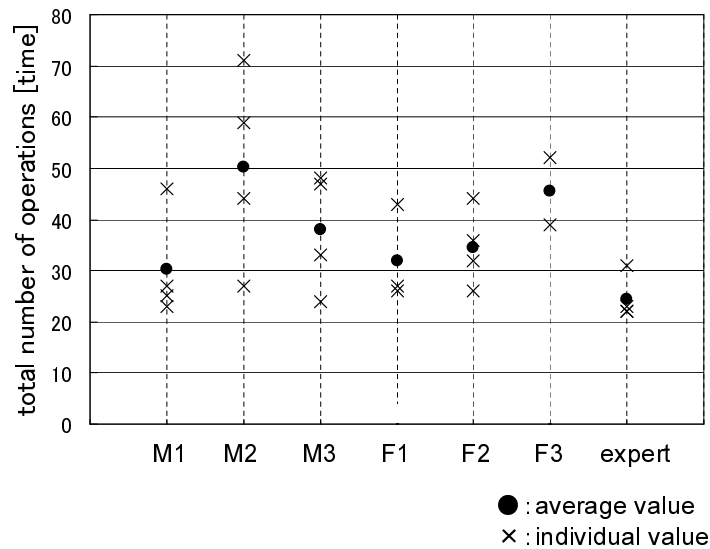


図 32 総操作数（連携サービスの編集）

## 5.4 連携サービスの削除に関する結果

既存の連携サービスを削除するというタスクは、全被験者が達成することができた。このタスクによって得られた有効性と効率性に関する実験結果を以下に示す。

### 5.4.1 有効性

タスク達成率の結果を図 33 に示す。縦軸はタスク達成率、横軸は被験者のグループを示している。タスク達成率の最低値は75%、最高値は100%であった。各グループのタスク達成率の平均は、M1 が100%、M2 が100%、M3 が約94%、F1 が100%、F2 が100%、F3 が100%、熟練者が100%であった。全被験者がタスクを達成することはできたが、減点条件に該当する被験者がM3に1名いた。

操作エラー率の結果を図 34 に示す。縦軸は操作エラー率、横軸は被験者のグループを示している。操作エラー率の最低値は0%、最高値は約17%であった。各グループの操作エラー率の平均は、M1 が0%、M2 が0%、M3 が約2%、F1 が0%、

F2 が 0% , F3 が約 6% , 熟練者が 0%であった . 最も悪い操作エラー率の平均値は , F3 の約 6% であるが , 他のタスクの結果に比べると低い値である .

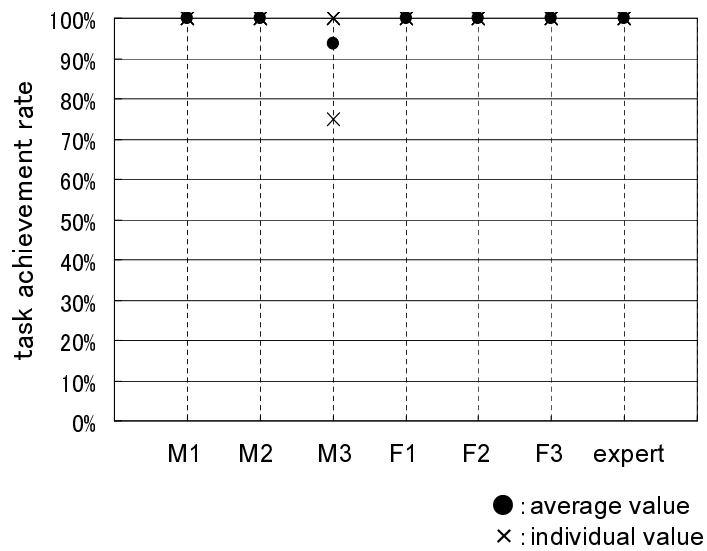


図 33 タスク達成率 ( 連携サービスの削除 )

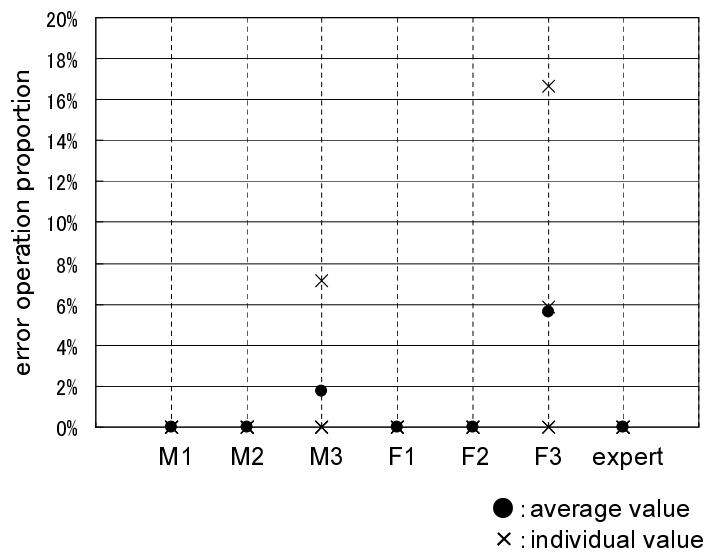


図 34 操作エラー率（連携サービスの削除）

#### 5.4.2 効率性

タスク達成時間の結果を図 35 に示す．縦軸はタスク達成時間，横軸は被験者のグループを示している．タスク達成時間の最短値は 7 秒，最長値は 94 秒であった．各グループのタスク達成時間の平均は，M1 が約 27 秒，M2 が約 15 秒，M3 が 30 秒，F1 が約 18 秒，F2 が約 24 秒，F3 が約 52 秒，熟練者が約 13 秒であった．よって，F3 は他のグループに比べると，タスクの達成までに時間がかかっていることがわかる．

総操作数の結果を図 36 に示す．縦軸は総操作数，横軸は被験者のグループを示している．総操作数の最低値は 4 回，最高値は 17 回となった．各グループの総操作数の平均は，M1 が 9 回，M2 が約 8 回，M3 が約 11 回，F1 が約 10 回，F2 が約 8 回，F3 が約 11 回，熟練者が約 10 回であった．



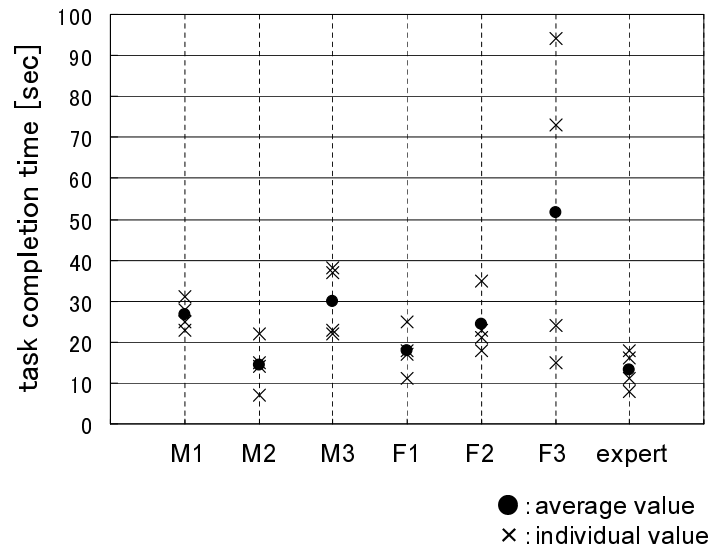


図 35 タスク達成時間（連携サービスの削除）

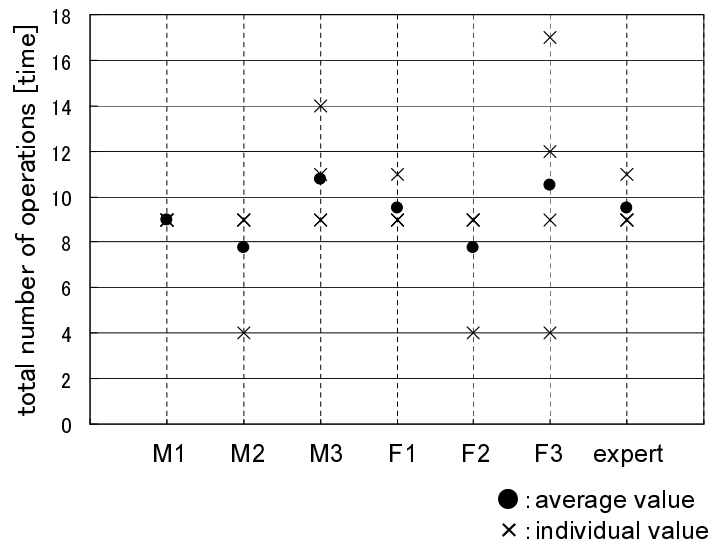


図 36 総操作数（連携サービスの削除）

## 5.5 満足度

熟練者はBANBEEを使い慣れているため、満足度の測定は行わなかった。よって、エンドユーザのみ満足度の測定を行った。SUS質問紙の回答結果を図37に示す。縦軸はSUSスコア、横軸は被験者のグループを示している。SUSスコアの最低点は15点、最高点は約88点であった。各グループのSUSスコアの平均は、M1が約81点、M2が約68点、M3が約71点、F1が約81点、F2が約76点、F3が約41点であった。よって、F3のSUSスコアの平均は、他のグループに比べて低い値であった。

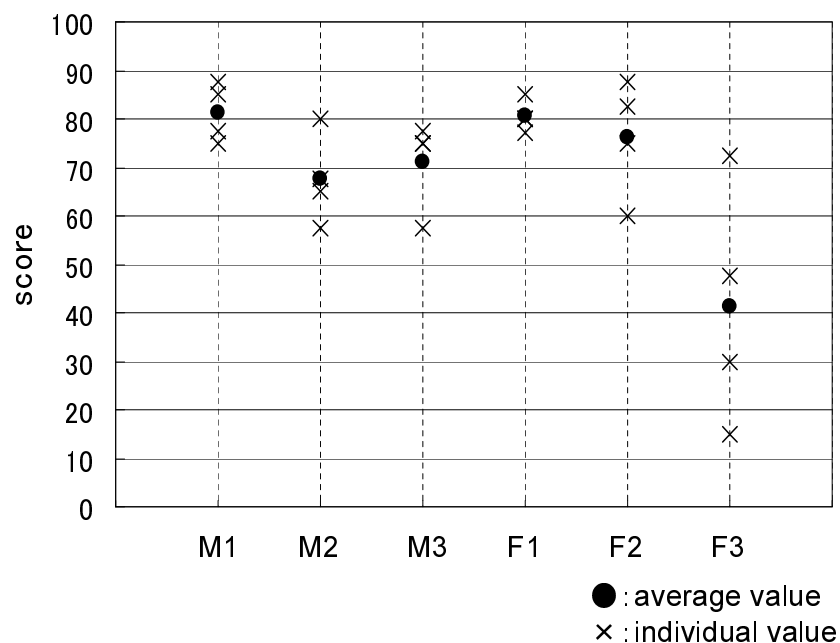


図 37 SUS スコア

自由記述アンケートの回答結果を以下に示す。

#### システムの良かった点

- タッチパネルは操作方法を学ぶ手間が減ると感じた。
- 音声案内があるとマニュアルを覚えなくてもよいので便利だった。
- イラスト付きのアイコンがわかりやすく、また、大きくて見やすかった。
- 必要な操作をまとめて入力できる自動挿入機能が便利だと思った。
- 普段から使用する家電の操作を連携サービスに組み込めるのが良かった。

#### システムの悪かった点

- システムを使う前にマニュアルを読む必要があると感じた。
- 作成した連携サービスファイルに名前が付けられない。
- 音声案内機能は ON と OFF の設定ができるほうが良い。
- 操作フロー編集機能の「削除」「初期化」の実行確認が無い。
- 操作フロー編集機能の「削除」では、削除する対象が選べない。
- 操作フロー編集機能の「入れ替え」「移動」がわかりにくい。

## その他の意見

- 連携サービスの動作テストを行った後、テストを行う前の状態に戻す機能が欲しかった。
- 行った編集操作を取り消すアンドゥ機能が欲しい。
- 操作フローのアイコンに触れることで、その操作の編集が行えるようにしてほしい。
- システムに詳しい人が予め連携サービスを作っておけば、病人などの寝たきりの人や身体障害者の人には有効だと思う。
- 高齢者や機会が苦手な人など、先入観を持っている人は毛嫌いすると思う。
- このようなシステムが欲しいと持っている人には有用であると思う。逆に、使いたいと思わない人は、いくら便利で学習しやすくても使わないと思う。
- 始めは使いこなせなくても、何度か使えば使えるようになると思う。
- マニュアルを読まずに使えるシステムになると良いと思う。
- 家電の操作数が10個以上の連携サービスだと、実用性が増すと思う。

## 6. 考察

### 6.1 連携サービスの作成に関する考察

全被験者が連携サービスを作成するタスクを達成することができたが、そのうち2名はタスクを進めるための助言を必要とした。しかし、その2名も連携サービスを作成するという作業は自力で達成しており、助言が必要だったのは、作成した連携サービスの動作テストを行うという作業に関してであった。よって、連携サービス作成機能の問題点は無かったと考えるが、連携サービスの動作テスト機能については、改善の必要があるといえる。

BAMBEE で連携サービスの動作テストを行う場合は、サービスリストの数字ボタンを押していき、対象の連携サービスが見つかるまで探す必要がある。これは、数字ボタンを押さないと連携サービスの内容がわからないためである。これでは時間もかかり、操作数も増えるので、連携サービスの動作テスト機能の有効性と効率性を下げる原因となっている。

この問題点を解決するための方法として、連携サービスに名前を付けるという方法が、自由記述アンケートによって得られた。連携サービスに名前が付いていれば、その名前から内容をある程度予想することができるということである。

この解決方法を実現するとすれば、作成した連携サービスに名前を付ける機能を実装する必要がある。考えられる実装案としては、GUIに仮想キーボードを表示させるという方法があり、それならばタッチパネルでも操作することができる。この機能を実装した場合、作成した連携サービスの利用に関する有効性と効率性は向上する。しかし、連携サービスの作成には名前を付けるという作業が加わるため、連携サービス作成機能の有効性と効率性は下がる、そこで、「作成」と「利用」のどちらが重要かを考える必要がある。後述の6.2節と6.3節でも述べるが、作成した連携サービスに名前が付けられるようになれば、連携サービスの編集・削除機能の有効性と効率性が向上すると考える。よって、「作成」よりも「利用」を重視し、作成した連携サービスに名前を付ける機能を実装することで、BAMBEE全体の有効性と効率性は、向上すると考える。

## 6.2 連携サービスの編集に関する考察

連携サービスを編集するタスクを失敗した被験者は3名いたが、その3名も連携サービスを編集するという作業は自力で達成しており、失敗したのは、編集した連携サービスの動作テストを行うという作業であった。よって、実質は全被験者が連携サービスを編集することができたといえる。

しかし、実験中に考え込む被験者が多かったことから、このタスクは他のタスクに比べると、難易度の高いタスクであったといえる。難易度が高くなった原因としては、家電の操作の順番を入れ替えるという作業があったことだと考える。この作業では、操作フロー編集機能の「移動」もしくは「入れ替え」を使う必要があるが、これらの機能の使い方を被験者がよく理解できていなかったことが、実験中の様子からわかった。また、自由記述アンケートでも、これらの機能が理解しづらいという意見が出ていた。よって、被験者がこの2つの操作フロー編集機能の使い方を正しく理解することができれば、今回、タスクに失敗した被験者でも、連携サービスの編集を行うことができると考える。実際に、これらの操作フロー編集機能について熟知している熟練者のタスク達成率は100%であったので、エンドユーザもシステムに慣れればその域に達すると考えられる。とはいえ、システムに慣れれば扱えるようになるということは当然であるので、操作フロー編集機能の「移動」と「入れ替え」に関しては、より理解しやすいようにGUIを改善する必要があるといえる。

実験で見つかった連携サービス編集機能の更なる問題点としては、6.1節で述べた問題点と同じで、編集する連携サービスを選ぶ時に、サービスリストの数字ボタンを押さないと、連携サービスの内容がわからないということである。これでは有効性と効率性が悪い。しかし、連携サービスファイルに名前が付けられるようになることで、編集する連携サービスを探す時間と操作数が減少し、連携サービス編集機能の有効性と効率性が向上すると考える。

### 6.3 連携サービスの削除に関する考察

1名を除いた被験者全員のタスク達成率が100%であったことを考えると、最も難易度の低いタスクであったといえる。評価実験で見つかった連携サービス削除機能の問題点としては、6.1節で述べた問題点と同じで、削除する連携サービスを選ぶ時に、サービスリストの数字ボタンを押さないと、連携サービスの内容がわからないということである。これでは有効性と効率性が悪い。しかし、作成した連携サービスに名前が付けられるようになることで、削除する連携サービスを探す時間と操作数が減少し、連携サービス削除機能の有効性と効率性が向上すると考える。

### 6.4 満足度に関する考察

有効性と効率性に関する実験結果で最も悪い結果が出たグループはF3であり、SUSスコアに関してもF3が最も悪かった。よって、BAMBEEを上手く使えなかったことが原因で、F3のSUSスコアが低くなったと考えられる。また、F3の被験者から自由記述アンケートで「連携サービスを使いたいと思えるエンドユーザには有用なシステムであるが、使いたいと思わない人はいくら便利でも使わないと思う」という意見が挙がった。確かに、連携サービスのような新しい操作方を学習する意思の無い高齢者などにとっては、BAMBEEは必要の無いものだといえ、満足度も低くなる。しかし、そういったエンドユーザの中にも、何度かBAMBEEを利用することで使い方をより理解し、連携サービスを使ってみようという考えになる人も出ると考える。

## 7. まとめ

本論文では、まず、2章で既存の連携サービス作成法を述べた。そして、プログラミングの知識が無いエンドユーザは、連携サービスを作成することができないという問題点を挙げた。次に、3章でその問題点を解決するためのシステムであるBAMBEEを提案した。次に、実装したBAMBEEを利用することで、エンドユーザが連携サービスの作成・編集・削除を行えるかどうかを確認するため、4章に示したユーザビリティテストを実施した。その結果、5章に示すように、エンドユーザが連携サービスの作成・編集・削除を行えたことを確認できた。



## 謝辞

本研究を進めるにあたり、多くの方々に御指導と御協力を頂きました。ここにお世話になった方々への感謝の意を表したいと思います。

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、本研究の主指導教官を担当して頂き、また、研究に限らず研究室内での活動を支えて頂きました。そして何より、入学してから半年以上経っていたにもかかわらず、快く研究室に受け入れてくれたことに感謝いたします。ありがとうございました。

奈良先端科学技術大学院大学 情報科学研究科 伊藤 実 教授 には、本研究の副指導教官を担当して頂き、本研究の発表において、御意見や御指摘を頂きました。ありがとうございました。

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、本研究の副指導教官を担当して頂き、発表練習では御意見や御指摘を頂きました。また、研究に限らず研究室内での活動も支えて頂きました。ありがとうございました。

神戸大学 大学院工学研究科 中村 匡秀 准教授には、本研究の副指導教官を担当して頂き、また、研究活動では最もお世話になりました。中村先生のおかげで、私たち Web サービス班のメンバーは、ここまで辿り着くことができました。神戸大学の学生への指導だけでもお忙しいのに、私たちへの指導を続けて下さったことには、心から感謝しています。ありがとうございました。

神戸大学 大学院工学研究科 井垣 宏 特命助教には、中村先生と共に本研究に関する指導をして頂きました。私たち Web サービス班のメンバーがここまで辿り着くことができたのは、井垣先生の御指導のおかげでもあります。ありがとうございました。

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、研究室内での活動を支えていただきました。ありがとうございました。

ソフトウェア工学講座の皆様には、本研究について多くの御協力を頂きました。特に亀井 靖高 様には論文の内容に関する御指摘を頂き、感謝しております。皆様には研究活動以外の様々な場面でも御助力して頂き、充実した時間を過ごすことができました。ありがとうございました。

インターネットアーキテクチャ講座の皆様には、約半年間お世話になりました。

短い間でしたが，非常に充実した時間が過ごせ，人間的に成長することができました．皆様との出会いが無ければ，今の自分はありません．ありがとうございました．

以下の方々には実験に御協力して頂きました．お忙しい中，ありがとうございました．高井 亮太 様，横田 直彦 様，高田 純 様，田村 晃一 様，伊原 彰紀 様，津ざき 憲治 様，筒井 昭博 様，田中 俊之 様，荒瀬 大 様，伊原 豊實 様，溝川 牧雄 様，井立 廣美 様，掘田 敏雄 様，呂 悠妃 様，乾 純子 様，前田 くれあ 様，永田 裕樹子 様，山尾 法子 様，浜田 真理 様，橋本 洋子 様，金岡 恵 様，加藤 晴子 様，渡邊 之子 様，松田 美智代 様，伊原 とも江 様，三井 康平 様，大西 洋司 様，榊原 弘記 様，福岡 佑介 様（順不同）

そして最後に両親へ，心からの「ありがとう」を．

## 参考文献

- [1] 松下電工 ライフィニティ, <http://biz.national.jp/Ebox/kahs/>
- [2] 東芝ネットワーク家電 フェミニティ, <http://feminity.toshiba.co.jp/feminity/>
- [3] Digital Living Network Alliance, <http://www.dlna.org/jp/industry/>
- [4] ECHONET CONSORTIUM, <http://www.echonet.gr.jp/>
- [5] 筒井 章博, 藤井 伸朗, 川村 龍太郎, 依田 育生, “5. 次世代ホームネットワーク技術 (<小特集>次世代ネットワーク技術の標準化動向)”, 電子情報通信学会誌, Vol.89, No.12, pp.1067-1072, Dec. 2006.
- [6] 井垣 宏, 中村 匡秀, 玉田 春昭, 松本 健一, “サービス指向アーキテクチャを用いたネットワーク家電連携サービスの開発”, 情報処理学会論文誌, Vol.46, No.2, pp.314-326, February 2005.
- [7] 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.
- [8] スギヤマエレクトロニクス クロッサム 2+USB, <http://www.sugi-ele.co.jp/top.htm>
- [9] W3C Web Service Activity, <http://www.w3.org/2002/ws/>
- [10] XML-RPC Home Page, <http://www.xmlrpc.com/>
- [11] W3C XML Protocol Working Group, <http://www.w3.org/2000/xmlrpc/Group/>
- [12] W3C Web Services Description Working Group, <http://www.w3.org/2002/ws/desc/>
- [13] 三菱電機エンジニアリング株式会社 超音波表面弾性波方式タッチパネルモニター TSD-T124-M, <http://touch-monitor.mee.co.jp/t124m.html>

- [14] SOAP::Lite for Perl, <http://www.soaplite.com/>
- [15] 宮澤 永光, 亀井 昭宏, “マーケティング辞典”, 同文館出版, 2003
- [16] ISO 9241-11:1998, Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability
- [17] 黒須 正明, “ユーザビリティテスト ユーザ中心のものづくりに向けて”, 井立出版, 2003
- [18] Brooke, J., “SUS: A quick and dirty usability scale”, Usability evaluation in industry, London: Taylor& Francis, pp.189-194, 1996

## 付録

### A. BAMBEE 操作マニュアル

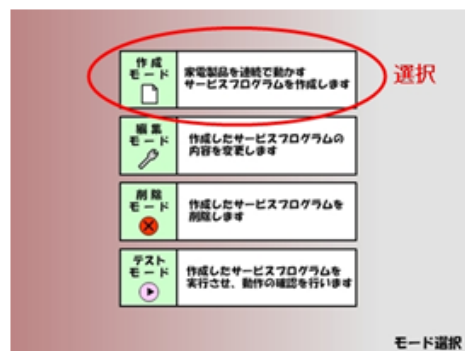
# 操作マニュアル

## 目次

1. 連携サービスの作成 .....	3
2. 各種編集機能 .....	
2.1. 削除機能 .....	6
2.2. 初期化機能 .....	6
2.3. 入れ替え機能 .....	7
2.4. 移動機能 .....	8
2.5. 自動挿入機能（強制挿入） .....	9
2.6. 自動挿入機能（選択挿入） .....	10
3. 連携サービスの編集 .....	11
4. 連携サービスの削除 .....	14
5. 連携サービスの実行 .....	16

## 1. 連携サービスの作成

この章では連携サービスの作成について説明します。

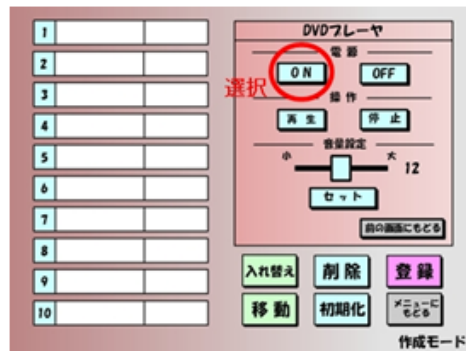


まずはメニュー画面から「作成モード」を選んでください。

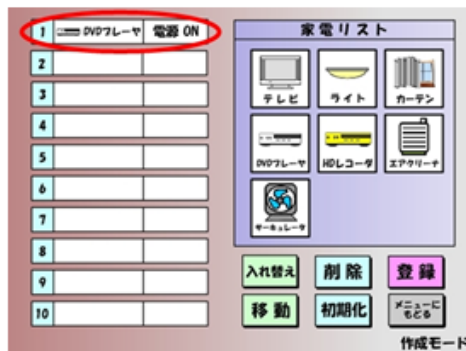


そうするとこのような画面になります。右側の赤丸で囲まれた家電リストから、操作させたい家電を選んでください。





家電を選択すると、次はこのように家電の動作を選択する画面になります。  
例として、ここでは DVD プレーヤの「電源 ON」を選択します。



そうすると、先ほど選んだ「DVDプレーヤの電源をONする」という操作が  
左側に表示されます。

以上のようにして、他の家電の操作も同じように追加していきます。



必要な操作を全て追加し終わった後は、最後に「登録ボタン」を押します。  
 それにより、連携サービスの作成が完了します。  
 そしてこれ以降、作成した連携サービスの実行・編集・削除が出来るようになります。

※保存しておける連携サービスの数は最大で12個です。それを超えないように  
 不要な連携サービスは削除してください。

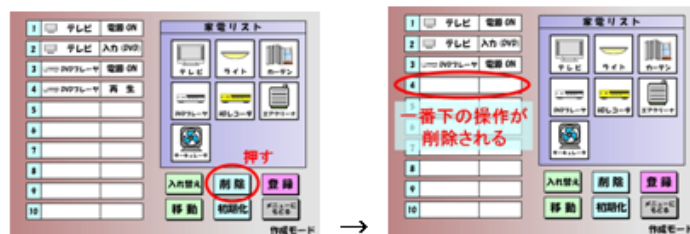
※「入れ替え」「移動」「削除」「初期化」などの各種編集機能については  
 2章をご覧ください。

## 2. 各種編集機能

### 2.1. 削除機能

最後に追加した家電の操作を削除する機能です。

※どの操作を削除するかは選べません。



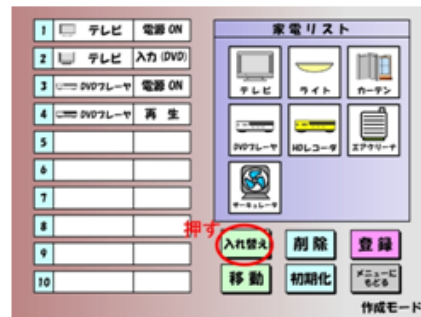
### 2.2. 初期化機能

追加した家電の操作を全て削除（初期化）する機能です。

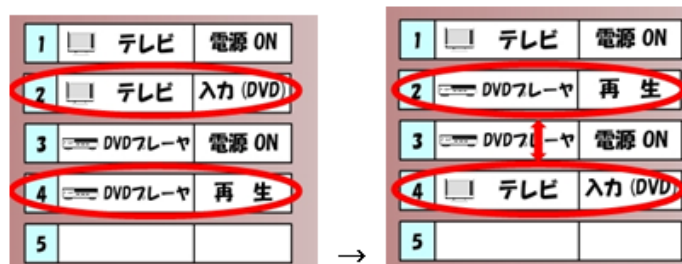


### 2. 3. 入れ替え機能

選択した2つの操作の順番を入れ替える機能です。



はじめに「入れ替えボタン」を押します。

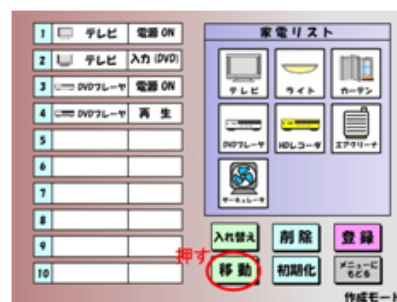


「入れ替えボタン」を押した後、図のように2つの操作を選ぶことで、順番が入れ替わります。

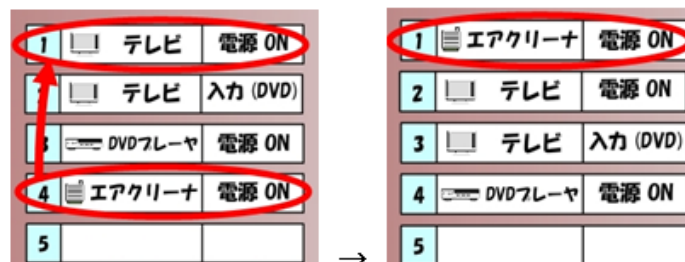
#### 2. 4. 移動機能

選択した操作の順番を移動させる機能です。

入れ替え機能と違い、他の操作の順番も繰り上がったり繰り下がったりします。



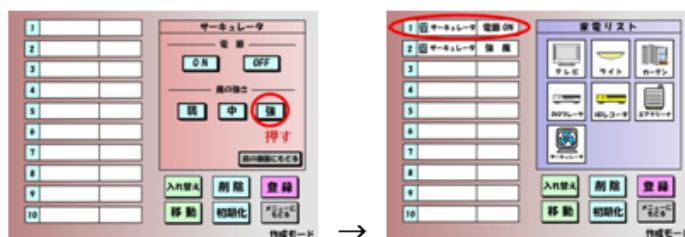
はじめに「移動ボタン」を押します。



「移動ボタン」を押した後、まずは移動させたい操作を選び、次に移動先の場所を選ぶことで、選ばれた操作が移動します。

## 2. 5. 自動挿入機能 (強制挿入)

選んだ操作に必要な操作が足りなかった場合、必要な操作を強制的に挿入する機能です。



例えば図のように、サーキュレータの電源を ON にせずに風量を強風にしても、強制的に「電源ON」の操作が挿入されます。

## 2. 6. 自動挿入機能（選択挿入）

選んだ操作に必要な操作が足りなかった場合、必要な操作を挿入するかどうかを選べる機能です。

例えば、DVD プレーヤを再生する場合には

- ・テレビの電源を ON にする
- ・テレビの入力を「DVD」にする
- ・DVD プレーヤの電源を ON にする（最低限必要な操作）

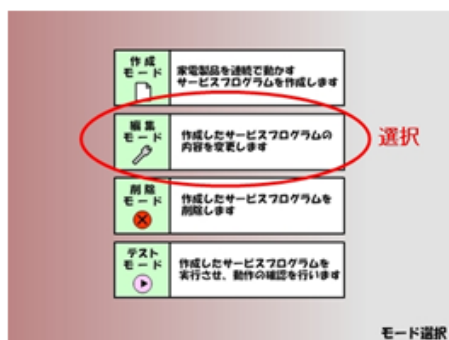
という3つの操作が必要です。上の2つの操作は無くても問題ありませんが、その2つの操作がないと、DVD が再生されてもテレビに映像は映されません。

必要な操作が足りないまま DVD プレーヤの「再生ボタン」を押すと、以下の図のような選択窓が現れます。そして、「はい」を選ぶと必要な操作が全て挿入され、「いいえ」を選ぶと最低限必要な操作のみが挿入されます。



### 3. 連携サービスの編集

この章では連携サービスの編集について説明します。



まずはメニュー画面から「編集モード」を選んでください。



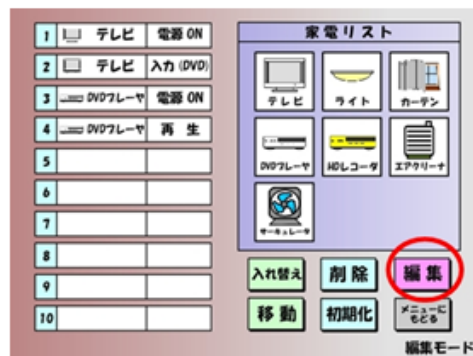


そうするとこのような画面になります。右側のサービスリストに現在保存されている連携サービスの番号が表示されます。この図で言えば、10個の連携サービスが保存されているということになります。

※数字の値が大きいほど作成した日時が新しいようになっています。  
 (1番の連携サービスが最も昔に作成され、12番の連携サービスが最も新しい)

ここで、例えば「7番」のボタンを押すと、左側にその番号の連携サービスの内容が表示されます。このようにして、番号のボタンを押して連携サービスの内容を確認しつつ、編集したい連携サービスを探します。

編集したい連携サービスが見つかったら、「選択ボタン」を押してください。

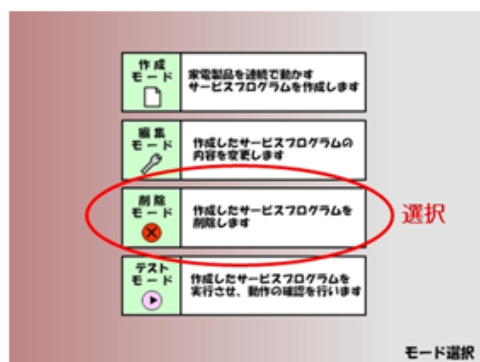


編集したい連携サービスを選ぶと、このような画面になります。  
 あとは作成のときと同じようにして、家電の操作を追加したり削除したりして  
 ください。最後に「編集ボタン」を押すと編集が完了します。

※編集を中断したいときは、右下の「メニューにもどるボタン」を押して  
 ください。

## 4. 連携サービスの削除

この章では連携サービスの削除について説明します。



まずはメニュー画面から「削除モード」を選んでください。



そうするとこのような画面になります。右側のサービスリストに現在保存されている連携サービスの番号が表示されます。この図で言えば、10個の連携サービスが保存されているということになります。

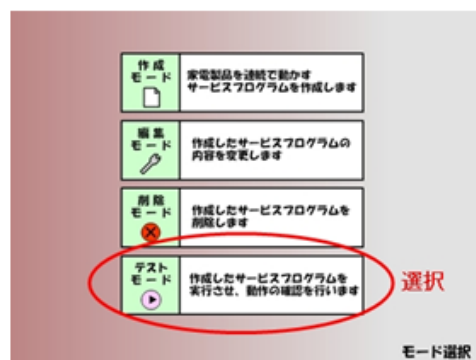
※数字の値が大きいほど作成した日時が新しいようになっています。  
 (1番の連携サービスが最も昔に作成され、12番の連携サービスが最も新しい)

ここで、例えば「7番」のボタンを押すと、左側にその番号の連携サービスの内容が表示されます。このようにして、右側にある番号のボタンを押して連携サービスの内容を確認しつつ、削除したい連携サービスを探します。

削除したい連携サービスが見つかったら、「削除ボタン」を押してください。  
 それにより、連携サービスの削除が完了します。

## 5. 連携サービスの実行

この章では連携サービスの実行について説明します。



まずはメニュー画面から「テストモード」を選んでください。



そうするとこのような画面になります。右側のサービスリストに現在保存されている連携サービスの番号が表示されます。この図で言えば、10個の連携サービスが保存されているということになります。

※数字の値が大きいほど作成した日時が新しいようになっています。  
 (1番の連携サービスが最も昔に作成され、12番の連携サービスが最も新しい)

ここで、例えば「7番」のボタンを押すと、左側にその番号の連携サービスの内容が表示されます。このようにして、右側にある番号のボタンを押して連携サービスの内容を確認しつつ、実行したい連携サービスを探します。

実行したい連携サービスが見つかったら、「実行ボタン」を押してください。その連携サービスが実行されます。

※実行確認を終えてメニューに戻りたい場合は、右下の「メニューにもどるボタン」を押してください。