

ホームネットワークシステムにおけるパーソナルリモコン作成実験

徳田 啓介[†] 稲田 卓也[†] 松本 真佑[†] 中村 匡秀[†]

[†] 神戸大学 〒657-8501 神戸市灘区六甲台町 1-1

E-mail: [†]{tokuda,inada}@ws.cs.kobe-u.ac.jp, ^{††}{shinsuke,masa-n}@cs.kobe-u.ac.jp

あらまし 我々はホームネットワークシステムにおいて、ユーザの嗜好に適応したパーソナルリモコンを実現可能な実行プラットフォームを開発している。本稿では、先行研究において開発したパーソナルリモコンのレイアウト作成支援ツール、Personal Remocon Creator (PRCreator) を拡張し、アクションの作成支援機能を追加する。この拡張により、パーソナルリモコン作成に係る全ての過程を、専門知識を持たないエンドユーザでも容易に達成できると考える。評価実験として、あらかじめ指定された家電操作要求を満たすリモコンを自由に作成してもらい、インタビューを行う。その結果に基づき、個人でリモコンを作成するというパーソナルリモコンの概念そのものに対する評価、PRCreator の使用感に対する評価、及び、リモコン定義言語の記述能力や応答速度といった実行プラットフォームの評価の3つの観点から提案手法に対する考察を行う。

キーワード パーソナルリモコン, ヒューマンインタフェース, パーソナルリモコン実行プラットフォーム, GUI

Experiment of Creation for Personal Remote Controller in Home Network System

Keisuke TOKUDA[†], Takuya INADA[†], Shinsuke MATSUMOTO[†], and Masahide NAKAMURA[†]

[†] Kobe University Rokkoudaityou 1-1, Nada-ku, Kobe, Hyogo, 657-8501 Japan

E-mail: [†]{tokuda,inada}@ws.cs.kobe-u.ac.jp, ^{††}{shinsuke,masa-n}@cs.kobe-u.ac.jp

Abstract We have been studying an execution platform of the personal remote controller for the home network system (HNS). Our previous work have also implemented “Personal Remocon Creator” which supports layout creation of the personal remote controller. This paper extends the layout creation tool to support action definition. This extension allows end-users to easily create their personalized controllers more intuitively without specialized knowledge and skills. We also conduct an experimental evaluation, where the subjects create own controllers that have some specified functions for operating home appliances. Finally we discuss from the following three perspectives; the concept of personalized remote controller, the usability of controller creation tool, the performance and the capability of controller execution platform.

Key words Personal remote controller, human interface, personal remote controller platform, GUI

1. はじめに

我々は先行研究 [1] において、個人の嗜好や宅内環境に合わせたインタフェースで家電機器を操作できる個人適応型リモコンパーソナルリモコンを提案し、パーソナルリモコンの実行プラットフォームを開発した。このプラットフォームは、エンドユーザが作成したリモコンの定義情報を読み取り、携帯端末上にパーソナルリモコンを実現する。リモコンの定義情報は、ボタン配置や背景色などのレイアウトを定義する言語 PRL (Personal Remocon Layout Language) と、ボタン押下時の家電操作アクションを定義する言語 PRAL (Personal Remocon

Action Language) の2つの言語で記述される。エンドユーザは PRL と PRAL をテキストエディタなどを用いて記述することで、自身の好みに応じたパーソナルリモコンを自由に作成、カスタマイズできる。

先行研究 [2] では、PRL で記述されるレイアウト情報の作成支援ツール Personal Remocon Creator (PRCreator) を提案した。本ツールを利用することで、PRL を直接記述することなく、GUI による直感的な操作でレイアウトを作成することができる。また PRCreator は作成したレイアウト情報の共有機能を持っており、別のユーザが作成したレイアウトを参考にカスタマイズできる機能を持つ。

現在パーソナルリモコンには、以下に示す 2 つの課題が残されている。まず、先行研究 [2] で開発した PRCreator はリモコンレイアウトの作成支援、すなわち PRL の作成支援に留まっており、ボタン押下時のアクション情報を記述する PRAL の作成支援が必須課題の一つである。この課題を達成することで、パーソナルリモコン作成に係る全ての過程を、専門知識を持たないエンドユーザでも容易に達成できると考える。最後の課題は、提案実行プラットフォームと作成支援ツールを含めた、パーソナルリモコン作成の全過程の評価実験の実施である。

本稿では、PRAL の作成を支援するために PRCreator の拡張を行い、この支援ツールを用いた評価実験を行う。実験では 10 名の被験者を対象に、あらかじめ指定された家電操作要求を満たすリモコンを自由に作成してもらう。評価項目は、リモコン作成に要した工数と作成されたりリモコンに対する満足度、及び実験後に実施したインタビューである。これらの評価項目に基づき、個人でリモコンを作成するというパーソナルリモコンの概念そのものに対する評価、PRCreator の使用感に対する評価、及びリモコン定義言語の記述能力や応答速度といった実行プラットフォームの評価の 3 つの観点から提案手法に対する考察を行う。

2. 準備

2.1 ホームネットワークシステム

ホームネットワークシステム (HNS) とは、宅内の家電やセンサをネットワークに接続することによって、ネットワークから計算機による家電機器制御を可能とするシステムである。HNS を利用することで、宅外からの家電モニタリングや遠隔操作、複数家電の連携制御等の付加価値サービスを実行できる。

我々の研究グループは、実際の家電機器を用いた HNS 環境、CS27-HNS を構築している。CS27-HNS では、様々な種類の家電をネットワーク越しに標準的な方法で操作するため、各家電の機能を Web サービス [3] として公開している。Web サービスとして公開された家電は、単純な URI へのアクセスにより操作することができる。例えば、TV の電源をつけるためには <http://cs27-hns/TVservice/on> にアクセスし、TV のチャンネルを 8 に変更するには <http://cs27-hns/TVservice/channel¶m0=8> にアクセスするだけで家電を操作できる。

2.2 パーソナルリモコン実行プラットフォーム

我々は先行研究 [1] において、ユーザが自由にカスタマイズ可能なリモコン、パーソナルリモコンの実行プラットフォームを構築した。このプラットフォームを利用することで、エンドユーザは自身の嗜好や宅内環境に適応したインタフェースを利用して、宅内の家電操作を行うことができる。

パーソナルリモコンの作成と利用の流れ、及び実行プラットフォームのアーキテクチャを図 1 に示す。エンドユーザは、まずテキストエディタあるいは支援ツールを用いて、後述するリモコンの定義情報が記述されたファイルを作成する。実行プラットフォームは作成された定義ファイルを読み取り、XML Parser を介してリモコン定義情報を格納するデータベースに登

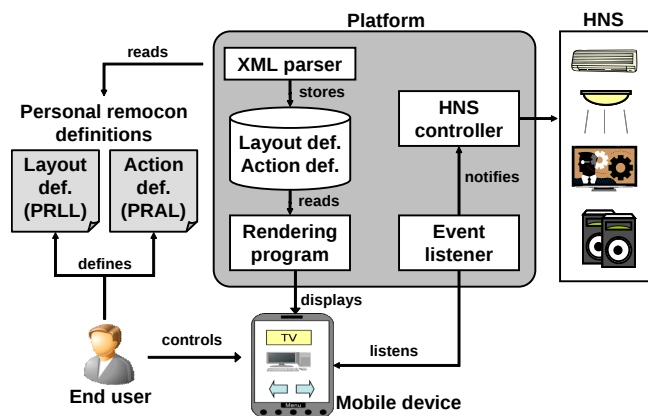


図 1 パーソナルリモコン実行プラットフォームのアーキテクチャ

録する。登録されたりリモコン情報は、Rendering Program により、ユーザの持つ携帯端末上にリモコンの形で描画される。リモコンの操作イベントは Event Listener に通知され、HNS Controller により HNS 内の家電機器操作が実行される。

2.3 パーソナルリモコン定義言語

エンドユーザが定義するリモコン情報は、以下の 2 種類の言語を用いて定義される。

PRL (Personal Remocon Layout Language) : リモコンのレイアウト情報を定義する XML 形式の構造化言語である。1 レイアウトは複数のスクリーンで構成され、1 スクリーンは複数のボタンから構成される。各ボタンには配置や色、画像ファイルといったパラメタが定義される。またボタンには可視 / 不可視パラメタが付与されており、ボタン押下に応じて見た目を変えることが可能である。ボタン押下の際の振る舞いは、次に説明するアクション記述言語により定義される。

PRAL (Personal Remocon Action Language) : PRL で定義された各ボタンの押下時アクションを定義する XML 形式の構造化言語である。定義可能なアクションは、HNS 上の家電操作、スクリーンの遷移、携帯端末上での効果音の再生である。単一家電の操作に限らず、1 ボタンでの複数家電の制御や、画面遷移と効果音の再生といった複数アクションの定義も可能である。

2.4 研究の目的

先行研究 [2] では、PRL の作成支援ツールとして PRCreator の開発を行った。PRCreator を利用することで、エンドユーザは GUI による直感的な操作でレイアウトを作成できるようになった。一方で、アクションを定義する PRAL の作成支援については残された課題となっていた。

本研究では、まず PRCreator を拡張し、PRAL 記述の支援を可能とする機能を追加する。これにより、専門知識を持たないエンドユーザでもパーソナルリモコンの全過程を容易に達成できると考える。さらに提案実行プラットフォーム及び PRCreator を利用したパーソナルリモコン作成全過程の評価実験を行い、提案手法の有効性を確かめる。

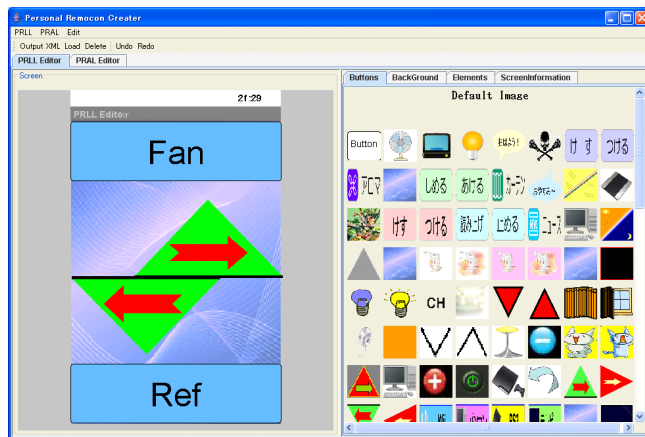


図 2 PRL Editor のレイアウト作成画面

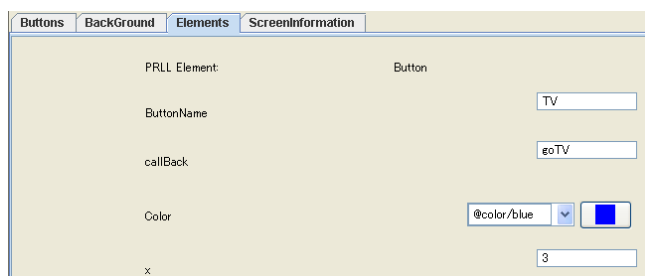


図 3 PRL 要素入力フォーム

3. Personal Remocon Creater

3.1 概要

Personal Remocon Creater (PRCreator) は、パーソナルリモコン作成を支援する GUI アプリケーションである。この GUI アプリケーションのシステムは主に、PRL Editor [2] と PRAL Editor の 2 つの機能から構築される。

3.2 PRL Editor

PRL 記述を支援する機能である。ボタンを視覚的に配置できる、レイアウトの変更を簡単に行える、PRL の要素を簡単に記述できるといった特徴を持つ。以下で PRL Editor の持つ主な機能を説明する。

ドラッグ&ドロップによるボタン配置：レイアウトをドラッグ&ドロップなどの操作で視覚的に作成、変更できる。ユーザは、図 2 の右側に示される画像を、左側のスクリーンに自由に配置することでレイアウトを自由に構築する。利用できる画像に関しては、ユーザが自分で用意した画像とあらかじめ用意された画像の 2 種類が存在する。また、配置した画像の右下をマウスでドラッグすることで画像のサイズを変更できる。

PRL 要素入力フォーム：ユーザは図 3 に示される入力フォームで PRL 要素をコンボボックスや自動記述などで容易に記述できる。設定できる情報は、ボタンの色、テキストと可視/不可視パラメタといったボタンの見た目とボタン押下時のアクションである。アクションの具体的な内容に関しては、次に説明する PRAL Editor によって設定する。

3.3 PRAL Editor

PRAL 記述を支援する機能である。PRAL の要素を容易に記

Type	Call	Comment	Delete
ref	TVOff	テレビを消す	Delete
ref	LightOff	ライトを消す	Delete

Type	Call	Comment	Add
ref	LightOff	ライトを消す	Add

Type	SoundFile	Comment	Clear
sound	goodNight		Clear

Type	Screen	Comment	Clear
forwardAction	TOP		Clear

図 4 PRAL 要素入力フォーム

述できる、一度記述した家電操作 URI を再利用できるといった特徴を持つ。以下で PRAL Editor の持つ主な機能を説明する。

PRAL 要素入力フォーム：ユーザは図 4 に示される入力フォームで PRAL 要素を容易に記述できる。ActionDescription の入力フォームでは複数のアクションを識別するための名前を設定する。Home Appliance Operation の入力フォームで、TV をつける・照明を消灯するといった家電操作を設定する。設定した家電操作の情報は TasksList に追加されていく。Device Action の入力フォームで、効果音や画面遷移を設定する。効果音に関しては、ユーザが自分で用意した音楽ファイルを利用できる。作成したアクションはアクションリストに登録される。

家電操作 URI 参照機能：一度記述した家電操作のための URI を再利用するための機能である。具体的には、一度作成したアクションの名前を記述するだけで、指定したアクションで設定した家電操作を実行できる。例えば、図 4 において、宅内の照明を消灯し、TV の電源を消す “Good Night” アクションを作成している。この場合、既に “TVOff” アクションと “LightOff” アクションが作成されていれば、その名前を記述するだけでアクションの設定ができる。<http://cs27-hns/Lightservice/off> といった照明消灯や TV 電源を切るのための URI を記述する必要がなくなる。一度作成した PRAL ファイルを読み込み既成のアクションとして利用することも可能である。

3.4 実装

Personal Remocon Creater はオープンソースソフトウェアである DroidDraw [4] をベースに機能を拡張し、実装した。利用した DroidDraw は version 18.1 である。開発言語は Java で総コード行数は約 11,000 行、開発工数は 80 人日である。

4. 評価実験

4.1 実験概要

パーソナルリモコンの有用性を確かめるために、リモコン作成の一連の流れを通した評価実験を行う。実験では被験者に PRCreator を利用してもらい、指定された機能要求を満たすリモコンを作成してもらう。リモコン作成に要した工数と作成されたリモコンに対する満足度、及び実験後に実施したインタビューの評価項目に基づき、以下の3つの観点で評価を行う。1. 個人でリモコンの作成を行うというパーソナルリモコンの概念そのものに対する評価、2. PRCreator の使いやすさ、3. リモコン定義言語の記述能力。被験者は学士課程と修士課程の20代男性10名である。

4.2 実験タスク

実施してもらったタスクは、以下の4つである。

レイアウト設計タスク：リモコンレイアウトの画面設計、及び画面に用いる画像素材の作成を行う。リモコンが満たすべき機能要求は以下の通りである。TV に対して、電源 ON / OFF・TV モードと DVD モードへの入力切替・音量操作・チャンネル操作ができること。照明に対して点灯 / 消灯ができること。扇風機に対して、電源の ON / OFF・スイング操作・風量操作ができること。なお、画面設計方法は各被験者の自由とした。

レイアウト作成タスク：レイアウト設計タスクで作成した画面設計に基づき、実際に PRL Editor を利用してレイアウトを作成してもらう。なお、家電操作に関するアクション定義は、あらかじめ用意された PRAL ファイルを利用する。

レイアウト変更タスク：レイアウト作成タスクで作成したリモコンレイアウトに対して、PRL Editor を利用して以下の追加・変更・削除を行う。TV の入力切替を TV モードと DVD モードから、ゲームモードと PC モードへ変更する。操作可能な照明機器の数を追加する。操作可能対象機器から扇風機を削除する。

アクション作成タスク：PRAL Editor を利用して、以下の2つのアクションを定義する。1つ目は、宅内の照明を全て消灯し、TV の電源を切り、カーテンを閉める“Good Night”アクション。2つ目は、宅内の照明を全て点灯し、TV の電源を入れ、カーテンを開ける“Good Morning”アクション。なお、ボタン押下時の効果音再生に関しては、ユーザが自由に設定してよいものとする。レイアウトの設定に関しては、あらかじめ用意された PRL ファイルを利用する。

全てのタスクの実施後に、各被験者に自分で作成したパーソナルリモコンを利用してもらい、実際に家電を操作してもらった。その後インタビューを行い、PRCreator の使用感やリモコン定義言語の記述能力等に関して意見をもらった。各タスクの評価項目は、レイアウトの設計時間とレイアウトの作成時間・変更時間、及びアクションの作成時間である。

4.3 実験環境

実験環境として、本研究室で研究・開発している HNS 環境、CS27-HNS を使用した。パーソナルリモコンの利用端末として、Android 1.6 搭載の Android dev Phone1、Android 2.1 搭載

表 1 レイアウトとアクションの作成に要した時間（単位：分）

被験者	レイアウト			アクション 作成
	設計	作成	変更	
A	57	22	8	23
B	57	5	9	26
C	77	40	24	8
D	57	19	16	33
E	210	12	30	10
F	330	30	30	20
G	132	150	45	19
H	144	137	19	22
I	83	227	24	23
J	114	116	29	35
平均	126	76	23	22

の ZiiO7、及び Android 2.2 搭載の GALAXY Tab を用いた。

4.4 実験結果

実験結果を表 1 に、作成されたレイアウトの一部を図 5 に、実験後にに行ったアンケートの結果を表 2 に示す。

レイアウトに関するタスクに対しての平均工数は、画面設計は約 2 時間、PRCreator を利用したレイアウト作成は約 76 分、変更は約 23 分であった。被験者のタイプとしては、最低限の機能を満たすリモコンを短時間で作成する被験者（A～D）や、画面設計からレイアウト作成まで数時間を費やす、こだわりを持った被験者（G～J）などが確認できる。また被験者（E, F）は画面設計に 3 時間から 5 時間と多くの工数を費やしていたが、実際のレイアウト作成はそれぞれ 12 分、30 分と他の被験者と比べて短かった。レイアウト完成後のレイアウト変更タスクに対する工数は、全ての被験者で 1 時間未満であった。

アクションの作成工数は平均 22 分であった。読み込み対象のファイル配置ミスや家電操作 URI の記述ミスなど細かなミスが多く、全体的に予想以上の時間を要していた。一方トラブルの無かった被験者 I は、8 分と短い時間でアクション作成を終えていた。

5. 考察

5.1 パーソナルリモコン全体に対する考察

エンドユーザ自らリモコンを作成するという行為に対して考察する。実験ではレイアウトの設計からアクション作成まで、平均で合計 4 時間以上の工数が必要であった。しかしながら、アンケートの結果からは「愛着が沸く」、「カスタマイズできて楽しい」といったパーソナルリモコンの満足感に関する肯定的な意見が得られた。また、リモコンの利便性や使いやすさという観点からも「利用頻度の高いボタンを使いやすい位置に配置できる」、「宅内環境に柔軟に適應できる」、「複数家電を 1 ボタンで連携制御できる」、「宅内の複数のリモコンを 1 つのリモコンに集約できる」といった肯定的な意見が多く得られた。これらの点から、パーソナルリモコンの有用性が確かめられたと考える。

実験に対する制約として、本実験の被験者は全て計算機に関する知識が豊富な学生であったため、リモコンを自身で作成・



図 5 作成されたパーソナルリモコンのレイアウト

表 2 アンケート結果

パーソナルリモコン全体に対する評価	
肯定	自分で作ったものなので愛着が沸く．
肯定	好みに合わせてカスタマイズできて楽しい．
肯定	利用頻度の高いボタンを使いやすい位置に配置できる．
肯定	宅内環境は多様であり，個人適応型リモコンには意義がある．
肯定	ボタン 1 つで複数家電を操作できるて便利．
肯定	宅内の複数のリモコンを 1 つのリモコンに集約できる．
否定	レイアウト作成が趣味に走りすぎて使いにくかった．
実行プラットフォームに対する評価	
改善	家電の現状態を取得できる機能が欲しい．
改善	ジェスチャなどの携帯端末の機能を利用した操作を使いたい．
否定	ボタン押下時の反応が鈍く操作感が悪い．
PRCreator に対する評価	
肯定	視覚的にレイアウトを作成できるのは直観的で分かりやすい．
肯定	家電操作アクションのプリセット機能が非常に便利．
肯定	効果音など遊び機能があって，より愛着が沸く．
改善	家電 URI 入力に対するスペルミス機能が欲しい．
改善	デフォルトレイアウトを用意して欲しい．
否定	GUI エディタとしての必須機能の不足．

カスタマイズするという行為に抵抗が少なかったということが考えられる．より多くの年齢層の被験者に実験を行ってもらい，より客観的な評価実験を行うことは今後の課題の一つである．

5.2 実行プラットフォームに対する考察

プラットフォームへの拡張要求として，家電状態を取得できる機能が欲しいという意見が得られた．実験で用いた CS27-HNS では，家電機器の現状態や，温度・湿度などの宅内環境をネッ

トワーク経由で取得可能である [5]．この状態取得は機器操作と同様に URI アクセスで呼び出すことが可能であり，提案パーソナルリモコンに組み込むことも比較的容易である．この拡張により，携帯端末での機器モニタリングや消費電力振り返りなどの機能を持った，単なる家電操作のためのリモコンに留まらない「HNS 向け個人適応型・統合インタフェース」への拡張が可能である．

次に，リモコン定義言語の記述能力について考察する．アンケートにより，「ジェスチャを利用した操作機能が欲しい」といった定義言語の記述能力に関する改善意見が得られた．現在のリモコン定義言語は従来のリモコンと同様，ボタン押下による操作のみを想定していた．しかし提案実行プラットフォームは Android 端末での利用を想定しており，端末に搭載された加速度センサを利用できるように拡張することも可能である．これにより，リモコン本体のひねる動作や上下左右に振る動作などに家電機器操作を割り当てることで，より直感的で使いやすいリモコンを提供できると考えられる．

次に実行プラットフォームのパフォーマンスについて考察する．否定的な意見として「ボタン押下時の反応が鈍く操作感が悪い」という意見が目立った．実験で利用した CS27-HNS 環境では，各種家電機器を Web サービスとしてラップしており，URI 呼び出しにより家電の機器操作が実行される．家電のサービス化により，ネットワーク越しの家電操作が可能となる一方で，無視できない通信オーバーヘッドが発生する．これは CS27-HNS 環境に依存した問題であり，その性能改善は本研究の対象外となるが，家電同士を直接ネットワークに繋げるよう

なスマートホーム環境では大きな問題にならないと考える。

5.3 PRCreator に対する考察

レイアウトの設計・作成・変更、及びアクション作成を含めた、リモコン作成全行程の平均時間は約 4 時間強であり、ユーザにとっては大きな負担であるように見受けられる。その理由について調査したところ、レイアウト作成に 1 時間以上を要していた被験者 (G~J) は、より自分の好みに合ったリモコンを目指して、レイアウトの試行錯誤を繰り返していた。また図 5 に示すように、被験者 J は強いこだわりを持ってレイアウトを作成しており、被験者同士のレビューの中でも最も好評であった。このような職人気質の強いユーザにとっては、リモコン作成の負担が大きくても、高い満足度が得られると考えられる。

一方、短い時間でレイアウト作成を終えていた被験者 (A~F) に関しては、ほぼ事前設計の内容と同じレイアウトを作成しており、その平均作成工数は約 21 分程度であった。また、アンケートの結果からも、「視覚的にレイアウトを作成できる」、「アクションのプリセット機能が便利」、「効果音等の機能があり、より愛着が沸く」などの肯定的な意見が得られた。これらの結果から、ある程度の工数は要するものの、おおむね PRCreator を利用することで満足 of いくリモコンを作成できると考えられる。

アクション作成に対する平均工数は平均 22 分であった。機器操作をプリセットで用意していた点については肯定的な意見が多かったが、プリセットにない操作に対する URI 記述ミスが多発しており、この点に関する改善が必須課題である。また、コピー&ペーストやオブジェクトの前面 / 背面設定ができないなど、GUI エディタとしての機能不足に関する改善要求も挙げられており、これらに対する改善を今後加える予定である。

6. 関連研究

Crespo と Bier [6] は、専門的な知識を持たない非プログラマを対象とした Web アプリケーション作成支援システム、WebWriter を提案している。WebWriter では、まずハイパーリンクやチェックボックス等の GUI コンポーネントを配置して Web ページを作成する。同様に、CGI による動的コンテンツ (チャット Web アプリのメッセージフィールドなど) の入出力コンポーネントを配置することで、Web アプリケーションが作成される。“WebWriter” は、非プログラマ向けの GUI 作成支援という点で本研究と一致する部分は多く、リモコン GUI への応用も可能である。しかしながら、本研究ではレイアウト定義とアクション定義を別ファイルとして切り離すという設計方針を取っている。この定義の分離により、家電の追加や入れ替えなどの宅内環境の変更に対して、レイアウトを意識することなく既存のアクション定義を修正・再利用可能である。

ユーザの嗜好や環境に応じたインタフェースを提供するための研究も数多く報告されている [7] ~ [9]。Liu らの研究 [7] や Khushraj と Lassila の研究 [8] は、ユーザのプロファイルや現状態、レイアウトの操作履歴などのユーザの情報を利用して、自動的にそのユーザにとって使いやすいインタフェースを提供するものである。また Kryzstof ら [9] は、ユーザの好みや使

用端末に応じて、自動的に使いやすいインタフェースに切り替わるシステムを提案している。例えば、同じ「プリンタの選択を行う」という操作であっても、ユーザがマウスを利用している場合は、プリンター一覧がドロップダウンメニューで提示され、タッチパネルを利用している場合にはリストとして提示される。これらの手法はインタフェース側が自動的にユーザに適應するものであるが、我々の提案するパーソナルリモコンはユーザ自らがインタフェースをカスタマイズするという点で異なる。これらの研究は互いに敵対するものではなく、レイアウトの自動適應機能とユーザによるカスタマイズ機能を融合した仕組みを用意することで、より使いやすく愛着のある個人適應型インタフェースを提供できると考えられる。

7. おわりに

本稿では、パーソナルリモコン作成全工程支援を目的として、レイアウト作成支援ツール PRCreator を改良し、ボタン押下時のアクション設定を支援する機能を追加・拡張した。評価実験の結果、PRCreator を利用することで、高い満足度と利便性を持つパーソナルリモコンを作成できることが確認できた。

今後の課題として、GUI エディタとしての必須機能の追加、家電モニタリングのためのレイアウト定義言語の拡張、より直感的な家電操作を提供するためのアクション定義言語の拡張などが挙げられる。また、ユーザの現状態に応じて最適なレイアウトを自動で提供するような仕組みとの融合を現在検討中である。

謝辞 この研究の一部は、科学技術研究費 (基盤研究 B 23300009, 若手研究 B 21700077, 研究活動スタート支援 22800042), および、ひょうご科学技術協会の助成を受けて行われている。

文 献

- [1] 徳田啓介, 稲田卓也, 松本真佑, 中村匡秀, “ホームネットワークシステムのためのパーソナルリモコン開発フレームワーク,” 電子情報通信学会技術研究報告, vol.110, no.458, pp.7-12, 2011.
- [2] 徳田啓介, 稲田卓也, 松本真佑, 中村匡秀, “ホームネットワークシステムにおけるパーソナルリモコン作成 GUI の実装と評価,” 電子情報通信学会技術研究報告, vol.111, no.152, pp.13-18, 2011.
- [3] “W3C web service activity”. <http://www.w3.org/2002/ws/>.
- [4] “Droiddraw”. <http://droiddraw.org/>.
- [5] M. Nakamura, S. Matsuo, S. Matsumoto, H. Sakamoto, and H. Igaki, “Application framework for efficient development of sensor as a service for home network system,” Int'l Conf. Services Computing (SCC), pp.576-583, 2011.
- [6] A. Crespo and E.A. Bier, “Webwriter: A browser-based editor for constructing web applications,” Int'l J. Computer and Telecommunications Networking, vol.28, no.7, pp.1291-1306, 1996.
- [7] J. Liu, C.K. Wong, and K.K. Hui, “An adaptive user interface based on personalized learning,” IEEE Intelligent Systems, vol.18, no.2, pp.52-57, 2005.
- [8] D. Khushraj and O. Lassila, “Ontological approach to generating personalized user interfaces for web services,” Proc. Int'l Conf. The Semantic Web, vol.3729, pp.916-927, 2005.
- [9] K.Z. Gajos, D.S. Weld, and J.O. Wobbrock, “Automatically generating personalized user interfaces with supple,” Artificial Intelligence, vol.174, no.12-13, pp.910-950, 2010.