

住宅内の機器・サービスを統合するための ビジュアルプログラミング環境とシステム

橋岡 良^{1,a)} 平井 重行^{2,b)}

概要: 家電や様々なセンサ、インターネットサービス（天気予報やニュース、SNS など）、コンテンツまでを様々な組み合わせで統合的に連携・動作させることを目的とした機器やサービスが次々に登場し、住宅のスマートハウス化を進めている。本研究は、その流れにおいてエンドユーザが様々な機器やサービスをより柔軟で自由に組み合わせられるようにすべく、直感的でわかりやすい操作を実現するビジュアルプログラミング環境を開発し、その内部処理を行うためのシステムも構築する。本稿では、そのシステム概要とビジュアルプログラミング環境について述べ、本システムの利用例を示す。

Visual Programming Environment and its system for Coordinating Appliances and Services in a Smart House

RYO HASHIOKA^{1,a)} SHIGEYUKI HIRAI^{2,b)}

Abstract: Various types of appliances and equipment, including sensors, displays, and internet services (e.g., email, social networking services, and news), can improve the smartness of a house. To facilitate cooperation among these devices, we propose a visual programming environment that is more intuitive, flexible, and simple, which allows end users to access additional programming services to produce a smart house. This paper describes the design, functionalities and current implementation of the visual programming environment, and shows examples with our system.

1. はじめに

スマートハウスの研究は、Xanadu House[1] を初めとし、様々な実験住宅プロジェクトが行われてきている [2]-[6]。これらでは実際の住宅内へ IT 技術を埋め込むことから、それらを活用したサービス・システムの構築、人の生活行動を計測しながらインタラクティブに動作するアプリケーションの開発まで、基礎的な研究から応用研究まで様々な展開されている。また、住宅に限らず異なるスマート環境間の連携を試みる研究 [7] や、複数のスマート環境フレ-

ムワークを用いて単一のスマート環境を構築する研究 [8]、多様な状況下の機器指定を実現するインタラクション手法の研究 [9] など、環境の統合や環境構築手法そのものを対象とした研究も行われている。そのような中で、家電機器を中心に複数の機器を連携させるような統合制御を行うためのプログラミング手法およびそのユーザインタフェースの研究がある [10]-[13]。最近では、エンドユーザが住宅内で扱う機器はセンサ類を含んだ家電や住宅設備に留まらず、スマートフォンの普及に伴って様々なインターネットサービス（例えばニュースや天気予報、SNS など）までもが操作対象となりつつあり、幅広い統合や柔軟な連携を実現する必要が出てきている。

このような状況のもと、我々は京都産業大学内に建築された実験住宅 E Home（くすいーほむ）[14][15] を研究プラットフォームとして、住宅で扱う生活を拡張する研究 [16]-[22] を様々な行っている。本研究は、従来の家電や

¹ 京都産業大学大学院 先端情報学研究科
Division of Frontier Informatics, Graduate School of Kyoto Sangyo University

² 京都産業大学 コンピュータ理工学部
Faculty of Computer Science and Engineering, Kyoto Sangyo University

a) i1358077@cc.kyoto-su.ac.jp

b) hirai@cse.kyoto-su.ac.jp

住宅設備だけでなくインターネットサービスをも対象に、さらにはスマートハウスでの新たなシステムやサービスまでを含め、エンドユーザが自由自在に動作を設定・連携させることができるプログラミング環境とその動作を支えるシステムについて扱っている。そして、エンドユーザにとって容易に機器の連携や設定が行えるプログラミング環境を考慮し、直感的でわかりやすい操作を実現するビジュアルプログラミング環境に着目したシステムを構築している。本稿では、我々の研究システムの全体概要を設計、およびプログラミング環境の具体的な動作とプログラム例を示すと共に、どのような形で導入・利用できるのかをシナリオ例として示す。

2. 関連研究や機器・サービス

2.1 スマート環境を実現するリモコン機器

従来から、学習型や複数の AV 機器などのリモコンを統合する形の赤外線リモコンは多数存在していた。最近では、iRemocon^{*1}や Pluto^{*2}のように WiFi やイーサネットなどネットワーク対応し、スマートフォンからでも操作可能な赤外線リモコン機器が幾つか登場している。iRemocon については、開発者向け情報が公開されており、スマートフォンアプリに限らず外部プログラムを通じて様々な機器を統合、連携して制御することも可能である。これらリモコン機器は、ネットワーク対応していない既存家電製品をコントロールする上では欠かせないものであり、本研究でもこれらを通じて機器制御ができるよう対応する。

2.2 スマート環境のためのネットワークプロトコル

スマートハウス環境を構築するためのネットワークプロトコルとしては、UPnP^{*3}や DLNA^{*4}、ECHONET/ECHONET Lite^{*5}などが規格策定され、実装・実用化が進んでいる。また、それらをまとめて様々な住宅設備や家電、センサ類を扱えるようにする住宅 API^{*6}のようなものも開発が進められている。本研究においても、これらに対応した形で統合的にプログラミングが可能な環境を構築する。

2.3 家電プログラミングやスマート環境構築に関する研究

個別にリモコンで操作していた家電機器を複数対象に連動制御するための、プログラミング環境についての研究がある。例えば、増井らによる実世界指向 GUI プログラミング環境 [10] では、家電機器の個別機能を表したバーコードを連続して読み込むことで、連携する機器・機能のプロ

グラミングを行う手法を提案している。福地らは pin 型タグを用いてユーザが機器やセンサにピン接続することでそれらの関連付けを行う実世界でのプログラミング環境 [11] を提案している。大和田らは JavaScript ベースの開発環境 Kadecot [12] を提供し、家電連携を容易にすることで機器とコンテンツを結びつけたスマートフォンアプリを簡単に開発できるようにしている。さらに、現時点で住宅内の家電製品などは対象ではないが、ネットワーク上のサービス同士やガジェットなどとの連携をルール記述の形でエンドユーザが記述して、ある種のプログラミングを行える IFTTT^{*7}や Zapier^{*8}などのサービスがインターネット上で提供されている。本研究では、これら研究や既存サービスとは違い、ビジュアルプログラミング環境を中心としたプログラミング手段を一般ユーザに提供する。一方で、Meliones らはビジュアルプログラミング環境のひとつである LabVIEW 上で動作する家電を連携させる仕組み [13] を開発している。我々のビジュアルプログラミング環境はユーザがより使いやすい環境を目指すために独自で開発し、より拡張性を増すためにサーバシステムも構築している。

3. システム設計と概要

本研究システムはエンドユーザが利用するクライアント端末上で動作するビジュアルプログラミング環境と、アプリケーションサーバとで主に構成され、アプリケーションサーバは住宅内の様々な機器・センサ類、またインターネット上の様々なサービスとの間で情報のやりとりを行う(図 1 参照)。エンドユーザは PC やスマートフォン、タブレット機器などの端末を用いて、機器やサービスの連携をデータフロー型のビジュアルプログラミング環境でプログラム作成し、そのプログラムは XML もしくは JSON 形式で WebSocket を介してアプリケーションサーバへ送信、データベースに保存される。アプリケーションサーバ上では、機器やサービスを統合制御するプログラムが動作しており、プログラムに従って状況に応じた機器制御やサービスを連携動作させる仕組みになっている。

3.1 アプリケーションサーバ

アプリケーションサーバは、Web サーバ機能と本システムで作成したプログラムを管理・保存するデータベース、およびそのプログラム実行機能とで構成する。開発環境は Java1.6 で、WebSocket 通信には Jetty 8.0.4.v20111024 を用いている。

Web サーバとしては、クライアント端末(ビジュアルプログラミング環境)との間の通信に WebSocket を用いる。ここでは、クライアント端末側で作成・編集されたプログラムがアップロードされてくるとデータベースに保存す

^{*1} iRemocon : <http://i-remocon.com/>

^{*2} Plute : <https://pluto.io/>

^{*3} UPnP : <http://www.upnp.org/>

^{*4} DLNA : <http://www.dlna.org/>

^{*5} ECHONET : <http://www.echonet.gr.jp/>

^{*6} 住宅 API : <http://www.daiwahouse.co.jp/lab/HousingAPI/>

^{*7} IFTTT : <https://ifttt.com/>

^{*8} Zapier : <https://zapier.com/>

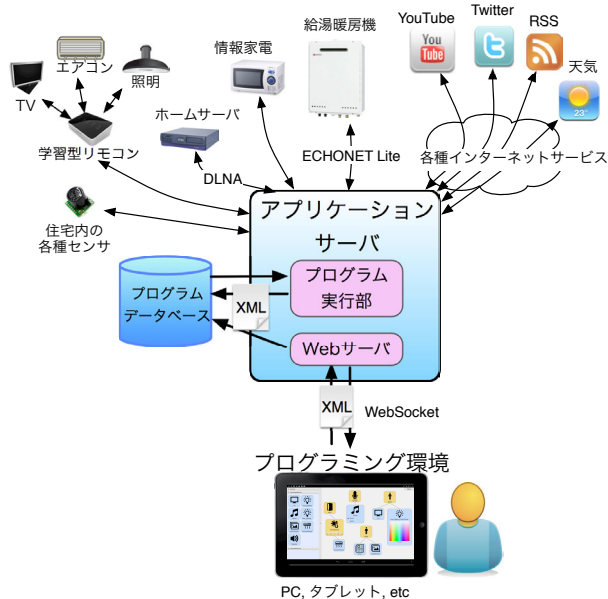


図 1 システム概要図

る。また、プログラム実行状態やセンサの値などの情報は都度プログラム実行部からクライアント端末へ WebSocket のプッシュ機能を用いて送信する。WebSocket は従来の HTTP の通信と違い、クライアントからサーバへの Pull 機能しかないのではなく、サーバからクライアントへの Push 機能があり、そのオーバーヘッドが少ないことが特徴であることから、本システムのサーバ-クライアント間の通信に用いる。

ビジュアルプログラミング環境で作成されたプログラムは、XML もしくは JSON 形式で記述され、サーバ側で保存・管理・実行される。クライアント端末上でプログラミング環境が起動されると、サーバからプログラム一覧や対応機器のリストなどが送られてビジュアルプログラミング環境上で表示される。

プログラム実行部は常時稼働しており、データベース上にあるプログラムのうち「実行」と指示されたものを読み込んで実行する。ここでは、UPnP や DLNA, ECHONET Lite, 住宅 API などのネットワークプロトコルによる家電や住宅設備類、そして統合型リモコン機器を用いた家電制御、その他様々なインターネット上のサービスを連携対象として実行する。また、Arduino や Raspberry Pi などを用いて日曜大工的に住宅へ組み込んだセンサ機器などについても、OSC^{*9}などを用いたカスタム仕様にも柔軟に対応する。

3.2 プログラミング環境

クライアント端末上で動作するプログラミング環境は、Max/MSP や PureData, vvvv などのデータフロー型ビジュアルプログラミング環境として構成している。今回、

^{*9} OSC (OpenSoundControl) : <http://opensoundcontrol.org/>

我々が開発したビジュアルプログラミング環境を図 2 に示す。図 2 の環境は Android 4.4.2 上で動作し、Java1.7 の javafx を用いて開発している。

図 2 に示す画面内部では、左側に機器や設備、サービスなどを表すオブジェクトがカテゴリ分けされた状態で表示され、右側にプログラミングを行うプログラムフィールドがある。ユーザは左側のオブジェクトリストからドラッグアンドドロップでプログラムフィールドへオブジェクトを配置し、2つのオブジェクト間をパッチコードで接続することでプログラムを作成する。このパッチコード間は、基本的にトリガー情報が流れており、そのトリガーを発する条件は、各オブジェクトにユーザが設定した内容に応じる。条件設定については、次章で詳細を述べる。個別オブジェクトの状態やセンサの値などはサーバ側から WebSocket の Push 通信によって受信し、対応するオブジェクトの状態がわかるようになっている。なお、ここではパッチコード同士で接続された一連のオブジェクト群をここではプログラムと呼ぶため、図 2 のプログラムフィールドには全部で 4 つのプログラムが描かれている。これらエンドユーザが記述したプログラムは、プログラム作成者のユーザ ID と各プログラムに割り当てられた UUID と共に XML のデータとしてサーバに送信し保存される。



図 2 プログラミング環境概要図

なお、図 2 に示した Android 端末上で動作している環境とは別に、PC 上で動作している環境も開発している。

4. ビジュアルプログラミング環境の説明とプログラム例

4.1 本システムのプログラミング環境

現時点でのビジュアルプログラミング環境の外観は図 2 の通りである。ここでは、オレンジ色のオブジェクトは、温度や扉の開閉など環境情報を知らせる機能、人の行動センシングに基づいて動作する機能、時間に伴って処理を行う機能などがあり、総じて入力オブジェクトと呼んでいる。水色のオブジェクトは、操作する機器や情報提示出力する機器・サービスを表す出力オブジェクトを表している。

入力オブジェクトは、別の入力オブジェクトと接続できるインレットがあり、他の入力オブジェクトや出力オブジェクトと接続できるアウトレットがある。出力オブジェクトは、入力オブジェクトと接続できるインレットがある。これらは、オブジェクトに応じた機器やサービスの設定を行える。オブジェクトははじめ図 3 の左のようにアイコンと名称だけが表示されている。その状態のオブジェクトを pinch-out すると、オブジェクトの設定を行うための GUI が追加される。再び pinch-in すると追加された GUI は削除されるが、設定された値はオブジェクト内に保持されている。

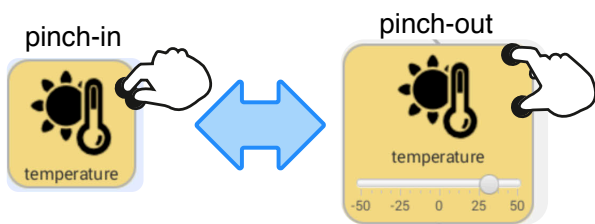


図 3 オブジェクトの設定画面の開き方

入力オブジェクトや出力オブジェクトの例とその設定表示が面を図 4 に示す。図 4 の a) は気温が何度かときにトリガー情報が流れるかをスライダーを用いて設定できる。b) はトリガー情報を受け取ったときに音楽を再生するか停止するかを設定でき、図では再生する設定を行っている。c) はライト（Philips の hue）の調光設定をカラーピッカーを用いてできる。d) は時刻が 6:00 から 18:10 の間にトリガー情報を流す設定をレンジスライダーを用いて行っている。現在は図 4 の 4 種類の GUI を実装したする。

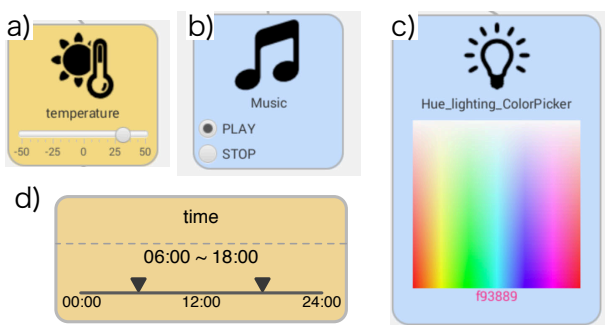


図 4 オブジェクト設定表示の例

a) : 温度センサ, b) : 音楽プレーヤ, c) : ライト, d) : 時間

エンドユーザは図 2 画面の左側に列挙されているオブジェクトをプログラムフィールドにドラッグアンドドロップし、プログラムフィールド上に存在するオブジェクトをパッチコード（線）で繋ぐ事で入出力の関係を記述する（プログラミングを行う）。パッチコードは図 5 のように、繋ぎたい 2 つのオブジェクトをタッチし、タッチしたオブジェクトをぶつけることで接続できる。

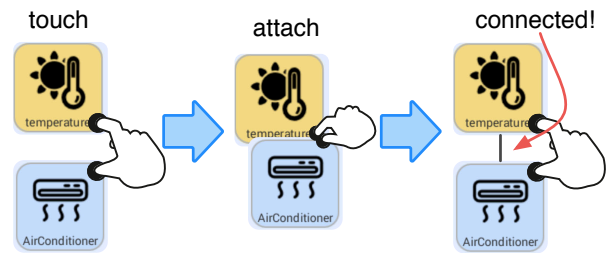


図 5 オブジェクトの接続方法

パッチコードで繋がれたオブジェクト群は先頭（上側）から状態を確認してゆき、入力オブジェクトの場合は機器やサービスが設定に合致していればアウトレットからトリガーを出力する。そして、その下流で接続された入力オブジェクト全てが設定に合致した状態になれば、出力オブジェクトにトリガーが入力され、機器やサービスの制御が行われる。なお、機器やサービスがオブジェクトの設定条件に合致した状態になっている場合には、オブジェクトの外枠と文字が赤くなる（図 2 と図 6 参照）。

4.2 プログラム例

本システムのプログラミング環境で作成した簡単なプログラム例を図 6 に示す。このプログラムは扉が開くと空調の電源を切るプログラム（図 6 の左）の例であり、扉を表す door オブジェクトとエアコンを表す AirConditioner オブジェクトが接続されている。door オブジェクトは入力オブジェクトで、スライダーで close か open の状態設定が行える。close を設定すれば、扉が閉じてあれば設定と合致した状態となり、open の場合は扉が開けば合致する状態となる。AirConditioner オブジェクトは出力オブジェクトで、設定項目（On/Off など）の制御を行う。このプログラムでは、扉が閉まれば画面上で door オブジェクトの枠と文字が赤くなり、エアコンの電源を切る動作となる。

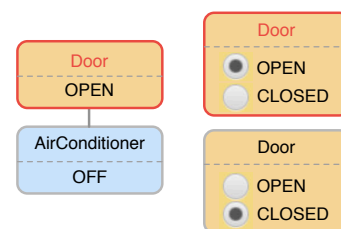


図 6 ドア開閉のプログラム例

次に、別のプログラムとして図 7 と図 8 の例を示す。

図 7 は 17:00 から翌朝 5:00 までの間、人がリビングルームに居れば照明がつくプログラムである。time オブジェクトは図 4 の d) にもあるようにスライダー上の 2 つのノブで時間範囲の時刻設定を行い、設定した時刻範囲内で条件合致と判断する。place オブジェクトも住宅内の部屋を選択しておき、その場所に人がいれば条件合致となる（赤外線焦電センサなどの手段で人の検出を行う）。light オブ

ジェクトもスライドバーで照明操作を設定することができる。このプログラムは time オブジェクトと place オブジェクトとの両方が設定条件に合致しておれば、light オブジェクトの設定操作が実行される。

図 8 では、複数の入力オブジェクトと出力オブジェクトが接続され、複数の処理が一つにまとめられている例である。place オブジェクトと temperature オブジェクトがそれぞれ人の検出と設定温度範囲の条件判定を処理する入力オブジェクトである。ここでは place オブジェクトでリビングルームに人がいるかを判定し、その結果に応じて接続された Music オブジェクトが音楽を再生する。一方で、place オブジェクトから temperature オブジェクトへ接続され、さらに AirConditioner オブジェクトへ接続された部分では、人がリビングルームにいと共に室温が 20～40 度の範囲にある場合に冷房を ON にする、という処理を意味する。



図 7 プログラム例 1

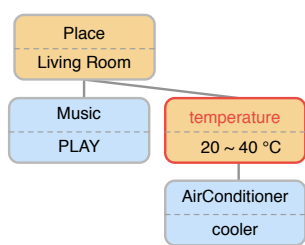


図 8 プログラム例 2

5. 本システムの利用シナリオ

本システムを住宅に導入すれば、従来の家電や様々なセンサなどが統合的に扱え、エンドユーザが個人や家族で扱うインターネットサービスまでも含めてのプログラム記述や操作が可能となる。本章では、本システムが住宅に導入された際の利用例をシナリオと共に紹介する。

5.1 家電機器の統合制御例

H さんは映画が好きで、自宅では頻繁に液晶プロジェクタで映画を視聴している。住宅はスマート環境の導入が進んでおりソファには座っている状態を検出できるセンサがあり、部屋のマイクを通じて音声認識機能もある。H さんは以前は映画を視聴する度にメディアプレーヤーやプロジェクタなど複数機器のリモコンを操作していた。あまり使わない機能のボタンが多数あるリモコンを見て、その操作を手間を感じた H さんは、本研究システムを用いて図 9 のようなプログラムを記述した。プログラムの内容は、ソファに座り、音声認識で「DVD」という発話が認識された場合に、スクリーンが降りて液晶プロジェクタの電源が付き、映画が再生され、別に音楽が再生されていたら音楽を停止し、照明を消すプログラムである。このようなプロ

グラムを一旦記述して実行設定しておく事で、部屋の中ではプログラムが動作し続ける。これにより、H さんはソファに座って「DVD」と発話するだけで DVD プレーヤーが再生して映画を見ることができるようになった。

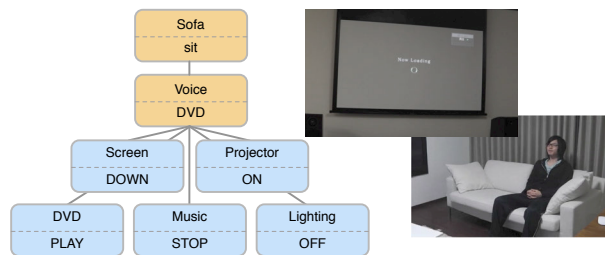


図 9 家電機器統合制御の例

5.2 インターネットサービスの統合と情報提示例

R さんは毎朝、外出する前に、スマートフォンで天気予報や最寄り駅の電車の時刻を確認し、次の電車までに時間があれば Twitter のタイムラインを確認している。これらのサービスを閲覧するためにはスマートフォンを取り出し、複数のアプリケーションを切り替えて情報を得る必要がある。そこで R さんは図 10 に示すプログラムを本システムを用いて記述した。ここでは玄関に行けば、天気予報と Twitter のタイムライン、電車の時刻を玄関で表示するプログラムである。この住宅の玄関では床面がディスプレイとして機能するように設定されており、表示する複数種類の情報はレイアウトされた状態で表示される。このようなプログラムを記述することで、R さんはスマートフォンを操作せずとも、玄関で靴を履く際に天気予報や次の電車の時刻、Twitter のタイムラインが確認できるようになった。



図 10 統合情報提示の例

5.3 機器とサービスを用いた複数プログラムの統合例

A さんは天候が雨だったり、気温が低かったりと A さんに合った環境でないと仕事が捗らないことがある。普段は仕事に集中するために、部屋のライトの色を青空をイメージした明るめの青色にしたり、暖房をつけることで対処し

ている。しかし、仕事に集中してから天候や気温が気になることもあり、仕事を中断してライトやエアコンの操作をすることが非効率に感じた。そこで、Aさんは図11に示すプログラムを本システムを用いて記述した。ここでは仕事部屋にいるとき、天気が雨であればライトを明るい青色にし、気温が15度以下であれば暖房をつけるプログラムである。このようなプログラムを記述することで、Aさんは仕事環境を気にすることなく、仕事に集中できるようになった。

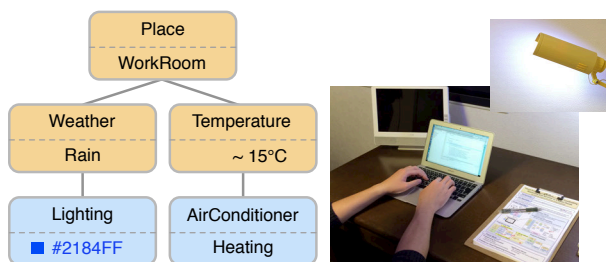


図 11 複数プログラムの統合例

6. おわりに

本研究は、スマートハウス研究の1つとして、家電や住宅設備などの既存機器や、様々なインターネットサービスを連携、統合的に制御するシステムと、そのためのエンドユーザが用いるプログラミング環境を構築した。本稿では、本システムの概要と設計について述べ、ビジュアルプログラミング環境の動作とプログラム例と共に、具体的にスマートハウスでどのように利用できるかのシナリオも示した。そして、この研究テーマは、我々の研究プラットフォームである実験住宅E Home（くすいーほーむ）において、実際に動作するシステムとして構築している。今後は、プログラミング環境の評価実験を行うとともに、より多くの家電機器やインターネットサービスと連携すべく、DLNAやUPnPの機能を組み込むと共にECHONET Liteや住宅APIなどにも対応していく予定である。

参考文献

- [1] Mason, Roy., Jennings, Lane., Evans, Robert. The Computerized Home of Tomorrow and How It Can Be Yours Today!, Acropolis Books, 1983.
- [2] Cory D. Kidd, et al. : The Aware Home: A Living Laboratory for Ubiquitous Computing Research, Proc. of the Second International Workshop on Cooperative Buildings - CoBuild' 99. Position paper, 1999.
- [3] S. S. Intille, K. Larson, J. Beaudin, E. Munguia Tapia, P. Kaushik, J. Nawyn, and T.J. McLeish, "The Place-Lab: a live-in laboratory for pervasive computing research (Video)," Proc. of Pervasive2005, 2005.
- [4] B.de Ruyter, et al. Ambient Intelligence Research in HomeLab: Engineering the User Experience, Ambient Intelligence, Springer, pp.49-61, 2005.
- [5] 上田博唯, 山崎達也. ユビキタスホーム: 日常生活支援の

- ための住環境知能化の試み, ロボット学会論文誌, Vol.25, pp.10-16, 2007.
- [6] 椎尾一郎, 元岡展久, 塚田浩二, 神原啓介, 太田裕治. Ocha House とユビキタスコンピューティング, ヒューマンインタフェース学会誌, Vol.12, No.1, pp.7-12, 2010.
- [7] 岩崎陽平, 榎堀優, 藤原茂雄, 田中宏一, 西尾信彦, 河口信夫. REST に基づく異種スマート環境間のセキュアな連携基盤. マルチメディア, 分散, 協調とモバイル (DICOMO2008) シンポジウム論文集 2008.
- [8] Yu Enokibori, Nobuhiko Nishio. Smart Environment Suite: To build smart environments with collaboration among multi-smart environment frameworks. Late Breaking Results at Pervasive 2009.
- [9] 伊藤昌毅, 橋爪克弥, 河田恭生, 天目直哉, 伊藤友隆, 井村和博, 西條晃平, 中澤仁, 高汐一紀, 徳田英幸. Chameleon: 多様な状況下の機器指定を実現する複数インタラクション統合技術. 情報処理学会論文誌 Vol.52, No.4, pp.1571-1585, 2011.
- [10] Masui Toshiyuki, Itiro Siio. Real-world graphical user interfaces. In Handheld and Ubiquitous Computing. Springer Berlin Heidelberg pp72-84, 2000.
- [11] Kentaro Fukuchi, Takeo Igarashi, Maki Sugimoto, Charith Fernando, Masahiko Inami. Push-pins: Design-by-user Approach to Home Automation Programming. International Conference on Ubiquitous Computing, workshop 5, 2009.
- [12] Shigeru Owada, Fumiaki Tokuhisa. Kadecot: HTML5-based visual novels development system for smart homes. Consumer Electronics, 2012 IEEE 1st Global Conference on, pp17-19, 2012.
- [13] Apostolos Meliones, Dimitrios Giannakis. Visual Programming of an Interactive Smart Home Application using LabVIEW homes. In INDIN 2013, IEEE (2013), 655-660.
- [14] 平井重行, 上田博唯. 京都産業大学の生活型実験住宅E Home（くすいーほーむ）について. 電子情報通信学会 信学技報, Vol.110, No.35, MVE2010-12. pp.43-50, 2010.
- [15] 平井重行, 上田博唯. 実験住宅E Home（くすいーほーむ）でのユーザエクスペリエンス研究へ向けて. 計測自動制御学会 SI2011 講演論文集, 2011.
- [16] 大西諒, 平井重行. RFID を用いた浴室内行動計測の基礎検討, 情報処理学会論文誌 Vol.49, No.6, pp.1932-1941, 2008.
- [17] 林宏憲, 大西諒, 平井重行. 一般住宅用浴室におけるミストを利用した立体的映像表現. エンタテインメントコンピューティング 2007 論文集, pp.75-76, 2007.
- [18] 榎原吉伸, 林宏憲, 平井重行. TubTouch: 湯水の影響や自由形状への適用を考慮した浴槽タッチ UI 環境, 情報処理学会論文誌 Vol.54, No.4, pp.1538-1550, 2013.
- [19] Shunsuke Morioka, Hirotada Ueda. Cooking Support System Utilizing Built-in Cameras and Projectors, Proc. of IAPR Conference on Machine Vision Applications, pp.271-274, 2011.
- [20] Shigeyuki Hirai, Keigo Shima. Multi-touch Wall Display System Using Multiple Laser Range Scanners, Proc. of Interactive Tabletops and Surfaces 2011, pp.266-267, 2011.
- [21] Shigeyuki Hirai, Yoshinobu Sakakibara, Seiho Hayakawa. Bathcratch: Touch and Sound-Based DJ Controller Implemented on a Bathtub, In Proc. of 9th International Conference ACE 2012, pp.44-56, 2012.
- [22] 松浦祐樹, 平井重行. マルチタッチ壁ディスプレイを用いた実寸大電子書架システム. インタラクション 2013 インタラクティブ発表 3EXB-52, 2013.