

目標温湿度環境への遷移を 最小コストで実現する家電制御システム

宵 憲治^{1,a)} Khaled El-Fakih^{2,b)} 安本 慶一^{1,c)}

受付日 2015年10月2日, 採録日 2016年2月23日

概要: 本論文では, 様々な家電デバイスを効率良く制御するコンテキストウェアシステムの実現を目的に, 目標温湿度環境への遷移を最小コストで実現する家電制御システムを提案する. 提案システムは, 著者らが以前から開発してきた任意の2つのコンテキスト間を最小コストで遷移させる家電操作系列をシミュレーションにより求めるツール PathSim を用いて, 実空間における温湿度を目標の値に最小コストで遷移させる. 本システムを様々な環境で実現できるように, 任意の家電の操作系列を記述可能な家電制御言語を定義し, PathSim の出力を提案言語の形式に変換する機能と, 提案言語で記述された家電制御シナリオを実行し, 赤外線リモコン学習デバイス IRKit を用いて各対象家電を制御する機能を持つシステムを実装した. 本システムの有用性を示す実証実験として, 実際のスマートホーム環境でシステムを稼働させ, 目標温湿度環境への遷移の際の消費電力, 遷移時間を測定し, PathSim によるシミュレーション結果と比較した. その結果, 消費電力, 遷移時間ともに 10%以下の誤差に収まることを確認した.

キーワード: コンテキストウェアシステム, スマートホーム, 省エネ家電制御, ホームオートメーション

An Appliance Control System for Reaching Target Temperature and Humidity with Minimal Cost

KENJI YOI^{1,a)} KHALED EL-FAKIH^{2,b)} KEIICHI YASUMOTO^{1,c)}

Received: October 2, 2015, Accepted: February 23, 2016

Abstract: In this paper, aiming to realize a context-aware system for controlling various home appliances in a home, we propose an appliance control system for reaching target temperature and humidity with minimal energy cost. The proposed system changes the temperature and humidity of the room to the target values by using a tool called PathSim developed in our previous research which derives an appliance operation sequence for moving between two any context at the minimum cost through simulation. To realize this system, we defined a device control language to represent appliance control sequences derived by using PathSim. In addition, appliances can be operated via the wifi-infrared gateway (called IRKit) from the home network. We confirmed effectiveness of this system by experiments in actual smart home. This result shows discrepancy between real environment and simulation-based values of power consumption is 10% or less.

Keywords: context-aware system, smart home, energy-saving appliance control, home automation

1. はじめに

ユビタスコンピューティングシステム [1] やコンテキストウェアシステム [2] は, 周囲の環境をセンシングし, それらに適応するように動作するシステムである. 近年, ユビタスコンピューティングシステムを実現する多くの研究 [3] が行われている.

¹ 奈良先端科学技術大学院大学情報科学研究科
Nara Institute of Science and Technology, Ikoma, Nara 630-0192, Japan

² Department of Computer Science and Engineering,
American University of Sharjah

^{a)} yoi.kenji.yc6@is.naist.jp

^{b)} kelfakih@aus.edu

^{c)} yasumoto@is.naist.jp

ユビキタスコンピューティングシステムが実装された空間や家はスマートホーム（またはスマートスペース）[4]と呼ばれている。スマートホームには、センサデバイスや家電デバイス等が分散して設置されている。センサデバイスはスマートホームを構成する各空間の温度や湿度といった空間の物理的特性、ユーザの位置情報や行動等の状況をセンシングする。また各デバイスは、スマートホーム内のユーザにサービスを提供するためにネットワークを通して操作される。

スマートホームは、現在のコンテキストから、ユーザの好みのコンテキストに到達するように、デバイスを適応的に操作する必要があるが、一般的に、複数のデバイスがスマートホーム内に混在している場合、スマートホームをあるコンテキストから異なるコンテキストに遷移させるデバイスの操作列は複数存在する。そして、操作列ごとに操作するデバイスや操作する順番が異なるため、消費電力量や動作時間がそれぞれ異なる。

たとえば、スマートホーム内の温度と湿度の組をコンテキストとし、温度 13°C と湿度 20% (13°C, 20%) というコンテキストから、(23°C, 50%) のような異なるコンテキストに遷移させる場合を考える。このとき、スマートホーム内に、エアコン、加湿器、オイルヒータが設置されていると、温度を上昇させるためにエアコンやオイルヒータが使用でき、また、湿度を上昇させるために加湿器を使用することができる。つまり、目標のコンテキストへ到達するために有効な家電デバイスが 3 種類存在し、これらのうち 1 種のみを使用する場合、複数同時に使用する場合等があり、さらにデバイスを制御する順番や動作させる時間の長さ等を考慮すると、膨大な数の異なる操作列が存在する。この中から最小（または準最小）コストの操作列を自動的に見つけることができれば、ユーザにとって非常に有益だと考えられる。

著者らは、文献 [5] において、スマートホーム内の温湿度の組をコンテキストとし、2つのコンテキスト間を最小コストで遷移させる家電操作系列を、シミュレーションによる遷移コストの動的導出と A*アルゴリズムにより求めるツール PathSim を提案し、PathSim によって導出されたデバイス操作列が、人が直感的な方法でデバイスを操作する場合と比べ、デバイスの合計消費電力量を最大で 45% 削減できることを確認している。

本論文では、前述の PathSim がシミュレーションにより導出したデバイス操作列を実際のスマートホームに適用するための、実環境での実行系を構築することで任意のコンテキスト遷移を最小コストで行う家電制御システムを構築することを目的とする。

提案システムは、温湿度センサがスマートホーム内外の温湿度を数秒間隔で計測し、ホームサーバにデータをアップロードしており、リアルタイムに現在の温湿度が確認で

きる環境を想定している。まず、対象とするスマートホームの部屋の間取り等の空間情報、スマートホーム内外の現在の温湿度から、外気がスマートホーム内の温湿度に及ぼす熱量等を計算し、PathSim を用いて目標の温湿度に最小コストで遷移させるためのデバイス操作列を求める。求められたデバイス操作列に基づいて、各家電を制御しなければならないが、実環境において家電を制御する環境は様々なものが想定される。本システムでは、あらゆるスマートホーム環境において統一した方法で家電の制御を行うため、PathSim によって求めたデバイス操作列を、家電への命令の種類やタイミング、温湿度センサの観測の範囲等を定義した中間言語である「家電制御言語」に基づいた家電制御スクリプトに変換する。求められた結果は、家電への命令の種類やタイミング、温湿度センサの観測の範囲等を定義した家電制御言語に基づいて家電制御スクリプトに変換される。次に、変換されたデバイス操作列に基づいて、IRKit [6] を用いた家電操作統合システムによって家電を自動制御することで、対象空間内を最小コストで目標の温湿度に遷移させる。本システムは、スマートホーム内外の温湿度を確認するための温湿度センサとホームサーバ、IRKit、赤外線リモコン対応家電があれば実現可能であり、一般家庭への導入も容易であると考えられる。

奈良先端科学技術大学院大学内にある実際のスマートホーム設備内で本システムの動作実験を行った結果、PathSim のシミュレーションにより導き出された消費電力量、遷移時間と比べ、10%以下の誤差でコンテキストの遷移ができることを確認した。

2. 関連研究

近年、スマートホームとコンテキストウェアシステムに関する様々な研究が行われている [7], [8]。MavHome [7] では、住民の行動予測に基づいて、家庭内の状態（コンテキスト）をユーザが快適なものに遷移させるコンテキストウェアサービスを提案している。しかしながら、サービスを提供する際に、エネルギー効率の良いデバイス制御を考慮していない。また、Kashimoto ら [9] は、省エネのために期間ごとに利用可能なエネルギー量が制限されているという仮定のもと、各家電のエネルギー消費量、ユーザの状況、ユーザの嗜好等を考慮して、居住者の快適さのレベルが最大化されるように、スマートホーム内の家電に利用可能なエネルギー量を準最適配分する手法を提案している。しかしながら、この研究においても、各環境変数の値を変えられるデバイスが 1 つだけと仮定しており、複数のデバイスを協調させながら、省エネルギーに特定のコンテキストに遷移することは考えていない。

HEMS [10], [11] (Home Energy Management System) は、スマートホーム内の家電機器の電力消費量を見える化したり、自動制御することでエネルギーを節約するための

管理システムであり、日本国内外の様々な企業や研究者が普及に向けて研究を行っている。しかし、現在の HEMS は、電力可視化と遠隔制御による無駄遣い防止や、蓄電、売電に基づいた電気料金最小化のための家電の自動制御は実現されているが、コンテキストの遷移におけるエネルギー効率の最適化は行われておらず、また、配線工事やシステムの設置による導入コストの高さから、一般家庭への普及には至っていないのが現状である。

本論文では、スマートホーム内のコンテキスト遷移のコスト最小化に着目し、目的の温湿度までのエネルギー効率の最も良い家電制御と、一般家庭への普及のため、容易に導入可能なシステムの実現を目指している。

3. 家電制御システムの構成

本章では、提案する家電制御システムの全体構成を示し、処理の流れに沿ってシステムの構成要素の詳細を記述していく。

まず、提案システムの概要図を図 1 に示す。

システムは、デバイス操作列導出ツール PathSim、デバイス操作列変換プログラム、家電制御スクリプト実行プログラム、家電操作統合システムに分かれている。まず、目標の温湿度であるコンテキストを設定し、PathSim により、現在のコンテキストから目標のコンテキストに遷移するためのデバイス操作列を導出する。そして、PathSim が導出したデバイス操作列をデバイス操作列変換プログラムによって中間言語である家電制御スクリプト（家電制御言語）に変換する、次に、家電制御スクリプト実行プログラムによって、IRKit [6] を用いて構築した家電操作統合システムに対して家電制御信号を送ることで、デバイス操作列に応じた各家電の制御が可能になる。

3.1 コンテキストの定義

本研究で想定するコンテキストは、文献 [6] をもとに定義しており、以下にその概要を示す。コンテキストは、対象

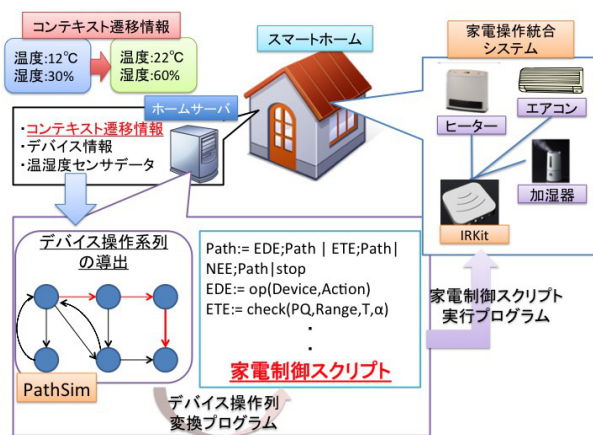


図 1 システム構成図

Fig. 1 System architecture.

とする空間内に存在するデバイスとその属性の集合と空間内の温湿度の値の組からなる。空間内には、エアコン、加湿器等の空調デバイスが設置されており、これらのデバイスの集合を $D = \{d_1, \dots, d_m\}$ と表す。各デバイス d_i は、それぞれ固有の属性 $A^{d_i}(j)$, $j = 1, \dots, n$ を持つ。たとえば、エアコン (AC) の属性は、電源状態 (Status)、モード (Mode) 等の属性を持ち、それぞれ $A^{AC}(1) = \text{Status}$, $A^{AC}(2) = \text{Mode}$ のように表される。また、デバイスの属性 $A^{d_i}(j)$ は、それぞれ、定義域 $Dom(A^{d_i}(j))$ を持つ。たとえば、エアコンの属性は $Dom(A^{AC}(1)) = \{ON, OFF\}$, $Dom(A^{AC}(2)) = \{Heat, Cold, Dehumidifier\}$ のような定義域をそれぞれ持つ。

また、温湿度の値は、温度を t 、湿度を h としてそれぞれ表す。これらの属性の値は、デバイスの動作を通して変化し、センサデバイスなどから取得された値が設定される。

対象空間のコンテキストの遷移は、ノードの集合 V とエッジ (有向辺) の集合 E からなる有向グラフ $G = (V, E)$ で表すことができる。ここで、ノードはコンテキスト、有向辺は 2 つのコンテキスト間の遷移に対応している。また、各有向辺には、2 つのコンテキスト間の遷移を可能にするイベントとコストが関連付けられる。しかしながら、デバイスの属性、温湿度の属性を連続値として扱うとコンテキストとして扱う範囲が無限になる。また、使用するデバイスの仕様や性能 (センシング可能な値の範囲および分解能) に合わせて上限と下限を持った離散値として扱うことが可能であるが、設定可能な、または取得可能な値をすべてコンテキストとして扱うと、同様にコンテキスト空間は膨大になり、コンテキスト空間を探索するためのシステムの負担が大きくなる。この問題に対し、本論文では、コンテキストの各属性を、有限個の連続的な範囲 (レンジ) に適切に分割することで対処する。たとえば、温度のレンジは、 $(-\infty, 0]$, $(0, 10]$, $(10, 20]$, $(20, 30]$, $(30, \infty)$ のような有限個のレンジに区切ることができる。

本研究では、環境デバイスイベント、環境遷移イベントという 2 種類のイベントを考える。環境デバイスイベントは、エアコンや加湿器といった対象空間内の環境に影響を与えるデバイスの属性を設定するイベントである。一方、環境遷移イベントは、温度や湿度といった環境に関する属性があるレンジから異なるレンジに遷移することによるコンテキストの遷移を表すイベントである。本研究では、温湿度のみに限定している。また、コンテキスト間の遷移にかかる合計消費電力量をエッジのコストとする。

3.2 デバイス操作列導出ツール PathSim

ここでは、スマートホームにおいて、あるコンテキストから任意の別のコンテキストへ遷移する最小コストのデバイスの操作列を導出するツール PathSim [5] の概要について述べる。

PathSim は、A*アルゴリズムを基としたアルゴリズムにより最小コストのデバイス操作列を導出する。A*アルゴリズムは有限個のノードとコストが付されたエッジで構成されたグラフに対してのみ適用可能なアルゴリズムである。スマートスペースのコンテキストの遷移は、各コンテキストをノードの集合 V 、コンテキスト間の遷移をエッジの集合 E とする有向グラフとして表すことができる。しかしながら、一般的にコンテキストの各変数がとりうる値の範囲は連続または無限であるため、そのままでは A*アルゴリズムを適用することができない。そこで、A*アルゴリズムを適用できるように、コンテキストを構成する各属性（変数）の値域を有限個の範囲に分割し、離散的なコンテキスト遷移モデルとして表す。また、A*アルゴリズムが適用されるグラフでは、ノード間を結ぶエッジ（およびコスト）の集合があらかじめ与えられる必要があるが、本論文で対象とするコンテキストの遷移では、エッジの存在およびエッジのコストが未知である。これは、対象空間の特性（広さや壁の材質等）、空調家電の性能等により、隣接するコンテキストへの遷移にかかるコストが変わることによる。そのため、探索を行う各コンテキストにおいて、隣接するコンテキストへのエッジの存在およびそのコストを動的に調べる必要がある。PathSim は、各コンテキストにおいて実行可能なデバイスの操作をシミュレーションにより模擬実行することで、イベントによって遷移する隣接コンテキストと遷移に必要なコストを導出する。そして、A*アルゴリズムに基づきコンテキストの探索を繰り返すことで最小コストで目的のコンテキストに到達するパス（操作列）を導出する。

3.3 デバイス操作列変換プログラム

PathSim によって導出された目標の温湿度へ到達する最小コストのデバイス操作列は、デバイス操作列変換プログラムによって、以下に示す家電制御言語に基づいた家電制御スクリプトに変換する。本論文で示す家電制御言語は、本システムの実現にあたって新たに提案するものであり、システムの導入環境に応じてデバイスの種類の追加等、文法の拡張が容易にできるものとなっている。

- 1: $Path ::= EE"; Path | EI"; Path | NE"; Path | "stop"$
- 2: $EDE ::= "op(" Device"; " ActionList")"$
- 3: $ETE ::= "check(" RangeList"; " T"; " a")"$
- 4: $Device ::= "ac" | "hm" | "dh" | "ht" | \dots$
- 5: $ActionList ::= Action | Action"; " ActionList$
- 6: $Action ::= ON | "off" | "tmp:=" Val | "hmd:=" Val | \dots$
- 7: $ON ::= "on" | "on(" Mode")"$
- 8: $Mode ::= "cl" | "dh" | "ht" | \dots$
- 9: $RangeList ::= Range | Range"; " RangeList$
- 10: $Range ::= Type "<=" Val | Val "<=" Type$
- 11: $Type ::= "tmp" | "hmd" | "illum" | \dots$

- 12: $T ::= (integer\ number)$
- 13: $a ::= (integer\ number)$
- 14: $Val ::= (real\ number)$

家電制御言語では”（ダブルクォーテーション）で囲まれている単語を終端記号、そうでないものを非終端記号として記述している。まず、あるコンテキスト間の遷移を表すデバイス操作列 Path は環境デバイスイベント (EDE) と環境遷移イベント (ETE) の2種類のイベントの繰返しで構成されており、“stop” が出現するまでそれぞれのイベントの操作を行っていく。EDE はそれぞれの家電の制御を行うイベントであり、操作を行う家電デバイスの種類 (Device) とそのデバイスの操作 (Action) で構成される。Device は、スマートホーム内に存在するエアコンや加湿器、除湿機といった家電デバイスの種類が定義されており、Action には on や off、エアコンならば冷房や暖房、除湿といった家電デバイスの操作のモードが定義されている。また、ETE はコンテキストが遷移したかどうかを確かめるために環境の観測を行うイベントであり、観測する値の種類 (Type)、値の範囲 (Range) と観測を行うタイミング (T, α) で構成されている。EDE が読み込まれ、家電に信号が送られてから、T 秒後に Type の値が Range の範囲内にあるかどうかを観測し、範囲内であれば次のイベントに移り、範囲外であれば α 秒後にもう一度観測を行い、それを繰返す。また、所定回数の繰返しでも、Range の範囲に到達できなかった場合、エラーを出力して停止する。以下に簡単なスクリプト例を示す。

- 1: $op(ac, on(mode(cl)));$ (1)
- 2: $check(temp, 22 \leq x \leq 24, 60, 10);$ (2)
- 3: $op(dh, on);$ (3)
- 4: $check(humid, 52.5 \leq x \leq 55, 60, 10);$ (4)
- 5: $stop$ (5)

(1), (3) は EDE で、(2), (4) は ETE となっている。まず、(1) のイベントは、Device が ac、Action が on(mode(cl)) であるので、エアコンを冷房モードで稼働させることを表す。(2) は、(1) のイベントから 60 秒後に温度が 22°C から 24°C の間に遷移しているかどうかを観測し、なければ ld、その後 10 秒ごとに観測を続ける。(2) の観測後、(3) で加湿器を稼働させ、(4) で、(3) のイベント後から 60 秒後の湿度が 52.5% から 55.0% の間にあるかどうかを観測し、条件を満たせば stop を読み込み終了する。

3.4 家電操作統合システム

家電制御スクリプトに変換されたデバイス操作列は、家電制御スクリプト実行プログラムによって家電操作統合システム (図 2) に送信され、システムが各家電に制御信号を送信することでデバイス操作列の家電制御を実行する。

家電操作統合システムは、IRKit [6] を用いて構築している。IRKit は WiFi 機能の付いたオープンソースな赤外線リモコン学習デバイスで、エアコンやテレビ、ライト等、赤外線で作動できる家電を、WiFi を通じて操作することができるようにするものである。開発用 API が公開されており、家電の赤外線リモコンの赤外線信号の学習や、学習した信号の家電への送信等が可能である。本システムでは、複数の IRKit と、複数の家電の複数のモードを扱うた

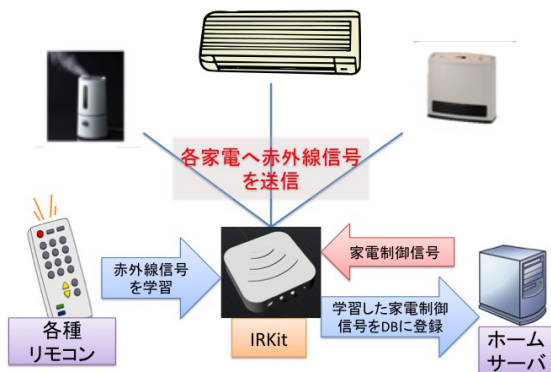


図 2 家電操作統合システム
Fig. 2 Appliance control system.

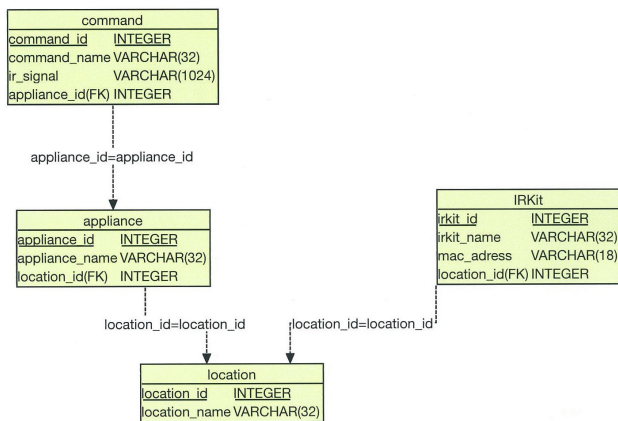


図 3 家電制御信号統合データベース ER 図
Fig. 3 ER model of appliance control signal database.

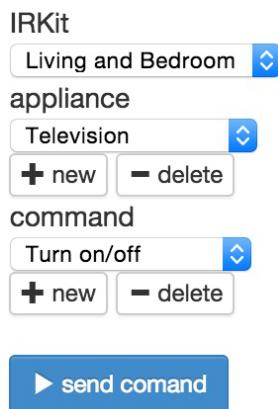


図 4 データベース登録用 WebClient イメージ
Fig. 4 WebClient.

め、図 3 の ER 図に示す家電制御信号統合データベースを構築している。

location テーブルはデバイスの場所の情報を location_id, location_name として保持しており、location_id に appliance テーブルと IRKit テーブルが紐付けられている。appliance テーブルは location_id に対して家電デバイスの情報を appliance_id, appliance_name として保持し、IRKit テーブルは、location_id に対して IRKit デバイスの情報を irkit_id, irkit_name, mac_adress として保持している。さらに、application_id に command テーブルが紐付けられており、家電デバイスの複数のモードとそれに対する赤外線信号を command_id, command_name, ir_signal として保持している。本研究では、データベースへの家電の追加と削除、リモコンの赤外線信号の追加と削除の簡易化のため、図 4 に示す WebClient を作成している。これを用いれば、IRKit デバイスと家電デバイスの登録、リモコンの赤外線信号の学習と登録と削除をブラウザ上で手軽に行うことが可能である。

4. 家電制御システムの評価実験

本章では、実際のスマートホーム設備を用いて、本システムの動作確認を行う。また、PathSim がシミュレーションによって導出した消費電力量および遷移時間と、本システムを動作させたときにかかった消費電力量および遷移時間を比較し、その誤差を調査する。これにより、PathSim を含めた提案システムの有用性を評価する。

4.1 実験環境

実験で用いるスマートホーム設備として、図 5 に示すスマートホーム間取り図内の赤色の線で囲まれたエリア (W: 6,100 cm, D: 6,410 cm, H: 2,350 cm) を対象とする。このエリアは、2 部屋間の引き戸を取り外し 1 部屋として扱っている。木製の壁と天井の熱伝導率を熱伝導する面積、壁と天井の厚さ、材質の熱伝導率から概算し 0.15 W/(m・K) に、ガラス窓の熱伝導率を同様に概算し 0.2 W/(m・K) と

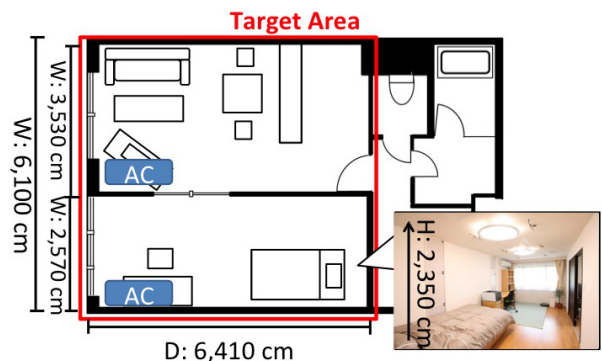


図 5 対象とするスマートホーム
Fig. 5 Setting of smart home.

した。また、流入出する水蒸気量を計算するために、予備実験として図 5 の対象空間において湿度の変化量を実際に計測し、1 秒あたりの換気量 $Vol_{ven} = 0.0762$ を導出した。

使用するデバイスとして、図中の位置に設置された 2 台のエアコン（東芝製 RAS-281EDR）を同時に使用する。エアコンは、冷房、暖房、ドライ（除湿）という 3 つのモードを備えており、それぞれの性能は表 1 に示すとおりである。

4.2 実験手法

本実験では、PathSim がシミュレーションにより導出した消費電力量とコンテキスト遷移時間と、実環境で稼働させたときの消費電力量とコンテキスト遷移時間の比較を行うため、以下の手順で本システムの実証を行った。本実験は、外気温 25.6°C、湿度 68% の環境下で行った。

- (1) 部屋を換気し、室内の温湿度を外気の温湿度にあわせる。
- (2) 窓とカーテンを閉め、システムを実行し、現在の室内の温湿度から目標の温湿度までの遷移を行い、消費電力量とコンテキスト遷移時間を計測する。今回の実験で設定した目標の温湿度は温度 20°C から 21°C、湿度 52.5% から 55.0% である。
- (3) 以上の手順で 3 回の計測を行い、消費電力量とコンテ

表 1 実験で使用するデバイスの設定

Table 1 Setting of the appliance to use in the experiment.

	消費電力	W: 冷/暖房能力, L: 除湿量
エアコン RAS-281EDR	COLD: 990 W, HOT: 1225 W, DH: 155 W	COLD: 3.4kW, HOT: 4.7kW, DH: 1.5L/h

表 2 目標の温湿度に到達するまでに経由したコンテキスト

Table 2 All context in the transition of temperature and humidity.

コンテキスト	温度 (°C)	湿度 (/%)	デバイス操作
a	25.5	67.5	$A^{AC}(1) = ON, A^{AC}(2) = Cold$
b	20.5	76.9	$A^{AC}(1) = ON, A^{AC}(2) = Dehumidifier$
c	21.4	53.8	$A^{AC}(1) = ON, A^{AC}(2) = Cold$
d	20.6	58.8	$A^{AC}(1) = ON, A^{AC}(2) = Dehumidifier$
e	20.9	53.8	$A^{AC}(1) = OFF$

表 3 実験結果

Table 3 The result of experiment.

		コンテキスト遷移				合計消費電力量
		a-b	b-c	c-d	d-e	
シミュレーション値	消費電力量 (Wh)	199.2	15.5	38.3	4.2	257.2
	遷移時間 (s)	998	494	192	128	1812
実測値	消費電力量 (Wh)	207.4	22.0	41.1	4.5	278.0
	遷移時間 (s)	998	674	192	128	1992
誤差	消費電力量	+5.3%	+29.5%	+6.8%	+6.7%	+7.5%
	遷移時間	0%	+26.7%	0%	0%	+9.0%

キスト遷移時間の平均値を求める。

4.3 実験結果

以下では、コンテキスト x からコンテキスト y への遷移を x-y と表す。この実験で導出されたデバイス操作列により目標の温湿度に到達するまでに遷移したコンテキストを表 2 に、コンテキスト遷移を図 6 に示す。本実験では、2 台のエアコンはすべて同じ操作が行われたため、表 2 で、エアコン 2 台をまとめて AC と表している。また、実験の結果を表 3 に示す。

表 3 の合計消費電力量を見ると、PathSim のシミュレーションによる値と比べ 10% 以下の誤差で収まっていることが分かる。それぞれのコンテキスト遷移における結果を比較すると、コンテキスト遷移 b-c における誤差が大きくなっており、その他の遷移は消費電力量の誤差はあるものの遷移時間に関してはシミュレーションと等しくなってい

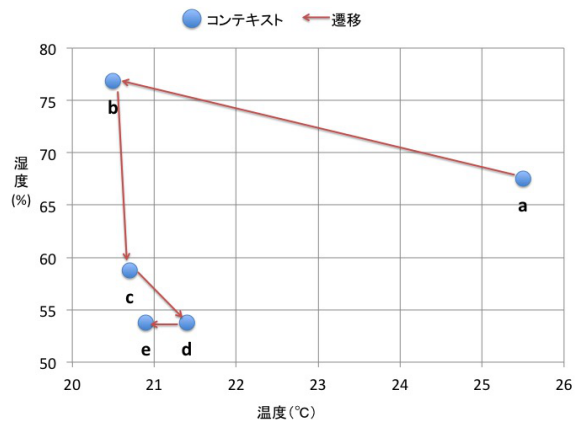


図 6 コンテキスト遷移
Fig. 6 Transition of context.

る。表 2 によるとコンテキスト遷移 b-c においては、エアコンを除湿モードで稼働させていることが分かる。よって、除湿による湿度変化がしにくかったと考えられる。すべてのコンテキスト遷移において、シミュレーションより、実測値の方が消費電力量が大きくなっているのは、家電自体の劣化や汚れ等によりカタログスペックから導出した消費電力モデルと実際の家電との性相違原因だと考えられる。また、湿度の変化に関しては、室内のエリアごとに湿度の違いにより、計測位置の湿度が反映されにくかった等の原因が考えられる。今後、温湿度を正確に計測する手法についても検討が必要である。

5. まとめ

本論文では、著者らが提案したスマートホーム内のコンテキスト遷移を最小コストで行うデバイス操作列導出ツール PathSim を用い、目標温湿度環境への遷移を最小コストで実現する家電制御システムを提案した。本システムは PathSim が導出したデバイス操作列を、デバイス操作列変換プログラムによって家電制御言語に基づいた家電制御スクリプトに変換し、家電制御スクリプト実行プログラムによって IRKit を用いた家電操作統合システムに送信することで、目的の温湿度への最小コストでの遷移を実現している。また、本システムは、温湿度センサ、ホームサーバ、IRKit、赤外線リモコン対応家電があれば実現可能であるので、一般家庭にも容易に導入が可能である。実際のスマートホーム環境を用いて行った実証実験により、本システムが PathSim がシミュレーションによって導出した消費電力量、遷移時間と比べ、10%以下の誤差で目標温湿度への遷移ができていることが分かった。著者らは文献 [5] において、PathSim によって導出されたデバイス操作列が、人が直感的な方法で家電を操作した場合と比べて目標温湿度への遷移にかかる合計消費電力量を最大で約 45%削減できていることを示しており、実証実験の結果の誤差を考慮したとしても、本システムは十分実用的であるといえる。

今後の課題としては、PathSim のシミュレーションによる値と本システムとの誤差の要因の調査、ユーザがより扱いやすくなるようなユーザインタフェースの構築、赤外線リモコン対応家電デバイスを増やした環境で、様々な季節での実証実験を行っていく必要があると考えている。

参考文献

- [1] Weiser, M.: Some computer science issues in ubiquitous computing, *Comm. ACM*, Vol.36, No.7, pp.75-84 (1993).
- [2] Henriksen, K. and Indulska, J.: Developing context-aware pervasive computing applications: Models and approach, *Pervasive and mobile computing*, Vol.2, No.1, pp.37-64 (2006).
- [3] Baldauf, M., Dustdar, S. and Rosenberg, F.: A survey on context-aware systems, *International Journal of Ad*

Hoc and Ubiquitous Computing, Vol.2, No.4, pp.263-277 (2007).

- [4] Edwards, W.K. and Grinter, R.E.: At home with ubiquitous computing: Seven challenges, *UbiComp 2001: Ubiquitous Computing*, pp.256-272, Springer (2001).
- [5] Mizumoto, T., El-Fakih, K. and Yasumoto, K.: PathSim: A Tool for Finding Minimal Energy Device Operation Sequence for Reaching a Target Context in a Smart-Home, *Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC)*, pp.64-71, IEEE (2013).
- [6] Ohtsuka, M.: IRKit — opensource infrared remote controller (2014).
- [7] Cook, D.J., Youngblood, M., Heierman III, E.O., Gopalratnam, K., Rao, S., Litvin, A. and Khawaja, F.: MavHome: An agent-based smart home, *2013 IEEE international conference on pervasive computing and communications (PerCom)*, pp.521-521, IEEE Computer Society (2003).
- [8] A system for collecting activity annotations for home energy management, *Pervasive and Mobile Computing*, Vol.15, pp.153-165 (2014).
- [9] Kashimoto, Y., Ogura, K., Yamamoto, S., Yasumoto, K. and Ito, M.: Saving energy in smart homes with minimal comfort level reduction, *Workshop Proc. IEEE PerCom 2013*, pp.372-376 (2013).
- [10] Hayashi, H., Tsukamoto, Y. and Mochizuki, S.: HEMS (ホームエネルギーマネジメントシステム), *The journal of the Institute of Electrical Engineers of Japan*, Vol.132, No.10, pp.695-697 (2012).
- [11] Iwafune, Y.: Home Energy Management System (Japanese Title: これからの HEMS), *Journal of The Institute of Electrical Engineers of Japan*, Vol.133, pp.809-812 (2013).



宵 憲治 (学生会員)

1991 年生。2014 年大阪大学基礎工学部情報科学科卒業。同年奈良先端科学技術大学院大学情報科学研究科博士前期課程入学。現在、同大学院博士前期課程。ユビキタスコンピューティングに関する研究に従事。



Khaled El-Fakih

is an associate professor of Computer Science and Engineering at the American University of Sharjah, UAE. He received his PhD degree in computer science from the University of Ottawa, Canada, in 2002. He spent one year's sabbatical leave at Verimag Laboratory, University Joseph Fourier, France. He was a visiting researcher at Osaka University, University of Ottawa, University of Verona, University of Sao Paulo, Nara Institute of Science and Technology, and Tomsk State University. He acted as a program co-chair of the 2008 International Conference on Formal Techniques for Networked and Distributed Systems and of the 2015 International Conference on Testing Software and Systems. He conducted research in software testing, synthesis of distributed systems, optimization and application of genetic algorithms, and he published many papers in these areas.



安本 慶一 (正会員)

1991年大阪大学基礎工学部情報工学科卒業。1995年同大学大学院博士後期課程退学後、滋賀大学経済学部助手。2002年より奈良先端科学技術大学院大学情報科学研究科助教授。博士(工学)。現在、同校同研究科教授。モバイル、ユビキタスコンピューティングに関する研究に従事。ACM, IEEE 各会員。