

スマートスペースにおける最小コストでのコンテキスト遷移を可能にするデバイス操作系列導出ツールの提案

水本 旭洋¹ Khaled El-Fakih² 安本 慶一¹

概要：本論文では、スマートハウスにおいて省エネルギーに様々なデバイスをコンテキストウェア制御することを目的に、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デバイス操作シーケンスを導出するツール *PathSim* を提案する。*PathSim* は、スマートスペースにおけるコンテキストの各構成要素の値域を有限個の範囲に分割することで、コンテキストを有限個の要素からなる空間として扱う。そして、各コンテキストにおいて任意のデバイスの操作イベントを実行した時に遷移するコンテキストをシミュレーションにより求めていくことで、最小コストで目的のコンテキストに遷移するデバイスの操作シーケンスを A* アルゴリズムに基づいて導出する。スマートスペースにおける典型的な要求仕様を想定したケーススタディを通して評価実験を行った結果、*PathSim* によって導出された操作シーケンスが、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 72% 削減できることを確認した。

1. はじめに

近年、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ユビキ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] を実現する多くの研究 [3] が行われている。これらのシステムが実装された空間や家はスマートスペース（またはスマートホーム）[4] と呼ばれている。スマートスペースには、温度や湿度、ユーザの位置情報や行動などの状況をセンシングするセンサデバイスやネットワークから制御可能な家電デバイスなどが分散して設置されている。これらの物理的特性、ユーザの状況、そして家電デバイスの状態などのそれぞれを変数として表したとき、これら変数の値の組はスマートスペースの **コンテキスト** と呼ばれる。スマートスペースがユーザに適切なサービスを提供できるように、ユーザは自分の好みに合わせた要求仕様を設定する必要がある。ユーザの要求はコンテキストを構成する各変数の値の範囲を用いて定義される。要求仕様は、システムが動作するトリガーとなる条件（以下、**事前条件**と表す）とシステムの動作によって提供または維持される状態の条件（以下、**事後条件**と表す）の組（性質）で与えられる。要求仕様に基づいたサービスを実現するために、スマートスペースは、事前条件を満たすコンテキストから、事後条件を満たすコンテキストに到

達するようにデバイスを操作する必要がある。一般的に、複数のデバイスがスマートスペース内に設置されている場合、あるコンテキストから異なるコンテキストにスマートスペースを遷移させるデバイスの操作シーケンスは多数存在する。そして、シーケンスごとに操作するデバイスや操作する順番などが異なるため、消費電力量や動作時間がそれぞれ異なる。そこで、膨大なシーケンスの中から最適または準最適なコストのシーケンスを自動的に見つけることが出来れば、ユーザにとって非常に有益だと考えられる。

本論文では、省エネルギーにコンテキストウェアなデバイス操作を実現することを目的とし、スマートスペースにおいて、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シーケンスを導出するツール *PathSim* を提案する。*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シーケンスを導出する。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取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）の集合が予め与えられる必要があるが、

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

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

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。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（イベントと呼ぶ）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（操作シーケンス）を導出する。

典型的なスマートハウスの要求仕様を考慮したケーススタディにより、PathSim の有用性を評価した。結果、PathSim で導出したシーケンスは、直観的な手法でコンテキストを遷移させる場合と比べ、消費電力量が 72% 少ないことを確認した。

2. スマートスペースにおける最小コストデバイス操作問題

本章では、想定するスマートスペースおよび要求仕様の定義を行う。そして、任意のコンテキストから特定のコンテキストへスマートスペースを遷移させるための最小コストのデバイス操作シーケンスを導出する問題を定義する。

2.1 スマートスペース

2.1.1 コンテキスト

スマートスペースは 1 つまたは複数の（有限個の）物理的な位置（または部屋）の集合 $R = \{r_1, \dots, r_k\}$ を持つとする。スマートスペースには、テレビ、エアコン、加湿器など様々なデバイスが設置されている。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。例えば、テレビの場合、電源状態 (Status), チャンネル (Current-Channel), 音量 (Volume-Level) などの属性を持ち、それぞれ $A^{TV}(1) = \text{Status}$, $A^{TV}(2) = \text{Current-Channel}$, and $A^{TV}(3) = \text{Volume-Level}$ 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 $\text{Dom}(A^{d_i}(j))$ を持つ。例えば、テレビの属性は、 $\text{Dom}(A^{TV}(1)) = \{\text{ON}, \text{OFF}\}$, $\text{Dom}(A^{TV}(2)) = \{1, \dots, 300\}$, and $\text{Dom}(A^{TV}(3)) = \{1, \dots, 100\}$ のような定義域をそれぞれ持つ。デバイス d_i の現在の状態 c は、デバイスの属性値のベクトル $\mathbf{v}_c^{d_i} = \langle \mathbf{v}_c^{d_i}(1), \dots, \mathbf{v}_c^{d_i}(n) \rangle$ ($\mathbf{v}_c^{d_i}(j) \in \text{Dom}(A^{d_i}(j))$, $j = 1, \dots, n$) で表される。例えば、テレビの電源状態が “ON”, チャンネルが 10ch, 音量が 30 の場合、 $\mathbf{v}_c^{TV} = \langle \mathbf{v}_c^{TV}(1) = \text{ON}, \mathbf{v}_c^{TV}(2) = 10, \mathbf{v}_c^{TV}(3) = 30 \rangle$ のように表すことができる。スマートスペース内に設置されている m 個のデバイスの現在の状態は、 $\langle \mathbf{v}_c^{d_1}, \dots, \mathbf{v}_c^{d_m} \rangle$ で表される。

加えて、スマートスペースに l 人のユーザ $u_1, \dots, u_l$ がい

るものとする。スマートスペース内のユーザの位置は、部屋単位または空間単位で表す。すなわち、各ユーザ u_i は、 R の要素のいずれかを $u_i.\text{pos}$ に保持する。また、ユーザの行動などの他の属性は $u_i.\text{attr}$ として保持されるとする。ユーザ u_i がスマートスペース内に居ない場合は、 $u_i.\text{pos}$ と $u_i.\text{attr}$ は、 null である。ユーザ u_i の現在の状態も、 $\mathbf{v}_c^{u_i} = \langle u_i.\text{pos}, u_i.\text{attr} \rangle$ のように属性値のベクトルとして表し、 l 人のユーザに関する状態を、 $\langle \mathbf{v}_c^{u_1}, \dots, \mathbf{v}_c^{u_l} \rangle$ と表す。

さらに、スマートスペースは、各位置 $r_1, \dots, r_k$ ごとに環境に関する属性を持つものとする。これらの属性として、温度、湿度、照度、騒音レベルなどが考えられるが、本論文では、簡単化のため、温度と湿度のみを扱い、位置 r_i の温度を $r_i.\text{temp}$, 湿度を $r_i.\text{humidity}$ 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。位置 r_i の現在の状態を、 $\mathbf{v}_c^{r_i} = \langle r_i.\text{temp}, r_i.\text{humidity} \rangle$ のように表し、 k 個の位置に分割されたスマートスペース内の現在の環境を、 $\langle \mathbf{v}_c^{r_1}, \dots, \mathbf{v}_c^{r_k} \rangle$ という属性値ベクトルで表す。

以上のことを踏まえて、 m 個のデバイス、 l 人のユーザ、 k ヶ所の位置の状態を保持するスマートスペースの状態（コンテキスト） c を、 $c = \langle \mathbf{v}_c^{d_1}, \dots, \mathbf{v}_c^{d_m}, \mathbf{v}_c^{u_1}, \dots, \mathbf{v}_c^{u_l}, \mathbf{v}_c^{r_1}, \dots, \mathbf{v}_c^{r_k} \rangle$ のように表す。

2.1.2 コンテキスト遷移

スマートスペース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、ノードはスマートスペースのコンテキスト、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、ユーザの位置、環境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。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。例えば、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。同様にユーザの位置に関してもスマートスペースの空間全体を適切に分割することで、有限個の部分空間の集合 R として扱う。

本研究では、**非環境イベント**、**環境デバイスイベント**、**環境遷移イベント**という 3 種類のイベントを考える。非環境イベントは、テレビのようなスマートスペース内の環境に影響を与えないデバイスの属性に値を設定するイベントである。環境デバイスイベントは、エアコンや加湿器といったスマートスペース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。

デバイスの属性を直接設定できる非環境/環境デバイスイベントを、値 $VALUE$ とデバイス d_i の属性 j を用いて、 $E\{A^{d_i}(j) \leftarrow VALUE\}$ と表す。また、値を直接設定できない環境遷移イベントを、遷移前のコンテキスト c と遷移先のコンテキスト c' を用いて、 $E\{c \rightarrow c'\}$ と表す。

2.2 要求仕様

与えられたスマートスペースで満たされるべき性質の集合を要求仕様と呼び、 $S = \{pr_1, \dots, pr_h\}$ と表す。また、各性質 pr_i を事前条件と事後条件の組として以下のように表す。

$$pr_i = (Pre_i, Post_i) \quad (1)$$

ここで、 Pre_i と $Post_i$ はコンテキストを構成する変数（属性）および定数、論理演算子、 $>, \geq, \leq, <, =$ のような比較演算子からなる線形不等式や集合論の記号 $\in$ などを用いた式で記述されるものとする。例えば、“Alice がリビングルームにいる時、リビングルームの温度と湿度は、それぞれ 27℃ と 60% であるべき” という性質は、以下のように記述することができる。

$$Pre : u_{Alice}.pos = r_{Living}$$

$$Post : r_{Living}.temp \in \{[26, 27], [27, 28]\}$$

$$\wedge r_{Living}.humidity \in \{[59, 60], [60, 61]\} \quad (2)$$

ここで、温度と湿度のレンジは、それぞれ摂氏 1℃、相対湿度 1% で分割されている。

2.3 問題定義

ここでは、あるコンテキストから任意の別のコンテキストに最小コストで遷移できるイベントシーケンスを導出する問題を定義する。

イベント e によるコンテキスト c から c' への遷移を $c \xrightarrow{e} c'$ と表す。 $path(c_s, c_t) = \langle e_1, \dots, e_n \rangle$ は、ソースコンテキスト c_s から対象コンテキスト c_t にスマートスペースを遷移させるイベントシーケンスを表す。 $c_1, \dots, c_{n+1}$ は、 c_s と c_t の間で経由する中間コンテキストを表す。ここで、 $c_1 = c_s$ 、 $c_{n+1} = c_t$ として扱う。 c_s から c_t への遷移可能なイベントシーケンスの条件を以下に記す。

$$\begin{aligned} & \exists c_2, \dots, \exists c_n, \exists e_1, \dots, \exists e_n \\ & c_s \xrightarrow{e_1} c_2 \wedge c_2 \xrightarrow{e_2} c_3 \wedge \dots \wedge c_n \xrightarrow{e_n} c_t \\ & \wedge e_1 \in E(c_s) \wedge \dots \wedge e_n \in E(c_n) \end{aligned} \quad (3)$$

ここで、 $E(c)$ は、コンテキスト c で実行可能なイベントの集合を表す。

本論文では、コンテキスト遷移に必要な消費電力量の最小化を目的にする。そのため、 $e.energy$ をイベント e の消費電力量とすると、目的関数は以下のように表される。

$$\text{minimize } \sum_{i=1}^n e_i.energy \quad \text{subject to (3)} \quad (4)$$

3. PathSim : 最小コストデバイス操作シーケンス導出ツール

本章では、ソースコンテキスト c_s から対象コンテキスト c_t にスマートスペースを遷移させる最小コストのデバイス操作シーケンス（イベントシーケンス）を導出する PathSim の詳細について述べる。

PathSim は、与えられたコンテキストから対象コンテキストに最小コストで到達できるイベントシーケンスを導出するために、まず、現在のコンテキスト c_s からシーケンスの順番を考慮する必要の無い非環境イベントを全て反映した後のコンテキスト c_r までのイベントのシーケンス $non_env_path(c_s, c_r)$ （以下、非環境イベントシーケンスと表す）を導出する。その後、環境イベントのみによって構成される c_r から対象コンテキスト c_t へのイベントシーケンス $env_path(c_r, c_t)$ （以下、環境イベントシーケンスと表す）を導出する。PathSim は、それら 2 つのシーケンスを統合することで、 c_s から c_t に最小コストで遷移可能なパス $path(c_s, c_t)$ を導出する。以下の節で、非環境イベントシーケンスと環境イベントシーケンスの導出手法の詳細を述べる。

3.1 非環境イベントシーケンスの導出法

PathSim は、 c_s と c_t の非環境デバイスの各属性を比較し、その差に基づいて、非環境イベントシーケンス $non_env_path(c_s, c_r)$ を導出する。PathSim は、最初に各非環境デバイス $d_i \in D$ について、属性値ベクトル $\mathbf{v}_{c_s}^{d_i}$ と $\mathbf{v}_{c_t}^{d_i}$ を比較する。 j 番目の属性に差がある場合、すなわち、 $\mathbf{v}_{c_s}^{d_i}(j) \neq \mathbf{v}_{c_t}^{d_i}(j)$ である場合には、イベント $E\{A^{d_i}(j) \leftarrow \mathbf{v}_{c_t}^{d_i}(j)\}$ をイベントシーケンスに追加する。

3.2 環境イベントシーケンスの導出法

スマートスペースのコンテキストの遷移は、コンテキストをノード、有向辺をイベントとコストとした有向グラフで表すことができる。グラフの 2 つのノード間の最小コストパスを見つける問題は最短経路問題と呼ばれ、その効率の良い解法として A* アルゴリズム [5] が知られている。提案するアルゴリズムでは、A* アルゴリズムで使用する有向グラフにおける各辺の存在が予め分かっているということを考慮して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のエッジとコストを動的に調べるように A* アルゴリズムを改変する。また、コスト関数として本研究に適したものを採用する。

環境イベントに関するパス導出アルゴリズムは、(a) イベントジェネレータ、(b) コンテキスト遷移シミュレータ、(c) 2 種類のコスト関数、(d) アルゴリズムのメイン関数である PathFinder、という 5 つの関数で構成される。以下でそれぞれの関数について述べる。

3.2.1 イベントジェネレータ

この関数は、任意のコンテキスト c において実行可能なイベントの集合を生成する関数である。単に c において実行可能なイベントを全て生成する方法では、解空間と計算時間が肥大化してしまう。そのため、対象コンテキスト c_t に到達させるために、 c において有効なイベントのみを生成するようにする。例えば c より c_t の方が温度のレンジが高い場合には、ヒーターの電源を入れたり、エアコンのモードを暖房にするイベントが生成される。

3.2.2 コンテキスト遷移シミュレータ

この関数は、現在のコンテキスト c とイベント e が与えられた時、 c において e を実行することで到達する隣接コンテキスト c' を出力する。イベント e が環境デバイスイベントである場合には、コンテキスト c' は e に関連した属性以外は c と同じ属性値を持つ。一方、 e が環境遷移イベントである場合には、スマートスペース内の温度と湿度の変化をシミュレーションし、 c の温度か湿度が隣接するレンジに遷移した後のコンテキスト c' を導出する。

シミュレータは空間内の温度 t_{mp} が隣接する温度レンジの中央値 $t_{mp_{nei}}$ に到達するまでの時間 $T_{tmp \rightarrow tmp_{nei}}$ と、湿度 hmd が隣接する湿度レンジの中央値 hmd_{nei} に到達するまでの時間 $T_{hmd \rightarrow hmd_{nei}}$ をスマートスペースの容積（サイズ）、壁の熱伝導率、外気温、デバイスが発する熱や水蒸気などを考慮して計算する。

$T_{tmp \rightarrow tmp_{nei}}$ と $T_{hmd \rightarrow hmd_{nei}}$ を計算した後、両方の値を比較する。この時、 $T_{tmp \rightarrow tmp_{nei}} < T_{hmd \rightarrow hmd_{nei}}$ である場合には、温度が湿度より先に隣接するレンジに到達するため、シミュレータは以下の式を用いて現在の湿度 hmd の値を更新する。

$$hmd \leftarrow hmd + \frac{((M_{dev} + M_{ext}) \times T_{tmp \rightarrow tmp_{nei}})}{Vol} \quad (5)$$

ここで、 Vol はスマートスペースの容積を表し、 M_{dev} は c において稼働中のデバイスが発する水蒸気量、そして、 M_{ext} は外部の空気によって流出（または流入）する水蒸気量を表している。

一方、 $T_{tmp \rightarrow tmp_{nei}} > T_{hmd \rightarrow hmd_{nei}}$ である場合には、シミュレータは以下の式を使用して現在の温度 t_{mp} を更新する。

$$t_{mp} \leftarrow t_{mp} + \frac{((Q_{dev} + Q_{ext}) \times T_{hmd \rightarrow hmd_{nei}})}{C} \quad (6)$$

ここで、 C は熱容量を表し、 Q_{dev} は c において稼働中のデバイスが発する熱量を、そして、 Q_{ext} は外気温から壁を通して伝わる熱量である。更新された t_{mp} と hmd の値は、到達したコンテキスト c' に、 $c'.t_{mp}$ と $c'.hmd$ としてそれぞれ保存される。

その後、シミュレータは、以下の式を用いて、時間 T と動作中の全デバイスの消費電力量を乗算することでコストを計算する。そして、計算したコストを $e.cost$ に保持する。

ここで T は、 $T_{tmp \rightarrow tmp_{nei}}$ と $T_{hmd \rightarrow hmd_{nei}}$ のうち小さい方の時間であり、 $power(d)$ は動作中のデバイス d の消費電力を表す。

$$e.cost = T \times \sum_{d_i \in D} power(d_i) \quad (7)$$

3.2.3 コスト関数

各シーケンスのコストを評価するために、 $ActCost(c, c')$ と $EstCost(c, c_t)$ という2つのコスト関数を使用する。 $ActCost(c, c')$ はコンテキスト c から c' に遷移する際の、イベントの合計コストを計算する関数である。前節で説明したように、コンテキスト遷移シミュレータを用いて、コンテキストの遷移に掛かるコストを計算できる。そのため、 c から c' までのパスに含まれる全てのイベントのコストを足し合わせることでパスのコストを計算できる。 $EstCost(c, c_t)$ はコンテキスト c から対象コンテキスト c_t へのコストを推定する。 $EstCost()$ は、 c から c_t への遷移コストを、 c と c_t の温度と湿度の差を考慮して、 c から有効な全てのデバイスを動作させた場合のデバイスが発する熱量 Q_{dev} と水蒸気量 M_{dev} を用いて、 c_t に遷移する時間および消費電力を計算することでコストの推定を行う。

3.2.4 PathFinder：環境イベントシーケンス導出関数

PathSim のメイン関数である PathFinder は、コンテキスト c_r から c_t への最小コストパス（イベントシーケンス）を A* アルゴリズムに基づいて導出する。

PathFinder によって導出される各ノードは、それぞれ、コンテキスト $v.c$ 、親ノード $v.p$ 、 $v.p$ から $v.c$ に遷移するために実行されたイベント $v.e$ 、 $v.p$ から $v.c$ に遷移するために掛かるコスト $v.cost$ 、そして v_r から v までのコストと v から v_t への推定コストから計算される合計コスト $v.cost_{total}$ を保持する。 v_{cur} を現在のノード、 v_{nei} を v_{cur} からイベント $v_{nei}.e$ によって到達するノード、そして、 v_{old} を v_{nei} より先に導出された同じコンテキストを持つノードと定義する。また、 v_r と v_t を、それぞれソースコンテキスト c_r を持つソースノードと対象コンテキストを持つ c_t を持つ対象ノードとして定義する。

PathFinder は、まず、現在のノードが保持するコンテキスト $v_{cur}.c$ において実行可能な環境イベントの集合 E を生成する。集合 E には環境遷移イベントも含まれており、このイベントは、現在のコンテキスト $v_{cur}.c$ から、温度と（または）湿度のレンジが異なる隣接コンテキスト $v_{nei}.c$ にスマートスペースが遷移することを表す。各環境イベントに対して、PathFinder はイベントを実行した後の隣接コンテキスト v_{nei} を導出する。その後、 $v_s.c$ から $v_{nei}.c$ に到達するためのコストを $ActCost(v_s, c_{nei})$ 関数を用いて計算し、 $v_{nei}.c$ から $v_t.c$ に到達するための推定コストを $EstCost(v_{nei}, v_t)$ 関数を用いて計算する。PathFinder は、計算された2つのコストを合計することで、 $v_s.c$ から $v_t.c$

に $v_{nei,c}$ を経由して到達するための合計コストを決定する。PathFinder は、最小合計コストを持つノードを v_{cur} として再帰的に選択し、選択された v_{cur} ごとに隣接ノードを導出する。この処理を v_t が v_{cur} として選択されるまで行う。

3.3 非環境/環境イベントシーケンスの結合

$non_env_path(c_s, c_r)$ と $end_path(c_r, c_t)$ という 2 つのシーケンスを導出した後に、 $non_end_path(c_s, c_r)$ の後に $env_path(c_r, c_t)$ を結合することでデバイス操作シーケンス $path(c_s, c_t)$ を獲得できる。

4. ケーススタディによる評価

本章では、様々なスマートホームの要求仕様とそれらを満たす多くのソース/対象コンテキストを考慮したケーススタディにより評価を行う。

4.1 実験設定

以下に、本ケーススタディで使用するスマートホームのデバイス、要求仕様、実験方法を記す。

4.1.1 対象とするスマートホームおよび使用するデバイス

本ケーススタディでは、図 1 で表したスマートホームのフロアプランを想定する。簡単化のため、本ケーススタディでは図内の赤色の線で囲まれた 1 つの部屋のみを想定し、壁と天井の熱伝導率を $0.15 W/(m \cdot K)$ 、ガラス窓の熱伝導率を $0.2 W/(m \cdot K)$ とした。また、外気温を日本の 8 月の平均気温である $27^\circ C$ に固定した。

使用するデバイスとして、エアコンとヒーターをそれぞれ 1 台ずつ、そして、加湿器を 2 台使用する。エアコンは東芝製 RAS-281EDR、ヒーターは De'Longhi 製 TDD0915W を想定し、加湿器は ALCOLLE 製 ASH-600 と Panasonic 製 F-VXF45 という性能の異なる 2 つの加湿器を想定した。これらのデバイスは、目標の温度/湿度に到達するまで最大出力で動作すると仮定する。また、これらのデバイスの消費電力と性能をカタログスペックに基づいて設定する。

4.1.2 要求仕様およびソース/対象コンテキスト

本ケーススタディでは、表 1 に記述した prA, prB, prC, prD という 4 つの性質を含んだ要求仕様を使用する。性質の事前条件および事後条件を満たすコンテキストは多く存在する。そのため、本実験では各事後条件に 1 つの対象コンテキストを生成し、対象コンテキストの温度と湿度のレンジを基準に、 k レンジずつ離れたソースコンテキストを生成する。この際、(1) 温度と湿度が高いコンテキスト、(2) 温度が高く、湿度が低いコンテキスト、(3) 温度が低く、湿度が高いコンテキスト、(4) 温度と湿度が低いコンテキスト、という 4 つのソースコンテキストを生成した。本実験では、温度と湿度のレンジを $2^\circ C$ と 5% で分割し、 k の値を $k = 1, \dots, 3$ として、それぞれ対象コンテキストとソースコンテキストを生成した。また、生成されたソースコンテキストと対象コンテキストに

おいて、全てのデバイスの電源状態を OFF に設定した。

4.1.3 実験手法

本実験では、生成した全てのソースコンテキストから対応する対象コンテキストに遷移させるデバイス操作シーケンスと消費電力量を PathSim を用いて導出する。また、一般的な手法を利用したシーケンスも導出し、PathSim を用いて導出したシーケンスと比較する。一般的な手法は、ソースコンテキスト内の温度/湿度のレンジが、対象コンテキストの温度/湿度のレンジと異なる場合に、関連する全ての有効なデバイスを動作させ（または、必要に応じて適切なモードに設定）、有効では無いデバイスの電源を切る（または、最小電力で維持させる）ような直感的なデバイス制御法である。

4.2 実験結果

実験の結果を図 2 と図 3 に示す。図 2 は、 $k = 1, \dots, 3$ に対して各手法で導出されたシーケンスの平均消費電力量を表している。図 2 から、 k の値に関係なく、PathSim の平均消費電力量は一般的な手法より低いことが明らかである。また、 $k = 1, \dots, 3$ に対する全シーケンスの平均消費電力量は、PathSim で $0.07 kWh$ 、一般的な手法で $0.25 kWh$ であった。そのため、PathSim は一般的な手法と比べて、平均消費電力量を 72% 削減できていることが分かる。

図 3 は、PathSim と一般的な手法で導出したシーケンスの消費電力の、 $k = 1, 2, 3$ ごとの累積分布関数 (CDF: Cumulative Distribution Function) を表している。一般的な手法は、50% またはより多くのシーケンスで $0.4 kWh$ 以上の電力を消費しており、一方、PathSim には、 $0.4 kWh$ より多く電力を消費するシーケンスは存在しない。さらに、 $k = 3$ の時、PathSim は、シーケンスの約 40% で $0.2 kWh$ 以下の消費電力で遷移を行え、これに対し、一般的な手法では、 $0.2 kWh$ 以下で遷移が可能なシーケンスが約 10% ほどしかない。以上の結果から、PathSim は、2 つのコンテキスト間の遷移に対するエネルギー効率のよいイベントシーケンスを導出でき、一般的な手法と比べて大幅に消費電力量を削減できていることが分かる。

5. 関連研究

提案した PathSim に関連する (1) スマートホーム/コンテキストアウェアシステム、(2) コンテキストの遷移に関するモデル化、に関する既存研究について概要を述べる。

スマートホーム [6] とコンテキストアウェアシステムに関する研究 [3], [8] が多く存在する。MavHome [6] では、家の状態 (コンテキスト) を感知し、自動的に快適なサービスをユーザに提供する知的エージェントの実現を目的に、住民の行動予測に基づいたコンテキストア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。柏

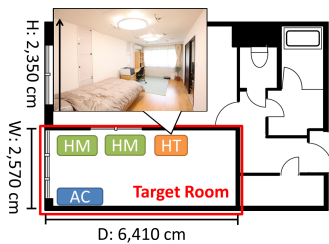


図 1 対象とするスマートホーム

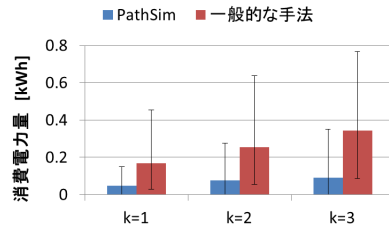


図 2 $k = 1, 2, 3$ に対して導出したシーケンスの平均, 最大, 最小エネルギー消費量

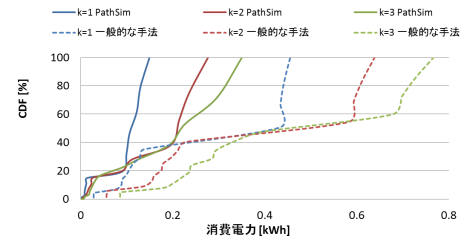


図 3 消費電力の累積分布関数 (CDF : Cumulative Distribution Function)

表 1 ケーススタディで使用する要求仕様

性質	事前条件 <i>Pre</i>	事後条件 <i>Post</i>
<i>prA</i>	$room.temp \notin (26^\circ C, 28^\circ C] \vee room.humid \notin (55\%, 60\%]$	$room.temp \in (26^\circ C, 28^\circ C] \wedge room.humid \in (55\%, 60\%]$
<i>prB</i>	$room.temp \notin (24^\circ C, 26^\circ C] \vee room.humid \notin (50\%, 55\%]$	$room.temp \in (24^\circ C, 26^\circ C] \wedge room.humid \in (50\%, 55\%]$
<i>prC</i>	$room.temp \notin (22^\circ C, 24^\circ C] \vee room.humid \notin (45\%, 50\%]$	$room.temp \in (22^\circ C, 24^\circ C] \wedge room.humid \in (45\%, 50\%]$
<i>prD</i>	$room.temp \notin (20^\circ C, 22^\circ C] \vee room.humid \notin (40\%, 45\%]$	$room.temp \in (20^\circ C, 22^\circ C] \wedge room.humid \in (40\%, 45\%]$

本ら [8] 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, ユーザの嗜好などを考慮して, 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, この研究では, 各環境変数を変化できるデバイスは 1 つだけだと仮定しており, 複数のデバイスを協調させながら, 最小消費電力量で特定のコンテキストに遷移することは考えていない.

コンテキストのモデル化と推論を扱ういくつかの研究が存在する. 文献 [7] では, 基本的なモデルとしてオブジェクトロールベースモデル, 空間モデル, オントロジーベースモデル, という 3 つのモデルが紹介されている. これらのモデルは, 形式的に任意のコンテキストを表すことができる. しかしながら, これらのモデルでは, コンテキスト間の遷移に関しては形式的に定義していない.

上記で説明したように, 本論文で対象とする問題を解決することを目的とした既存研究は存在しない. 我々が知る限り, 提案手法は, 最小消費電力量でスマートスペースを特定のコンテキストに遷移できる複数デバイス操作シーケンスを導出する最初の手法である.

6. まとめ

本論文では, コンテキストウェアなスマートスペースシステムにおいて, 2 つのコンテキスト間の遷移に対する最小コストのデバイス操作シーケンスを導出するツール *PathSim* を提案した. *PathSim* は, スマートスペースの変数の定義域を有限個に適切に分割し, (a) コンテキストと関連するイベントを最適に選択/決定し, (b) 選択したイベントを実行した時のコンテキストの遷移と関連する電力消費量をシミュレートし, そして (c) A*アルゴリズムに基づいて最小コストのシーケンスを求めることで最小コ

ストのデバイス操作シーケンスを導出する. *PathSim* と一般的な手法を用いた評価実験により, *PathSim* は, 一般的な手法よりおよそ 72% 少ない消費電力量を持つシーケンスを導出できることが分かった.

今後は, *PathSim* がより精度の高いデバイス操作シーケンスを導出できるように, 各種デバイスの消費電力モデルや出力モデルを実測値を用いて構築することを検討している.

参考文献

- [1] M. Weiser: "Some computer science issues in ubiquitous computing," *Communication of the ACM*, Vol. 36, No. 7, pp. 75-84, 1993.
- [2] K. Henriksena and J. Indulskab: "Developing context-aware pervasive computing applications: Models and approach," *Pervasive and Mobile Computing*, Vol. 2, No. 1, pp. 37-64, 2006.
- [3] M. Baldauf, S. Dustdar, and F. Rosenberg: "A survey on context-aware systems," *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 2, No. 4, pp. 263-277, 2007.
- [4] W. K. Edwards and R. E. Grinter: "At Home with Ubiquitous Computing: Seven Challenges," *Proc. of UbiComp 2001*, LNCS 2201, pp. 256-272, 2001.
- [5] P.E. Hart, N. J. Nilsson, B. Raphael: "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Trans. on Systems Science and Cybernetics* SSC4 (2): pp. 100-107 (1968).
- [6] Cook, D. J., Youngblood, M., Heierman III, E. O., Gopalratnam, K., Rao, S., Litvin, A., and Khawaja, F., "MavHome: An agent-based smart home," *Proc. of IEEE PerCom 2003*, pp. 521-524, 2003.
- [7] Bettini, C., Brdiczka, O., Henriksen, K., Indulka, J., Nicklas, D., Ranganathan, A., and Riboni, D., "A survey of context modelling and reasoning techniques," *Pervasive and Mobile Computing*, 6(2), pp. 161-180, 2010.
- [8] Kashimoto, Y., Ogura, K., Yamamoto, S., Yasumoto, K., Ito, M.: "Saving Energy in Smart Homes with Minimal Comfort Level Reduction," *Workshop Proc. of IEEE PerCom 2013*, pp. 372-376, 2013.