

## Tag and Think : モノに添付したセンサノードのためのモノ自身の推定

前川 卓也<sup>†1</sup> 柳沢 豊<sup>†1</sup> 岡留 剛<sup>†1</sup>

あらかじめモノに関する情報はいっさい与えずに、センサノードをモノに添付するだけで、添付されたモノが何であるか、そしてそのモノが現在どのような状態にあるかを推定するためのフレームワーク *Tag and Think* を提案する。特に本論文では、人の実世界に関する知識によって構築されたモノのモデルと、センサノードから取得されたセンサデータを用いてモノのタイプを推定する手法について紹介する。提案手法では、人によって作成されたノイズの含まれるモデルにロバストな推定アルゴリズム適用することで高い推定率を目指す。また、実環境において提案手法の評価を行う。

### Tag and Think: Inferring Object Type for Object Attached Sensor Nodes

TAKUYA MAEKAWA,<sup>†1</sup> YUTAKA YANAGISAWA<sup>†1</sup>  
and TAKESHI OKADOME<sup>†1</sup>

The system framework, *Tag and Think*, proposed here enables us to introduce ubiquitous environments easily and simply by attaching sensor nodes to physical objects without any information about the objects. In the framework, we present a method that uses real-world knowledge constructed from a person's common knowledge to infer types of physical indoor objects and their states automatically from sensor data produced by the sensor nodes. The method aims to achieve high inference accuracy by applying a robust inference algorithm to the noisy model built by a person. We also evaluate the method in a real environment.

<sup>†1</sup> NTT コミュニケーション科学基礎研究所  
NTT Communication Science Laboratories

### 1. はじめに

家庭屋内環境においてセンサノードをモノに添付し、モノのタイプごとにモノの状態や状態変化に応じた状況依存サービスを提供するための研究が多く行われている。ドアというタイプのモノが深夜に長時間開け放たれていたら、ユーザに通知を行うなどがそのサービスの一例である。しかし、数百のセンサノードが家庭内のモノに添付された場合、それぞれのノードにそのノードが添付されているモノのタイプを手作業で与えることは困難である。無線技術、センシング技術、省電力化技術の進展により、小型で安価なセンサノードをばらまくように使える時代がまもなくやってくる。我々は、そのような時代を見越して、モノに汎用的なセンサノードを添付するだけで、そのノードがどのタイプのモノに添付されているか、そのモノがどのような状態にあるかを自動的に推定するフレームワークである *Tag and Think* (TnT) を提案する。

TnT では、汎用的なセンサノードがさまざまなモノに添付された環境において一定期間センサデータを収集したあと、あらかじめ用意したモノのタイプごとのモデルから、収集したデータを最も出力しうるモデルを決定することでモノのタイプの推定を行う。さらに、モデルを用いてモノにどのような状態変化が起こったかも同時に推定する。本研究における汎用的なセンサノードは、加速度センサや照度センサなどの一般的なセンサを含み、すべて同一の機能を持つものであるとする。汎用的なセンサノードは特定のモノ専用のノードに比べ、生産の際にコスト的に優位なだけでなく、実際にセンサネットワーク環境を構築する際、大量のモノにセンサノードを添付するだけでよいため、導入の労力を抑えられる点でも有利である。また、以降では我々は加速度と照度センサを搭載したセンサノードを想定する。これらのセンサは現時点で最も一般的といえ、さらにエンタテインメント用やセキュリティ用として需要が高く、将来的にさらなる低価格化も見込めるため、これらを採用した。本論文では、特に人の経験や実世界に関する知識を用いて屋内にあるモノのタイプとその状態を推定する手法を紹介し、その評価を行う。

### 2. 背景と目的

近年、安価になってきたセンサノードを家庭にある大量のモノに添付することで、モノの状態や状態変化を取得することが可能となってきた<sup>6)</sup>。また、それにより状況依存サービスの実現が容易となりつつある。状況依存サービスは主に下記の2つの方法から実現される。(1) センサノードが添付されたモノの状態や状態変化をトリガとしたサービスをユーザ

に提供する<sup>7)</sup>。(2) 複数のセンサノードの出力と教師信号を用いて Activity of Daily Living (ADL) を推定し, ADL をトリガとしたサービスをユーザに提供する。方法 (2) の ADL の推定には教師あり学習が用いられる場合が多い。たとえば, 文献 9) ではモノの状態変化を取得できるセンサを用いて教師あり学習により ADL の推定を行っている。教師信号は, ユーザがそのユーザの生活にラベリングすることで作成される (たとえば, 7:45 から 8:00 までは朝食を作っていたなど)。方法 (2) を実現するために必要なことは, センサノードのモノへの添付と, ユーザによる生活行動のラベリングである。方法 (1) は, モノの名前とその状態や状態変化に応じたルールを用いてサービスを提供することが多い (たとえば, 人がイスに座れば照明をつけるなど)。方法 (1) を実現するために必要なことは, センサノードのモノへの添付, センサノードとそれが添付するモノのタイプの関連付け, および, モノの状態や状態変化とそれらに対応するセンサデータとの関連付けである。モノの状態や状態変化とそれらに対応するセンサデータとの関連付けとは, たとえば, ドアに添付したノードから得たセンサデータのある特徴的な変化は「ドアが開いた」という状態変化に対応するといった関連付けである。これにより, あるモノに対してある状態変化が起これば, それに対応するサービスを提供するというルールを記述できる。本論文の焦点は方法 (1) を実現するための, センサノードとそれが添付するモノのタイプの関連付け, および, モノの状態や状態変化とそれらに対応するセンサデータとの関連付けである。

一方, 低コストなセンサ環境の実現のため, エンドユーザによる屋内設置型センサの導入に関する調査研究がこれまでに進められている。文献 3) では, 実験においてバーコードとバーコードリーダを用いて手動でセンサノードとそれが添付するモノのタイプの関連付けを行っていたが, ノードとモノのタイプを関連付けるという行為自体の意味を理解できないエンドユーザがいた。また, 10 個のセンサノードを設置するのに平均して 84 分もの時間を要していた (ただしこれは関連付けのみに要した時間ではない)。本研究では, エンドユーザによるセンサ環境導入を前提とし, このようなエンドユーザの負担となりうるセンサノードとそれが添付するモノのタイプの関連付けを推定により自動で行うことを目的とする。さらに, 状況依存サービス実現のために, センサデータからモノの状態や状態変化を推定する。

### 3. アプローチ

センサノードが添付されたモノの状態や状態変化をトリガとしたサービスをユーザに提供するためには, センサデータからモノの状態や状態変化を推定しなければならない。逆にいえば, 状態や状態変化を推定できるモノでなければ, その状態と状態変化をトリガとして

用いることはできない。我々は, モノの状態変化 (動作) の特徴から, モノのタイプを下記の 3 種類に分類した。

(C1) 靴や歯ブラシなど, 特徴的な繰返し動作を持つモノ。

(C2) ドアや引き出しといった, モノのタイプに特徴的な動作がセンサの特徴的な出力の組合せからなるモノ。

(C3) テーブルや定規, ケトルなど, モノのタイプに特徴的な動作がない, またはセンサで特徴的な動作を検知できないモノ。

C1 のカテゴリに属するモノの特徴的な繰返し動作を検出するには, ADL 推定の研究における ‘歩行’ や ‘自転車での走行’ などの繰返し運動を推定する研究<sup>2)</sup> を利用できるだろう。つまり, 人が靴や歯ブラシなどを用いる際の加速度信号のフーリエ成分を用いてそれらのモノをモデル化することで, C1 のカテゴリに属するモノのタイプと状態変化は推定できる。C2 に属するモノのタイプの特徴的な動作は特徴的な周波数成分などを持たず, 特徴的な運動や照度変化などからなり, それらの運動や照度変化は抽象化することで表現できる。たとえば, ドアが開くという状態変化には, 回転運動と照度変化がともなう, などである。本論文では, 特に C2 に属するモノの状態変化の推定と, タイプの推定を行う。C3 に属するモノのタイプの推定は困難だが, そもそもその動作も推定できないため, そのタイプと動作 (状態変化) を用いた状況依存サービスを行うことは不可能である。C3 に属するモノは, 2 章で述べたモノのタイプを必要としない ADL を用いた状況依存サービスに用いられるだろう。

我々は C2 に属するモノのタイプ, 状態, 状態変化を推定するために, あらかじめモノのタイプごとにモデルを用意する。簡単なアイデアとして教師信号を用いた学習によりモデルを作成することが考えられる。しかし, エンドユーザの家の環境はさまざまであるという問題がある。たとえば, 家によって照明の具合は違ふし, 同じタイプのモノにもサイズや重さは多様である。さらに, ノードを添付する向きや角度が異なれば得られるセンサデータも異なる。すなわち, 学習によりモデルを作成するならば, エンドユーザの家で得られたセンサデータからエンドユーザが教師信号を作成しなければならない。これは, センサノードが大量にある環境においては, ユーザにとって多大な労力となるだろう。また, 作成されたモデルはそのエンドユーザの家庭でしか使えない。

ここで, 世界中の屋内環境で利用可能な C2 に属するモノのモデルについて考えてみよう。たとえば, ドアには開と閉の 2 状態があり, ドアが開から閉に状態が変化するときには, 加速度センサが回転を検知すると同時に照度センサが照度の変化を検知する。このモデルは,

誰もが容易に想像しうるモデルであり、世界中のほとんどのドアに共通する性質を反映しているであろう。もちろん実環境では、そのようなモデルに厳密に合致しない現象も起こるだろう。しかし、そのような現象に対してはロバストな推定アルゴリズムを利用することで対処できると考える。我々は一般的な知識を用いて上記のようなモデルを作成することができる。そのモデルは多くの環境で利用可能であるが、現実世界で起こる現象をすべて厳密に記述できていない。一方、教師あり学習を用いて厳密なモデルを作成することができる。そのモデルは作成にコストがかかるうえ、特定の環境でしか利用できない。我々は、容易に作成可能だが不完全なモデルとロバストなアルゴリズムを用いる手法が現実的であると考え。

本論文では、経験と実世界に関する知識に基づいて作成された上記のようなモノのモデルをモノのタイプや状態変化の推定に用いることを想定する。そして、実験により実験参加者に作成してもらったモデルがモノのタイプや状態変化の推定にどの程度有効かを検証する。そのために、次章では誰もが簡単に記述できるようなモデルの記述方式を提案する。さらに5章では、手作業で作成されたモデルとセンサデータを用いてモノのタイプと状態変化を推定する手法を提案する。

#### 4. モノのタイプのモデル

我々は、C2に属するモノをモデル化するために状態遷移図を選んだ。状態遷移図を用いて実世界の現象を記述する先行研究はいくつか存在する。文献5)は、状態遷移図を用いて家庭内の状態をモデル化している。たとえば、浴室の利用状況を表す状態遷移図は、‘入浴中’と‘入浴中でない’の2つの状態間を、浴室のドアの開閉などといったイベントにより遷移するようモデル化されている。状態遷移図を用いることで、モノの状態と状態変化を直感的に記述できると考える。以降では、状態変化を状態遷移と呼ぶ。

##### 4.1 モノのタイプの状態遷移図

図1は、後述する実験で得られたドアタイプの状態遷移図を図式化したものである。‘Open’の遷移から出ている吹き出しは、その遷移が起こる際にセンサノードによって検知されるであろうイベントの集合を時系列で表している（以降ではイベントの集合をイベント集合と呼ぶ）。遷移はイベント集合のセットからなっており、‘Open’の例では3つのイベント集合を示している。ただし、‘acc.rotate’, ‘illumi.bright’, ‘illumi.dark’は、加速度センサにより検知される回転運動のイベント、照度センサにより検知される照度上昇のイベント、照度下降のイベントをそれぞれ表す。また、‘person.walk’と‘person.point’は、人の歩行と、人が1カ所で何らかの動作をしている現象を表し、モノに添付しているセンサノード以外の

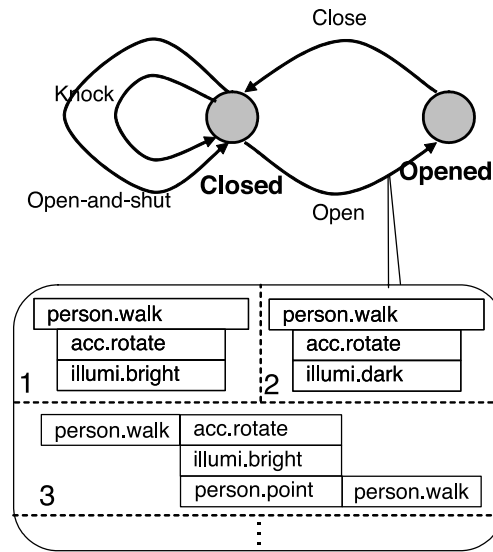


図1 ドアの状態遷移図  
Fig. 1 Diagram for the door.

デバイスから得られるイベントである。イベント集合の詳細な記述方法については後述するが、この例では、“Open”の遷移に含まれる3通りのケースを示しており、1つ目と2つ目は人が歩きながらドアを開けて通り抜けていくケースを、3つ目は人がドアの前まで歩いてきて立ち止まってからドアを開けドアから離れて歩いていくケースを表している。このように、1つの遷移に対して複数のケースを記述できるため、さまざまな状況を考慮し表現することができる。

##### 4.2 センサデータと状態遷移図

センサデータと状態遷移図の関係について説明しよう。遷移図における遷移は、センサデータにおける振幅が大きく変動している区間に対応する（以降ではその区間を変化と呼ぶ）。遷移図における状態は、センサデータにおいて振幅がほとんど変動していない区間に対応する。図2は、ノードを添付したドアが開き、閉じ、開閉する遷移を2回繰り返した際に得られたX軸方向の加速度、Z軸方向の加速度、照度である。Y軸方向の加速度センサの波形は省略している。 $ax_i$  および  $az_i$  ( $i = 1, \dots, 6$ ) は、XおよびZ方向の加速度センサから得られた変化であり、 $ai_i$  は照度センサから得られた変化である。加速度センサの変

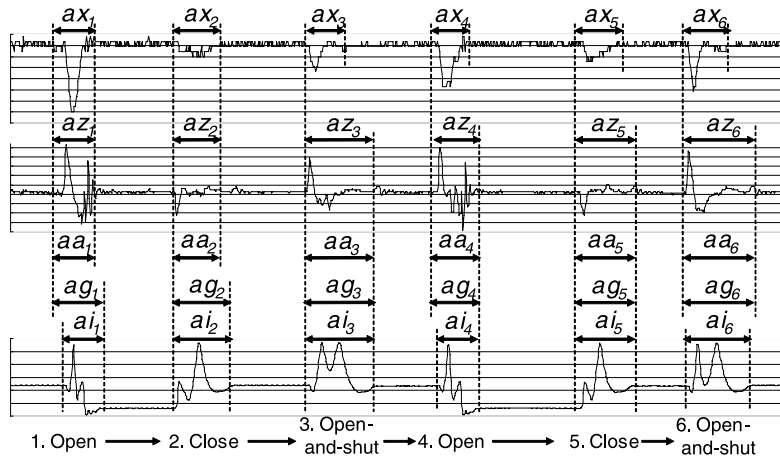


図 2 ドアに添付されたセンサノードから取得した信号  
Fig. 2 Signals from a sensor node attached to a door.

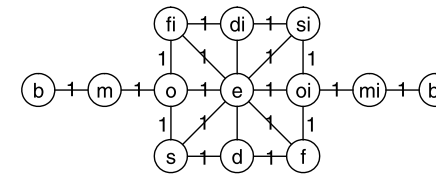
化である  $aa_i$  ( $i = 1, \dots, 6$ ) は,  $ax_i, ay_i, az_i$  の和集合である.  $ag_i$  ( $i = 1, \dots, 6$ ) は, 変化の集合であり, 以降では変化集合と呼ぶ. たとえば,  $ag_1$  は  $aa_1$  と  $ai_1$  からなる. 変化集合は遷移 (イベント集合) に対応し, たとえば,  $ag_1$  と  $ag_4$  は ‘Open’ に対応している. タイプの推定を行う際, 変化集合とイベント集合の比較を行う. イベント集合はシンボル (‘acc.rotate’ や ‘illumi.bright’ など) で記述されているため, 比較の際は変化を抽象化する (シンボルに変換する). たとえば,  $aa_1$  は ‘acc.rotate’ に抽象化される. さらに, 変化集合を時系列順に並べることで,  $ag_1, ag_2, ag_3, \dots$  といった変化集合系列を得ることができる. 次章で説明する推定手法では, (1) センサデータから変化を抽出し, (2) 変化を抽象化し, (3) 変化をグループ化し, (4) グループ化した変化集合を時系列順に並べ, (5) 変化集合系列に最も合致するもっともらしい状態遷移図を決定する.

### 4.3 イベント集合

イベント集合の記述方法について説明する. まず, 我々が用意したイベントについて紹介する. ‘illumi.bright’, ‘illumi.dark’, ‘illumi.bright\_dark’, ‘illumi.dark\_bright’ を照度センサが取得できるイベントとして用意した. ‘illumi.bright\_dark’ は, 照度が上昇した後に元の照度に戻る現象を表す. また, ‘acc.slide\_horizontal’, ‘acc.rotate’, ‘acc.drop’, ‘acc.fall’, ‘acc.raise’, ‘acc.vibrate’, ‘acc.other’ を加速度センサが取得できるイベントとして用意し

X before Y	
X equal Y	
X meets Y	
X overlaps Y	
X during Y	
X starts Y	
X finishes Y	

(a)



(b)

図 3 (a) Allen の時区間関係表現と (b) 時区間関係間の距離  
Fig. 3 (a) Allen's interval logic and (b) distance between interval relations.

た. これらはそれぞれ, ‘水平方向への移動’, ‘回転運動’, ‘落下’, ‘倒れること’, ‘鉛直上向きの移動’, ‘微小振動’, ‘その他の動き’を表す. これらは, 加速度センサデータの解析により高い精度で検知できるイベントである. ただし, 解析の際の積分累積誤差によりイベントの初期動作のみ高い精度で検知できる. たとえば, 図 2 の ‘Open-and-shut’ では, ドアが回転運動をした直後に, 反対方向に回転運動をする. しかし, 累積誤差のため反対方向の回転が起こったことを正確に検知することは困難である. それゆえ, これらの運動を表現するときは最初に起こった運動 (‘acc.rotate’) のみを用いる. ‘person.walk’ と ‘person.point’ を他のデバイスから取得できる人の歩行に関するイベントとして用意した.

イベント集合のイベント間には時間関係がある. たとえば, 図 1 の ‘Open’ の 1 つ目のイベント集合では, ‘person.walk’ が起こっている間に, ‘acc.rotate’ と ‘illumi.bright’ が同時に起こっている. このようなイベント間の時間的な関係は Allen の時区間関係表現<sup>1)</sup> を用いて表現する. Allen は図 3 (a) に示す 13 通りの時区間関係を定義している (‘equal’ 以外にはそれぞれインパースの関係が存在する). たとえば, ‘X before Y’ では, 黒の矩形が

イベント X の、白の矩形がイベント Y の時間的な範囲を表し、イベント X のあとにイベント Y が起こっていることを表現している。このようなシンボルとシンボル間の時間関係を用いたイベント集合の表現は直感的に分かりやすい。

イベント集合はその生起確率を持つ。生起確率はモノのタイプの推定において重要である。たとえば、モノ A によく起こる状態変化  $t_A$  に対応する状態変化の特徴が、モノ B のめったに起こらない状態変化  $t_B$  のそれと非常に似ているとしよう。そして、モノ A にセンサノードが添付されており、状態変化  $t_A$  ばかりが起こったとする。このとき、モノのモデルに状態変化の生起確率が含まれていなければ、そのセンサノードがモノ A か B に添付しているかを正しく判断することは難しい。そこで、経験や実世界の知識により、頻度をよく起こるかそうでないかの 2 値で記述する。たとえば後述する実験で作成された遷移図では、ドアの前で立ち話をした後にドアを開けるという行為に対応するイベント集合は、生起頻度が低いとされている。

#### 4.4 共通する遷移

モノには可搬性と可動性といった性質を持つものがあり、それらに共通して起こる遷移が存在する。たとえばコップは可搬性のあるもので、人が持ち歩いたり座りながらその位置を移動させたりする遷移などが存在する。イスは可動性のあるもので、人が押して移動させたり倒れたりする遷移が存在する。また、すべてのモノに共通して起こる遷移も存在する。以上の遷移をすべて考慮すると、状態遷移図は複雑となり記述の際に大きな負担となる。そこで、記述したいモノに可搬性や可動性があると明示すれば、そのモノのすべての状態に対応する遷移が存在すると解釈する。以下に、可搬性があるモノ、可動性があるモノ、すべてのモノに共通して起こる遷移を示す。

可搬性 { 倒れる, 持ち歩く (人の移動あり), 移動する (人の移動なし), 落ちる }

可動性 { 倒れる, 移動する (人の移動あり) }

全てのモノ { 人がぶつかる, 人が近づいて照度が変化する }

ただし、これらの遷移は、上記した遷移の内容からも分かるようにすべてループバックの遷移とする。すなわち、状態が変化しない遷移とする。

## 5. 推定手法

### 5.1 変化区間の抽出

音声認識の研究において、音声信号から音声区間 (voice activity) を抽出する研究は多く行われている<sup>8)</sup>。これらの研究では、発話区間とそれ以外の区間のモデルをあらかじめ教

師つき学習によって作成しておき、テストデータがどちらのモデルに近いかどうかを判定する。我々もこの方法と同様のアプローチをとる。しかし、エンドユーザが学習のために変化している区間のデータを用意することは困難であるため、何も起こっていないときのセンサデータから作成したモデルとテストデータとの比較からテストデータが変化を含むかどうかを判定する (何も起こっていないときのデータは、住人が外出している間に取得すればよい)。モデルはセンサごとに、ある長さのセンサデータセグメントのフーリエ成分を用いて Gaussian Mixture Model (GMM) で作成する。実際にセンサデータセグメントが変化を含むかどうかは、そのセグメントのフーリエ成分のモデルに対する対数尤度が閾値より小さいかどうかで判定する。

### 5.2 変化の抽象化

抽出された変化は、4.3 節で述べたイベントに抽象化される。‘acc.rotate’ と ‘acc.slide.horizontal’ の検出は、ルンゲクッタ法を用いて加速度を 2 回積分し、センサノードの軌跡を求めることで行っている。‘acc.drop’ と ‘acc.fall’ は、センサノードの軌跡と鉛直方向の加速度を求めることで検出する。‘acc.raise’ の検出は、軌跡の方向が鉛直上向きかどうかを判定することで行っている。‘acc.vibrate’ は、加速度がほとんど高周波成分かどうかで判定する。

本研究で用いた照度センサは、0 から 1024 までの値を照度として出力する。変化の前後で照度が 20 以上上がった場合は ‘illumi.bright’ とし、20 以上下がった場合は ‘illumi.dark’ とした。照度が 20 以上上がってから元の照度に戻った場合は ‘illumi.bright\_dark’ とし、照度が 20 以上下がってから元の照度に戻った場合は ‘illumi.dark\_bright’ とした。

### 5.3 モノのタイプの推定

変化集合系列とモデル (状態遷移図) を比較する手段として、高速でロバストである Viterbi アルゴリズム<sup>10)</sup> を用いる。Viterbi アルゴリズムは、Hidden Markov Model (HMM) が既知の場合に、与えられた観測系列から最尤推定になる状態系列を動的計画法で求める。本手法では、HMM がモノのタイプの状態遷移図に、観測系列が変化集合系列に、状態系列がイベント集合系列 (状態遷移系列) に対応する。Viterbi アルゴリズムにより、変化集合系列 ( $AG = ag_1, ag_2, \dots, ag_T$ ) の尤度  $P(D, AG)$  をそれぞれの状態遷移図  $D = (A, ST)$  に対して求め、尤度が最も大きい遷移図に対応するタイプを推定結果として出力する。以降の説明で用いる主な記号とその定義を表 1 にまとめている。以下で、

$A = \{a_i\}$  :  $a_i = 1/N$  は初期に状態  $s_i$  ( $1 \leq i \leq N$ ) に存在する確率,

$ST = \{t_{n,m}\}$  :  $t_{n,m}$  は状態  $s_n$  から  $s_m$  への遷移 ( $1 \leq n \leq N; 1 \leq m \leq N$ ),

表 1 用いられる主な記号とその定義  
Table 1 Symbols and their definition.

記号	定義
$ag_i$	$i$ 番目の変化集合
$s_n$	状態遷移図における状態
$t_{n,m}$	状態遷移図における状態 $s_n$ から $s_m$ への遷移
$eg_i$	遷移に含まれるイベント集合
$e_i$	イベント集合の生起頻度

$t_{n,m} = \{eg_i\} : eg_i$  は  $t_{n,m}$  が含むイベント集合 ( $i \geq 0$ ) とする .

Viterbi アルゴリズムでは  $P(D, AG)$  は下記のように求められる .

$$P(D, AG) = \max_{1 \leq i \leq N} (p_{i,T+1}),$$

$$p_{i,j} = \max_{1 \leq k \leq N} (p_{k,j-1} + Sim(t_{k,i}, ag_{j-1}, Th)),$$

$$(1 \leq i \leq N; 2 \leq j \leq T+1),$$

$$Sim(t_{n,m}, ag_i, Th) = \max_{j, e_j > Th} (sim(ag_i, eg_j) \cdot e_j).$$

ここで,  $Th$  は閾値である .  $j = 1$  のとき  $p_{i,0}$  は  $a_i$  である .  $sim(ag_i, eg_j)$  は,  $ag_i$  と  $eg_j$  の類似度であり,  $e_i$  は  $eg_i$  の生起頻度である . また,  $p_{i,j}$  の計算において  $p_{k,j-1}$  と  $Sim(t_{k,i}, ag_{j-1}, Th)$  の和を用いている . 一般的に Viterbi アルゴリズムでは積を用いるが, 我々は  $Sim$  を変化集合とイベント集合の距離 (の逆数) としているため, DP マッチングと同様に和を用いている . したがって,  $P(D, AG)$  は厳密には観測系列とモデルの適合の度合いであるが, 便宜上尤度と呼ぶ .

我々は,  $ag_i$  と  $eg_j$  の類似度を  $ag_i$  を  $eg_j$  に変換するためのコストの逆数としている . 文献 4) では時区間関係間の距離が定義されており, 類似映像検索などに用いられている . 図 3 (b) は, 13 の時区間関係間の距離を表している ('b' は 'before' を 'bi' は 'before のインパース' を表す) . 我々はこの距離を変換のためのコストとして用いる . また, イベントを追加または削除するコストを 20, イベントを他のイベントに変換するコスト (ただし同じセンサのイベントにしか変換できない) を 5 とした . これらの値は事前実験でよい結果を得たものを採用した . たとえば変化集合 'illumi.dark meets acc.rotate' をイベント集合 'illumi.dark equal acc.raise' に変換するには, 7 (meets を equal に変換するコスト 2 と acc.rotate を acc.raise に変換するコスト 5) のコストがかかる . また, 変化集合とイベン

ト集合が完全に一致するときはそれらの間のコストを 0.5 とする . ここで簡潔に変化集合からイベント集合に変換するコストの計算方法を説明する . まず, 変化集合とイベント集合において, それらに含まれる変化とイベントの種類と数がそれぞれ合致するように変化の追加・削除・変換の作業を行う . この過程では全探索により最もコストが小さくなる作業を採用する . そして, 変化集合における変化間の時間関係の変換を行う . このときも, 同じ種類の変化が 2 つ以上あるときは全探索を行う .

モノのタイプを推定するときは, 多くの遷移図に対して上記の手続きを実行する必要があるため, 閾値  $Th$  より低い生起頻度のイベント集合は無視する . 尤度が最も高い  $D_{max}$  を求めたあと,  $Th = 0$  として  $D_{max}$  に対して上記の手順を再度適用することで求めた最も尤度の高い状態遷移系列を状態遷移の推定結果として出力する . ここで, 手作業で作成された状態遷移図では, イベント集合の生起頻度が 'high' が 'low' で表現されている . 我々は,  $e_{(low)} < Th < e_{(high)}$  とした ( $Th > 0$ ) .

## 6. 評価

本章では実験により, 実験参加者に作成されたモデルの有効性と欠点, 提案するモデルがどのようなモノをモデリングするのに適しているかを明らかにする .

### 6.1 実験環境

センサノードは, 我々の研究グループが開発したもので, 一般的に普及している照度センサ, 三軸加速度センサを搭載する . これらのセンサから取得したデータをデータ保存用のサーバシステムに送信するために, 通信モジュールを搭載している . 約 60 msec ごとにその間にセンシングしたデータの平均値を送信する . 図 4 (a) にセンサノードを示す . 図 4 (b) に示すようなセンサネットワークシステムを備えたオフィス環境実験室を構築した . この部屋では, 午前 9 時から午後 5 時ごろまで 1 人から 4 人程度の作業員 (研究者ではない) が各々の作業や本センサネットワークシステムの整備などを行っている . 部屋の床にはセンサフロアシステム (Vstone, VS-SF99) を設置しており, 人の歩行を検知できる . 床圧センサはまだ一般的であるとはいえず, コピキタス環境の導入の妨げになると考えるが, 作業者に負担を強いることなく歩行データを収集できるため, 実験では歩行検出のために床圧センサを用いた . 実際には, スリッパや靴にセンサノードを添付することで人の歩行を検知してもよいだろう .

図 4 (b) は, センサノードを添付したモノの位置も示している . C2 と C3 に属すると考えられるモノにセンサノードを添付した . C2 に属するモノとしては, イスの背もたれ (1),

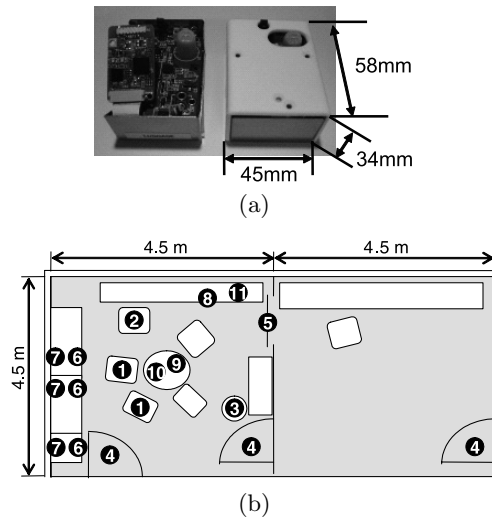


図 4 (a) センサノードと (b) センサ部屋  
Fig. 4 (a) Sensor node, (b) Sensing room.

車輪付き回転イスの背もたれ (2), ふた付きごみ箱のふたの内側 (3), ドア (4), スライドドア (5), ロッカーの扉の外側 (6), ロッカーの内部 (7), 引き出しの外側 (8) に添付した. C3 に属すると考えられるモノとしては, カップ (9), 机 (10) と目覚まし時計 (11) に添付した. 3 つあるドアのうち, 2 つのドアは同じ製品である. また, 2 つの背もたれのあるイスは同じ製品である. この C2 と C3 へのカテゴリ分けは, 我々が実験前に暫定的に行ったものである.

## 6.2 モデルの作成

3 人の実験参加者にモデルの作成方法を記述した文書を読んでもらった後, ドローソフト (Microsoft PowerPoint) を使ってモデルを描いてもらった. その後, 我々がそのモデルを TnT で用いるフォーマットに変換した. 参加者 A と B は情報技術関係の職業経験がある. 参加者 C は, PC の利用経験が乏しく, PowerPoint の使い方から説明した. 参加者にはそれぞれ図 4 (b) に示した 11 のモノに加え, 13 のモデルを作成してもらった (ロッカー扉の内側, 下方向に開くロッカー扉の外側, 下方向に開くロッカー扉の内側, 引き出しの内部, 車輪付き回転イスのシート, 背もたれのあるイスのシート, 車輪なし回転イスの背もたれ, 車輪なし回転イスのシート, ロッキングチェアの背もたれ, ロッキングチェアのシート, 窓

枠, 鏡台の鏡, キャビネットの上部に取り付けられている上方向に開く鏡). これらは, 家具オンラインショップ KAGOO (<http://www.kokugai.com/kagoo>) が扱う約 4,000 点の商品から, ユーザがそれを利用したときに特徴的な動作が起こると考えられるモノをすべて選んだ. ただし, 異なるモノでも同じ動作や役割を持つと考えられるモノは区別しなかった. たとえば, 机の引き出しとロッカーの引き出しは同じものとした. また, 遷移図の形状 (状態数や遷移) を決定する際の参考にしてもらうため, 参加者に 1.4 GByte のコーパス (ニューヨークタイムズ) から抽出した名詞 (モノ) に関連する上位 10 の頻出動詞を提示した. たとえば, イスに対しては, 'sit down', 'take a seat', 'get up' などを与えた. また, 状態と遷移は自由に名付けてもらい, 後で我々が統一した. モデル作成の際に特に注意してもらった点は以下のとおりである. (1) 遷移は考え出せばきりがなくなるため, 世界中のどの環境でも起こりうるような遷移やイベント集合を書く. (2) さまざまなシチュエーションを想像してイベント集合を書く. (3) 加速度センサには初期動作のみしか認識できないという制限がある. そして最後に作成するモデルの使い道を説明した.

参加者 A と B は状態遷移図とイベント集合の記述方法を簡単に理解できたとコメントした. 参加者 C は状態遷移図自体の概念が理解できなかったため口頭で再度説明した. 参加者は, 24 のモデルを作成するのにそれぞれ 7 日, 4 日, 10 日かかった. 3 人が作った遷移図における, 遷移あたりの生起確率が高いイベント集合の数の平均は 5.8, 3.4, 3.6 だった. (1) の条件と動詞を与えたことにより, 参加者が作成した遷移図の形状にはあまり違いが見られなかった. 例外を 1 つ紹介する. 参加者 B が作成したごみ箱の遷移図は, 'Opened' と 'Closed' の 2 つの状態を含んでいたのに対して, 参加者 A が作成したごみ箱の遷移図は 1 つの状態と 'ゴミを捨てる' というループバックの遷移からなっていた. 付録に図 4 (b) に示すオブジェクトの状態遷移図の一部を示しておく.

## 6.3 実験

我々は, 連続する 5 日間に図 4 (b) に示す 18 のセンサノードからセンサデータを収集した. そのデータを用いて, センサノードが添付しているであろうモノのタイプに対応するモデルを 6.2 節で作成した 24 のモデルから推定した. このとき, 推定に使用するセンサデータの日数を変更して手法を実行した (1 日分と 2 日分). つまり, 1 日分のセンサデータを用いて手法を 5 回実行する試行と, 連続する 2 日分のセンサデータを用いて手法を 4 回実行する試行 (1 と 2 日目, 2 と 3 日目, 3 と 4 日目, および, 4 と 5 日目) である. 表 2 にその結果のタイプごとの平均を示す. タイプの推定が成功したモノに対して遷移の推定も行った. その評価は作業者らが室内を記録したビデオ映像を見ながら行った. このとき, 4.4 節

表 2 モノのタイプ (o) と遷移 (t) の推定精度 (%)  
 Table 2 Accuracies of object type (o) and transition (t) inferences (%).

参加者	o/t	日数	背もたれ 付きイス	回転 イス	ゴミ箱 のふた	ドア	スライド ドア	ロッカー の扉	ロッカー 内部	引き出し	カップ	机	目覚し 時計
A	o	1	80	100	100	93.33	80	80	0	60	40	0	0
A	o	2	87.5	100	100	100	100	91.67	0	50	50	0	0
A	t	1	76.92	47.83	97.5	92.31	79.07	92.31	-	87.5	81.25	-	-
A	t	2	82.06	50.34	97.01	97.4	68.37	92.93	-	71.43	78.95	-	-
B	o	1	80	100	80	93.33	100	0	0	20	60	0	0
B	o	2	87.5	100	100	100	100	0	0	25	50	0	0
B	t	1	84	34.78	96.97	90.38	66.07	-	-	100	81.82	-	-
B	t	2	84.02	36.05	94.03	95.17	64.29	-	-	85.71	78.95	-	-
C	o	1	60	100	60	86.67	80	80	0	60	60	0	0
C	o	2	62.5	100	100	100	100	91.67	0	75	75	0	0
C	t	1	81.82	41.3	96.67	93.1	53.33	93.84	-	78.57	60.61	-	-
C	t	2	81.31	42.86	97.01	97.03	52.08	93.27	-	70.83	79.59	-	-

の共通する遷移は無視した。また、推定アルゴリズムは PentiumD 2.8GHz, 2GB メモリを搭載した PC 上で実行した。

イス, ドア, ごみ箱, ロッカーの扉といったモノは, 高いタイプの推定精度を達成した。参加者からは, これらのモデルは簡単に作成できたという意見があった。しかし, 参加者 B の結果ではロッカー扉の精度が低い (ロッカー扉がドアと間違えて推定されていた)。これはモデルの書き分けに起因している。参加者 A と C は, モノのモデルがモノのタイプの推定に利用されることを知っていたため, ロッカー扉とドアの特徴を考慮してモデルに違いが出るように書いていた (ロッカーを開けるときは, ロッカーの前に立ち止まることが多い。一方, ドアを開けるときは歩きながら開けることが多い)。参加者 B もその事実は知っていたが, 参加者 B は書き分けに対してあまり注意を払っていなかった。モデルの書き分けは, タイプの推定精度に大きな影響を与える。

推定に利用するデータ量 (日数) が増えているにもかかわらず, 遷移の推定精度が大きく下がっているモノのタイプがあった。これは, 遷移の推定結果がタイプの推定に失敗した結果を含まないことに起因する。たとえば, まず, 1 日分のデータを用いたモノのタイプの推定において, モデルに定義されていなかった現象の多発により大きく推定が失敗していた日 (A) があると仮定する。このとき, 1 日分のデータを用いた遷移の推定結果の平均には, その A の結果は含まれない。しかし, 2 日分のデータを用いた実験において, その大きく失敗していた日 A を含めた 2 日間のデータセット (A + B) を用いたモノのタイプの推定

が, 他の 1 日のデータ B の貢献により成功していたら, A の結果が遷移の推定結果に含まれるため, 推定結果の精度は低くなる。

タイプの推定が高い確率で成功していたモノは, スライドドアと回転イスを除いて, 遷移の推定も高い確率で成功していた。スライドドアに関しては, 部屋の照明が消された状態でスライドドアが何度も開閉される日があった。このとき, スライドドアに添付したノードは照度の変化を検知することができなかつたため, 遷移図において照度変化のイベントを含むスライドドアの開閉を正しく推定できなかつたと考える。また回転イスに関しては, 作業者がイスをまったく動かすことなく立ち上がる現象を検知することができなかつたため, 精度が低かつたと推測される。もし, ノードの小型化や薄型化により, イスのシートの上にセンサを設置できれば, その現象は容易に検知できるだろう。また, 2 人の参加者から引き出しの外側のモデル作成に苦労したという感想をもらった。なぜなら, 人が引き出しを開けたり閉じたりするとき, その人が引き出しの周囲にもたらす照度変化はさまざまなパターンが考えられるからである。実際, 引き出しのタイプの推定精度は低かつた。引き出しの内部にノードを設置していれば, その推定精度は高くなつたであろう (追加実験によりこの仮定の正当性は確かめている)。また, ロッカーの内部に添付したセンサノードはほとんど動かないため, そのタイプの推定精度も低い。以上のように, ノードを添付する場所によっては同じモノでも精度が異なる場合があることが結論できる。

明らかに C3 に属する机と目覚まし時計はタイプの推定ができなかつた。参加者は特にこ



これらのモノのモデルの作成に苦労していた。また、これらのモノは適切にモデル化できていない可能性があるという感想があった。これらは状態遷移図によるモデル化には適していないだろう。また、ほとんどのカップのモデルでは、机上にあるカップを持ち上げて飲むという動作しかモデル化できていなかったため、そのタイプの推定精度も低かった。

以上から、そのモノに起こりうる主要な遷移を用意したイベントで記述できるモノは、参加者にはモデル化しやすく、モデルの書き分けさえ注意すればその推定精度も十分高いことが分かった。当然だが、遷移をイベントで記述が困難なモノ（C3に属するモノ）は、モデル化も困難であり、推定精度も低い。またC2に属するモノでも、ノードを添付する場所によって推定精度が異なることが分かったため、エンドユーザがノードを添付する際には精度が高い場所に添付するように指導する必要があるだろう。本論文で行ったモデル作成実験では、KAGOOで取り扱われている4,000点の家具のうち、特徴的な動作を持つと考えられる24種類の家具（家具の部位）のタイプのモデルを作成してもらった。これらの家具のみが存在する世界では、推定実験において高い精度を達成していたタイプのモノの推定は高い確率で成功するだろう。実験中には作成されたモデルにおいて定義されていなかった変化集合も多く起こっていた。しかし、系列の一貫性を考慮するViterbiアルゴリズムは、そのような変化集合のうちほとんどの変化集合の推定に成功していた。たとえば、ロッカー扉に添付したセンサノードから得た51%の変化集合が、参加者Aの作成したロッカー扉のモデルに含まれるイベント集合に完全に合致していなかった。提案手法と状態遷移図で記述されたモデルの組合せにより、人手で作成された実世界のモノを完全に表現できていないモデルを用いても高い推定精度を達成することができた。この結果は、C2に属するモノのモデルの作成および配布に2つの可能性があることを示している。1つ目は、訓練された人がモデルを作成し、それをエンドユーザに配布する。これにより、今回の実験より高い精度で推定が可能になると考えられるが、コストがかかる。2つ目は、エンドに近いユーザがモデルを共同で作成し、エンドユーザ同士でそれを共有する。推定精度は、今回の実験と同程度、または協調作業の効果である程度は高くなると考えられ、コストはかからない。

## 7. 考 察

TnTのスケラビリティについて述べる。現在のアルゴリズムは、1日に得られた変化集合と1つのモデルの比較に約0.1秒から15秒かかる（実験では平均で約3.98秒）。また、世の中に存在するC2に属するモノのタイプの数は明らかでないが、4,000種類の家具の検討から、多くても50程度ではないかと考えている。家庭内に500個のノードが添付されて

いる場合、比較に要する時間は約28時間で済む。

次に、提案した記述方式の記述力について検証する。しかし、そのためには世の中のすべてのモノに対して記述方式で表現可能か吟味しなければならない。そこで、ここではViterbiアルゴリズムにより同じ遷移図と認識されない遷移図の数を算出する。簡単のため、いくつかの仮定を設定する。(1) イベント集合には加速度のイベントと照度のイベントがそれぞれ最大で1つずつしか含まれない。(2) 加速度のイベントと照度のイベントは同時に起こる。(3) イベント集合における歩行イベントの組合せは、モデル作成実験においてよく見られた6つのケースのみとする（たとえば、person.walkだけのケース、person.walkの直後にperson.pointが起こるケースなど）。(4) 1つの遷移には4つのイベント集合が含まれる。(5) イベント集合の生起確率はすべて等しい。細かい論述は省略するが、上記の仮定から663種類の他と確実に区別できる遷移図が記述できる。この数は、上述の50に比べて十分に多いが、実環境ではよく似たイベント集合を持つモノも存在する。また、ロッキングチェアのモデル作成において、人がチェアを揺らす際の加速度イベントを我々の用意した6つのイベントでは記述できていなかった（acc.otherを使っていた）。さらに、モデルの作成において、車輪付き回転イスを回転させる現象と、回転させながらスライドさせる現象を書き分けることができないという意見があった。このような問題を解決するためにも、我々はモデルの記述力強化に取り組んでいく予定である。たとえば、ノードに搭載するセンサを増やせば記述できるイベント数が増えるし、変化の解析も詳細に行える。

## 8. おわりに

本論文では、モノにセンサノードを添付するだけで、そのモノのタイプとモノに起こった状態変化を推定するフレームワークTag and Thinkを提案した。特に本論文では人の経験や実世界に関する知識によって作成されたモデルによりモノのタイプを推定する方法を提案し、実験によりその高い精度を確かめた。今後は、人によって作成されたモデルの改良や、既存手法や本手法で推定が困難なモノのタイプについても検証していきたい。また、今回行った実験では訓練されたモデルの作成者を用意するのが困難であったため、エンドユーザを想定した被験者にモデルを作成してもらい、モデル記述方式の理解のしやすさを確かめた。今後は筆者らを含む訓練された人が作成したモデルを用いた実験を行うことにより、本論文で行った実験結果との比較を行う予定である。

謝辞 NTTコミュニケーション科学基礎研究所 Erik McDermott氏、石塚健太郎氏に多くの有益なコメントをいただいた。心より感謝申し上げます。

## 参 考 文 献

- 1) Allen, J.F.: Maintaining knowledge about temporal intervals, *Comm. ACM*, Vol.26, No.11, pp.832-843 (1983).
- 2) Bao, L. and Intille, S.S.: Activity recognition from user-annotated acceleration data, *Proc. PERVASIVE 2004*, pp.1-17 (2004).
- 3) Beckmann, C., Consolvo, S. and LaMarca, A.: Some assembly required: Supporting end-user sensor installation in domestic ubiquitous computing environments, *Proc. Ubicomp 2004*, pp.107-124 (2004).
- 4) Freksa, C.: Conceptual neighborhood and its role in temporal and spatial reasoning, *Proc. IMACS Workshop on Decision Support Systems and Qualitative Reasoning*, pp.181-187 (1991).
- 5) Fujinami, K., Yamabe, T. and Nakajima, T.: Bazaar: A conceptual framework for physical space applications, *Proc. Int'l Symp. on Ubiquitous Computing Systems*, pp.174-191 (2004).
- 6) Intille, S.S., et al.: Tools for studying behavior and technology in natural settings, *Proc. UbiComp 2003*, pp.157-174 (2003).
- 7) Kidd, C., et al.: The aware home: A living laboratory for ubiquitous computing research, *Proc. CoBuild99*, pp.191-198 (1999).
- 8) Sohn, J., Kim, N.-S. and Sung, W.: A statistical model-based voice activity detection, *IEEE Signal Processing Letters*, Vol.6, pp.1-3 (1999).
- 9) Tapia, E.M., Intille, S.S. and Larson, K.: Activity recognition in the home using simple and ubiquitous sensors, *Proc. Pervasive 2004*, pp.158-175 (2004).
- 10) Viterbi, A.J.: Error bounds for convolutional codes and an asymptotically optimal decoding algorithm, *IEEE Trans. Information Theory*, Vol.IT-13, pp.260-269 (1967).

## 付 録

図4(b)に示すオブジェクトの状態遷移図を図5に示す。状態遷移における(1→2)は、状態1から2への状態遷移を表す。ページの都合上、典型的なケースの遷移のみを示している。また、イベントの表記には省略形を用いており、たとえば“a.ro”は“acc.rotate”を示す。

(平成19年9月3日受付)

(平成20年2月5日採録)



前川 卓也 (正会員)

2003年大阪大学大学院工学部情報システム工学科卒業。2006年同大学院情報科学研究科博士後期課程修了。同年日本電信電話株式会社入社。情報科学博士。コピキタスコンピューティング, Web 情報処理の研究に興味を持つ。



柳沢 豊 (正会員)

1994年3月大阪大学工学部情報システム工学科卒業。1998年3月同大学院工学研究科博士後期課程修了。1998年4月日本電信電話株式会社入社。以来、空間データベース, センサネットワーク, 位置取得デバイス等に関連する研究に従事。現在, NTT コミュニケーション科学基礎研究所研究主任。博士(工学)。情報処理学会ほか3学会の会員。



岡留 剛 (正会員)

1988年東京大学大学院情報科学科博士課程修了。同年NTT基礎研究所入所。以来、人間の情報出力過程と調音運動の解明・形式言語の学習・センサネットワークを利用した意味処理技術の研究に従事。博士(理学)。現在, NTT コミュニケーション科学基礎研究所主幹研究員(グループリーダー)。実世界セマンティックス, コピキタスコンピューティングに興味を持つ。ACM, 日本認知科学会各会員。

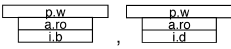
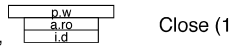
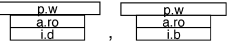
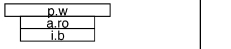

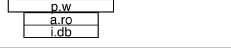


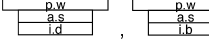
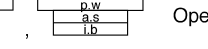
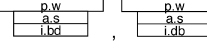
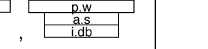
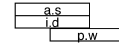
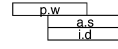
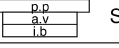

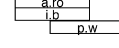
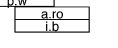
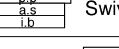
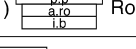


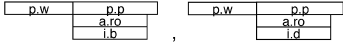

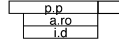
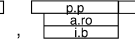
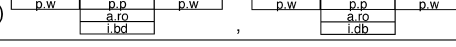

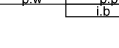
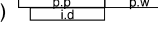
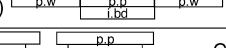
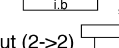
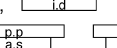
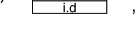
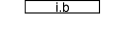
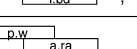
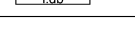
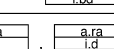
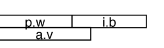
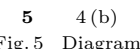
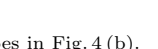
	状態	遷移とそのイベント集合
door	1 Opened 2 Closed	Open (2->1)  ,  Close (1->2)  ,  Open and shut (2->2)  , 
sliding door	1 Opened 2 Closed	Open (2->1)  ,  Close (1->2)  ,  Open and shut (2->2)  , 
back of straight chair	1 Sat 2 Not sat	Get up(1->2)  Sit(2->1)  Rock(1->1)  Slide(1->1) 
back of swivel chair	1 Sat 2 Not sat	Get up(1->2)  Sit(2->1)  Slide(1->1)  Swivel(1->1)  Rock(1->1) 
table	1 Stable	Put something(1->1) 
outside of locker door	1 Opened 2 Closed	Open (2->1)  ,  Close (1->2)  ,  Open and shut (2->2)  , 
locker interior	1 Opened 2 Closed	Open (2->1)  Close (1->2)  Open and shut (2->2) 
outside of drawer	1 Opened 2 Closed	Open (2->1)  ,  Close (1->2)  ,  Open and shut (2->2)  , 
trash can lid	1 Stable	Throw out (1->1) 
cup	1 Stable	Drink (1->1) 
alarm clock	1 Stable	Stop (1->1)  Set (1->1) 

図 5 図 4 (b) に示したオブジェクトの状態遷移図  
Fig.5 Diagrams of object types in Fig.4 (b).