

## センサネットワークにおける ECA ルールを用いたデバイス制御手法

植田 裕規<sup>†</sup> 横田 裕介<sup>††</sup> 大久保 英嗣<sup>††</sup>

<sup>†</sup> 立命館大学大学院理工学研究科 <sup>††</sup> 立命館大学情報理工学部

### 1 はじめに

本研究では、センサネットワークから取得した環境情報に基づき、アクチュエータや情報家電といったデバイス群の制御を行うためのフレームワークを提案する。

本研究で対象とするシステムにおいて、ユーザの望むサービスを提供するためには、何らかの形式で要求を記述し、システムに登録する必要がある。しかし、ネットワーク上には、センサやデバイスが多数存在することが考えられるため、ユーザがこれらの機能や位置のすべてを把握して記述することは困難である。また、既存のシステムは、センサネットワークを対象としたものが少ない。これらの問題を解決するために、本研究では、センサネットワークを用いたデバイス制御における動的な構成変化に対応するフレームワークを提案している。本フレームワークは、ECA(Event-Condition-Action)ルールを用いた動作定義の記述を採用している。この定義に従って記述することで、センサネットワークやデバイスに対して抽象的な問い合わせや動作定義を行うことが可能となる。

以下、本稿では、本フレームワークの概要と記述可能な動作定義について述べる。

### 2 センサネットワークにおけるデバイス制御

本研究では、センシングデータをイベントとしたデバイス群を制御するためのフレームワークを提案する。本フレームワークは、次に示す 3 つの機能要件を満たす。

#### 1. システム構成の動的な変化に対応

現在、センサネットワークやデバイスには様々な機能や種類が存在している。今後もこれらの機能や種類は増加することが考えられる。そのため、稼働中のシステムに新たなセンサネットワークやデバイスを追加・削除したいという要求が発生することが考えられる。しかし、これによりシステムの構成が変化した場合に、膨大な数のセンサやデバイスについて事前に把握することは困難である。したがって、これらをオブジェクトとして管理するディレクトリサービスを提供することで、空間に対する記述をサポートする。これにより、ユーザがセンサネットワークやデバイスを管理する負担を軽減する。

#### 2. センサネットワークを適用した様々な環境に対応

既存のフレームワークでは、センサネットワークが設置されている環境を想定しているものは少ない。センサネットワークを利用した環境は、センサ群を

1 つのグループとして扱うことが可能であり、センサ単体を対象とした環境と比較して、空間に対する問い合わせが容易に実現できる。また、センサネットワークは、基地局によるセンシングデータの集約を行うため、ローカルネットワークとセンサネットワークのレイヤを分離することが可能である。これにより、システムの効率化が見込める。システム構築における機器の設置においても、センサネットワークは、アドホックに通信を行い自動的にネットワークを構築するため、多数のセンサデバイスを個別にネットワークに組み込む場合と比べて、作業負担が少なくなる。これらの理由から、センサネットワークを利用することで多くの利点を得ることができる。本フレームワークでは、このような環境に対応するために、センサネットワークに適したフレームワークを提供することで、開発者とエンドユーザの作業負担を減らすことを目指す。具体的には、センサネットワークを容易に設置するための API と複数のセンサネットワークへ同時に問い合わせ可能な擬似 SQL による問い合わせ機構を提供する。

#### 3. システム構成を意識しない抽象度の高い記述

前述のように膨大な数のセンサやデバイスを事前にすべて把握することは困難である。本フレームワークは、エンドユーザが事前にこれらを把握する必要のない、抽象度の高い動作定義が記述可能な環境を提供する。これを実現するために、場所と時間を用いた空間に対する記述を可能とし、動作定義の際にはディレクトリサービスによって管理されるセンサネットワークやデバイスの情報を提供する。また、擬似 SQL による問い合わせ記述を組み合わせることでより詳細な環境条件の設定を可能にする。これらをサポートすることで、エンドユーザによる動作定義の記述効率の向上を図る。

### 3 ECA ルールによる動作定義の記述

本フレームワークでは、抽象度の高い動作定義の記述を実現するために、ECA(Event-Condition-Action)ルールを用いたイベント駆動型のルール記述を採用している。ECA ルールは、ユーザの要求をイベントとアクションに分けて記述することで、直感的に動作を定義することが可能なためである。

#### 3.1 デバイスの動作定義の構文

本フレームワークにおいて提供される動作定義の構文を図 1 に示し、特徴となるセンサネットワーク環境を想定した動作定義について述べる。

本フレームワークのイベント検出の設定は、イベントの発生場所と時間を組み合わせることで決定する。こ

A device control technique using ECA rule for sensor networks

Yuki Ueda<sup>†</sup>, Yusuke Yokota<sup>††</sup>, and Eiji Okubo<sup>††</sup>

<sup>†</sup> Graduate School of Science and Engineering, Ritsumeikan Univ.

<sup>††</sup> College of Information Science and Engineering, Ritsumeikan Univ.

```

DEFINE 動作定義名 [定義ユーザ名]
EVENT 発生イベント名(AND, OR の記述も可能)
  WHEN [開始年月日時刻 – 終了年月日時刻]
  WHERE 場所名
IF [(擬似SQLによる問い合わせ)(> | < | = | >=
| <=)(任意の値)](AND, OR の記述も可能)
THEN
  DEVICE 制御デバイス名
  WHERE 場所名
  DO 操作
  INFORMATION 操作に必要な情報

```

図1 デバイスの動作定義の構文

の場所は、本フレームワークにより提供される API によって、記述時のシステム構成で定義可能な場所を取得できる。この場所を指定することで、その場所で定義可能なイベントを限定する。同様に、THEN 項のデバイス制御に関する記述に関しても、ディレクトリサービスから提供されるデバイスや場所の情報を指定することで抽象的な記述が可能となる。IF 項に表記される擬似 SQL は、イベント発生時の環境状況を条件に指定してデバイス制御の条件付けを行う。この擬似 SQL の構文を図2に示す。

擬似 SQL では、集合演算関数またはセンシングデータの種類を指定することでデータを取得する。この際に、FROM に記述したセンサネットワークの設置場所の情報に基づいて、問い合わせの発行対象となる複数のセンサネットワークを特定し、それぞれのネットワークの仕様に合わせた問い合わせメッセージを送信する。これによって、複数のセンサネットワークを仮想的な一つのセンサネットワークとして扱うことが可能である。

### 3.2 制御記述の例

図1のEVENT, IF, THEN は、それぞれECA ルールのEvent, Condition, Action に相当する。

EVENT を記述するには、場所と時間を指定しその条件でシステムに登録されている定義可能なイベントを選択する。この際の場所の指定には、登録されている場所、ユーザの位置などが選択可能である。例えば、部屋Aに入ったときにイベントを検知する場合は、以下のように記述することができる。

```

EVENT ENTER_ROOM_A WHEN 2009/01/15/09:00
- 2009/01/30/23:59 WHERE ROOM_A

```

なお、図1に記載されているように、AND と OR による論理演算も可能である。

IF 項では、広範囲におよぶ環境状況を条件とすることができる。例えば、部屋AとBの平均温度を条件としたい場合、

```

IF (AVG(temperature) FROM ROOM_A, ROOM_B
WHERE temperature <= 50) <= 20

```

のように記述する。このような問い合わせに対して、デ

```

集合演算関数 (データの種類) | データの種類
FROM センサネットワークの設置場所 [WHERE (条件式)]

```

図2 擬似SQLの構文

バイス制御を行う条件を不等号と値で指定することで、「部屋AとBの50度以下の値のうち平均温度が20度以下のときデバイス制御を実行する」というような記述ができる。このとき、50度以下と指定する理由は、他のデバイスや環境の影響で局所的にセンサが誤検知する可能性が考えられ、それらを除く必要があるためである。

THEN 項は、EVENT 項と同様に登録されている場所を指定することで、制御可能なデバイスを限定し指定する。制御したいデバイスが部屋Aの空調機器である場合を例に挙げる。空調機器は、冷暖房の機能を備えており、温度を指定することで室温を調節するものとする。この場合の記述例は次のようになる。

```

THEN DEVICE AIR_CONDITIONER
WHERE ROOM_A DO COOLING INFORMATION 20

```

この例におけるINFORMATIONは、制御に必要な情報を値として渡すために用いる。したがって、ここでは20度を表す値を指定している。どのような情報を必要とするかは、デバイスをシステムに登録する際に情報としてフレームワークが要求する。

### 3.3 考察

ECA ルールを採用した既存のフレームワークとして[1]が挙げられる。このフレームワークは、プラグイン形式を用いることで各センサやデバイスの機能を動的に管理することが可能である。しかし、センサネットワークの利用を想定しておらず、本フレームワークのように環境状況を条件に指定する場合、それぞれのセンサごとに値と条件を設定する必要があり記述の負担が大きくなってしまふ。また、動作定義の記述には、ルールエディタが提供されており、記述候補を提示することで記述の負担の軽減を図っているが、本フレームワークのような場所条件の指定による定義可能なイベントの限定などをサポートしていない。そのため、膨大な数となることが予想される環境では、ユーザにとってこれらの把握が困難であると考えられる。

本フレームワークでは、場所に合わせた定義やセンサネットワークを想定した記述を可能にすることで、上記のような負荷が軽減できると考えている。

## 4 おわりに

本稿では、センサネットワークから取得可能なセンシングデータを用いて、特定の空間に設置されたデバイスを制御するためのフレームワークについて述べた。今後は、提案フレームワークのプロトタイプの実装と評価を行う予定である。

### 参考文献

- [1] “Wearable Toolkit ウェブ サイト,” <http://wearable-toolkit.com>