

## 多種多様なデータストリームを用いた家電制御システムの構築

鈴木 誠 二<sup>†1</sup> 峰野 博 史<sup>†1</sup>  
太田 賢<sup>†3</sup> 水野 忠 則<sup>†2</sup>

インターネット、無線技術、センサ技術などの発達に伴い、小型化・低価格化したセンサノードを宅内に設置することで、多種多様なデータを収集することが可能となってきた。その結果、収集したデータを利用して温度・湿度による住環境制御や健康モニタリングによる体調管理など、より柔軟かつ質の高いサービスを実現できる可能性が現実味を帯びている。このような家庭内での高付加価値サービスを実現するために、単一のセンサデータだけでなく複数のセンサデータを統合的に利用してユーザの状況を把握する研究が注目されつつある。しかし、センサデータだけでなく、天気やニュース、ソーシャルデータといった多種多様なデータストリームも組み合わせてユーザの状況を把握する研究に関してはまだ十分に行われているとは言い難い。

本研究では、ユーザの趣味趣向や状況に応じたサービスの提供に関して、ホームネットワークにおける多種多様なデータストリームを利用した家電制御システムを検討する。特にデータストリームの統合処理に関して、データストリーム処理を目的として設計された DSMS を利用することで、柔軟なデータストリームの追加や削除、制御条件の更新などを実現し、多種多様なデータストリームを活用した家電制御を実現できるプロトタイプシステムを構築した。

### Development of Data Streams-based Home Appliances Control System using DSMS

SEIJI SUZUKI,<sup>†1</sup> HIROSHI MINENO,<sup>†1</sup> KEN OHTA<sup>†3</sup>  
and TADANORI MIZUNO<sup>†2</sup>

With the development of small and low-cost wireless sensor node, it is easy to gather wide variety of data in home environment using wireless sensor networks (WSN). Although the WSNs are currently being assumed in a variety of applications ranging from home appliances control to health monitoring, and from home to industry, there is no system assumed using the heterogeneous sensor data streams as integrated system.

In this paper, we propose a data streams-based home appliances control system using data stream management system (DSMS). This system provides an

integrated environment for heterogeneous sensor data streams. We can easily add and delete the data streams to change the event-detection conditions for controlling home appliances.

#### 1. はじめに

近年、最新データを自発的に配信するストリーム型情報源であるデータストリームが多数存在し、その数は年々増加し続けている。例えば、株・金融情報、天気、ニュースなどのオンラインデータや、システムのログデータ、カメラやマイクなどのマルチメディアデータだけでなく、温度、照度、モーションなどのセンサデータはデータストリームとして考えることができる。また、twitter や mixi といった、個人のつぶやきや書き込みなどもデータストリームとして考えることができる。データストリームの増加の理由として、デバイス技術やネットワーク技術の発展・一般社会への浸透が考えられる。デバイスの小型化・低価格化により、センサノードなどのデバイスを大量に設置し、多種多様なセンシングデータを収集することが可能となった。また、ネットワークの技術の発展・普及により、インターネット上でニュースや新聞を見たり、音楽や動画を楽しんだりなど多種多様なオンライン情報を活用する社会が一般的になりつつある。

このように多種多様なデータストリームが存在するため、それらを利用してユーザの趣味趣向や状況に応じたサービスを提供しようとする動きに注目が集まっている<sup>1)-3)</sup>。家庭内においても、小型化・低価格化したセンサノードを設置することで、温度、湿度、照度、音、においなど、多種多様なデータを収集することが可能となり、温度・湿度による自動環境制御や健康モニタリングによる体調管理など、より質の高いサービスの実現が求められている。

ユーザの趣味趣向や状況に応じたサービスを提供するためには、単一のデータを用いるよりも複数のデータを統合して利用するほうが的確なサービスを提供することができる<sup>4)</sup>。近年、このような家庭内での高付加価値サービスを実現するために、単一のセンサデータだけでなく複数のセンサデータを統合的に利用してユーザの状況を把握する研究が増加の傾

<sup>†1</sup> 静岡大学情報学部

Department of Computer Science, Shizuoka University

<sup>†2</sup> 静岡大学創造科学技術大学院

Graduate School of Science and Technology Shizuoka University

<sup>†3</sup> 株式会社 NTT ドコモ 先進技術研究所

NTT DOCOMO, Inc. Research Laboratories

向を見せている<sup>5)</sup>。しかしながら、これらの研究は、センサノードの開発やセンサデータの解析方法、異なる複数のセンサノードからセンサデータをどのように管理するかなど、センサデータだけを対象としており天気やニュース、ソーシャルデータといったセンサデータ以外の多種多様なデータストリームを組み合わせてユーザの状況に沿ったサービスを提供する研究に関してはまだ十分であるとは言いがたい。そこでセンサデータ以外のデータストリームもセンサデータと組み合わせて利活用し、よりユーザの趣味趣向を考慮したサービスを提供する環境の実現が必要であると考えた。

本研究では、ユーザの趣味趣向や状況に応じたサービスの提供に関してホームネットワークに焦点を置き、多種多様なデータストリームを利用した家電制御を実現するための環境の構築について検討する。特にデータストリームの統合処理に関して、データストリーム処理を目的として設計された DSMS (Data Stream Management System) を利用することで、柔軟なデータストリームの追加や削除、制御条件の更新などを実現する。本環境においてユーザは、家電製品を制御する条件として多種多様なストリームデータを選択し、自分の好きな条件を柔軟に設定できる。

以降、2章で関連研究について述べ、3章では提案する DSMS を用いたデータストリームと家電の連携システムの設計を行う。4章では、DSMS として StreamSpinner を用いたプロトタイプシステムの実装について述べ、5章にて本検討のまとめと今後の課題について述べる。

## 2. 関連研究

現在、種類の違うセンサノードから収集するセンサデータを利用して、複合イベント判定を行うための研究はいくつか行われている<sup>6)~9)</sup>。複数のセンサデータを連携させ、複合イベント判定を行う方法は、図1、図2、図3に示す三つの方式に分類することができる。

方式1は、各ゲートウェイで行った単一イベントの判定結果を集めて複合イベント判定を行う方法である。サーバは、ユーザから複合イベントの問合せを受け取ると、複合イベントを複数の単一イベントに分解する。サーバは、単一イベント判定に利用するセンサデータを管理するゲートウェイに単一イベントを送信する。ゲートウェイはサーバから受け取った単一イベントをもとに単一イベント判定を行う。ゲートウェイは単一イベント判定の結果をサーバに返す。サーバは、各ゲートウェイから受け取った単一イベント判定の結果を組み合わせ複合イベント判定を行う。このように、方式1は各ゲートウェイで行った単一イベント判定の結果を組み合わせ複合イベント判定を行う。

表1 複合イベント処理の方式ごとの比較

	方式1	方式2	方式3
サーバの負荷	大	小	中
ゲートウェイの負荷	中	大	小
サーバと各ゲートウェイの連携	単純	複雑	単純
遅延	小	中	大
新しいゲートウェイの追加の容易性	やや容易である	容易でない	容易である

方式2は、複数のゲートウェイで木構造を造り複合イベント判定を行う方法である。サーバは、ユーザから複合イベントの問合せを受け取ると、その複合イベントをゲートウェイAに送信する。複合イベントの問合せを受け取ったゲートウェイAは、複合イベントからゲートウェイAが管理するセンサノードのデータで判定できる単一イベントと残りの複合イベントに分解する。単一イベントをゲートウェイAに登録し、残りの複合イベントを他のゲートウェイに送信する。この処理を複合イベントがすべて単一イベントに分解されるまで行う。ゲートウェイは、登録した単一イベントと、複合イベントの問合せを送信したゲートウェイの条件が満たされたときに、イベント情報を問合せ送信元のゲートウェイに送信する。ゲートウェイAが条件を満たした場合、ユーザが要求を出した複合イベントの条件が満たされたこととなり、サーバにそのイベント情報を知らせる。このように、方式2は複数のゲートウェイで木構造を形成して複合イベント判定を行う。

方式3は、DBMSを用いてイベント判定を行う方法である。すべてのゲートウェイは、収集したセンサデータをDBMSに送信する。DBMSは、受信したデータをデータベースに書き込む。サーバは、ユーザから受け取った複合イベントの問合せをもとにDBMSに複合イベントの条件を満たす新規データを定期的に要求する。DBMSは、条件を満たすデータがDBMS内に存在する場合、サーバに該当データを送信する。サーバがDBMSから該当データを受け取った場合、複合イベント条件が満たされたこととなり、ユーザにイベント情報を知らせる。このように、方式3はDBMSを用いて複合イベント判定を行う。

上記の3つの方式の特徴の違いを表1に示す。方式1の利点は、イベント情報の通信回数が少ないこととディスク書き込みを行わないことにより他の方式と比べて複合イベント判定による遅延が小さい点である。欠点は、サーバが複数のゲートウェイとメッセージのやり取りを行う必要があるため、他の方式と比べてサーバの負荷が大きい点である。方式2の利点は、サーバはイベント判定時、複数のゲートウェイを監視する必要がないため、他の方式と比べてサーバの負荷が小さい点である。欠点は、ゲートウェイが単一イベントの判定や他のゲートウェイと連携をとる必要があるため、他の方式と比べてゲートウェイの負担が大き

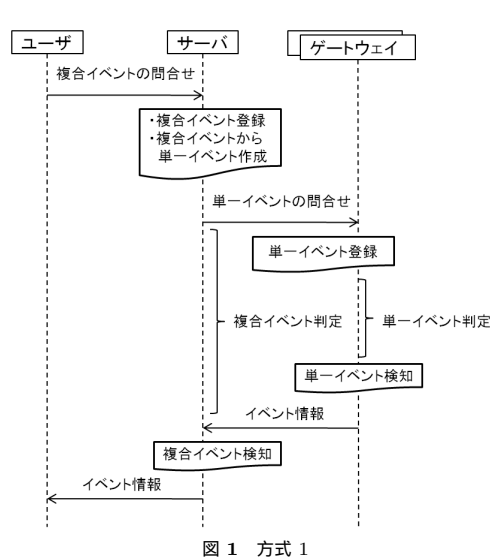


図1 方式1

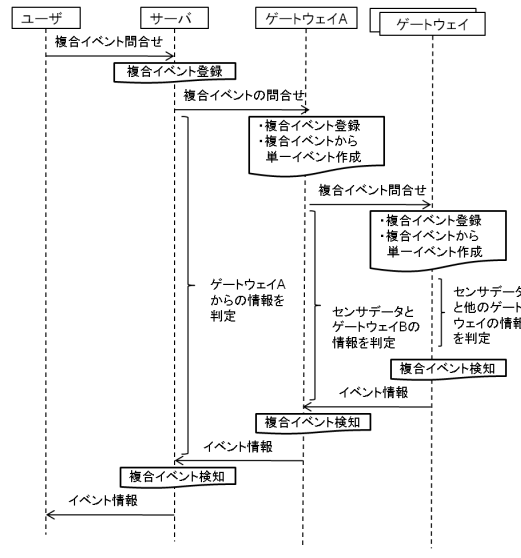


図2 方式2

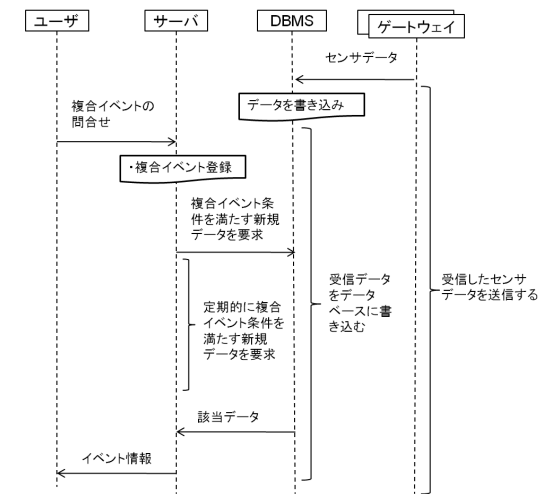


図3 方式3

い点である。方式3の利点は、サーバがゲートウェイの監視をする必要がないため他の方式と比べてサーバの負担が小さい点と、ゲートウェイは、センサデータをDBMSに書き込む以外の処理が必要ないためゲートウェイの負担が小さい点である。また、新しいゲートウェイの追加時は、新しいテーブルを追加するだけでいいため他の方式と比べて容易である。欠点は、DBMSにデータを書き込みことにより発生する遅延と、データベースにデータを要求するタイミングが定期的であるため発生する遅延により、他の方式と比べて複合イベント判定による遅延が大きい点である。

本システムでは、多種多様なデータストリームを利用するために、新しいデータストリームの追加が容易である必要がある。そのため、新しいデータストリームの追加が最も容易である方式3が一番適していると考えられる。しかし、DBMSを用いてデータストリームを処理する場合、ディスク書き込みにより発生する遅延により、処理が間に合わないということが考えられる。そこで本システムを実現するために、新しいデータの追加が容易であり、高頻度に送信されるデータストリームの処理にも対応できるデータ処理システムが必要となる。

### 3. 設 計

#### 3.1 概 要

本システムでは、多種多様なデータストリームを用いて家電制御を行うことを想定している。提案するシステムを図4に示す。ユーザは、制御する家電の情報と条件を設定する。システムでは設定された問合せに従ってイベント判定を行う。条件が満たされた時、設定された家電に制御命令を送信する。多種多様なタイプのデータストリームを利用するためには、新しいデータストリームの追加の容易性が求められる。そこで、ゲートウェイはデータストリームからデータを収集してサーバに送るのみとすることで、設計を単純化し新しいデータストリームの追加を容易にする。このような方式では、従来データを全て蓄積管理できるDBMSが用いられる。しかし、DBMSのデータ処理モデルでは、ディスク書き込みによる遅延が発生するため、即時性が求められるイベントに関しては不向きである。そこで、我々は、データストリーム処理を目的として設計されたデータ処理システムであるDSMSを用いてデータストリームの処理を行うことにした<sup>10)</sup>。本システムにおける機能モジュールの

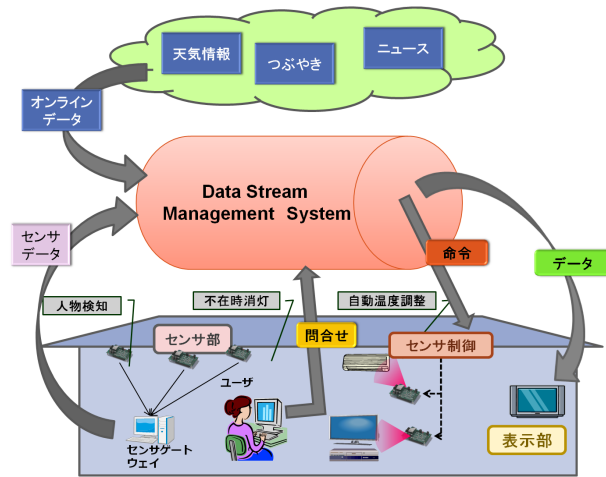


図 4 データストリームを用いた家電制御システム

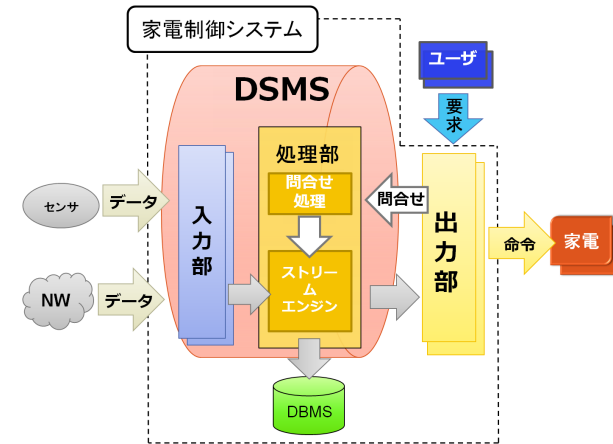


図 5 提案システムの機能モジュール構成

構成を図 5 に示す。

### 3.2 DSMS

DSMS は、データストリーム処理を対象とした駆動型アーキテクチャである。DSMS には大きく以下の二つの特徴がある。

一つ目の特徴はデータの処理方法である。DBMS は、新規データを処理する場合、まずデータをデータベースに書き込み、蓄積後に問合せ要求を発行し検索結果を得るため、データベースへの書き込みとアクセスによる遅延が発生する。このような遅延を避けるために、DSMS では、到着データをウィンドウと呼ばれるメモリ領域に一時的に保存し、メモリ上で問合せによるイベント判定を行う。ディスクへの書き込みやアクセスを行わず、全てメモリ上で処理を行うため、DBMS と比べて結果を出力するのが速い。しかし、DBMS と違いデータを永続的に保存することはできないため、新規データを保存したい場合は、イベント判定後にデータを DBMS に送り蓄積する必要がある。

二つ目の特徴は問合せ要求である。データストリーム処理では、ユーザはあらかじめ問合せをシステムに登録しておき、出力すべきデータが到着した場合、結果を出力することが望まれる。DBMS は基本的に、一度の問合せ要求に対して、一度の結果出力を行う。そのため、DBMS を用いてデータストリーム処理を行う場合、定期的に問合せを発行する必要がある。

表 2 DSMS と DBMS の比較

	DBMS	DSMS
処理速度	遅い	速い
蓄積	できる	できない
連続イベント判定	定期的	即時と定期的

ある。これに対して DSMS は、CQ (Continuous Query) と呼ばれる問合せ方式を用いる。CQ は、ユーザが問合せ要求を発行するとシステムにその問合せを登録し、新規データが到着する度にシステム内部で登録された問合せを実行する。そのため、DBMS のように何度も問合せ要求を発行する必要がなく、連続的なデータストリーム処理に向いていると考える。

以上より、DSMS はデータストリームのような新規データに有効であり、DBMS は蓄積された過去データの処理に有効であると考えられる。DSMS と DBMS の比較結果を表 2 に示す。表 2 より DSMS と DBMS は、相互補完の関係にあるといえる。

### 3.3 処理部

ユーザからの問合せを解析する問合せ処理と、問合せ処理結果に従ってデータストリームの処理を行うストリームエンジンは、既存のものを用いる。本システムでは、ユーザは家電ごとに好きな条件を設定する。そのため、大量の問合せに対応するストリームエンジンの導

入が適していると考えられる。また、問合せは、ユーザにがより自由な条件設定を可能とするため、新規データが到着する度に起動する問合せ型や、定期的に問合せを実行するタイマー型、その両方で起動する問合せなど様々なタイミングで実行可能な問合せ形式が必要となる。問合せ記述方式としては SQL ライクな記述方式を想定している。

### 3.4 入力部

多種多様なデータストリームを扱うためには、あらかじめ送信されるデータストリームの形式に合わせた入力部を作成する必要がある。入力部はデータストリームの数だけ必要となる。センサノードやインターネットなど外部から収集したデータストリームを属性ごとに切り分け、ストリームエンジンにデータを送る。また入力部は、外部からのデータを収集するだけでなく、自らが現在時間などのデータを作成してストリームエンジンに送ることことも考えられる。本システムでは、稼働中に新しいデータストリームの追加が行われることが想定される。データストリームの追加のためにシステム全体を止めることはできないため、入力部では、データストリームの変動によってシステムを稼働したまま入力部の追加・変更・削除を容易に行うことが要求される。

### 3.5 出力部

出力部では、DSMS に問合せ要求を送信する機能、問合せ結果を受信する機能、受信データに従って家電に制御命令を送る機能が必要となる。ユーザから制御する家電の情報と条件を受け取る。受け取った条件に従い、問合せを DSMS に送信し、DSMS から問合せ結果を受信するまで待ち状態となる。DSMS から結果を受信した場合、制御要求が出された家電に指定された制御命令を送信する。ユーザが自由に条件の変更が行えるように、入力部同様システムを稼働したまま出力部の追加・変更・削除を容易に行うことが要求される。

### 3.6 家電制御シーケンス

家電制御シーケンスは図 6 のように行われる。DSMS は、ゲートウェイやインターネットなど多種多様なデータストリームからデータを収集する。ユーザは、操作する家電と操作条件を出力部に入力する。出力部は、ユーザの設定した条件を DSMS に登録する。DSMS は、収集しているデータがユーザの設定した条件を満たすか判定を行う。条件が満たされた場合、DSMS は該当データを出力部に送信する。出力部は、ユーザが設定した家電に向けて制御命令を送信する。

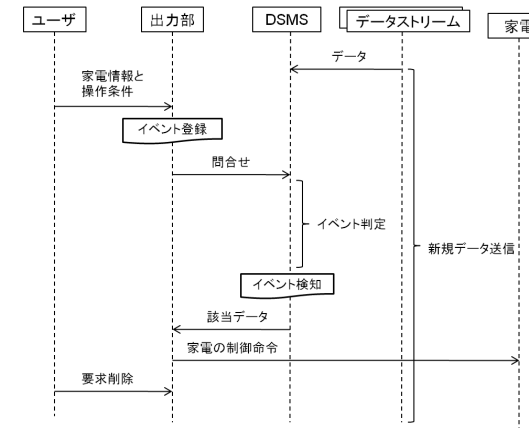


図 6 家電制御シーケンス



図 7 センサノード

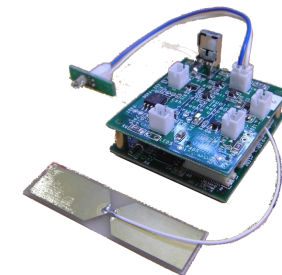


図 8 学習型赤外線リモコンノード

## 4. プロトタイプシステムの開発

### 4.1 外部モジュール

センサノード センサノードは、環境センサとして株式会社ルネサスソリューションズ<sup>11)</sup>製の無線センサノード(図7)を使用した。このセンサノードは、温度、照度、湿度、モーション、湿度を収集することができる。また、このセンサノードはセンサノード間で無線通信を行うことが可能であり、センサネットワークを形成する。そして、室内に設置した全てのセンサのデータは、センサネットワークを介してシンクノードに送信される。このセンサ

表 3 既存の DSMS

名前	問合せタイプ	記述方法	分散化	その他
STREAM	到着型 タイマー型	SQLライク	×	Load shedding (Random)
TelegraphCQ	到着型	SQLライク	×	Eddyによる動的最適化
Aurora	到着型	XML-QL	×	Load shedding(Semantic)
NiagaraCQ	到着型 タイマー型	NIAGARACQ COMMAND LANGUAGE	×	複数問い合わせ最適化
Borealis	到着型	XML-QL	○	Aurora + Medusa(分散化)
StreamSpinner	到着型 タイマー型	SQLライク	○	分散化 複数問い合わせ最適化

```
Master master_source1
[INSERT INTO table_name]
SELECT attribute1,...
FROM source1[window_size]...
WHERE predicate1 AND ...
```

図 9 StreamSpinner の問合せ記述方式

ノードを部屋の天井に 18 個設置することで、部屋全体の環境情報の収集を行った。

センサゲートウェイ センサゲートウェイは、UbuntuOS を搭載したノート PC 上で実装を行った。この PC とセンサデータが集まるシンクノードとは、常にシリアルケーブルで接続状態にあり、シリアル通信を行いデータを収集する。収集したデータは UDP 通信を用いて DSMS の入力部に送信される。

学習型赤外線リモコンノード 本論文の実装では、家電製品への制御命令を送信する手段として学習型赤外線リモコンノード(図 8)を使用した。この学習型赤外線リモコンノードは、家電のリモコンで使用する赤外線信号を記憶させることができる。シリアル通信を用いて PC から学習型赤外線リモコンノードに制御命令を送ることで、記憶した赤外線信号を家電に向けて送信することができる。本論文の実装では、照明などの赤外線信号をあらかじめ学習型赤外線リモコンノードに学習させ、制御を行った。また、この学習型赤外線リモコンノードは個々が通信機能を持っていて、無線通信を用いることでシリアルケーブルでつながれた学習型赤外線リモコンノードから別の学習型赤外線リモコンノードに制御命令を送信することができる。そのため、制御する家電製品一つ一つに学習型赤外線リモコンノードを組み合わせることで、複数の家電を制御する環境を構築することが可能となる。

#### 4.2 家電制御システム

StreamSpinner 本システムで使用する DSMS を決定するために、既存の DSMS<sup>(12), (13)</sup>

の調査を行った。既存の DSMS の例を表 3 に示す。

本システムでは、大量の問合せに対応できるストリームエンジンと、柔軟な問合せ記述方式が必要となる。以上の要求を満たす DSMS として本システムでは StreamSpinner<sup>(14), (15)</sup> を用いてシステムを構築する。StreamSpinner には二つの特徴がある。一つは、複数問い合わせ最適化である。複数問い合わせ最適化は、複数の問合せから共通演算を抽出し、本来複数回必要だった処理を一度で済ませる手法である。複数問い合わせ最適化を用いることで大量の問合せに対応することができる。もう一つは、問合せの実行タイミングを明示的に指定できる点である。StreamSpinner の問合せ言語は、新規データが到着する度に起動する問合せや、定期的に行うタイマー型、その両方で起動する問合せなど問合せの実行タイミングを自由に指定できるため、よりユーザは自由な条件設定を図 9 のような SQL ライクの言語で行うことができる。

入力部 本論文の実装では、三種類の入力部を実装した。一つ目は、UDP 通信を行うための入力部である。この入力部は、センサデータを受け取る役割を負う。UDP 通信により受信したデータは、属性ごとに切り分けられ、ストリームエンジンに送られる。二つ目は、天気情報を収集するための入力部である。この入力部では、一時間ごとに yahoo の天気 html ソースを取得する。収集した html ソースから、一日の天気情報だけを取り出し、ストリームエンジンに送られる。入力部は、データストリームの変動によってシステムを稼働したまま入力部の追加・変更・削除を容易に行うことができる。三つ目は、定期的に現在時間を配信する入力部である。この入力部は、定期的の日付、現在時間、曜日を配信するため、タイマー型の条件を設定するとき用いられる。

出力部 出力部では、DSMS に問合せ要求を送信する機能、問合せ結果を受信する機能、受信データに従って家電に制御命令を送る機能を実装した。ユーザから制御する家電の情報と条件を受け取り、受け取った条件に従い問合せを StreamSpinner に送信する。送信後は、StreamSpinner から問合せ結果を受信するまで待ち状態となる。結果を受信した場合、

```
Master SensorData
SELECT SensorData.*
FROM SensorData, Weather
WHERE SensorData.motion = 1
      AND Weather.type = "Rainy"
```

図 10 傘忘れ防止

```
Master SensorData
SELECT SensorData.*
FROM SensorData, Weather
WHERE SensorData.Timestamp > "05:00:00"
      AND SensorData.temperature > 20
      AND SensorData.Humidity < 30
      AND Weather.type = "Sunny"
```

図 11 洗濯機

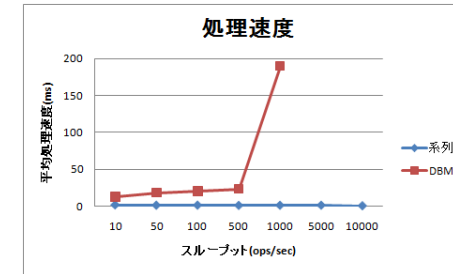


図 12 負荷実験の結果

制御要求が出された家電に指定された制御命令を学習型赤外線リモコンノードに送信する。出力部は、入力部同様システムを稼働したまま出力部の追加・変更・削除を容易に行うことが可能である。

TCP コネクションサーバ 携帯電話を用いて外部から家電の制御を行うことを想定し、携帯電話と StreamSpinner をつなぐ TCP コネクションサーバの実装を行った。携帯電話からの StreamSpinner への問合せを直接送信することはできないが、TCP コネクションサーバを経由することで、携帯電話からの StreamSpinner への問合せを送信することができる。携帯電話から問合せ要求が送信されると、TCP コネクションサーバはソケットを引数にスレッドを作成する。スレッドの一つ一つが出力部プログラムとなる。TCP コネクションサーバにより携帯電話を用いることで外部から家電の制御を行うことができる。

#### 4.3 動作シナリオ

以上の環境を用いて以下の三つのシナリオを行った。

一つ目のシナリオは、センサデータとオンライン上の天候データを利用したステレオ制御による傘忘れ防止機能を作成した。このシナリオは、センサデータとオンラインデータを統合利用することを目的として行った。玄関に設置されたセンサノードから収集するモーションにより、人が玄関にいることを検知する。また、オンライン上の天候データを利用することで、その日は傘を持って移動するほうがいいということを検知する。この二つの情報を組み合わせることにより、雨が降る可能性がある場合、家を出る時にステレオで音楽を流すことで、ユーザに傘を持ち歩くことを促す傘忘れ防止機能を作成した。この機能を実現するために用いた問合せを図 10 に示す。実際にこの傘忘れ防止機能を動作させ、正しく動作することを確認した。

二つ目のシナリオは、天気データと環境センサを利用した洗濯機の制御を行った。このシナリオは、センサデータとオンラインデータを統合利用と家電制御の条件の動的な変更を目的として行った。このシナリオはユーザが朝起きた時に洗濯日和である場合のみ、洗濯が完

了して干すだけの状態になっているとうれしいであろうという想定のもと行った。オンライン上のデータから天候が雨ではないことを確認する。ベランダに設置したセンサノードから時間、温度、湿度を確認する。これらの情報を組み合わせることにより、洗濯日和の日だけ稼働する洗濯機を作成した。この機能を実現するために用いた問合せを図 11 に示す。実際にこの機能を動作させ、正しく動くことを確認した。また、洗濯物のたまり具合によっては、ある程度洗濯日和ではなくても洗濯を行いたいということが考えられる。そこで、洗濯機を動かす条件である温度と湿度の動的な変更を行った。条件の変更後に動作させた結果、正しく動くことを確認した。

三つ目のシナリオは、遠隔地からのエアコンの制御機能を作成した。このシナリオは、携帯電話と TCP コネクションサーバを利用することで、外部から家電を操作することを目的として行った。帰宅途中にエアコンをオンにして部屋を暖めておくということを想定し、問合せ条件は、すぐに満たすよう時間だけを用いて設定を行った。これにより、ユーザは家に帰る直前に外部からエアコンのスイッチをオンにできるため、帰宅時に家の中が寒いという状況を回避できる。

#### 4.4 評価

本システムでは、全てのゲートウェイからデータを DSMS に集めるため負荷が集中する。そこで、DSMS の負荷試験を行い実際に高頻度に送られてくるデータの処理が可能か検証した。負荷実験では、実験用に作成したデータ送信プログラムを用いて送信頻度を変えながらデータを送信し、データの処理速度として、データが入力部に到達してから出力部にデータが届くまでの時間を計測した。比較対象として DSMS と同様の実験を DBMS である postgres でも行った。負荷実験は、Windows 7 搭載の PC 上で行った。負荷実験の結果を図 12 に示す。

DBMS は、負荷が高くなると急激に遅延が発生し、データ処理が間に合わなくなった。対して、DSMS は、負荷を増加させても一定の速度でデータ処理を行うことができた。また、DSMS は、平均的に 2ms でデータ処理を行うことができ、DBMS よりも速くデータ処理を行うことができた。以上により、DSMS は DBMS と比べて、頻度の高いデータの処理に対して強く、また、高速に処理できることを確認した。

## 5. おわりに

本研究では、ユーザの趣味趣向や状況に応じたサービスの提供に関して、ホームネットワークにおける多種多様なデータストリームを利用した家電制御システムの検討を行った。特にデータストリームの統合処理に関して、データストリーム処理を目的として設計された DSMS を利用することで、柔軟なデータストリームの追加や削除、制御条件の更新などを実現し、多種多様なデータストリームを活用した家電制御を実現できるプロトタイプシステムを構築した。また、DSMS と DBMS の負荷実験を行い、DSMS を用いた本システムは、DBMS を用いて構築する場合と比べて、より高頻度のデータに対応できること、より素早くデータ処理が可能であることを確認した。本論文においては家電製品の制御方法として、学習型赤外線リモコンノードを用いて照明、洗濯機、エアコンの制御を行ったが、近年、家電製品のネットワーク化が進んでいるため、ネットワークを介して家電製品に直接制御命令を送信することが可能となり、より多くの種類の家電製品に対して本システムを利用することができると考えられる。

今後は、過去データの考慮<sup>16)</sup> やデータストリームの種類を増やすことで、より汎用的なシステムを目指すとともに、大量のデータストリームの処理と遠隔地のデータストリームの利用を考慮した分散 DSMS の利用を検討して行く。また、一般ユーザにとって SQL 記述方式はなじみがないものであるため、ユーザが使いやすいよう SQL 記述を隠したインターフェイスの提供も検討する。

## 参 考 文 献

- 1) 中澤仁, 徳田英幸: センサアクチュエータネットワークの情報処理基盤, 情報処理学会誌, Val.51, No.12, pp.1127-1135 (2010).
- 2) 植原啓介: プローブ情報システム: 車載センサを活用した環境情報の取得, 情報処理学会誌, Val.51, No.12, pp.1145-1149 (2010).
- 3) 小林亜令, 木實新一: 携帯電話を用いたセンシングの可能性と課題, 情報処理学会誌, Val.51, No.12, pp.1157-1163 (2010).

- 4) Jurgen Dunkel: On Complex Event Processing for Sensor Networks, ISADS, pp.1-6 (2009).
- 5) 石田建一, 佐藤誠: HEMS による家庭用冷暖房・照明エネルギーの削減効果の検証, 日本建築学会環境系論文集, No.595, pp.57-64 (2005).
- 6) Jagrati Agrawal, Yanlei Diao, Daniel Gyllstrom, and Neil Immerman: Efficient Pattern Marching over Event Streams, In Processing of the ACM SIGMOD International Conference on Management of Data (2008)
- 7) 幸島明男, 井上豊, 池田剛, 車谷浩一: センサイベント指向のサービス連携ミドルウェア: SENSORD, UBI, pp.37-44, (2006).
- 8) 伊藤崇洋, 小倉正利, 神谷英樹, 峰野博史, 石川憲洋, 水野忠則, :”携帯端末から制御可能なセンサ・家電連携環境の構築, DICOMO, pp.2000-2007 (2008).
- 9) 川島英之, 今井倫太, 遠山元道, 安西祐一郎: センサデータベースシステム KRAFT の設計と実装, 情報処理学会論文誌, Vol.45 No.SIG14(TOD 24) pp.39-53 (2004).
- 10) 森下民平: データストリーム管理システム, (株) シーエーシー技術プロダクツセンター技術研究グループ技術レポート (2007).
- 11) 株式会社ルネサスソリューションズ, <http://www.rso.renesas.com/>
- 12) Daniel J. Abadi, Carney, D., Cetintemel, U., Cherniack, M., Conway, C., Lee, S., Stonebraker, M., Tatbul, N. and Zdonik, S.: Auroa: A new model and architecture for data stream management, The VLDB Journal, Vol.12, No.2, pp.120-139, (2003).
- 13) Daniel J. Abadi, Yanif Ahmad, Magdalena Balazinska, Ugur Cetintemel, Mitch Cherniack, Jeong-Hyon Hwang, Wolfgang Lindner, Anurag S. Maskey, Alexander Rasin, Esther Ryzkina, Nesime Tatbul, Ying Xing, and Stan Zdonik, : The Design of the Borealis Stream Processing Engine, Proceedings of the CIDR Conference pp.277-289 (2005).
- 14) StreamSpinner <http://www.streamspinner.org/>
- 15) 北川博之, 川島英之, 天笠俊之: センシングデータの処理基盤, 情報処理学会誌, Val.51, No.12, pp.1119-1126 (2010).
- 16) 山田真一, 渡辺陽介, 北川博之, 天笠俊之, : データストリーム管理システム Harmonica の設計と実装, 情報処理学会論文誌, Vol.48 No.SIG14(TOD 35) pp.91-106 (2007).