

## センサデータベースシステム実現方式の一提案

川島 英之† 遠山 元道‡ 今井 倫太‡§ 安西 祐一郎 ‡

† 慶應義塾大学大学院 理工学研究科 開放環境科学専攻

‡ 慶應義塾大学 理工学部 情報工学科

§ 科学技術振興事業団さきがけ研究 21

E-mail: † kawasima@ayu.ics.keio.ac.jp, ‡ {toyama,michita,anzai}@ics.keio.ac.jp

本論文はコミュニケーションロボットや即時地震情報システムを支援するセンサデータベースシステムの実現手法を提案する。そのようなアプリケーションの要求は、(1) センサデータに対する類似演算の実現、(2) センサデータと関係データを結ぶモデルの実現、(3) 実時間クエリ応答の実現、そして(4) データの時間的一貫性の実現である。(1)を実現するために、センサデータを管理するための抽象データ型を導入する。(2)を実現するために関係データモデルと(1)で導入した抽象データ型を処理できる言語を導入する。(3)を実現するために、高速クエリ処理と実時間クエリ処理の2アプローチを導入する。(4)を実現するために、高速データ挿入処理とスレッドスケジューリング方式を導入する。

キーワード：抽象データ型、実時間クエリ処理、データの時間的一貫性

## Proposing a Design of Sensor Database System

Hideyuki KAWASHIMA† Motomichi TOYAMA‡ Michita IMAI  
‡§ Yuichiro ANZAI ‡

†School of Science for OPEN and Environmental Systems,  
Faculty of Science and Technology, Keio University.

‡Department of Information and Computer Science, Faculty of Science and Technology,  
Keio University.

§Precursory Research for Embryonic Science and Technology, Japan Science and Technology  
Corporation

E-mail: † kawasima@ayu.ics.keio.ac.jp, ‡ {toyama,michita,anzai}@ics.keio.ac.jp

This paper proposes a design of sensor database system that supports communication robots and real-time earthquake information systems. The requirements of these applications are: (1) similarity function for sensor data, (2) data model that connect sensor data and relational data, (3) real-time query reaction, and (4) temporal consistencies of data. To satisfy (1), an abstract data type for sensor data is incorporated. To satisfy (2), a language is incorporated that treat data incorporated at (1) and relational data. To satisfy (3), fast query processing and real-time query processing are incorporated. And to satisfy (4), fast data insertion processing and thread scheduling are incorporated.

keyword : Abstract Data Type , Real-Time Query Processing , Temporal Consistency of Data

## 1 はじめに

本論文の目的は、コミュニケーションロボット [1] や即時地震情報システム [2] を支援するデータベースシステムの実現手法を示すことである。コミュニケーションロボットや実時間地震防災などのアプリケーションは、実世界を周期的に認識し、認識された状況に適切な処置を発行する。実世界を認識するために、アプリケーションは時々刻々と生じるセンサデータのパタンを監視する。それゆえデータベースシステムがアプリケーションに提供すべきは、頻繁にシステムに到着するセンサデータを蓄えることと、そのセンサデータに対してパタン認識を適用する手段を提供することである。

頻繁にシステムに到着するセンサデータを蓄えるには、センサデータを効率的に管理できるデータ構造が必要になる。そしてセンサデータに対してパタン認識を適用するには、データベースシステムがセンサデータを管理する構造についてパタン認識をする手段を用意する必要がある。センサデータは頻繁に到着し、パタン認識は周期的に行われるから、クエリ処理には実時間性が要求される。さらにパタン認識に用いるデータが古ければ誤認識を引き起こすから、センサデータは新鮮でなければならない。複数のセンサから生じるデータについては同期が取れている必要もある。

センサデータを効率的に管理できるデータ構造としては抽象データ型が使われてきた [3]。抽象データ型をリレーショナルデータモデルに導入することで、リレーショナルデータモデルの柔軟性を活かすことができる。センサデータに対するパタン認識としては、距離尺度としてユークリッド距離 [4] やダイナミックタイムワーピング距離 [5][6] が使われてきた。クエリの実時間処理としては、リアルタイムシステム分野におけるタスクスケジューリング [7] やデータストリームプロセッサ開発 [8][9] において負荷制御が研究される一方、クエリ処理をインクリメンタルに実行することで、デッドラインまでにある程度の結果を返却する手法が研究されてきた [10][11]。クエリ処理時間を短縮するために、CPU キャッシュを意識したデータ構造も研究されてきた [12][13][14][15]。データの新鮮性や同期性については、センサデータの挿入処理を高速化する研究 [16] と、クエリスケジューリングに関する研究が行われてきた [17]。

各課題に対する従来研究は問題意識が異なっているが、これらを一緒に捉えてデータベースシステムを実現しなければ、本論文で対象とす

るアプリケーションの要求を満たせない。

そこで本論文では、各課題に対する従来研究のアプローチを統合した、新たなデータベースシステムの実現方式を提案する。センサデータを効率的に管理するために、センサデータ用の抽象データ型を導入する。そのデータ型におけるセンサデータの表現は時系列データとする。そしてセンサデータに対してパタン認識を適用するために、センサデータ用抽象データ型へのメソッドとして類似演算を導入する。類似演算時の類似尺度としては、ユークリッド距離とダイナミックタイムワーピング距離を導入する。実時間クエリ処理を実現するために高速クエリ処理と実時間クエリ処理の両アプローチを適用する。高速クエリ処理を実現するために次のアプローチをとる。類似演算処理については次元削減と空間索引の組み合わせにより探索の高速化を図る。バッファプールと索引構造をキャッシュコンシャスにすることでクエリ処理の高速化を図る。そして複数クエリ処理の最適化を導入することで無駄な処理を減らし、クエリ処理の高速化を図る。実時間クエリ処理を実現するために次のアプローチをとる。実時間クエリスケジューリング方式の中で周期的タスク処理に有効な方式を導入する。リレーショナルデータ処理と類似演算処理についてインプリサイスクエリ処理を導入する。データの時間的一貫性を保証するために次のアプローチをとる。データの時間的一貫性を考慮したスケジューリング方式を導入する。そして挿入処理を絶対的に高速化するために、複数台のリモートホストのメモリを永続デバイスとして WAL を実行する。

本論文の構成は次の通りである。2 節では研究の背景を述べ、問題を定式化する。3 節では提案するデータベースシステムについて述べる。4 節では関連研究について述べる。最後に 5 節で論文をまとめる。

## 2 背景と問題

### 2.1 研究の背景

#### 2.1.1 対象アプリケーション

本論文で対象とするデータベースシステムのアプリケーションは、コミュニケーションロボット [1] の行動決定や即時地震情報システム [2] の被害警報通知などの、実世界を認識し、認識された結果に適切な反応をする必要があるアプリケーションである。コミュニケーションロボットとは、人間とコミュニケーションすることを目

的に開発されたロボットであり, Robovie[1] 等がある。コミュニケーションロボットはセンサデータをパタンマッチングすることで人間の振舞を認識し, 認識された人間の振舞に対して適切な行動をすることで, 人間とのコミュニケーションを図る。もしもコミュニケーションロボットが人間の振舞に対して不適切な行動をしてしまったら, 人間はコミュニケーションロボットに対して「コミュニケーションが取れていない」と感じてしまうため, コミュニケーションロボットの目的は達成されない。即時地震情報システムとは, 大地震到達前にその到来情報を提供するシステムである。地震波には P 波と S 波の 2 種類があり, P 波の速さは約 6km/s, S 波の速さは約 3.5km/s である。揺れ幅は S 波の方が大きく, 平均して P 波の 5~10 倍である。即時地震情報システムは, P 波を利用して S 波伝播パターンを提供することにより, S 波到着前に列車や車を停止させ, 地震被害を軽減する [2]。即時地震情報システムの例として, 文献 [2] では, JR のコレダス, メキシコ市の SAS, 気象庁のナウキャスト等が挙げられている。

これらのアプリケーションに共通することは, 現在のセンサデータを基に, データベースシステムにクエリを発行し, 行動決定に必要な情報を引き出すことで, 適切な行動をすることである。例えばコミュニケーションロボットはデータベースシステムに次のようなクエリを発行する。

クエリ例: 「コミュニケーション相手が行っている, 現在の振舞は何であるか」そしてクエリ結果にふさわしい行動を, コミュニケーションロボットは実行する。

### 2.1.2 データベースシステムへの要求

コミュニケーションロボットや即時地震情報システムはデータベースシステムに対して次の 4 条件を満足することを求める。この理由を上記のクエリ例の実現を通して説明する。

(条件 1) センサデータに対する類似演算の実現

(条件 2) センサデータと関係データを結びモデルの実現

(条件 3) 実時間クエリ応答の実現

(条件 4) データの時間的一貫性の実現

クエリ例を実現するために, コミュニケーションロボットはセンサから受け取ったデータを, 事前に用意されている振舞テンプレートと比較し, 最も類似するテンプレートが表す振舞を, コミュニケーション中である人間の振舞として

認識する必要がある。それゆえデータベースシステムは条件 1 を満足する必要がある。さらに, その振舞の性質を詳細に記述するには「振舞の名前」や「振舞の友好性」などの複数カラムが必要になるから, センサデータだけではなく, リレーショナルデータを扱える必要がある。それゆえ条件 2 を満足する必要がある。そしてデータベースシステムの応答は実時間で行われる必要がある。さもなければコミュニケーションロボットは次行動決定が遅れてしまい, コミュニケーション相手の人間を不愉快にさせてしまう。それゆえ条件 3 が満足されなければならない。最後に, データの時間的一貫性が満足されなければならない, コミュニケーションロボットは古いデータや同期性の取れていないデータからコミュニケーション相手の振舞を認識してしまい, 時間的に間違った行動をする。これはコミュニケーション相手の人間を不愉快にするから, 条件 4 が満足されなければならない。

## 2.2 4 条件の定式化

ここではデータベースシステムに求められる 4 条件を定式化する。

### 2.2.1 条件 1 の定式化

センサデータに関する類似演算として, ユークリッド距離とダイナミックタイムワーピング距離の二つの類似尺度を定義する。

定義 1 (センサデータ) 本論文では, あるセンサデータを  $s$  と表記し,  $s$  を次のように定義する。

$$s = \langle at, v \rangle$$

ここで  $at$  はセンサデータがデータベースシステムに到着した時刻 (*Arrival Time*) を表し,  $v$  はセンサデータの値 (*Value*) を表す。そして,  $S_A$  を  $s$  の集合と定義する。 $S_A$  の  $i$  番目の  $s$  を  $S_{A_i}$  と表記する。

定義 2 (ユークリッド距離) センサデータストリーム  $S_A$  と  $S_B$  のユークリッド距離を  $D_{euclid}$  と表記し, 次のように定義する。

$$D_{euclid}(S_A, S_B) = \sqrt{\sum_{i=1}^n (S_{A_i} - S_{B_i})^2}$$

定義 3 (ダイナミックタイムワーピング距離) センサデータストリーム  $S_A$  と  $S_B$  のダイナミッ

クタイムワーピング距離を  $D_{dtw}$  と表記し，次のように定義する [6] .

$$D_{dtw}(S_A, S_B) = W[\sum_{i=1}^n \delta(w_i)]$$

センサデータを管理するデータ型を *sensor* とする . *sensor* のデータは  $S$  により表現され，*sensor* には  $D_{euclid}$  及び  $D_{dtw}$  を計算する演算と，SP 演算が用意される必要がある . また，同種の *sensor* を複数まとめてグループにできる . *sensor* を用いて，条件 1 を次のように定式化する .

$$\text{条件 1} = \textit{sensor} \text{ の実現} \quad (1)$$

### 2.2.2 条件 2 の定式化

SPJ 演算を適用可能な関係データを  $R$  と表記する . データベースシステムのデータモデルを  $M$  とする .  $M$  と  $R$  を用いて，条件 2 を次のように定式化する .

$$\text{条件 2} = \textit{sensor} \in M \text{ かつ } R \in M \text{ の実現} \quad (2)$$

### 2.2.3 条件 3 の定式化

本論文では次のように実時間性を定義する . これは文献 [18] の 4 つの分類に従えばファームリアルタイムと呼ばれる .

定義 4 (クエリ応答の実時間性) あるクエリ  $Q$  について，ユーザが定めた  $Q$  の終了時刻を  $Q_{deadline}$  とし， $Q_{finish}$  を  $Q$  に対するデータベースシステムの応答時刻とする .  $Q_{deadline}$  と  $Q_{finish}$  を用いて条件 3 を次のように定式化する .

$$\text{条件 3} = Q_{deadline} \geq Q_{finish} \text{ の実現} \quad (3)$$

### 2.2.4 条件 4 の定式化

データ時間的一貫性には，新鮮性と同期性の二つがある . これらを以下で定義する .

定義 5 (新鮮性)  $s_{read}$  を  $s$  がアプリケーションに読まれた時刻とする .  $s_{arrive}$  を  $s$  の到着時刻， $at$  とする .  $threshold_{freshness}$  をアプリケーションに指定される閾値とする . このときすべての  $s \in S$  について次式が成り立つならば，新鮮性が成り立つと定義する .

$$s_{read} - s_{arrive} \leq threshold_{freshness}$$

定義 6 (同期性)  $s_{iarrive}$  を，ある  $s$  である  $s_i$  がアプリケーションに読まれた時刻とする .

$$\max(|s_{iarrive} - s_{jarrive}|)$$

を， $n$  個の  $s$  の  $at$  の中で，差分の絶対値が最大のものとする .  $threshold_{synchronosness}$  をアプリケーションに指定される閾値とする . このときすべての  $s \in S$  について次式が成り立つならば，同期性が成り立つと定義する .

$$\max(|s_{iarrive} - s_{jarrive}|) \leq threshold_{synchronosness}$$

条件 4 を次のように定式化する .

$$\text{条件 4} = \text{新鮮性の成立かつ同期性の成立} \quad (4)$$

## 2.3 システム構成

データベースシステムには 3 種類のクエリ処理が用意される . センサからセンサデータを受けとり，それをデータベースシステムに追加することを連続的に行うクエリ処理をアペンドクエリと表記する . 周期的にセンサデータの監視とパターンマッチングをするクエリ処理をモニタクエリと表記する . そしてデータベースシステム内のデータを一度だけ閲覧して終了するクエリ処理をアドホッククエリと表記する .

データは全てローカルディスクに収められ，その一部が高速アクセスのためにメモリ上のバッファプールと呼ばれる領域に置かれる .

## 3 提案

本節では前節で定式化した 4 条件を満足するアプローチを述べる .

### 3.1 条件 1 を満足するアプローチ

*sensor* を実現するために，本論文では抽象データ型モデルを用いる . 抽象データ型は，[19] には次のように述べられている . “The combination of an atomic data type and its associated methods is called an bf abstract data type, or ADT” . 本論文では ADT をこの意味で用いる . そしてセンサ抽象データ型を  $SADT$  と表記する .

*sensor* 中  $S$  を実現するため， $SADT$  ではデータ型として  $s$  を要素とする時系列データ型を導入する .  $SADT$  のデータについて  $D_{euclid}$  と  $D_{dtw}$  を計算するために， $SADT$  内にそれぞれ  $euclid()$  と  $dtw()$  を用意する . また，*sensor*

```
CREATE TABLE Robovie (
  name char[8], // ロボビーの固有名称
  sonar_g_sadt(s1...s24), // 超音波センサ 1-24
  touch_g_sadt(t1...t8) // タッチセンサ 1-8
);
```

図 1: Robovie 用の表定義

```
CREATE TABLE behavior (
  behavior_name char[8], // 振舞の名前
  friendship int, // 振舞の友好度
  sonar_g_sadt[24], // 超音波センサ 24 個
  touch_g_sadt[8] // タッチセンサ 8 個
);
```

図 2: クエリ例 1 を実現するための表定義

```
CREATE CQ monitor_behavior {
  PERIOD = 3s;
  DEADLINE = 1s;
  SELECT behavior_name, friendship
  FROM behavior
  WHERE
  Robovie.name = 'RobovieA' AND
  SYNCHRONOUSNESS ON behavior.sonar < 50ms;
  behavior.sonar[0].dtw(Robovie.sonar[0])
  WITH
  RANGE [NOW-1s, NOW] AND
  FRESHNESS <= 50ms;
  ) <= 100
  AND ... AND
  SYNCHRONOUSNESS ON behavior.sonar <= 100ms;
  behavior.touch[7].euclid(Robovie.touch[7])
  WITH
  RANGE [NOW-0.1s, NOW] AND
  FRESHNESS <= 100ms;
  ) <= 3
};
```

図 3: クエリ例 1 の実現

中の  $S$  を指定して, SP 演算を掛けられるようにする.

### 3.2 条件 2 を満足するアプローチ

$SADT \in M$  かつ  $R \in M$  を実現するために, 本論文ではデータベースシステムのデータモデルとしてオブジェクトリレーショナルデータモデルを導入する. 導入したデータモデルにより, クエリ例が実現されることを示す.

#### 3.2.1 クエリ例の実現

クエリ例を実現するにあたり, まず Robovie[1] の表定義を図 1 に示す. Robovie はある種のロボットの呼称なので, 複数台の Robovie がいる場合には, Robovie<sub>*i*</sub> とか Robovir<sub>*j*</sub> 等という名前を付けて区別する. アペンドクエリは Robovie<sub>*n*</sub> の SADT にセンサデータを挿入していく.

図 2 に, 人間の振舞を表現する表を作成する DDL を示す. 表 behavior のタプルは属性として, 振舞の名前, 振舞の友好度, 超音波センサ, そしてタッチセンサをもつ. このうち, 超音波センサとタッチセンサは SADT で定義される.

表 behavior に対して, 図 3 に示したクエリを発行することで, クエリ例 1 が実現される.

図 3 の意味をおおまかに述べると, 次のようになる. 「RobovieA について, 現在の超音波センサ 1 から 24 の DTW 距離が近く, かつタッチセンサ 1 から 8 のユークリッド距離に近い振舞について, その名前と友好度を教えなさい. ただし, 超音波センサとタッチセンサについてはデータに同期性が, 全データには新鮮性が成り立つようにしなさい. そしてこのクエリは 3 秒

おきに実行し, 1 秒以内に答えを提供しなさい」

図 3 を詳細に述べると次のようになる. CREATE CQ 文はモニタクエリを作成する. この場合, モニタ名は monitor\_behavior である. PERIOD = 3s は, monitor\_behavior が 3 秒ごとに実行を開始することを要求する. DEADLINE = 1s は, monitor\_behavior の処理が, 処理開始から 1 秒以内に終了することを要求する. SYNCHRONOUSNESS ON behavior.sonar は, behavior.sonar がもつ 24 個のデータについて, 同期性を成立させることを要求する.

behavior.sonar[0].dtw(Robovie.sonar[0]) は識別子 0 の超音波センサと behavior.sonar[0] の  $D_{dtw}$  を計算することを要求する. RANGE は類似演算を実行する時間範囲を指定する. NOW はモニタクエリが起動した時刻を表す. FRESHNESS <= 100ms は, 新鮮性が 100ms であることを要求する.

これより,  $SADT \in M$  かつ  $R \in M$  が実現できる.

### 3.3 条件 3 を満足するアプローチ

$Q_{deadline} \geq Q_{finish}$  を実現するには, 3 つのアプローチが存在する. 1 つ目は, クエリ処理を高速化することで  $Q_{finish}$  を小さくするアプローチである. このアプローチは実時間性を考慮しないデータベースシステムにおけるクエリ処理高速化と等しい. 2 つ目は, 複数のクエリが同時に走っている際に, 各  $Q$  の  $Q_{deadline}$  を基準にしてクエリスケジューリングをすることで, 条件 3 の不成立を減らすアプローチである. そして 3 つ目は,  $Q_{deadline}$  までに  $Q$  の処理が終了しなかった場合に, 中間結果を返却することで, クエリ応答に意味を与える方式である. 優れたパフォーマンスのためには, これらすべて

てを同時に実行することが望ましい．それぞれについて以下で述べる．

### 3.3.1 高速クエリ処理

- 類似演算処理の高速化

類似演算処理を高速化するアプローチは2つある．1つは次元削減と空間索引を組み合わせる手法である [4][5][20]．もう1つは時系列アクティブ探索法 (TAS)[21] のようにシーケンシャルスキャンを高速化する手法である．

これらの方式の内，実時間性を保証するには空間索引方式の方が望ましい．なぜなら最悪実行時間を保証しやすいからである．TAS方式は高速だが，最悪実行時間を見積ることが難しい．それゆえ本論文で扱うデータベースシステムには，次元削減と空間索引の組み合わせを導入する．

次元削減方式には，PAA，APCA[4]，FFT，Wavelet，波形特徴 [20] 等が提案されているが，どれかの手法が最も優れているというわけではなく，データの性質に応じて，使い分ける必要がある [20]．そこで提案するデータベースシステムでは，複数の次元削減方式を用意し，索引作成時にユーザが次元削減方式を選択できるようにする．

空間索引には，R\*-tree[22]，SR-tree[23]，X-tree[24] 等，さまざまなものがある．そこで提案するデータベースシステムでは，複数の空間索引方式を用意し，索引作成時にユーザに空間索引方式を選択させる．

- キャッシュコンシャス

近年，CPU キャッシュのアクセス速度とメモリアクセス速度が極めて大きくなっている．CPU 速度の年間向上率は60%であるのに対し，メモリアクセス速度の年間向上率は10%しかない [15]．そこで，CPU キャッシュミスが減らせるようなバッファプールの構造と索引構造が研究されている．そこで，本論文で開発を行うデータベースシステムでも，バッファプールと索引構造をキャッシュコンシャスにする．バッファプール構造には PAX[12] を参考にし，空間索引の構造には CSB<sup>+</sup>tree[13] を参考にする．

- 複数クエリの最適化

提案するデータベースシステムでは，複数のモニタクエリと複数のアベンダクエリが同時に走る．それゆえ複数のモニタクエリが同じ処理結果を求める可能性がある．渡辺らの研究 [25] では，関係データに関するクエリ処理について，モニタクエリのクエリ内容が全く同一である場合，クエリ結果をキャッシングすることで，クエリ応答を高速化できることを実験により示している．

提案するデータベースシステムでは，関係データと SADT の両データ構造に関するクエリ処理について，モニタクエリのクエリ内容をキャッシングすることで，クエリ応答の高速化を図る．

### 3.3.2 実時間性に基づくクエリスケジューリング

実時間性に基づくタスクスケジューリング方式には，Rate-Monotonic(RM)，Deadline-Monotonic(DM)，そして Earliest Deadline First(EDF) がある．RM と DM は周期的タスクに適切な方式であり，EDF は非周期的タスクに適切な方式である．本論文で対象とするデータベースシステムでは，長時間にわたり周期的に処理をおこなうクエリが主であるので，スケジューリングポリシーには RM か DM を用いる．

### 3.3.3 インプリサイスクエリ処理

$Q_{finish} \geq Q_{deadline}$  である場合，全く答えを返さないよりも， $Q_{deadline}$  までに得られた中間結果を返却することが好ましい．そのようなクエリをインプリサイスクエリ処理と呼ぶ．関係データに関するインプリサイスクエリ処理の研究には APPROXIMATE[11] がある．空間索引により索引付けされたデータを実時間で返却するために，インクリメンタルに部分結果を返却する研究には，白石らの研究 [10] がある．提案するデータベースシステムにも，インクリメンタルクエリ処理が導入することで，スケジューリング不能の過負荷時にも，ある程度の結果を返却できるようになる．

## 3.4 条件4を満足するアプローチ

条件4を成立させるには，2つのアプローチがある．1つは永続化処理を高速化することで，

$s$  がデータベースシステムに到着してから、アクセス可能になるまでの時間を短縮するアプローチである。そしてもう1つは、アペンドクエリをスケジューリングすることで、モニタクエリに時間的一貫性の成り立つ  $s$  を読ませるアプローチである。

### 3.5 永続化処理の高速化

永続化処理を高速化するために、リモートメモリを永続化デバイスとして  $s$  を永続化する手法を我々は提案してきた [16][26]。ディスクではなくリモートメモリに対して永続化処理を実行することで、優れた新鮮性が得られる結果が得られている。

### 3.6 時間的一貫性に基づくスケジューリング

時間的一貫性に基づいてアペンドクエリとモニタクエリをスケジューリングすることで、複数のアペンドクエリと複数のモニタクエリが同時に走っている場合に、時間的一貫性を高められる可能性がある。条件3を成立させるためにもスケジューリングが必要であるので、提案するデータベースシステムにおいてパフォーマンスを高めるためには、条件3で求められる実時間性と、条件4で求められる時間的一貫性を同時に考慮して、スレッドスケジューリングをする必要がある。

## 4 関連研究

本節ではセンサデータもしくはデータストリームを扱う既存研究について述べる。

COUGAR[3] プロジェクトではセンサデータベースシステムを開発している。COUGARではセンサデータを扱うためにADTを導入している。センサデータを時系列データとして表現し、データを処理するメソッドとして、*getTemp()* と *detectAlarmTemp(Threshold)* を導入している。COUGARはデータモデルとしてオブジェクトリレーショナルデータモデルを採用しているため、本論文の対象とするデータである  $s$  を格納することが容易だと考えられる。しかしながら、類似演算は導入していないため、コミュニケーションロボットへの適用は困難である。COUGARは実時間性と時間的一貫性についてはアプローチをしていない。

TELEGRAPH[9]ではPostgreSQLを拡張し、ストリームプロセッサを開発している。対象とするデータモデルは関係データモデルであるため、条件1と条件2に対するアプローチはない。

AURORA[8]ではデータストリームプロセッサを開発している。AURORAでは、データにタイムスタンプを張り付けるが、センサデータを扱おうとはしていない。それゆえ類似演算も導入されていない。AURORAのデータモデルはリレーショナルデータモデルであり、オブジェクトは扱わない。それゆえ条件1と条件2に対するアプローチはない。AURORAでは実時間性を成立させるために、スケジューラと負荷制御機構に関するアプローチをしているので、条件3に対してアプローチをしている。データ時間的一貫性についてのアプローチはない。

STREAM[27]ではデータストリームプロセッサを開発している。STREAMではセンサデータを特別に扱おうとはしていない。データモデルとしてリレーショナルデータモデルを採用し、リレーションをストリームに変換する演算を導入している。実時間性とデータ時間的一貫性については、STREAMでは扱われていない。

## 5 まとめ

本論文ではセンサデータベースシステムの実現手法の一案を示した。そのようなデータベースシステムに要求される4条件を提示し、各条件に対するアプローチを詳述した。記述したアプローチ内容を実装し、客観的に明らかな手法で評価をとることが今後の課題である。

## 参考文献

- [1] 佐竹聡, 川島英之, 今井倫太. データベースを用いたコミュニケーションロボットシステムの構築. Vol. 103, No. 32, pp. 7-12, 2003.
- [2] 菊地正幸. リアルタイム地震学, 第3章. 東京大学出版会, 2003.
- [3] Philippe Bonnet, Johannes Gehrke, and Praveen Seshadri. Towards Sensor Database Systems. In *Mobile Data Management, Second International Conference, Lecture Notes in Computer Science 1987*, pp. 3-14, January 2001.
- [4] Eamonn J. Keogh, Kaushik Chakrabarti, Sharad Mehrotra, and Michael J. Pazzani. Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2001.

- [5] Eamonn J. Keogh. Exact Indexing of Dynamic Time Warping. In *Proceedings of International Conference on Very Large Databases*, pp. 406–417, 2002.
- [6] 大崎竜太, 上原邦昭. Dynamic Time Warping法を用いた身体運動の動作識別, 1998.
- [7] Jane W. S. Liu. *Real-Time Systems*. Prentice Hall, 2000.
- [8] Don Carney, Ugur Cetintemel, Mitch Cherniack, Christian Convey, Sangdon Lee, Greg Seidman, Michael Stonebraker, Nesime Tatbul, and Stan Zdonik. Monitoring Streams - A New Class of Data Management Applications. In *Proceedings of International Conference on Very Large Data Bases*, 2002.
- [9] Samuel R. Madden and Michael J. Franklin. Fjording the Stream: An Architecture for Queries over Streaming Sensor Data. In *18th International Conference on Data Engineering*, February 2002.
- [10] 白石陽, 安西祐一郎. 空間検索のためのインクリメンタルなデータ提供手法. 情報処理学会論文誌: データベース, Vol. 42, No. SIG 15(TOD 12), pp. 15–29, December 2001.
- [11] Susan V. Vrbsky. A Data Model for Approximate Query Processing of Real-Time Databases. In *Data and Knowledge Engineering*, Vol. 21, pp. 79–102, December 1996.
- [12] Anastassia Ailamaki, David J. DeWitt, Mark D. Hill, and Marios Skounakis. Weaving Relations for Cache Performance. In *Proceedings of International Conference on Very Large Databases*, pp. 169–180, 2002.
- [13] Jun Rao and Kenneth A. Ross. Making B+tree Cache Conscious in Main Memory. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2000.
- [14] 吉岡弘隆. OLTP 性能向上を目的としたメモリプロファイリングツール. データ工学ワークショップ (DEWS2003), 2003.
- [15] 本田喜久, 川島英之, 今井倫太, 安西祐一郎. CSS-tree の挿入処理の高速化. Vol. 71, No. 128, 2002.
- [16] Hideyuki Kawashima, Motomichi Toyama, Michita Imai, and Yuichiro Anzai. Providing Persistence for Sensor Data Streams with Temporal Consistency Conscious WAL. In *Proceeding of IASTED International Conference on Information Systems and Databases (ISDB 2002)*, pp. 13–18, September 2002.
- [17] Sasa Tomic, Susan V. Vrbsky, and Tracy Camp. A new measure of temporal consistency for derived objects in real-time database systems. *Information Sciences*, Vol. 124, No. 1–4, pp. 139–152, 2000.
- [18] Joakim Eriksson. Real-Time and Active Databases: A Survey. In *Lecture Notes in Computer Science 1553: Active, Real-Time and Temporal Database System*, pp. 1–23, September 1997.
- [19] Raghu Ramakrishnan and Johannes Gehrke. *DATABASE MANAGEMENT SYSTEMS*, chapter 23, p. 785. McGrawHill, 3 edition, 2003.
- [20] 川島英之, 遠山元道, 今井倫太, 安西祐一郎. 波形特徴を用いた類似シーケンス検索. Vol. 71, No. 128, 2002.
- [21] 木村昭悟, 柏野邦夫, 黒住隆行, 村瀬洋. 区分線形写像に基づく音や映像の高速探索-特徴系列の分解による次元削減の導入-. データ工学ワークショップ (DEWS2003), 2003.
- [22] Beckmann, Kriegel N., R H.P., Schneider, and B. Seeger. R\*-tree: An Efficient and Robust Access Method for Points and Rectangles. In *Proc. of ACM SIGMOD Int. Conf. on Management of Data*, pp. 322–331, 1990.
- [23] N Katayama and S Satoh. The SR-tree: An Index Structure for high-Dimensional Nearest Neighbor Queries. In *Proc. of ACM SIGMOD Int. Conf. on Management of Data*, pp. 369–380, 1997.
- [24] S. Berchtold, D.A Keim, and H.P Kriegel. The X-tree: An Index Structure for High Dimensional Data. In *Proceedings of International Conference on Very Large Databases*, pp. 28–39, 1996.
- [25] 渡辺陽介, 北川博之. 連続的問い合わせに対する複数問い合わせ最適化. データ工学ワークショップ (DEWS2003), 2003.
- [26] Hideyuki Kawashima, Motomichi Toyama, Michita Imai, and Yuichiro Anzai. Providing Persistence for Sensor Streams with Light Neighbor WAL. In *Proceedings of Pacific Rim International Symposium on Dependable Computing (PRDC2002)*, pp. 257–264, December 2002.
- [27] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwami, and Jennifer Widom. Models and Issues in Data Stream Systems. In *ACM PODS*, 2002.