

ストリーム統合システムを用いた実世界情報の統合利用

山田 真一[†] 渡辺 陽介^{††} 北川 博之^{††,†††}

[†] 筑波大学 第三学群 情報学類, 〒 305-8573 茨城県つくば市天王台 1-1-1

^{††} 筑波大学 システム情報工学研究科

^{†††} 筑波大学 計算科学研究センター

E-mail: {snic,watanabe}@kde.cs.tsukuba.ac.jp, kitagawa@cs.tsukuba.ac.jp

あらまし 近年, データストリームという配信形態で送られる情報が増加し, ストリームデータに対する問合せ処理が重要となっている. また, 光や温度, 位置などの実世界情報を取得するセンサも容易に利用できるようになり, 実世界情報とストリームデータや RDBMS の統合利用への要求が高まっている. そこで本研究では, 今まで我々の研究グループが開発してきたストリーム統合エンジンに基づいて, 実世界情報の統合利用環境を構築した. 本稿では, そのアーキテクチャについて述べる. また, 実際にその統合利用環境を用いて開発した, ストリームデータと実世界情報を統合利用するアプリケーションについて述べ, その性能を評価するために行った実験についても述べる.

キーワード データストリーム, ユビキタスコンピューティング, 情報統合

Integration of Real World Information by using Stream Integration System

Shinichi YAMADA[†], Yousuke WATANABE^{††}, and Hiroyuki KITAGAWA^{††,†††}

[†] College of Information Sciences, Third Cluster of Colleges, University of Tsukuba
Tennoudai, Tsukuba-shi, 305-8573 Japan

^{††} Graduate School of System and Information Engineering, University of Tsukuba

^{†††} Center for Computational Sciences, University of Tsukuba

E-mail: {snic,watanabe}@kde.cs.tsukuba.ac.jp, kitagawa@cs.tsukuba.ac.jp

Abstract Today, the amount of data delivered as data streams is increasing, and query processing of streams has become important. In addition, we can easily obtain real world information such as brightness, temperature, and positions of moving objects. Such information is usually modeled as streams. Thus, integrating real world information modeled as streams is attracting a lot of attentions. We have been developing an stream integration environment based on the query processing engine we implemented before. In this paper, we explain the architecture and an application using our system.

Key words Data Stream, Ubiquitous Computing, Information Integration

1. はじめに

ネットワークの発達・普及に伴って, 時々刻々と変化する事象に関する, 最新の情報を逐次配信するストリームという情報源が増えてきた. ストリームの例としては, 天気予報, ニュース, 株価情報などが挙げられる. このようなストリームの増加に伴って, ストリームに対する問合せ処理を実現するシステムの需要が高まっている.

最近では, ストリームに対する問合せ手法として連続的問合せが注目されている. 連続的問合せとは, ストリームから到着したデータに対して繰り返し問合せ処理を実行し, 前回実行時の処理結果からの差分を生成する手法である. 現在, ストリーム

に対する問合せとして連続的問合せを使用しているストリーム情報源統合システムは幾つかの研究グループ [9], [11] ~ [15] で開発されており, 我々の研究グループでも研究開発を行っている [1], [2].

一方, センサ機器の改良等によって, 温度や光, デバイスの位置などの実世界の情報がより容易に取得できるようになっている. 例として, 自発的にネットワークを構築する無線センサデバイス [7] が挙げられる. 無線センサデバイス同士はアドホックな通信を行い, デバイス間でマルチホップを繰り返すことによって, 近くに無いデバイスとも通信可能である. そのため, 遠くの場所の温度や光の情報などを手軽にモニタリングすることができる. また, GPS 等の普及により, ノートパソコンや, PDA

などの携帯端末の位置情報を安価で取得することができるようになった。このような温度や光、位置など実世界から時々刻々と取得されるデータは、一種のストリームであると考えられる。今後、家具や家電製品などあらゆるところにコンピュータチップやネットワークが搭載されるユビキタス社会に向けて、実世界情報に対する問合せ処理は益々重要となってくると考えられる。

本研究では、我々の開発しているストリーム統合エンジンをベースに、センサなど実世界から得られるストリームを対象とした実世界情報統合利用環境を構築した。具体的には、実世界情報をストリーム統合エンジンに取り込むためのモジュールを設計・開発した。また、本研究では実用的なアプリケーションを想定したデモシステムを作成した。本稿では、統合利用環境およびアプリケーションについて紹介し、その性能評価のために行った実験について述べる。

以下では、まず次節で実世界情報を取得する方法について述べる。3節で実世界情報統合環境について述べ、4節でそのアプリケーションについて説明する。5節では、アプリケーションのデモシステムにおける性能評価実験について述べ、6節で関連研究を紹介する。最後に結論として、まとめと今後の課題について述べる。

2. 実世界情報

本節では、本研究で用いる実世界情報について述べる。

2.1 センサネット

センサネットとは、センサ機器同士が無線通信を自発的にを行い、アドホックなネットワークを構築するというものである(図1)。そのようなセンサネットの機器の1つに Crossbow 社が販売している Mote [7] がある(図2)。実世界情報は、Mote に付属しているセンサによって取得する。取得できる情報には、温度、光、加速度などがある。センサネットと PC 間の通信は、PC に接続された基地局から行う。Mote には、センサデバイスだけでなく、CPU が搭載されており、プログラミングが可能となっている。

現在、Mote 上で動く OS として TinyOS [4] が開発されており、TinyOS 上で開発されているアプリケーションとして TinyDB [3] がある。センサの値を取得するために、TinyDB は TinySQL という SQL ライクな問合せ言語を提供している。TinyDB は、センサから得られる値やデバイスの ID などの属性をもつ仮想的なテーブル sensors を提供しており、利用者は sensors テーブルに対して問合せを記述する。以下に TinySQL 問合せの記述例を示す。

```
SELECT nodeid, temp
FROM sensor
WHERE temp >= 25
SAMPLE PERIOD 5s
```

この TinySQL 問合せは、Mote に付属している温度センサから取得した値が 25 以上ならば、その Mote の ID とそのときの温度を取得する問合せである。TinySQL における SELECT-FROM-WHERE 節は、SQL のそれとほぼ同様の効果を持つ。TinyDB は

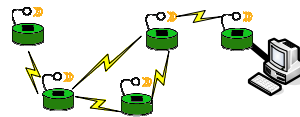


図1 アドホックネットワーク

Fig. 1 Ad hoc network



図2 MICA2DOT Mote

Fig. 2 MICA2DOT Mote

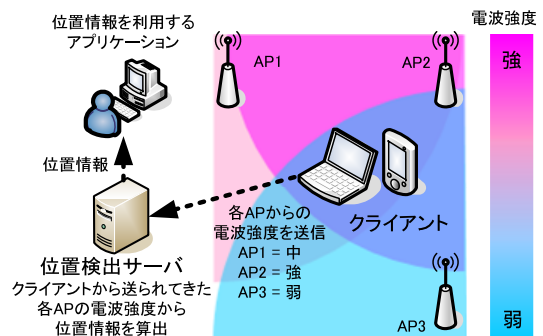


図3 無線 LAN を使った位置検出

Fig. 3 Position detection by using Wireless LAN devices

一定間隔で指定されたセンサの値をサンプリングし、利用者へ提供する。また、TinySQL の SAMPLE PERIOD 節でセンサのサンプリング間隔を指定することが可能である。

2.2 位置情報

移動オブジェクトの位置情報を取得する手段としては、GPS、RFID タグなどのハードウェアを用いる他に、無線 LAN を利用した位置検出がある。無線 LAN を利用した位置検出では、複数のアクセスポイントからの電波強度を利用して、現在の位置を算出する(図3)。理想的にアクセスポイントが設置されれば、精度の高い位置情報が期待できる。本研究では、位置情報を取得する方法として Ekahau 社の EPE(Ekahau Positioning Engine) [6] を用いる。EPE が位置情報を計算するためには、3 台以上のアクセスポイント、縮尺の正確な地図、そして事前の学習データが必要である。EPE は、各アクセスポイントからの電波強度以外にも、位置の遷移の情報や、通行可能な地点の情報なども考察して矛盾の無いように、正確な位置を計算する。取得できる位置情報の形態は、オブジェクトの現在位置を表す地図上の xy 座標値、または論理エリア名である。論理エリアは、地図上の 3 点以上の座標値で定義される領域で、利用者が自由に作成できる。

3. 実世界情報統合利用環境

本研究で実現した実世界情報の統合環境について述べる。問合せ処理については、我々が以前に開発したストリーム統合エンジン [1], [2] を用いて実現している。3.1 でエンジン部の概要について述べる。実世界情報の統合のために、本研究で新たに開発した部分については、3.2, 3.3 で述べる。

3.1 ストリーム統合エンジン

図4に示すように、エンジンは、問合せ解析器、問合せ最適化器、メディエータ、ビューマネージャから構成される。利用者から与えられた問合せは、問合せ解析器を経て問合せ最適化器へと渡される。問合せ最適化器で問合せの実行プランを生成し、メディエータに渡す。メディエータはその実行プランに従って、RDB か

ら必要なデータの取得や、ストリームから到着したデータの処理などを行う。メディエータにおける実行プランの評価は、時間の経過やストリームからの情報の到着などのイベントに応じて行われる。メディエータで生成された処理結果は直ちに利用者へと配信される。また、各種ストリームや RDB 等の情報源とエンジンの間にはラッパーが存在する。ラッパーの主な役割は各情報源で扱われている固有のデータ形式をエンジンの内部形式に変換することである。ストリームラッパーには新規到着情報を検出し、メディエータに通知する機能があり、RDB ラッパーにはメディエータからの SQL 問合せ要求を RDBMS へ渡し、結果を取得する機能がある。本研究では、実世界情報を統合するために 2 つのストリームラッパーを開発した。

ストリーム統合エンジンは、連続的問合せを処理する。連続的問合せとは、新規に到着した情報に対して問合せ処理を適用し、前回の処理実行時からの差分の結果を生成することを繰り返すものである。ストリームを RDB のリレーション、到着したストリームの 1 配信単位をタイムスタンプの付加された 1 タプルとしてモデル化しているため、問合せ記述には SQL に基づいた記述法を用いている。連続的問合せの記述例を以下に示す。

```

MASTER Sensor
SELECT Sensor.Light,
       SensorLocation.X, SensorLocation.Y
FROM Sensor[1msec], SensorLocation
WHERE SensorLocation.NodeId = Sensor.NodeId
    
```

この例は、光センサが動作して値を取得すると、その光センサの値とセンサの設置場所の情報を統合して通知する連続的問合せになっている。ただしここでは、Sensor は光センサから取得した値とそのセンサの ID を配信するストリームであるとし、SensorLocation はセンサの座標値とセンサ ID を格納した RDB とする。問合せの MASTER 節は、問合せ実行のきっかけとなるストリームを指定する。SELECT-FROM-WHERE 節は、SQL のそれとほぼ同等の意味を持つが、FROM 節には処理対象とするデータ範囲を決めるための時間幅 (ウィンドウ) を指定することが可能となっている。この例の場合、Sensor のウィンドウは 1 ミリ秒である。

3.2 センサネットラッパー

本研究では、センサネットから実世界情報を取得するためのモジュールとしてセンサネットラッパーを開発した (図 5)。センサネットラッパーは、TinyDB に TinySQL 問合せを登録し、センサネットから到着したデータをエンジンの内部形式に変換して、メディエータに配信する機能を持っている。

具体的なラッパーの動作を説明する。まず、メディエータの初期化時に、センサネットラッパーは TinySQL 問合せが記述されたストリーム定義ファイルから、TinySQL 問合せを読み込み、それを TinyDB に登録する。問合せを登録すると、TinySQL 問合せの SAMPLE PERIOD 節に記述した時間幅で、センサネットから結果が送られてくる。センサネットラッパーは TinyDB に問合せを登録するとともに、TinySQL 問合せの SELECT 節に列挙された属性名から、ストリームのスキーマを作成し、メディエータ

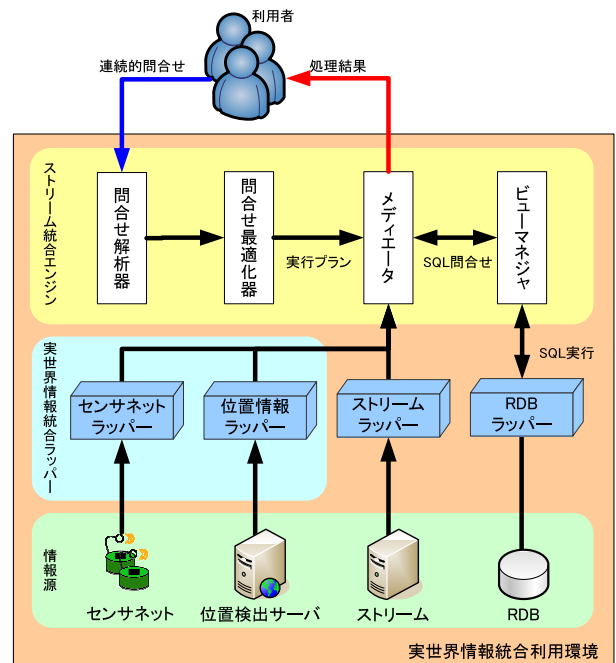


図 4 実世界情報統合利用環境

Fig. 4 System for Integrating Real World Information

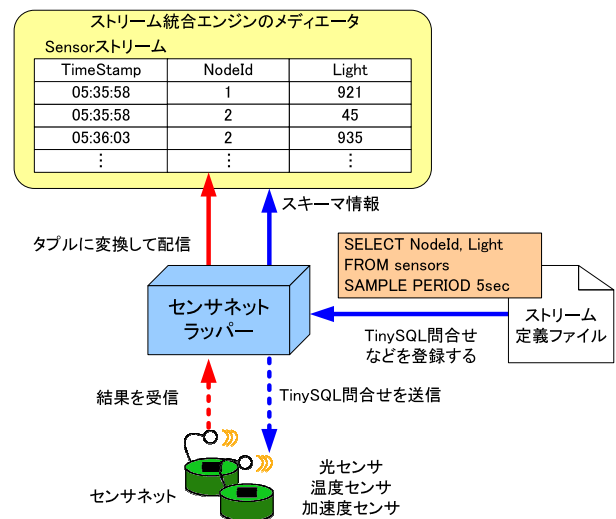


図 5 センサネットラッパー

Fig. 5 Sensor Net wrapper

に提供する。例えば、図 5 のように、5 秒毎にノード ID と光センサの値を取得する TinySQL 問合せをストリーム定義ファイルに記述すると、メディエータには、ノード ID と光センサの値を通知するストリームのスキーマが提供される。

センサネットからデータが到着すると、センサネットラッパーは、ストリームのスキーマ定義に合うようにそのデータからタプルを作成し、直ちにメディエータに配信する。図 5 の例では、5 秒毎にストリームデータが配信され、タプルが追加されていく。

本ラッパーでは、複数のストリーム定義ファイルを用意することで、複数の TinySQL 問合せが利用可能である。複数の TinySQL 問合せを利用した場合は、問合せ毎に別個のストリームとして利用者に提供される。

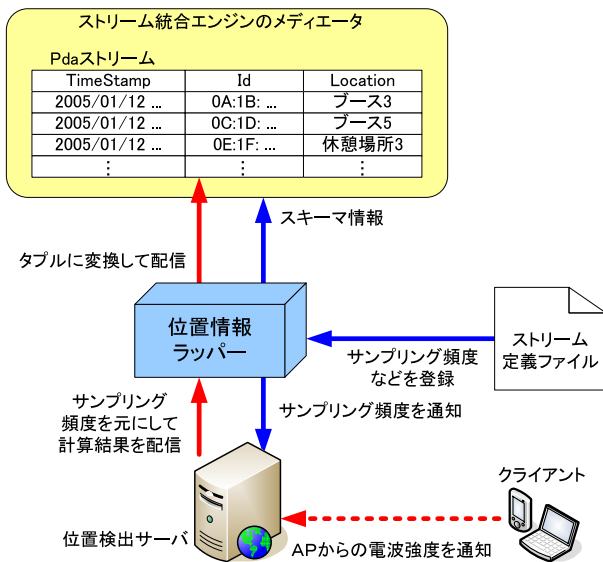


図6 位置情報ラッパー

Fig. 6 Position Detection wrapper

3.3 位置情報ラッパー

EPE サーバ (位置検出サーバ) からの位置情報の取得は、位置情報ラッパーによって行われる (図 6)。センサネットワークラッパーと同様に、ストリーム定義ファイルでの定義に従って、EPE サーバにサンプリング間隔を登録し、ストリームのスキーマをメディアータに提供する。EPE サーバから位置情報が届けられると、ストリームのスキーマ定義に合うようにタプルを作成し、直ちにメディアータに配信する。EPE サーバと通信するための Java のクラスがあらかじめ用意されているので、それを利用してサンプリング間隔の設定を行ったり、EPE が現在トラッキングしているデバイスの情報や、その位置情報などを取得したりする。位置情報は、EPE サーバに登録したサンプリング間隔に従って位置情報ラッパーに届けられる。図 6 では、EPE サーバから無線 LAN カードの物理アドレスと、論理エリア名を取得し、ストリームとして提供している。

4. 展示会支援アプリケーションの開発

本研究では、実世界情報統合利用環境を用いたアプリケーションのひとつとして、展示会における情報提供支援ツールを考えており、実際にデモシステムを実装した。

4.1 概要

大規模な展示会では、広い会場内に多種多様な出展者が新製品などのアピールをするために多くのブースを開設する。多数の出展物のある会場内を大勢の入場者が歩き回るため、会場内の状況が頻繁に変化し、人手によるモニタリングや情報提供は困難である。我々の実世界情報統合利用環境では、RDB に格納された出展物のコンテンツ等の情報と、入場者の位置情報や会場内のセンサ情報、場内アナウンス等のストリームとの統合を容易に実現可能である (図 7)。デモシステムでは、入場者の所持する携帯端末に対して、入場者の現在位置に応じた展示情報の提示や、入場者の興味に合った場内アナウンスの通知、休憩所の混雑状況の表示などを行っており、これらの統合要求を連続的

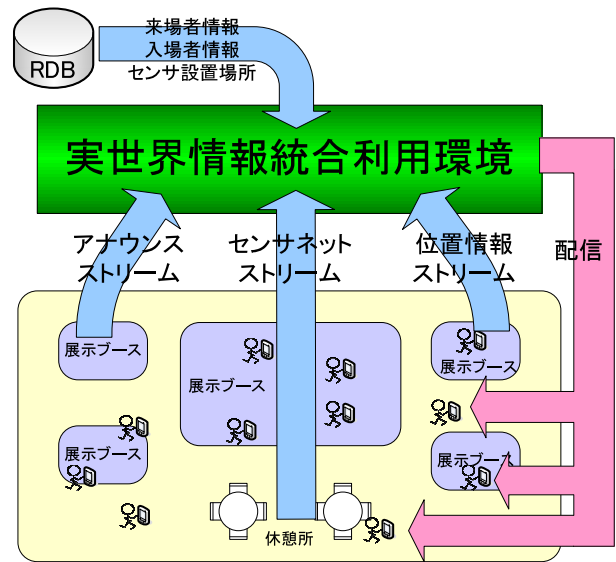


図7 展示会支援アプリケーション

Fig. 7 Exposition Supporting Application

問合せにより与えている。

4.2 デモシステム

デモシステムは Java(J2SE 1.4.2) を用いて実装され、現在はノート PC を携帯端末として使用している。

4.2.1 スキーマ定義

デモシステムでは、実際に図 8 のようなスキーマを持った RDB とストリームを想定している。システムの利用者は、出展者と入場者に分かれている。出展者は、会場内にそれぞれのブースを開設して展示を行う側で、展示会の開催中に入場者に対して様々なアナウンス (Announce ストリーム) を流すことができる。展示内容のコンテンツや出展者のブースの位置、出展物のカテゴリを表すキーワード等は事前に RDB へ登録されている (Display テーブル, Category テーブル)。入場者は出展物を見る側であり、携帯端末を持ち歩きながら、端末上の地図で情報提供を受けることができる。位置情報は、携帯端末の無線 LAN 装置によって EPE [6] が検出する (Pda ストリーム)。入場者の個人情報や興味のあるキーワードは、入場前に RDB に登録するものとする (Guest テーブル, Interest テーブル)。展示会場は、図 7 のように、いくつかの展示ブースと通路および休憩所からなっている。休憩所の椅子の地図上の座標が RDB に格納されており (Chair テーブル)、また、椅子の使用状況は、椅子の背もたれに当たる光の強弱として、Mote [7] の光センサが検知する (図 9)(Sensor ストリーム)。

4.2.2 連続的問合せ

デモシステムで用いられている連続的問合せについて説明する。図 10 はデモシステムで用いられている連続的問合せの処理要求を図示したものである。実際に使われている連続的問合せは以下で説明する。

- RDB
 - 出展者の情報
Display (Id, Name, Address, Title, Content, Location)
出展者ID, 出展者名, メールアドレス, 出展物のタイトル, 内容, 出展場所
 - 出展物の分類
Category (DisplayId, Keyword)
出展者ID, 出展物のキーワード
 - 入場者の情報
Guest (Id, Name, Address, PdaId)
入場者ID, 名前, メールアドレス, 所持しているPDAのID
 - 入場者の興味・関心事
Interest (GuestId, Keyword)
入場者ID, 関心があるもののキーワード
 - 休憩所の椅子の情報
Chair (NodeId, X, Y)
センサID, 地図上のX座標, 地図上のY座標
- ストリーム
 - PDAの位置情報
Pda (Id, Location)
PDAのID, 現在地の論理エリア名
 - 椅子の空き状態を知らせる
Sensor (NodeId, Light)
センサID, 光センサから取得した値
 - アナウンスを伝える
Announce (DisplayId, Title, Content)
出展者ID, アナウンスのタイトル, アナウンスの内容

図 8 RDB とストリームのスキーマ定義

Fig. 8 Schema definitions of RDBs and streams



図 9 椅子に取り付けられた Mote

Fig. 9 Mote on Chair

a) アナウンスを提供する

```

MASTER Announce
SELECT Announce.*, Display.Name, Display.Location
FROM Announce[1msec], Display, Guest,
      Category, Interest
WHERE Display.Id = Category.DisplayId
      AND Guest.Id = Interest.GuestId
      AND Category.Keyword = Interest.Keyword
      AND Display.Id = Announce.DisplayId
      AND Guest.Id = <MyId>
  
```

この問合せは、出展者側の発信したアナウンス情報から、各入場者にとって興味のある出展者のものだけを選び出して通知する、という問合せで、Announce ストリームと 4 つの RDB のデータの統合を行っている。

b) 場所にに応じた出展物の情報を提供する

```

MASTER Pda
SELECT Display.*
FROM Pda[1msec], Display, Guest
WHERE Pda.Location = Display.Location
      AND Guest.PdaId = Pda.Id
      AND Guest.Id = <MyId>
  
```

この問合せは、入場者の現在位置に応じた出展物の情報を提

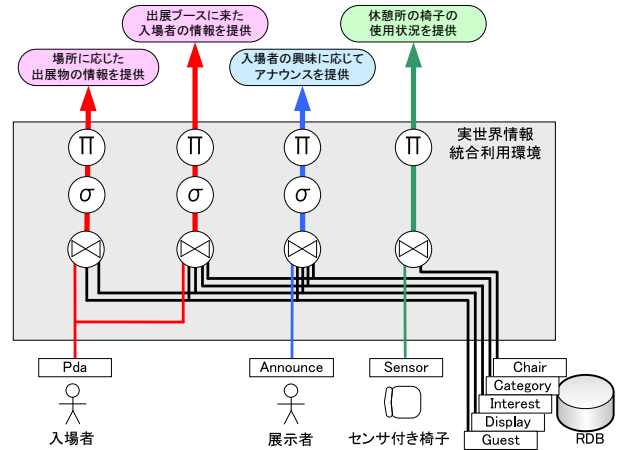


図 10 連続的問合せの処理の流れ

Fig. 10 Processing flow of the continuous query

示する、という問合せで、Pda ストリームと 3 つの RDB の統合を行っている。

c) 休憩所の椅子の使用状況を提供する

```

MASTER Sensor
SELECT Sensor.Light, Chair.X, Chair.Y
FROM Sensor[1msec], Chair
WHERE Chair.NodeId = Sensor.NodeId
  
```

この問合せは、休憩所の椅子の位置とその使用状況を通知する問合せである。Sensor ストリームと休憩所の椅子の場所の情報が入っている Chair テーブルを統合している。

d) 展示ブースに来た入場者の情報を提供する

```

MASTER Pda
SELECT Guest.*
FROM Pda[1msec], Guest, Display, Category, Interest
WHERE Guest.PdaId = Pda.Id
      AND Display.Location = Pda.Location
      AND Guest.Id = Interest.Id
      AND Category.Keyword = Interest.Keyword
      AND Display.Id = <MyId>
  
```

この問合せでは、出展物を見に来た入場者の情報を展示者側へ提供する。入場者の位置情報を提供する Pda ストリームと、4 つの RDB を統合している。

4.2.3 GUI

本デモシステムの GUI について述べる。GUI は Java の Swing を用いて構築されている。入場者側の GUI を図 11 に示す。図 11 では、我々の研究グループの研究室のレイアウト図を使用している。入場者側の GUI は 3 つの表示領域に分けられている。左側に展示会場の地図が配置されており、ユーザの位置や休憩場所などの情報を視覚的に理解できるようになっている。利用者の現在いるブースは緑色で表示される。右側の上部はテーブルになっており、システムからアナウンスや出展物に関するデータを受信すると、逐次行が追加されていく。テーブルの各行は、データのタイプ(アナウンスか出展内容か)、データの発信者、データのタイトル、データを配信した場所(論理エリア名)が入るようになっており、データのタイトルをマウスでクリックすると、右下の部分にそのデータの具体的な内容(配信されたコンテンツ)

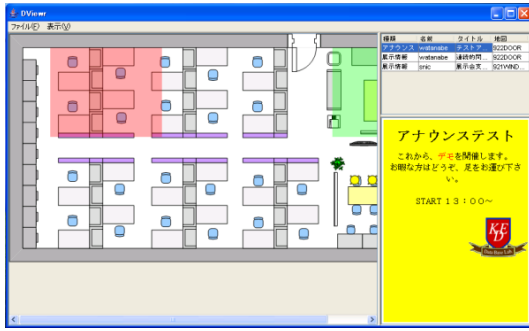


図 11 入場者側 GUI

Fig. 11 GUI of the application

CPU	Pentium4 2.8GHz
OS	Windows XP Professional SP2
メモリ	1GB
Java	J2SE 1.4.2
DBMS	MySQL 4.1

表 1 実験環境

Table 1 Experiment Environment

が表示されるようになっている。テーブル中の論理エリア名をクリックすると、左側の地図にその場所が赤色で表示される。

5. 性能評価実験

本研究で行った実験について説明する。

展示会では、その性質上たくさんの方が集まることが予測できる。本研究で想定しているアプリケーションでは、入場者各人が携帯端末を持ち歩くため、多くの連続的問合せを早く処理することが要求される。そこで本実験では、本デモシステムで使用される連続的問合せが多数登録されても、許容できる時間範囲でシステムが結果を生成することができるかどうかを検証した。本実験はいずれも表 1 に示す環境で行った。

以下で、実験の方法を説明する。本実験はシミュレーション実験となっている。展示ブースの数は 1000 で固定とし、入場者の人数を可変とした。システムを 30 分間動かし続け、そのときの平均応答遅延時間を算出する。ここで言う応答遅延時間とは、ストリームデータがシステムに到着してから、システムが問合せ結果を配信するまでの時間のことを指す。入場者の人数を 250 ~ 1250 まで 250 刻みで計 5 回の測定を行い、それぞれの平均応答遅延時間から検証を行う。入場者の人数は連続的問合せの登録数に対応している。本実験では、ストリームデータは人工的に生成したものをを用いている。実験 1 では 4.2.2 b) に示す連続的問合せを用い、実験 2 では 4.2.2 c) に示す連続的問合せを用いた。

5.1 実験 1

実験 1 では、位置情報 (Pda) ストリームに対する連続的問合せを用いて測定した。位置情報ストリームは 10 秒毎に配信されるように設定し、システムには 4.2.2 b) に示す連続的問合せのみを登録した。また、人の移動のシミュレートには乱数を用いた。

実験結果を図 12 に示す。図 12 は、入場者数 (システムに登録されている問合せ数) を x 軸に、各入場者数における 30 分間の

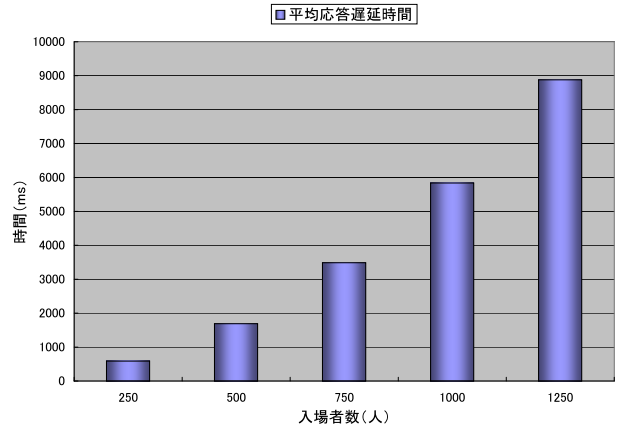


図 12 4.2.2 b) の問合せにおける平均応答遅延時間

Fig. 12 Average response-delay time relevant to the query of 4.2.2 b)

平均応答遅延時間を y 軸に取ったグラフである。これより、入場者数に応じて平均応答遅延時間が増えていることが分かる。この実験によって、図 12 の問合せを用いた場合、現システムではおよそ 1250 人分のモニタリングが約 9 秒の応答遅延時間で実現可能であることが確認できる。4.2.2 b) は、展示ブースにいるときにその出展物の情報を通知する問合せである。出展物の情報を通知してほしいと考えている利用者が、10 秒足らずでその場を移動するとは考え難いので、約 9 秒で結果が通知できるのであれば十分実用可能である。

5.2 実験 2

実験 2 では、センサネット (Sensor) ストリームに対する連続的問合せを行った。システムには、4.2.2 c) に示す連続的問合せを人数分登録する。センサの数を 500 個で固定し、センサから取得した値は乱数を返すようにした。また、センサネットのサンプリング間隔は 10 秒とした。

実験結果を図 13 に示す。図 12 と同様に、x 軸は入場者数、y 軸は平均応答遅延時間を表す。実験 1 の連続的問合せは、各入場者が各々の場所と興味に応じた別の結果を取得するのに対し、今回の連続的問合せでは、各入場者の得られる結果は同じである。ただし、センサの台数分のタプル (500 タプル) を一度に受け取るという点で異なる。図 13 から、入場者数が 1250 人程度のときは、センサのサンプリング間隔を 3 秒にしてもきちんと動作することが分かる。4.2.2 c) は、休憩所の椅子の使用状況を通知する連続的問合せである。休憩所の椅子の使用状況は、利用者が椅子を使用したいと思ったときに確認する程度なので、10 秒以内で結果が得られるのであれば十分実用可能である。また、この実験では処理時間に余裕があったので、入場者数が 5000 人の場合においても測定を行った。図 13 に示すように、5000 人分の問合せでも 10 秒以内で処理結果の配信が可能である。

6. 関連研究

本研究に関連する主な研究について述べる。

センサネットからセンサデータを取得する研究として、Cougar [5] や TinyDB [8] がある。共に、センサデバイスの制御や、センサの値の取得を SQL ライクな問合せ言語で行う機能

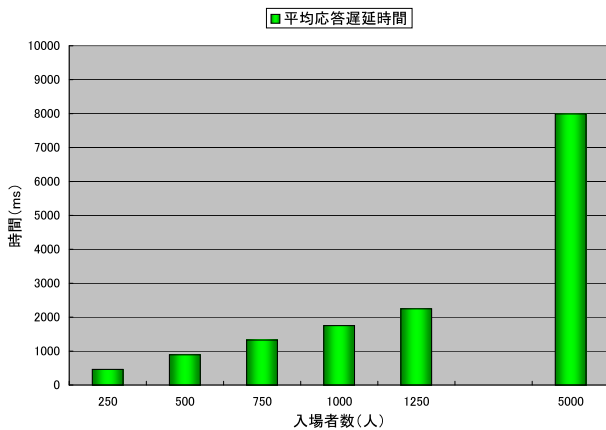


図 13 4.2.2 c) の問合せにおける平均応答遅延時間

Fig. 13 Average response-delay time relevant to the query of 4.2.2 c)

を提供している。これらの研究では、センサから取得したデータを既存の RDB のデータなどと統合利用できるような問合せ処理機能は提供されていない。

移動するオブジェクトに周辺の情報を提供するための空間データベース検索システムについての研究に [10] などがある。こちらは位置情報の利用に特化したシステムであるため、ストリームや他のセンサ情報との統合利用については考えられていない。

Borealis [9] は、ストリームとセンサネットワークの統合利用のためのシステムである。ストリームとセンサネットワークの統合においては、両者の性質の違いを考慮した問合せ最適化を行っている。本研究では、統合利用環境の構築だけでなく、センサネットワークと位置情報およびストリームを統合利用した現実的なアプリケーションについて考察し、実際にデモシステムの開発・性能評価を行った。

CSAVA [16] もまた、ストリームとセンサネットワークの統合利用について考えられている。ストリームとセンサネットワークを統合利用する状況として、サプライチェーンマネージメントを考えており、実際にプロトタイプシステムの開発を行っている。プロトタイプシステムでは、ストリーム処理エンジンとして TelegraphCQ [15] を、センサネットワークとして TinyDB を用いている。評価実験において有効性が確認されていない。

その他に、連続的問合せを用いたストリーム処理システムに関する研究について述べる。OpenCQ [11] は分散異種情報源に対する情報統合システムである。この連続的問合せは、問合せ、発火条件、終了条件から構成されており、発火条件が真になると、終了条件が満たされるまで繰り返し問合せを実行する。NiagaraCQ [12], [13] もまた、連続的問合せを用いたストリーム処理システムの 1 つである。連続的問合せがインクリメンタルに追加・削除される状況を想定しており、既存の実行プランを元に新しい実行プランを構築し、問合せ最適化を行っている。しかし、時間条件などを考慮していないので、ストリーム情報源に対する問合せとしては非現実的である。またその他に、Aurora [14] がある。Aurora は、ストリームデータの処理に QoS の概念を取り入れている。システムは利用者の与えた QoS 指定を元に、ク

オリティ低下を最小限に抑えるような処理を行う。ストリームデータの到着が重なってシステムの処理能力を超えたときは、入力データを減らして負荷を軽減する機能を持っている。

7. おわりに

本研究では、ストリーム統合エンジンにセンサ情報と位置情報をストリームとして取り込むモジュールを開発し、実世界情報統合利用環境を構築した。また、実世界情報統合利用環境を用いて展示会支援アプリケーションのデモシステムを開発し、デモシステム環境を想定した性能評価実験を行った。

今後の課題としては、より多様なセンサデバイスから取得できる実世界情報を統合することや、大規模な実証実験を行うことが上げられる。

謝辞 本研究の一部は、日本学術振興会特別研究員奨励費 (15・330)、科学研究費補助金特定領域研究 (2)(#16016205)、基盤研究 (B)(#15300027) による。

文 献

- [1] 渡辺陽介, 北川博之, 内山大悟. "問合せ最適化機構を備えたデータストリーム統合システムの開発", DEWS 2004.
- [2] 渡辺陽介, 北川博之. "連続的問合せに対する複数問合せ最適化手法", 電子情報通信学会論文誌, Vol.J87-D-I, No.10, pp.873-886, 2004 年 10 月.
- [3] TinyDB, <http://telegraph.cs.berkeley.edu/tinydb/>
- [4] TinyOS, <http://www.tinyos.net/>
- [5] Y. Yao and J. Gehrke. "Query Processing for Sensor Networks", Proc. Conference on Innovative Data Systems Research 2003.
- [6] Ekahau, <http://www.ekahau.com/>
- [7] Crossbow Inc, <http://www.xbow.com/>
- [8] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. "The Design of an Acquisitional Query Processor for Sensor Networks", Proc. ACM SIGMOD Conference, pp.491-502, 2003.
- [9] D. J. Abadi, Y. Ahmad, M. Balazinska, U. Cetintemel, M. Cherniack, J. H. Hwang, W. Lindner, A. S. Maskey, A. Rasin, E. Ryvkina, N. Tatbul, Y. Xing, and S. Zdonik. "The Design of the Borealis Stream Processing Engine", CIDR 2005.
- [10] Y. Ishikawa, H. Kitagawa, and T. Kawashima. Continual NeighborhoodTracking for Moving Objects Using Adaptive Distances, Proc. of International Database Engineering and Applications Symposium (IDEAS '02), Sept. 2002.
- [11] L. Liu, C. Pu, and W. Tang. "Continual Queries for Internet Scale Event-Driven Information Delivery", IEEE Trans. Knowledge and Data Engineering, vol.11, no.4, pp.610-628, 1999.
- [12] J. Chen, D. J. DeWitt, and J. F. Naughton. "Design and Evaluation of Alternative Selection Placement Strategies in Optimizing Continuous Queries", Proc. International Conference on Data Engineering, pp.345-356, 2002.
- [13] J. Chen, D. J. DeWitt, F. Tian, and Y. Wang. "NiagaraCQ: A Scalable Continuous Query System for Internet Databases", Proc. ACM SIGMOD Conference, pp.379-390, 2000.
- [14] D. J. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, and S.Zdonik, "Aurora: a new model and architecture for data stream management", VLDB Journal Vol.12, No.2, pp.120-139, 2003.
- [15] S. Chandrasekaran, O. Cooper, A. Deshpande, M. J. Franklin, J. M. Hellerstein, W. Hong, S. Krishnamurthy, S. Madden, V. Raman, F. Reiss, and M. Shah. "TelegraphCQ: Continuous Dataflow Processing for an Uncertain World", Proc. Conference on Innovative Data Systems Research 2003.
- [16] M. J. Franklin, S. R. Jeffery, S. Krishnamurthy, F. Reiss, S. Rizvi, E. Wu, O. Cooper, A. Edakkunni, and W. Hong. "Design Consideration for High Fan-in Systems: The HiFi Approach", Proc. Conference on Innovative Data Systems Research 2005.