

実世界情報ストリームの高度利用のための統合環境

An Integration Environment for Advanced Use of Real World Data Stream

山田 真一[♥] 渡辺 陽介[♥]
北川 博之[♦]

Shinichi YAMADA Yousuke WATANABE
Hiroyuki KITAGAWA

近年、自発的に情報を配信するストリームと呼ばれる情報源が増加し、ストリームデータに対する問合せ処理の重要性が高まっている。一方、通信技術やデバイス技術の発展により、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った。これらセンサから得られる情報は逐次配信されるため、一種のストリームと考えることができる。ユビキタス社会の到来に向けて、実世界情報に対する問合せ処理は今後益々重要となってくる。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、実世界情報統合利用環境を構築した。本稿では、そのアーキテクチャと共に、実際にその統合利用環境を用いて開発した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。

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 have implemented. In this paper, we explain the architecture and an application using our system. Also, we show results of experiments that we have done to measure the performance of our system.

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ストリーム統合システム[6]~[8]が幾つかの研究グループで開発されており、我々の研究グループでも研究開発を行っている[1]。

[♥] 学生会員 筑波大学システム情報工学研究科
snic.watanabe@kde.cs.tsukuba.ac.jp

[♦] 正会員 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 kitagawa@cs.tsukuba.ac.jp

一方、センサ機器の改良等によって、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。自発的にネットワークを構築する無線センサデバイス[5]が開発され、遠く離れた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可能となり、また、GPS等の普及により、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。このような理由から、実世界から得られる温度や光、位置などの情報とストリームや既存のRDBの統合利用は今後益々重要となってくる。

そこで本研究では、我々の開発しているストリーム統合システムをベースに、実世界情報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新たに、設計・開発した。また、本研究では実世界情報統合の現実的なアプリケーションを想定したプロトタイプ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。

2. 実世界情報

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。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[5]がある。実世界情報は、Moteに付属しているセンサによって取得される。取得できる情報には、温度、光、加速度などがある。また、MoteにはCPUが搭載されており、プログラミングが可能である。

現在、Mote上で動くOSとしてTinyOS[3]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[2]がある。利用者が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。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。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。TinyDBは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。

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。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[4]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慮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

3.1 ストリーム統合エンジン

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、我々がこれまでに開発したストリーム統合エンジン[1]を用いて実現している。図1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 等の情報源とメディエータ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メディエータ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。

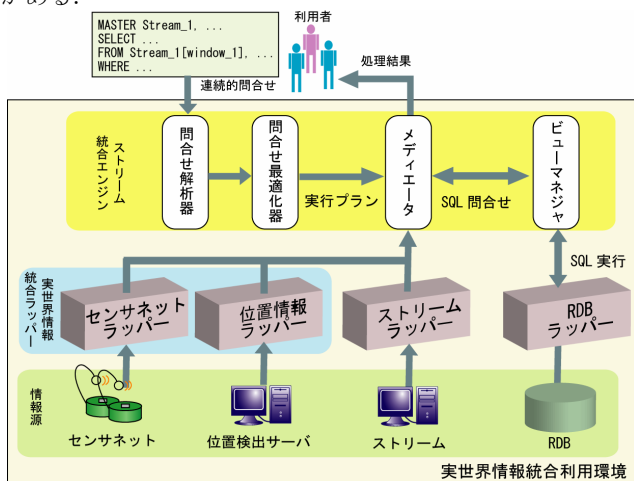


図1 実世界情報統合利用環境

Fig.1 System for Integration Real World Information

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、新規に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。連続的問合せは MASTER-SELECT-FROM-WHERE 節から成っている。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 (ウインドウ) を指定することが可能となっている。

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2)。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、TinyDB に登録する。TinySQL には取得したセンサの値の属性名と、サンプリングレートが記述されている。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ようになる。また、センサネットラッパーは 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、メディエータに提供する。例えば、図2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 問合せがストリーム定義ファイルに記述されていた場合は、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生成し、直ちにメディエータに配信する。図2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、複数個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。

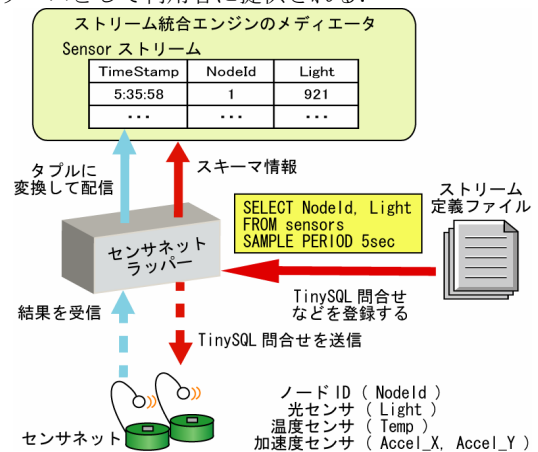


図2 センサネットラッパー

Fig.2 Sensor Net 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。センサネットラッパーと同様に、ストリーム定義ファイルの記述に従って、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エータに提供する。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、直ちにメディエータに配信する。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。

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

を想定しており、実際にプロトタイプシステムを実装した(図3)。

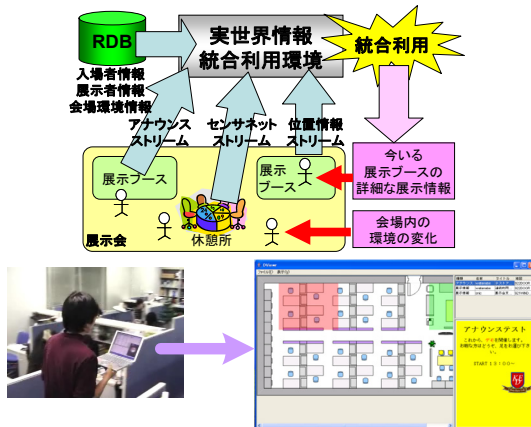


図3 展示会支援アプリケーション
Fig.3 Exposition Supporting Application

4.1 スキーマ定義

プロトタイプシステムでは、実際に図4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、展示内容のコンテンツや出展者のブースの位置などは事前にRDBへ登録されている(Displayテーブル)。また、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。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。入場者の個人情報などは、入場時にRDBに登録するものとする(Guestテーブル)。位置情報は、携帯端末の無線LAN装置によってEPE[4]が検出する(Pdaストリーム)。展示会場は、図3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成っている。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[5]の光センサが検知する(Sensorストリーム)。

- ・ 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, 出展者のタイトル, 内容, 出展場所
 - ・ 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
 - ・ 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
- ・ STREAM
 - ・ PDA の位置情報
Pda (Id, Location)
PDA の ID, 現在地のエリア名
 - ・ 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4 RDBとストリームのスキーマ定義

Fig.4 Schema definitions of RDBs and streams

4.2 連続的問合せ

プロトタイプ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5は、入場者の現在位置に応じた出展物の情報を提示する、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。図6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。実際には、これらの他に出品者側の発信したア

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選び出して通知する問合せと出展物を見に来た入場者の情報を展示者に通知する問合せが使用されているが、ここではスペースの都合上省略する。

```

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

図5 場所に応じた出展物の情報を提供

Fig.5 Provision of information about displayed objects

```

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

図6 休憩所の椅子の使用状況を提供

Fig.6 Provision of information which chairs are vacant

4.3 GUI

本プロトタイプシステムのGUIについて述べる。GUIはJavaのSwingを用いて構築されている。図3に示すように、入場者側のGUIは3つの表示領域に分けられている。左側の領域に展示会場のレイアウト図が配置されており、ユーザの位置や休憩場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。右上の領域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、データの発信者、データのタイトル、展示ブース名等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、右下の領域に配信されたコンテンツが表示される。展示ブース名をクリックすると、左側のレイアウト図にその場所が赤色で表示される。

5. 性能評価実験

展示会では、その性質上たくさんの人が集まることになる。入場者各人につき図5, 6のような連続的問合せが必要となるため、大量の連続的問合せを効率よく処理することが要求される。そこで本実験では、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。

表1 実験環境

Table 1 Experiment Environment	
CPU	Pentium 4 2.8GHz
OS	WindowsXP Professional SP2
Memory	1GB
Java	J2SE 1.4.2
DBMS	MySQL 4.1

以下で、実験の方法を説明する。本実験は人工的に生成したストリームデータで行った。展示ブースの数を1000、センサの数を500で固定とし、入場者の人数を可変とした。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、人の移動のシミュレートには乱数を用いた。また、センサから取得した値は乱数を返すようにし、サンプリング間隔は10秒とした。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回の計測を行い、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。実験1では図5に示す連続的問合せを用い、実験2では図6に示す連続的問合せを用いた。

5.1 実験1

実験1では、図5に示す位置情報(Pda)ストリームに対する連続的問合せを用いて測定した。

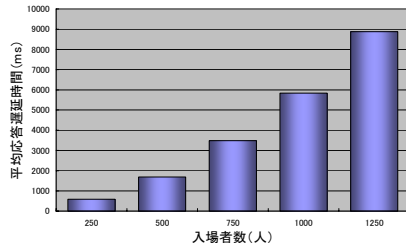


図7 図5の問合せにおける平均応答遅延時間

Fig.7 Average delay time relevant to the query of Fig.5

実験結果を図7に示す。図7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は増えていることがわかる。図5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。この実験によって、図5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可能であることが確認できる。

5.2 実験2

実験2では、図6に示すセンサネットワーク(Sensor)ストリームに対する連続的問合せを行った。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。

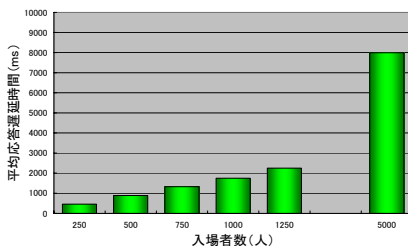


図8 図6の問合せにおける平均応答遅延時間

Fig.8 Average delay time relevant to the query of Fig.6

実験結果を図8に示す。図7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。図8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。図6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8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ことが確認できた。

6. まとめと今後の課題

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、プロトタイプ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. "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.
- [2] TinyDB, <http://telegraph.cs.berkeley.edu/tinydb/>
- [3] TinyOS, <http://www.tinyos.net/>
- [4] Ekahau, <http://www.ekahau.com/>
- [5] Crossbow, <http://xbow.com/>
- [6] 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.
- [7] 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 System Research 2003.
- [8] D. J. Abadi, Y. Ahmad, M. Balazinska, U. Cetintemel, M. Cherniack, J. H. Hwang, W. Linder, A. S. Maskey, A. Rasin, E. Ryzkina, N. Tatbul, Y. Xing, and S. Zdonik. "The Design of the Borealis Stream Processing Engine", Proc. CIDR pp.277-289, 2005.

山田 真一 Shinichi YAMADA

筑波大学大学院システム情報工学研究科在学中。2005 同大・第三学群情報学類卒業。実世界情報統合利用に関する研究に従事。日本データベース学会学生会員。

渡辺 陽介 Yousuke WATANABE

筑波大学大学院システム情報工学研究科在学中。2001 同大・第三学群情報学類卒業。異種分散情報源統合利用システムにおける配信型情報源の統合利用に関する研究に従事。日本データベース学会学生会員。

北川 博之 Hiroyuki KITAGAWA

筑波大学大学院システム情報工学研究科, 計算科学研究センター教授。1980 東京大学大学院理学系研究科修了。理学博士(東京大学)。異種情報源統合, 文書データベース, WWWの高度利用などに興味をもつ。著書「データベースシステム」(昭晃堂), 「Unnormalized Relational Data Model」(共著, Springer-Verlag)など。ACM SIGMOD 日本支部長, 日本データベース学会理事, ACM, IEEE-CS, 情報処理学会, 電子情報通信学会, 日本ソフトウェア科学会各会員。