

マルチモーダルセンサ情報活用のための 検索クエリと可視化フレームワーク

出口祐輝^{†1} 石川翔吾^{†2} 竹林洋一^{†3}

多様なセンサデータをネットワーク上で手軽に収集、可視化するための開発支援環境を構築した。時間情報付きというセンサデータの特性に合わせたデータベース構造と検索クエリ言語を設計し、複数の時系列データからのデータ抽出、分析を可能とした。本フレームワークを適用して実験的なサービスを構築し、試験運用した結果、センサを活用したサービスの開発、設計検証に有効であることを示した。

A Search Query and the Visualization Framework for Utilizing Multimodal Sensor Data

YUKI DEGUCHI^{†1} SHOGO ISHIKAWA^{†2}
YOICHI TAKEBAYASHI^{†3}

We have proposed a framework to collect and visualize a multimodal sensor data over the network. We designed a database structure and a query language to deal with flexibly sensor data with time-stamp. The framework enables to extract desired data from a multiple time-series data. Developing experimental services using the framework, we have shown the effectiveness of the proposed framework for efficient services development.

1. はじめに

近年センサデバイスやネットワークの発達に伴い、インターネット上で取り扱われるデータが爆発的に増大している。これら进行分析し、新しい情報サービスやマーケティング等へ活用しようとする取り組みが増えている。ビッグデータのビジネス活用を代表として、スマートフォンによる交通情報の収集[1]や、センサと SNS の連携による高齢者見守りサービス[2]、医療診断への活用[3]など、多方面で情報サービスとして活用されている。このような取り組みが増えてきている一方で、収集したデータから有用な情報を抽出し活用するためには、応用分野ごとに異なる方法論が必要となる。ビッグデータ分析の課題としても、プライバシー保護の問題や、非構造データによるデータ粒度の不一致に並んで、多量のデータを分析し、活用法を提案できるデータ分析技術者の不足などが取り上げられている [4]。そこで、多様なセンサデータを汎用的に収集し、手軽に可視化できる環境があれば、分析技術者でなくても試行錯誤しながら発展的にデータの活用法を探ることができる。

センサデータから有用な情報を抽出するにあたって、例えばカーナビゲーションシステムが GPS、ジャイロ、加速度センサから複合的に位置推定をしているように、単一のセンサデータよりも、異なる種類のセンサデータを組み合わせるほうが、センシング対象の状況をより正確に抽出で

きると推測される。また、センサデータは必ずセンシングした時間情報が付随する時系列データであり、複数のセンサデータを組み合わせるには、時間情報を考慮した仕組みが必要となる。

本稿では、異なる種類のセンサデータをネットワーク上で手軽に一元管理、統合、可視化する開発支援環境の構築について述べる。

2. センサ情報統合のためのフレームワークの必要性

2.1 現状の開発支援環境

大量のデータを簡単に、スケーラブルに管理できる開発支援環境として、近年ではクラウド型のデータベースシステムが普及しつつある。Amazon Relational Database Service[5]や、Amazon SimpleDB[6]等のサービスは、ネットワーク経由でクラウドへのデータの格納、検索が可能であり、遠隔地に点在するセンサのデータ収集に利用できる。これらのサービスは一般的なテーブル型（表）のデータ構造を持ち、SQL ライクなクエリ言語を用いてデータの検索を行う。

図 1 に示すように、センサデバイスから送られてくる連続的なデータは、センシングした時間情報付きの時系列データとして捉えることができる。一種類のセンサデータであれば、従来のテーブル型データ構造と SQL でも比較的容易に表現できる。しかし、複数の時系列データを表現するには、センサ用のキー列を設定して 1 テーブルに異種センサデータ同士を混在させるか、センサ毎にテーブルを分割することになるため、センサ同士の時間関係を考慮しなが

^{†1} 静岡大学 情報学研究科
Graduate School of Informatics, Shizuoka University

^{†2} 静岡大学 情報学部
Faculty of Informatics, Shizuoka University

^{†3} 静岡大学 創造科学技術大学院
Graduate School of Science and Technology, Shizuoka University

ら直感的にデータ抽出するには不向きである。

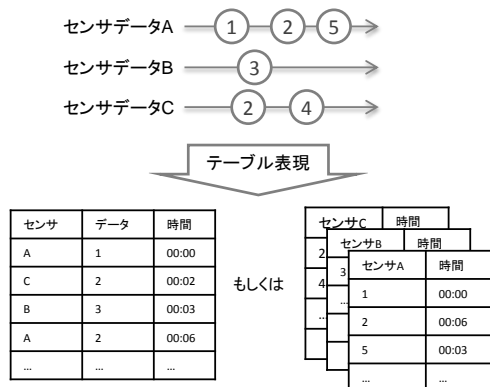


図 1 複数時系列データのテーブル表現の例

Figure 1 Examples of the table description of multi time-series data.

本フレームワークは、センサデータの特性である時間情報に焦点を当て、複数の時系列データの取り扱いに特化した開発支援環境となっている。

2.2 データ形式

GPS、加速度、ジャイロ、温度、IR、RFID、カメラ、マイクなど、様々な種類のセンサが日常的に利用されている。これら形式の異なるセンサデータを同じ枠組みの上で管理し、データ同士を組合せて迅速な可視化を実現するために、本フレームワークではセンサデータを Raw データそのままではなく、「状況 (Context)」と呼ぶ情報粒度で表現する。「状況」は、Raw データをセンサデバイス側で処理し、上位の意味を抽出した、有限個の状態を文字列、または数値で表現したデータである。表 1 に Raw データと状況抽出処理、状況データの例を示す。

表 1 状況データの例

Table 1 Examples of context data.

Raw データ	状況抽出処理	状況データ
スマートフォンの三軸加速度データ	歩数計アプリ	1, 2, 3... (歩数)
監視カメラ映像	画像処理	異常なし, 不審者発見, 不明
音声データ	音声認識	(認識結果の文字列)

これにより、本来の Raw データを抽象化し情報量を切り捨てることになるが、本フレームワークは、データ分析技術者でなくても容易にセンサデータの活用法を試行錯誤できるよう、この方式によって得られる以下の利点を優先している。

- 異種センサデータを汎用的に同一の DB、検索クエリで扱える
- 異種センサデータ同士の組合せが容易になり、単一データでは読み取れない上位の意味を抽出できる
- データとビューエンジンの結合処理を簡略化することができ、容易に可視化できる

- トラフィックの軽減、サーバのデータ容量削減による低コスト化

- 音声、映像などのデータを直接保存せず、文字、数値情報に丸め込むことによるプライバシーの向上

また、センサデバイス側にデータ処理を負担することになるが、多数のセンサを内蔵しているスマートフォンの普及や、オムロンのクラウド連携型ヘルスケアデバイス[7]の例に見るように、センサデバイスの処理能力向上に伴い、デバイス側での高度な情報処理が現実的なものとなっている。なお、開発支援環境としての汎用性を考慮して、数値、文字列情報であれば Raw データそのままの取り扱いも可能である。

2.3 センサデータの処理過程

図 2 に本フレームワークのデータ処理過程を示す。

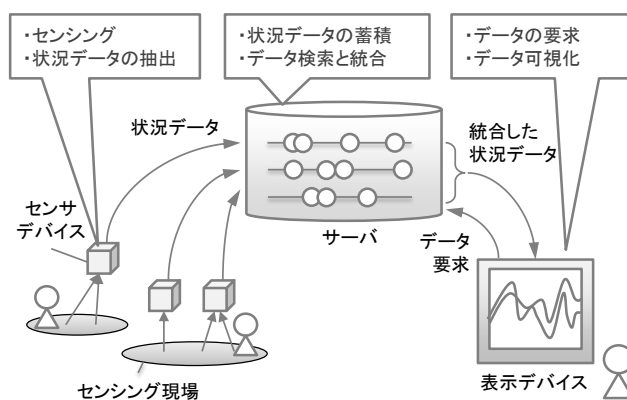


図 2 センサデータの処理過程

Figure 2 Processing of sensor data.

2.3.1 センサデバイスのデータ処理

センサデバイスは、センシングした Raw データからの状況データ抽出と、サーバへのデータ送信を担当している。

状況データの抽出アルゴリズムや、状況の判断基準は、センサの種類や環境によって異なるため、センサデバイス側に一任している。これにより異種センサや現場の差異を吸収し、大規模なセンサ情報の集約を可能とする。

2.3.2 サーバのデータ処理

サーバはセンサデバイスから送られてくる状況データの蓄積、及び表示デバイスから送られてくるデータ要求(検索クエリ)に従って、データの検索と統合を行い、表示デバイスへの返送を担当している。

2.3.3 表示デバイスのデータ処理

表示デバイスはセンサデータとビューエンジンの結合と可視化処理を担当する。センサデータだけを受信し、可視化処理を表示デバイス上で実行することで、トラフィックとサーバ負荷の軽減が期待できる。

表示デバイスから簡単にセンサデータの受信処理と可視化ができるよう、これらを代行する可視化ライブラリを提供している。

2.4 サーバのデータ構造

図3に示すのは本フレームワークのサーバのデータ構造である。時系列データを保存するTimeLineと、それらを管理する4階層のツリー構造から構成されている。

各TimeLineがセンサー一つの時系列データに相当し、センサからデータ(Value)が送られてくると、タイムスタンプ(TimeStamp)を付与し、TimePoint構造体として時系列順に蓄積していく。

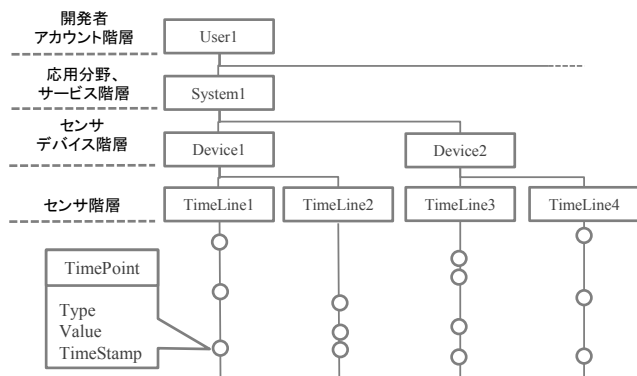


図3 本フレームワークのサーバデータ構造
Figure 3 Server data structure of our framework.

2.5 応用シナリオ

センサデータの集約、検索、可視化に特化した本フレームワークの特色を活かし、実際に情報サービスとして活用できる応用シナリオについて検討した。

2.5.1 医療診断、高齢者見守りサービス

日常生活のライフログを本フレームワークを用いて収集し、医療診断の材料に利用するなどの応用が期待できる。本フレームワークを用いれば、ライフログ中に埋もれた危険な生活パターンなどの抽出条件を簡単に指定できる。

また、ネットワークを通じて大勢の患者の情報を収集できるため、遠隔地に居ながら医師や親族が健康状態をチェックすることができる。

2.5.2 スマートホームシステム

家庭、オフィス、店舗などの住空間にセンサを埋め込み、電力の見える化や家電機器のコントロール等に活かすサービスへの応用が期待できる。これらの住空間は環境間で物的、人的な差異が大きく、各環境に合わせたシステム構成やサービスのカスタマイズが必要となる。

そこで本フレームワークによってセンサ情報を集約、可視化し、試行錯誤しながら環境毎に合わせたサービスの検討を支援する。

3. システム構成

3.1 全体のシステム概要

図4に本フレームワークの全体像を示す。本フレームワークは大きく分けて、センサデバイス、サーバアプリケー

ションである Contexts Management System, 表示デバイスと、その上で実行される可視化 Web コンテンツから構成される。

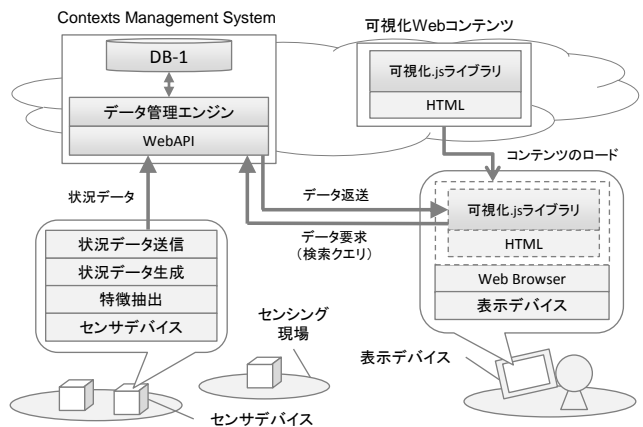


図4 システム構成
Figure 4 System architecture.

3.2 Contexts Management System (CxMS)

本フレームワークの中心となるサーバアプリケーションである。各センサデバイスから送られてくるセンサデータをデータベースへ蓄積し、また表示デバイスから送られてくる検索クエリに従い、センサデータを検索、統合して返送する役割を持つ。

本フレームワークを利用してサービス開発をする利用者(以下利用者と呼称)は、CxMSのWeb管理コンソールに管理者アカウントでログインすることで、データベースの閲覧、管理を行うことができる。なお、利用者が試行錯誤しながらサービス開発することを考慮して、CxMSのデータベースはスキーマフリーとなっており、データ構造をいつでも自由に追加、変更することができる。

3.3 WebAPI

各センサデバイス、表示デバイスと、CxMS間のインターフェースとなるウェブサービスであり、データの登録/更新/検索/削除要求を受け付けるサービスを提供する。各デバイス間はREST (REpresentational State Transfer) に基づいた通信を行う。要求はHTTP (HyperText Transfer Protocol) リクエストで受け付け、結果はJSON (JavaScript Object Nation) オブジェクトとして返送する。

例としてセンサデバイスから、図3におけるCxMSの『User1.System1.Device1.TimeLine1』階層へ、データ『SampleValue』を送るには、以下のURLにHTTPアクセスすることでデータを送信することができる。

`http://CxMSのURL/UPDATE?
path=User1.System1.Device1.TimeLine1&val=SampleValue`

HTTP は数多くの開発環境でサポートされており、IP ネットワークに接続できるセンサデバイスであれば、容易に CxMS との通信機能が組み込める。そのため、一般的なパーソナルコンピュータやモバイル機器を、センサデバイスと見立てて情報を収集することも可能となっている。

3.4 検索クエリ言語

3.4.1 検索クエリ言語の基本書式

表示デバイスから CxMS へデータ要求する際に送る検索クエリ言語は、広く普及している SQL をベースとして、図 3 に示す CxMS のデータ構造から効率良くデータ検索するためのドメイン固有言語となっている。基本的な書式を以下に示す。

```

http://CxMS の URL/GETDATA?query=
FROM User1.System1
SELECT Device1.TimeLine1.TTL(秒),
        Device1.TimeLine2.TTL(秒)
WHEN 2012-02-28 T 00:00:00 / 2012-03-02 T 23:59:59
WHERE Device1.TimeLine1.CONTEXT(x < 10)
        AND Device2.TimeLine2

```

(※下線部は予約語)

FROM 節, SELECT 節では、それぞれ取得したいセンサデータの User, System, Device, TimeLine 層の指定を行う。WHEN 節では取得したいデータの期間を ISO 8601 表記[8]により指定する。WHERE 節では TimeLine 同士による AND, OR, NOT 演算や、条件式による抽出条件、関数によるデータ加工の指定を行う。

3.4.2 条件式によるフィルタリング

センサデータに条件式を指定してフィルタリングを行うのが CONTEXT()関数である。引数として『x 比較演算子 値』の形式で条件式を指定する。

また、時系列データという特徴を生かして、時間的な文脈関係や状態変化を抽出するために、“状態 A から〇秒以内に状態 B へ変化した”というような条件を指定したい場合には、以下のように引数を指定する。

```
CONTEXT(条件式, 条件式@秒, 条件式@秒, ...)
```

例として、図 5 に示す心拍センサデータから、心拍が 30 秒以内に急激に上昇したデータのみを抽出したい場合には、WHERE 節を以下のように記述する。

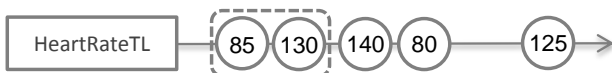


図 5 Context 関数によるデータ抽出例

Figure 5 Example of data extraction using Context function.

```

WHERE HealthDev.HeartRateTL.
CONTEXT(x < 90, x > 120 @ 30)

```

3.4.3 TimeLine 同士の AND, OR 演算

TimeLine 同士を組み合わせて AND, OR 演算しようとした場合、TimeLine 上の TimePoint はごく一瞬の一点であり、これが別の TimeLine 間で秒単位で重なる可能性は限りなく低い。TimePoint 同士の時間のズレをどこまで許容するのかが課題となるが、そこで本フレームワークでは、SELECT 節の TTL()関数(Time to Live)で、TimeLine 毎にセンサデータの有効期限を設定することで、センサデータを仮想的に連続的な値として扱い、AND, OR 演算を行なっている。

例として、図 6 に示すように機器 A の電源が ON の状態の時の消費電力計の値を抽出するには、SELECT 節と WHERE 節を以下のように記述する。

```

SELECT MachineA.PowerTL.TTL(60),
        WattChecker.WattTL.TTL(30),
WHERE MachineA.PowerTL.CONTEXT(x = 'ON')
        AND WattChecker.WattTL

```

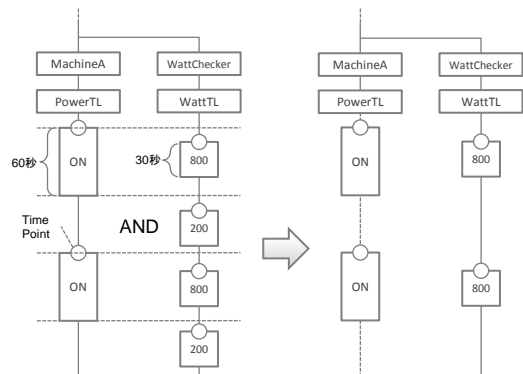


図 6 TimeLine 同士の AND 演算

Figure 6 AND operation between TimeLines.

3.5 可視化 Web コンテンツと可視化.js ライブラリ

センサデータを可視化した画面 (ビュー) は、CxMS とは独立した Web コンテンツとして提供され、表示デバイスのブラウザ上で実行される。

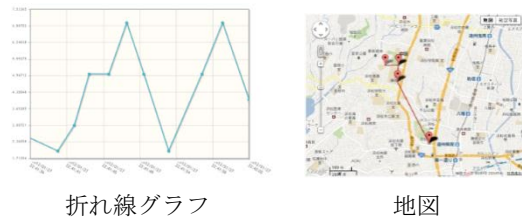


図 7 可視化機能の例

Figure 7 Examples of visualization function.

ビューエンジンとなる可視化.js ライブラリは、WebAPI との通信処理を代行し、受信した JSON データを可視化する Javascript ライブラリである。利用者は基本的な HTML の知識があれば、図 7 に示すようなライブラリの機能を利用して、データを可視化する Web コンテンツを自由にデザインすることができる。

4. 応用システムの実装事例

本フレームワークを利用して、試験的なサービスを構築、実験運用し、サービス構築にかかる時間的コストを評価した。介護施設にあるトレーニングマシンの成績自動記録を対象とし、システム構成は図 8 のようになっている。本サービス用に新たにトレーニングマシン用センサデバイスと、成績記録閲覧サイトを開発した。

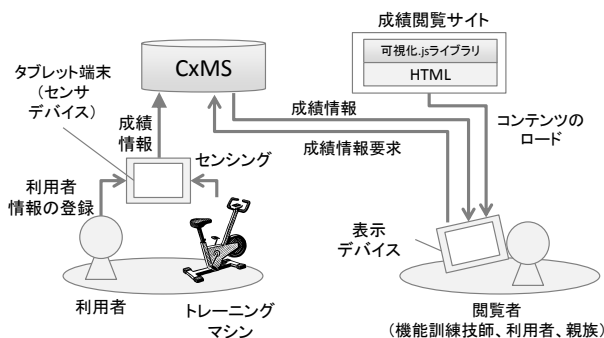


図 8 成績自動記録サービスの構成

Figure 8 Architecture of the results automatic record service.

図 9 に示すようなトレーニングマシンの稼働をセンシングし、成績記録として CxMS へデータを送信する。収集した成績記録を Web で可視化、共有することで、利用者のモチベーション向上や健康管理を狙ったサービスである。



図 9 トレーニングマシンの例

Figure 9 Examples of training machines.

本フレームワークを利用することで、本サービスを表 2 に示す期間で開発することができた。

これにより開発期間を短縮し、サービスを試用しながら現場で改良点をヒアリングするなど、センサを活用したサービス開発に貢献できる見通しを得た。

表 2 開発期間結果

Table 2 Result of development time.

開発品目	コード行数	開発言語	日数/人
センサデバイス	907	C#	5
成績閲覧サイト	326	HTML+Javascript	2
計			<u>7</u>

また、サービスを Web 上で実現しているため、成績記録を家族や医師へ共有するなど、幅広いユーザ層にサービス展開が可能である。

5. おわりに

本稿では、異なる種類のセンサデータをネットワーク上で管理、統合、可視化するフレームワークを提案した。時間情報付きというセンサデータの特徴に合わせたデータベース構造と検索クエリ言語により、複数の時系列データからのデータ抽出、分析が可能となる。

本フレームワークを利用して試験的なサービスを構築し、試験運用した結果、開発コスト削減、サービス設計の検証に有用であることを示した。センサデータの活用法を発展的に探り、センサを使った新しい情報サービス創出に貢献できる。

今後は検索クエリを含めた評価手法について検討し、新たに応用システムを実装して評価を進めてゆくと共に、時系列データを取り扱うフレームワークに必要な要素を検証してゆく。

謝辞 本研究を進めるにあたり、共同研究としてご協力頂いた株式会社 TOKAI コミュニケーションズの森藤真樹氏、栗田剛氏、実験環境をご提供頂いた TOKAI ライフプラス株式会社の玉城光氏に、謹んで感謝の意を表します。

参考文献

- 1) Arvind Thiagarajan, James Biagioni, Tomas Gerlich, Jakob Eriksson : Cooperative Transit Tracking using Smart-phones. SenSys '10 Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, pp.85-98(2010).
- 2) 中野裕介, 河上寛, 垂水浩幸 : ライフログを共有する Life Networking Service, 情報処理学会インタラクシオン 2012, pp.205-210(2012).
- 3) 伊藤達明, 石原達也, 中村幸博 他, : 医療分野におけるライフログ活用の展望, NTT 技術ジャーナル, pp33-36(2010.7).
- 4) 小栗秀暢, 松井くにお: データ分析の事業性の評価を行うプラットフォームの提案と考察, 研究報告ヒューマンコンピュータインタラクシオン (HCI), 2012-HCI-150, pp.1-8(2012-10-25).
- 5) Amazon Relational Database Service (Amazon RDS), <http://aws.amazon.com/jp/rds/>
- 6) Amazon SimpleDB, <http://aws.amazon.com/jp/simpledb/>
- 7) オムロンヘルスケア, <http://www.healthcare.omron.co.jp/>
- 8) ISO 8601:2004, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40874