

モバイル環境センシングにおける個人向けセンサボックス 実装フレームワークの提案

大櫛 章裕[†] 梶本 真佑[†] 佐伯 幸郎[†] 中村 匡秀[†]

[†] 神戸大学 〒 657-8501 神戸市灘区六甲台町 1-1

E-mail: [†]okushi@ws.cs.kobe-u.ac.jp, ^{††}{shinsuke,masa-n}@cs.kobe-u.ac.jp, ^{†††}sachio@carp.cs.kobe-u.ac.jp

あらまし ユーザがセンサを持ち歩き、自分の周辺の環境を測定し共有するモバイル環境センシングは、参加型センシングの1つの実現手法である。我々は、ユーザー一人ひとりが自前の測定装置（個人向けセンサボックス）を所有し、好きなセンサを組み合わせることで自由にモバイル環境センシングを行う環境を想定し研究を進めている。本稿では、個人向けセンサボックスのための柔軟なソフトウェアを実現するための個人向けセンサボックス実装フレームワークを提案する。提案フレームワークは、センサデバイス毎に異なる処理を実装する ConcreteSensor レイヤ、センサに共通する処理を抽象的に規定する AbstractSensor レイヤ、複数のセンサを集約しアプリケーションに計測データを提供する SensorBox レイヤの3層から構成される。提案フレームワークによって、ユーザは自分のセンサボックスを柔軟にカスタマイズでき、センサボックスを利用するアプリケーションを容易に開発可能となる。さらに本稿では、提案フレームワークを利用して、9種類の Phidget センサを内蔵した個人向けセンサボックスを試作する。

キーワード 環境センシング, モバイルセンシング, 参加型センシング

A Framework for Personal Sensor Box in Mobile Environmental Sensing

Akihiro OKUSHI[†], Shinsuke MATSUMOTO[†], Sachio SAIKI[†], and Masahide NAKAMURA[†]

[†] Kobe University Rokkoudai-cho 1-1, Nada-ku, Kobe, Hyogo, 657-8501 Japan

E-mail: [†]okushi@ws.cs.kobe-u.ac.jp, ^{††}{shinsuke,masa-n}@cs.kobe-u.ac.jp, ^{†††}sachio@carp.cs.kobe-u.ac.jp

Abstract The mobile environmental sensing, where mobile users measure surrounding information with hand-held sensors, is a powerful means of the participatory sensing. We are studying the *personal sensor box*, by which individual user performs personalized mobile sensing with favorite sensors. This paper presents an application framework for the personal sensor box that enhances the flexibility of the applications towards the variety of sensor configurations. The proposed framework consists of three layers: (1) ConcreteSensor Layer, wrapping sensor devices and implementing device-specific operations, (2) AbstractSensor Layer, abstracting the devices with common sensor operations, (3) SensorBox Layer, aggregating multiple sensors and providing measured data for various applications. The proposed framework allows a user to flexibly customize his sensor box, and to easily implement applications using the customized box. We develop a prototype of a personal sensor box with 9 kinds of Phidgets sensors.

Key words environmental sensing, mobile sensing, participatory sensing

1. はじめに

センシングデバイスの高性能化や小型化, ICT 技術の進歩に伴い, 複数のユーザが計測したデータを共有し有効活用する参加型センシングに関する研究が盛んに行われている [1]~[5]. ユーザが位置する局所的な情報を広域な範囲から収集し共有することで, 従来の据え置き型センサでは実現出来なかった様々な分野への応用が可能となる. 参加型センシングに関して, どのようにセンシングを行うかという観点から考えると, 周辺環

境を自動的にセンシングする方法と, 周辺環境の状況をユーザ自身が解釈しセンシングする方法の二つに大別できる.

前者は, 必要なセンサを内蔵した測定装置 (本稿では「センサボックス」と呼ぶ) を用いて自動的にセンシングを行う方法で, ユーザへの負担は少ない. しかしながら, 多くのアプリケーションでは目的に応じた単一 (あるいは少数の) センサを用いており, 推定できるコンテキストは比較的単純なものに限られる. 後者は「晴れている”, “曇りである” など, ユーザ自身がセンサとなって周辺環境の状況を解釈し共有する. 環境セ

ンサでは推定が難しい複雑なコンテキストを、人知を用いて柔軟に得ることができる。一方で、ユーザが主体となって入力する必要があるため、少なからずユーザへの負担になる。

そこで我々は、ユーザへの負担を抑え、かつ、複雑なコンテキストを推定可能とするためのモバイル環境センシングについて研究を行っている [6]。モバイル環境センシングでは、複数の環境センサを内蔵したセンサボックスを、個々のユーザが持ち運び、周辺環境を自動計測して計測データを共有する。計測データは様々なアプリケーションやサービスから横断的に利用される。自動センシングによりユーザへの負担を抑え、複数種類のセンサを用いることで高度なコンテキスト推定を行うことをねらう。将来的には、ユーザー一人ひとりが自前のセンサボックス（「個人向けセンサボックス」と呼ぶ）を所有し、好きなセンサで自由にモバイル環境センシングを行い、様々な用途やサービスに活用するシナリオを想定している。

個人向けセンサボックスでは、ユーザ個人が好きなセンサを自由に選択してカスタマイズするため、様々なセンサの種類やデバイスの組み合わせを柔軟に受け入れるソフトウェア設計が必要である。しかし従来は、アプリケーションに特化したセンサボックスを開発してきたため、あるアプリのセンサボックスを別のアプリに流用したり、内蔵センサを別のものに組み替えたりすることを、ソフトウェアにおいて想定していない。

そこで本研究では、個人向けセンサボックスのための柔軟なソフトウェアを実現するための個人向けセンサボックス実装フレームワークを提案する。提案フレームワークは大きく3つのレイヤから構成される。センサデバイス毎に異なる処理を実装する ConcreteSensor レイヤ、センサに共通する処理を抽象的に規定する AbstractSensor レイヤ、複数のセンサを集約しアプリケーションに計測データを提供する SensorBox レイヤである。ユーザは自分のセンサボックスの構成を定義ファイルに記述すると、フレームワークは適当な ConcreteSensor オブジェクトを生成して SensorBox に接続する。SensorBox は具体的なデバイスを気にすることなく抽象的な AbstractSensor から値を取得するため、個人によって異なるセンサの構成や組み合わせを柔軟に収容できる。センサボックスを利用するアプリケーションは、具体的なセンサデバイスの仕様や接続方法を意識せずに、SensorBox の API を通して統一的な方法で計測データを取得する。よって、様々なアプリケーションから横断的に計測データを利用可能になる。提案フレームワークによって、ユーザは自身のセンサボックスを柔軟にカスタマイズでき、カスタマイズしたセンサボックスを利用するアプリケーションを容易に開発可能となる。

また本論文では、提案フレームワークの有効性を検証するために、実際のセンサデバイスを用いたセンサボックスを試作した。具体的には Phidgets 社の9種類のセンサを内蔵したセンサボックスを製作し、提案フレームワークを用いてソフトウェア実装を行った。その結果、これらの9種類のセンサデータを記録するデータロガーを容易に実装できた。

2. 準備

2.1 参加型センシング

参加型センシングは、ある共通の目的の下ユーザがセンシングに参加し、計測データを共有して有効活用するセンササービス体系のひとつである。多くのユーザが計測した局所的な情報を広域から収集することにより、従来の据え置き型センサでは実現出来なかった様々な分野への応用が可能となる。具体的な適用分野としては、車に搭載された速度や GPS などのセンサを利用した交通分野での応用 [1], [2] が有名である。他にも、携帯デバイスを利用した食事写真の共有による健康支援 [3] や、車道沿いでの大気質の可視化 [4]、食料品店での商品価格の共有による買い物支援 [5] などが存在する。また SNS サービスを利用した災害情報やイベント情報のリアルタイムな共有も、参加型センシングの一つの形態であるといえる。Yahoo! Japan ではユーザ参加型のリアルタイムな天気共有サービス「みんなで実況！今の天気」[7] を実際に提供しており、今後 IT 産業界や様々な分野での応用、実用化が期待されている。

参加型センシングの実現方法は、ユーザがセンサを内蔵した計測装置（センサボックスと総称する）を持ち歩いて周辺環境を自動的にセンシングする方法と、人間がセンサとなって周辺環境の状況を解釈してセンシングする方法の2つに大別出来る。前者は車載センサや GPS を用いた交通センシングや、二酸化炭素濃度の計測による大気汚染の状況把握等が該当する。この方法は、自動でセンシングすることが可能であるためユーザへの負担が少ないが、計測対象が比較的単純なものに限定される。

後者は周辺の天気をユーザが観測して共有する天気共有サービスや、購入履歴の共有による商品価格の共有等が該当する。このセンシング方法は、目的に応じて入力を設定することが可能なため、複雑なコンテキストを容易に取得することが可能である。一方、参加者自身の入力が必要となるため、参加者に大きな負担が生じるという問題がある。

2.2 モバイル環境センシングと個人向けセンサボックス

我々の研究グループでは、センサボックスに複数の環境センサを組み込んでユーザが持ち運び、周囲の環境をセンサで自動計測するモバイル環境センシングについて研究を行っている。自動センシングによりユーザへの負担を抑えると同時に、複数種類のセンサを用いた高度なコンテキスト推定をねらっており、参加型センシングの強力な実現手段となりうる。我々の先行研究 [6] では、温度センサと湿度センサと気圧センサの組み合わせによって天候の変化を推測したり、音量センサと照度センサの組み合わせによって交通量の推測が可能であると考察している。

また、モバイル環境センシングは、個人行動の精密な振り返りや、ライフログの一手段としても非常に有効であることがわかっている。したがって、将来的には、ユーザー一人ひとりが自前のセンサボックス（個人向けセンサボックスと呼ぶ）を所有し、好きなセンサで自由にモバイル環境センシングを行い、様々な用途やサービスに活用するシナリオを想定している。目的・用途に応じた個人向けセンサボックスの構成方法や、ソフトウェアアーキテクチャについても研究を進めている。

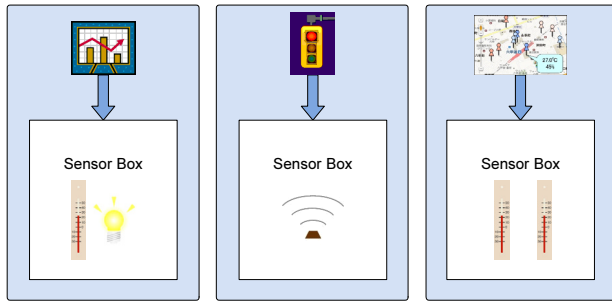


図 1 従来のセンサボックスのアーキテクチャ

2.3 課題

本論文では、特に個人向けセンサボックスのソフトウェアの構成方法に着目する。従来の多くのセンサ・アプリケーションでは、アプリケーションの目的や用途に応じて、専用のセンサボックスを作成する形態がとられる。図 1 にそのアーキテクチャを示す。センサボックスには目的に応じたセンサデバイスが内蔵され、特定のアプリケーションから利用される。アプリケーションとセンサボックス、センサデバイスは互いに密結合している。センサボックスのソフトウェアとしては、あるセンサボックスを別のアプリに流用したり、内蔵センサを別のものに組み替えたりすることを想定していない。

個人向けセンサボックスでは、ユーザ個人が好きなセンサを自由に選択してカスタマイズするため、様々なセンサの種類やデバイスの組み合わせを柔軟に受け入れるソフトウェア設計が必要である。したがって、この従来のアーキテクチャを個人向けセンサボックスのソフトウェアの実現に直接適用することは困難である。

3. 個人向けセンサボックス実装フレームワーク

前節で述べた課題を解決するために、本研究では個人向けセンサボックスのためのアプリケーションフレームワークを提案する。ユーザ毎に異なる多様なセンサ構成に対して柔軟に対応し、様々なアプリケーションが横断的にセンサボックスを活用できるフレームワークを目指す。

3.1 アーキテクチャ

提案フレームワークのアーキテクチャを図 2 に示す。提案フレームワークは次の 3 つのレイヤから構成される。

- **ConcreteSensor レイヤ**: センサデバイスをラップし、デバイスからセンサ値を取得するデバイス依存の処理を実装するレイヤ。
- **AbstractSensor レイヤ**: センサデバイスを抽象化し、センサ値の取得方法を統一するレイヤ。
- **SensorBox レイヤ**: 複数のセンサを集約し、上位のアプリケーションに計測データを提供するレイヤ。

以下の節で、各レイヤの詳細な説明を行う。

3.2 ConcreteSensor レイヤ

ConcreteSensor レイヤでは、センサデバイスから計測値を取得するソフトウェアを実装するレイヤである。センサデバイスの種類ごとにクラスを 1 つ作成し、デバイス毎に異なるセン

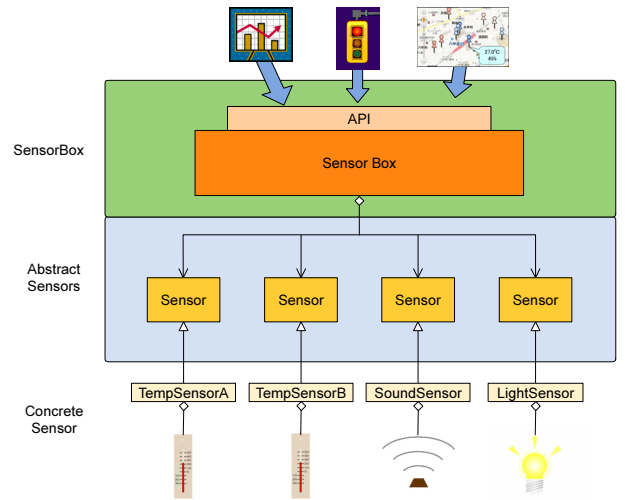


図 2 提案フレームワークのアーキテクチャ

サ値の取得処理を実装する。実装においては、デバイスベンダから提供されるライブラリや SDK を利用することを想定している。デバイスに依存する処理をこのレイヤに閉じ込め、センサボックスから独立させることにより、センサボックスは多様なセンサの組み合わせに対応できることが可能となる。センサボックスに新たなセンサデバイスを組み込む場合、そのデバイスに対応した ConcreteSensor クラスが存在すればそれを利用してオブジェクトを生成する。もし存在しなければ、そのデバイスに対応した ConcreteSensor クラスを新たに実装することになる。開発した ConcreteSensor クラスは、フレームワーク内のクラスライブラリとして管理し、多様なデバイスに対応させる。これにより、提案フレームワークの普及が促進される。

3.3 AbstractSensor レイヤ

AbstractSensor レイヤは、様々な種類のセンサデバイスを抽象的なセンサとして標準化するためのレイヤである。直感的に、全てのセンサは、問い合わせに対して何らかの値を計測して返すオブジェクトとして抽象化することができる。センサボックスは、直接 ConcreteSensor と通信するのではなく、この抽象化されたセンサと通信することで、具体的なセンサの組み合わせに依らず、統一的な方法でセンサ値を取得できるようになる。

また、具体的なセンサデバイスに厳密に対応していなくても、値を返すオブジェクトであれば論理的な AbstractSensor とすることができる。例えば、スマートフォンなど複数のセンサが搭載されたデバイスから、特定のセンサだけを抜き出す ConcreteSensor をつくり、それを抽象化することも可能である。また、時間と共に更新される Web 上の情報や時計なども、抽象化されたセンサとしてセンサボックス実装フレームワークに組み入れることが可能である。

3.4 SensorBox レイヤ

SensorBox レイヤは、個人向けセンサボックスを表現して実装する。1 つの SensorBox は AbstractSensor レイヤを通して 1 つ以上のセンサを所有し、センサの構成情報 (ID, 型, 仕様情報等) を管理する。各 AbstractSensor には計測値取得のための共通インターフェース `getValue()` が定義されており、リ

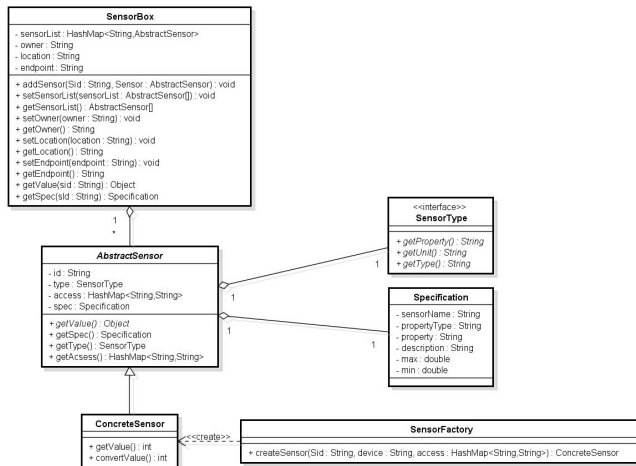


図 3 提案フレームワークのクラス図

クエストを行うと ConcreteSensor で計測されてセンサ値を取得できる。センサ値の計測値やセンサの構成情報は、API を通して、センサボックスを利用するアプリケーションに提供される。アプリケーションははじめに、センサボックス内のセンサ構成情報を取得する。次に、値を取得したいセンサの ID を指定して `getValue(id)` リクエストを行うことで値を取得できる。このとき、アプリケーションは具体的なセンサデバイスの実装や型番を意識することなく、統一的な方法でセンサを利用できる。

さらに、センサボックスのセンサデバイスが別のものに入れ替わったり、種類が増えたりした場合でも、アプリケーションからは同様の手段でセンサにアクセスできる。したがって、センサボックスは特定のアプリケーションやサービスに依らずに、様々なアプリケーションから横断的に利用可能になる。さらに、センサボックスの API をクラウドサービスとして公開することで複数のユーザ間での共有も容易となり、大規模な参加型センシングサービスへの応用も可能となる。

4. 提案フレームワークの設計と実装

本章では、提案フレームワークの具体的な設計と実装について説明する。

4.1 提案フレームワークのクラス図

図 3 に提案フレームワークのクラス図を示す。ConcreteSensor レイヤがクラス図の下段の ConcreteSensor クラスに、AbstractSensor レイヤはクラス図の中段の AbstractSensor クラスに、SensorBox レイヤはクラス図の上段の SensorBox クラスにそれぞれ対応する。

4.1.1 ConcreteSensor クラス

ConcreteSensor クラスは、温度センサや湿度センサ等、フレームワークで対応する具体的なセンサデバイスの製品ごとにクラスを作成する。主要なメソッドとして、センサデバイスから得られたデバイスに依存した計測値を、温度 () や湿度 (%) といった意味のある標準的な値に変換する `convertValue()` や、変換された値を返す `getValue()` が実装される。実装においては、デバイスベンダが提供するライブラリや SDK を利用する

ことを想定している。また、それぞれのセンサデバイスのマニュアルや仕様から、センサの種類 (SensorType) やセンサの仕様 (Specification) の情報を抜き出しておく。

4.1.2 AbstractSensor クラス

AbstractSensor クラスは、あらゆるセンサが備えるべき機能を抽象化するインターフェースを定義する。最も重要なメソッドは `getValue()` であり、これは各 ConcreteSensor で実装すべき抽象メソッドとして定義される。また、各センサオブジェクトの ID やセンサのタイプ (SensorType)、仕様 (Specification) 等の情報を取得するメソッドも実装する。これらの情報は、ConcreteSensor の実装時に定義される。

また、プロパティ `access` は、ConcreteSensor の物理デバイスにアクセスするために必要なパラメータを定義するものであり、キーと値のペアを持つハッシュマップで定義される。これは、センサデバイスとセンサボックスの接続形態に基づいて、センサボックスのユーザが外部から与えるものとする。例えば、センサボックスが PC であり、センサデバイスと USB 接続した場合には、シリアルインタフェースの ID 番号やポート番号などが `access` に渡される。

4.1.3 SensorBox クラス

SensorBox クラスは、収容するセンサ群 (AbstractSensor) をリスト `sensorList` で保持する。このクラスは、センサボックスの ID や所有者、設置場所、API のエンドポイントといったプロパティを持っており、メソッドを介してこれらの情報を取得できる。また、センサの動的な追加を行う `addSensor()` メソッドや、センサ `id` を指定してセンサ値を取得する `getValue(id)` メソッドが実装されている。さらに、センサの仕様やタイプも取得可能である。

SensorBox クラスのメソッドは API として上位のアプリケーションに公開される。Web サービスとしてクラウドにデプロイすることで、インターネットを介したセンサデータの共有が可能となる。これにより、大規模な参加型センシングを実現する手段とすることができる。

4.2 センサオブジェクトの動的生成

個人向けセンサボックスには、どのようなセンサデバイスが接続されるか、あらかじめ知ることが出来ない。したがって、センサボックスにどの ConcreteSensor を収容するかはアプリケーションの実行時に判断して、適当なオブジェクトを生成する必要がある。この仕組みを実現するために、提案フレームワークはセンサオブジェクトを動的に生成する仕組みを実装している。具体的には、ユーザがセンサボックスに接続するセンサデバイスの情報をセンサボックス定義ファイルという外部設定ファイルの形であらかじめ与える。フレームワークはこのファイルを実行時に読み込んで、リフレクション機構を用いて適切なセンサオブジェクトを生成する。

図 4 にセンサボックス定義ファイルの例を示す。この例では、`okushi01` というセンサボックスが定義されている。location 属性から据え置きでなく持ち運びのセンサボックスである。また、2 つのセンサ `sensor01`, `sensor02` が存在し、それぞれ `Phidgets1124` 型温度センサ、`Phidgets1133` 音量センサであること

```

<box>
  <boxId>okushi01</boxId>
  <owner>okushi</owner>
  <location>mobile</location>
  <endpoint>http://BoxSensorService/okushi01</endpoint>
  <sensors>
    <sensor>
      <sensorId>sensor01</sensorId>
      <device>Phidgets1124Temperature</device>
      <sensortype>Temperature</sensortype>
      <accessMethod>
        <interface>PhidgetTextLCD</interface>
        <id>39824</id>
        <port>0</port>
      </accessMethod>
      <samplingRate>1000</samplingRate>
    </sensor>
    <sensor>
      <sensorId>sensor02</sensorId>
      <device>Phidgets1133Sound</device>
      <sensortype>Sound</sensortype>
      <accessMethod>
        <interface>PhidgetTextLCD</interface>
        <id>39824</id>
        <port>1</port>
      </accessMethod>
      <samplingRate>10000</samplingRate>
    </sensor>
  </sensors>
</box>

```

図 4 センサボックス定義ファイル

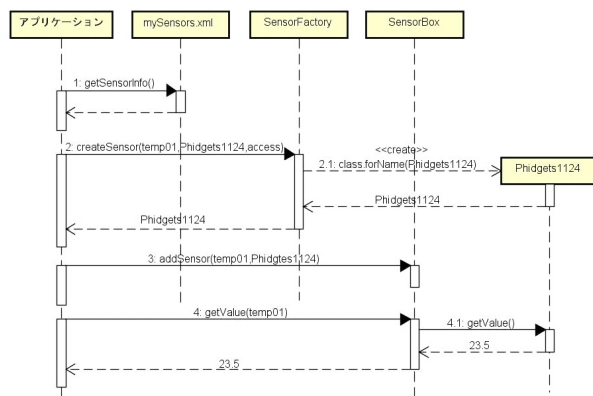


図 5 センサ登録・利用シナリオを表すシーケンス図

が定義されている。sensor01 は、PhidgetTextLCD (シリアル番号 39824) のインターフェースのポート 0 番に接続されている。この情報は、AbstractSensor の access プロパティにセットされる。また、samplingRate はサンプリングレートを msec で表し、1 秒間隔でサンプリングするように定義されている。

センサオブジェクトの動的生成を行うために、アプリケーションは SensorFactory というフレームワーク内のユーティリティクラスを利用する。アプリケーションは、はじめにセンサボックス定義ファイルをオープンし、必要なパラメータを抽出、SensorFactory に渡すことで、適切なセンサオブジェクトが生成される。アプリケーションはこれを SensorBox オブジェクトの sensorList に加える。この作業をセンサの個数分繰り返す。

4.3 センサの登録・利用シナリオ

提案するフレームワークにおいて、センサを登録し、センサ値を取得する例を示す。ここでは、Phidgets 社の Phidget-

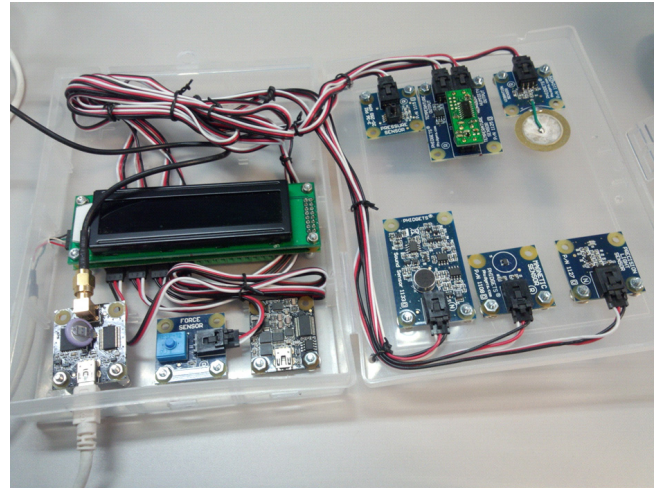


図 6 センサボックスのプロトタイプ

TextLCD (serial Number: 12345) と temperature Sensor (型番: Phidgets1124) を用いる。フレームワークには Phidgets1124 の ConcreteSensor クラスである Phidgets1124.class がすでに作成されているものとする。また、センサボックス定義ファイル mySensor.xml には、Phidgets1124 の接続ポートや id 等の情報が記載されているものとする。このとき、アプリケーションが mySensor.xml に基づいて、Phidgets1124 をセンサボックスに登録し、値を読み込むというシナリオを、図 5 のシーケンス図に示す。

まず、アプリケーションは mySensors.xml を読み込んで、作成するセンサの ID (temp01) やセンサデバイスの型 (Phidgets1124)、接続する Phidgets InterfaceKit のシリアル番号等の情報を取得する。取得した情報を SensorFactory に渡して、センサの作成を指示する。SensorFactory はセンサデバイスの型名を受けて、リフレクション Class.forName() 関数を用いて、Phidget1124.class のオブジェクトを生成する。このオブジェクトに対して、接続情報 access を設定し、アプリケーションに作成したインスタンスを返却する。次にアプリケーションはこのオブジェクトを SensorBox に SensorId と共に追加し、センサの動的生成・登録が完了する。

センサ値を取得する際には、センサの ID である temp01 を引数として SensorBox の getValue("temp01") を呼び出す。SensorBox は temp01 に対応するセンサの getValue() 関数からセンサ値を取得し、アプリケーションへ返却する。

4.4 フレームワークの実装

提案フレームワークは、Java 言語で実装した。有効行数は合計 5,273 行であった。

5. 個人向けセンサボックスの試作

提案フレームワークの有効性を評価するために、実際のセンサデバイスをを用いて、個人向けセンサボックスを試作した。

5.1 ハードウェア

今回作成したプロトタイプでは、Phidgets 社製 [8] の次の 9 種類の環境センサを用いた。

[2013-09-08T13:24:14],GPS,	light,	sound,	magnetic,	force,	gas_pressure,	temperature,	humidity,	vibration
[2013-09-08T13:24:14],[lat=24.403,lon=124.141,alt=11.1],	57,	63.26,	-5,	false,	101.25,	37.33,	43.66,	498.0
[2013-09-08T13:24:15],[lat=24.403,lon=124.141,alt=11.0],	60,	66.44,	-5,	false,	101.25,	37.33,	43.66,	497.0
[2013-09-08T13:24:16],[lat=24.403,lon=124.141,alt=11.0],	52,	63.95,	-5,	false,	101.5,	37.33,	43.66,	496.0
[2013-09-08T13:24:17],[lat=24.402,lon=124.141,alt=11.0],	60,	72.28,	-5,	false,	101.5,	37.33,	43.66,	497.0
[2013-09-08T13:24:18],[lat=24.402,lon=124.141,alt=10.9],	79,	70.53,	-5,	false,	101.5,	37.33,	43.66,	483.0

図 7 試作したセンサボックスで取得したセンサーログ

- 温度センサ (Humidity/Temperature Sensor 1125)
- 湿度センサ (Humidity/Temperature Sensor 1125)
- 気圧センサ (Pressure Sensor 1115)
- 振動センサ (Vibration Sensor 1104)
- 音量センサ (Sound Sensor 1133)
- 照度センサ (Light Sensor 1127)
- 磁気センサ (Magnetic Sensor 1127)
- 圧力センサ (Force Sensor 1106)
- GPS センサ (PhidgetGPS 1040)

これら 9 種類のセンサはアナログセンサであり、インターフェースボードである PhidgetTextLCD 1203 を介して、PC またはスマートフォンに USB で接続して使用する。図 6 に試作したセンサボックスの概観を示す。

5.2 ソフトウェア

作成したセンサボックスを用いたアプリケーションとして、9 種類のセンサから定期的に計測値を取得してログに出力するデータロガーを実装した。提案した個人向けセンサボックス実装フレームワークを利用し、Java 言語で実装した。

まず、上記 9 種類のセンサの ConcreteSensor クラスを、Phidgets が提供する SDK を用いて作成した。次に、各センサデバイスについて、センサを接続したインターフェースボードの ID と接続ポートを調べ、センサボックス定義ファイルに記述した。データロガーのアプリケーション実装では、内部で SensorBox オブジェクトを生成し、各種 API を呼び出すことで、各センサの計測値を取得する。取得したデータをコマ区切りテキストで出力するようにした。今回の実装では、1 秒おきにログを出力するようにした。

作成したセンサボックスとデータロガーを用いて実際に屋外でモバイル環境センシングを行い、9 種類のセンサデータが取得されていることを確認した。図 7 に取得したログの一部を示す。

6. ま と め

本稿では、モバイル環境センシングにおける個人向けセンサボックスのためのアプリケーションフレームワーク「個人向けセンサボックス実装フレームワーク」を提案した。提案フレームワークは、ConcreteSensor レイヤ、AbstractSensor レイヤ、SensorBox レイヤの 3 層から構成される。提案フレームワークでは、具体的なセンサデバイスの処理を抽象化し SensorBox と疎結合することによって、センサの追加や変更を容易にしている。これによって、ユーザ毎に異なるセンサの構成に柔軟に対応することが可能となる。提案フレームワークを Java で実装し、実際のセンサを用いたセンサボックスとデータロガーを試作した。さらに、実際にモバイル環境センシングを行い、センサーログを取得できたことを確認した。

今後の課題として、フレームワーク内で未実装の ConcreteSensor クラスの実装容易化や、センサボックスを用いた付加価値サービスのための API 開発が挙げられる。

謝 辞

この研究の一部は、科学技術研究費 (基盤研究 C 24500079, 基盤研究 B 23300009), および、積水ハウスの研究助成を受けて行われている。

文 献

- [1] R.K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T.F. Abdelzaher, “Greengps: A participatory sensing fuel-efficient maps application,” Proceedings of the 8th international conference on Mobile systems, applications, and services, pp.151–164, 2010.
- [2] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A.M. Bayen, M. Annavaram, and Q. Jacobson, “Virtual trip lines for distributed privacy-preserving traffic monitoring,” Proceedings of the 6th international conference on Mobile systems, applications, and services, pp.15–28, 2008.
- [3] S. Reddy, A. Parker, J. Hyman, J. Burke, D. Estrin, and M. Hansen, “Image browsing, processing, and clustering for participatory sensing: Lessons from a dietsense prototype,” Proceedings of the 4th workshop on Embedded networked sensors, pp.13–17, 2007.
- [4] P. Dutta, P.M. Aoki, N. Kumar, A. Mainwaring, C. Myers, W. Willett, and A. Woodruff, “Common sense: Participatory urban sensing using a network of handheld air quality monitors,” Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, pp.349–350, 2009.
- [5] L. Deng and L.P. Cox, “Livecompare: Grocery bargain hunting through participatory sensing,” Proceedings of the 10th workshop on Mobile Computing Systems and Applications, pp.4:1–4:6, 2009.
- [6] 大槲章裕, 徳永清輝, 裕本真佑, 中村匡秀, “個人向けモバイル環境センシングを活用した付加価値サービスの検討,” 電子情報通信学会技術研究報告, vol.112, no.307, pp.1–6, Nov. 2012.
- [7] YAHOO! JAPAN, “みんなで実況! 今の天気”. <http://weather.yahoo.co.jp/weather/promo/live/>.
- [8] Phidgets Inc., “Unique and easy to use USB interfaces”. <http://www.phidgets.com/>.