

センサネットワーク用ユーザインタフェースのメタデータによる動的生成

小倉正利^{†1} 峰野博史^{†4} 小佐野智之^{†3}
石川憲洋^{†3} 水野忠則^{†2}

近年様々な種類のセンサが開発され、身近な場所にセンサが普及し始めている。センサの検知情報は単体でも有用なデータを提供してくれるが、複数のセンサのデータの分析により、大きな価値を持つ情報となる。本論文ではセンサデバイスが提供するサービスをメタデータとして記述してセンサGWに持たせることでユーザ端末が様々なサービスを利用可能となるユーザインタフェースを開発した。また、センサGWに置くメタデータを作成するための補助ツールも開発した。これらによってユーザ端末でセンサデバイスを利用するためのインタフェースを動的に生成できるようになり、様々な複合サービスを設定可能な環境を構築できるようになった。

Automatic generation of User Interface from Meta-data for sensor network

MASATOSHI OGURA ^{†1}, HIROSHI MINENO^{†4}, TOMOYUKI OSANO^{†3}
, NORIHIRO ISHIKAWA^{†3} and TADANORI MIZUNO^{†2}

Recently the sensor of various types is developed, the sensor has started spreading in the familiar place. Information of inspection of the sensor offers the useful data even as a single unit, but, it becomes the information which has big value due to the analysis of the data of the plural sensors. In this research The service that the sensor device provides is described as meta data and it gives it to sensor GW. User Interface for which the user terminal was able to use various services was developed. A supplementary tool to make the meta data put on sensor GW was developed. It came to be dynamically generable in the user terminal from these. the interface to use the And we got possible to build the environment that could set various compound service

1. 研究背景

今日のセンサ技術は急速に進化し、自動ドアのセンサから健康状態を測るためのセンサなど多くの種類のセンサ機器が誕生している。近年のセンサ機器は通信機能を有し、ネットワークに接続して情報を共有することが可能となっている。ネットワークに接続されるセンサ機器は複数のセンサを搭載することもあり、センサネットワークシステムでは様々なセンサを搭載したセンサノードやアクチュエータノードが存在する。

本研究では以上のような様々なセンサを搭載したセンサノードやアクチュエータノードが存在する。本研究では以上のような様々なセンサを搭載するノードをセンサデバイスと呼ぶこととする。

物事の状態を捉える際、例えば動物の生態であればカメラなどの機器によって情報を視覚的に捉えることが可能であるが、体内外に取り付けられた生態センサデバイスは動物の身体の内部の変化まで詳細に捉えることができる。センサデバイスから得られるデータは単体でも情報を与えてくれるが、複数台のまとまったセンサ情報や他の種類のセンサ情報と組み合わせることでより有用な情報を得ることができる。生態センサの捉えたデータは他のセンサデバイスの捉えたデータと組み合わせ、分析することによって生命維持に関わる情報を得ることができ¹⁾、今後更に多種多様な分野で利用されていくと考えられている。

このようなセンサデバイスから得られる有用な情報を監視するためのモニタリングツールがいくつか開発されている。しかし組み合わせることでより

^{†1} 静岡大学大学院情報学研究所
Graduate School of Informatics, Shizuoka University
^{†2} 静岡大学創造科学技術大学院
Graduate School of Science and Technology, Shizuoka University
^{†3} (株)NTTドコモ
NTT DOCOMO, Inc.
^{†4} 静岡大学情報学部
Faculty of Informatics, Shizuoka University

詳細な情報が得られるセンサの利点を生かすためにはセンサデバイスの設定や複数のセンサデバイスを連携させたり複合イベントを設定するための GUI があると便利である。

本論文では複数の異なる種類のセンサデバイスの組み合わせを利用してするためにメタデータを用いて動的にユーザインタフェースを生成する仕組みを開発した。

2. 関連研究

2.1 PUCG

PUCG(P2P Universal Computing Consortium) は P2P ネットワークを利用して様々な機器を相互接続・運用する技術開発を目的として活動している標準化団体である。P2P オーバレイネットワークを用いることで異種ネットワーク環境上に分散している様々なデバイスをシームレスに接続し、移動端末を用いて様々なアプリケーションを実行することができる²⁾。

しかし、超小型センサノードは処理能力が低く、P2P プラットフォームの実装が困難であるため PUCG ではゲートウェイ (センサ GW) を用いて問題解決に取り組んでいる。センサ GW は、PUCG ネットワークとセンサネットワークにそれぞれ接続し、仮想ノードという形式で他のセンサノードとの通信を可能にしている。

このセンサ GW を用いることによって携帯電話などの身近な通信機器からセンサデバイスの制御を行うことができ、様々なサービスが利用可能になる。ここでセンサ GW を用いてイベントを生成するには、配下のセンサデバイスが提供するサービスを記述したメタデータを設定する。その際、どのセンサデバイスからの情報を取捨選択するのか、あるいは統合しながら収集していくのかなど利用者が柔軟に設定できるような情報を含めておく必要がある。

2.2 センサゲートウェイ (センサ GW)

複数の種類のセンサデバイスの情報を組み合わせて利用するためにセンサ GW を用いる。システムのイメージを図 (1) に示す。PUCG では各センサデバイスから送られてくる検知データは特定のデータベースに格納され、PUCG プロトコルを利用してクライアントと通信を行う。その際に仲介を行うのがセンサ GW である。センサ GW は自身の管理するセンサデバイスの情報を記述したメタデータを保持しており、クライアントはセンサ GW の提供するサービスを利用す

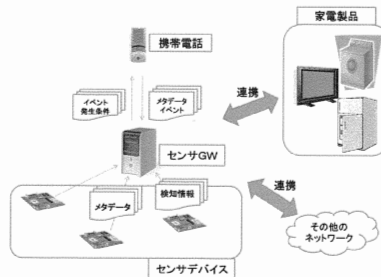


図 1 センサ GW 図

るためにメタデータが必要となる。本論文ではこのメタデータに記述されたセンサデバイスの情報を利用して GUI を作成する。

2.3 Nanomon

複数種類のセンサデバイスのモニタリングツールとしては Nanomon という研究がある³⁾。Nanomon では複数種類のセンサデバイスを同一のサーバのデータベース上で管理し、得られた情報を端末の種類にこだわること無く提供するためのツールである。表示方法には記述方式が統一された専用のデータをモニタリングを行うクライアント側に送り、あらかじめ用意されたパーツなどの組み合わせで表示を行う方法を取っている。データに記述された位置情報などから部屋内部の位置情報などを表示することも可能である。

その他センサのモニタリングツールにはセンサ開発元が提供する MOTEVIEW などがある。

これらは高度な GUI によってセンサデバイスの情報をモニタリングすることが可能であるが、設定ファイルなどを必要としており、ユーザは設定ファイルを取得せねばならず、GUI 開発側も設定ファイルを作成しなければならない。

3. メタデータを用いた GUI 生成

GUI の生成の基本となる情報として PUCG プロトコルで利用されるメタデータを使用する。メタデータは利用可能なセンサの詳細なデータであり、XML ベースで記述されている。

3.1 メタデータ

家電製品を個々のデバイスの種類として認識する場合には冷蔵庫やエアコンなど固有の名称を定義、デバイスの提供する機能は冷凍、録画、再生などがあり、サービスとして個別に定義する。これ等の情報を得る

ことでデバイス毎に特定のサービスを選択することが可能となる。

センサについても同様に、センサデバイスが取得したデータを基にセンサ GW を利用するためには、各センサデバイスの情報と送られてくるデータについての情報が必須となる。PUCC のセンサ GW ではセンサの詳細とサービスについて記述したメタデータを利用して特定センサデバイス毎に特定のデータを要求することができる。

図 (2) にメタデータの書式の一例を示す。

センサ GW を利用するクライアントはメタデータを受け取り、そこに記述された機能に基づいてセンシングされたデータの取得、あるいはデータの値に基づいたイベントの設定を行う。メタデータは xml のフォーマットを用いて、大きく分けて 2 つのデータが記述される。

まず一つ目はセンサネットワーク情報で、

- センサデバイスの識別情報
- センサノード情報
- 搭載されているセンサの種類・仕様

が記述される。

二つ目はセンサ GW 側が提供可能なサービス情報で、

- センサが測定した値を示すローデータ
- イベントデータフォーマット
- イベントデータリスト

が記述される。

イベントデータリストとはセンサデバイスから送られてくるデータから状態変数を基に判断されるイベントについてまとめたモノである。メタデータにはデバイス名とデバイスが利用可能なサービスの名前が記述されており、サービスの内容を知ることが可能となる。

機器をまとめて一括制御する場合、機器の一括制御のための仮想的なデバイスのみがデバイスのメタデータを持つ。メタデータには個々の機器が持っていたメタデータと同様に仮想的なデバイスを一つの機器と見なし内容を記述する。その際状態変数の定義については、仮想的なデバイスが提供するサービスに関連する状態変数について定義し、機器が持つ状態変数と 2 重管理になるものは記述しない。このモデルでプリミティブデバイスとして記述される内容は個々の機器が持っていたデバイス用のメタデータとなる。

センサデバイスが各所に設置されることによってサービスの幅も広がり、アプリケーションも増加すると考えられる。各センサ毎に個別の GUI を作成して配布する場合、ユーザ側では利用時にダウンロードに手間がかかりサービス開始までに時間がかかってしまう

というデメリットがある。高度なサービスで日常的に何度も利用するアプリケーションであれば、GUI プログラムをダウンロードしてメモリに保存しておく方が利用しやすい。

しかし、単純なサービスで場所毎に異なるアプリケーションの場合には、余分なメモリを消費することになり、その数が増えて行くに連れてプログラムを選択する煩わしさがある。そこでメタデータに記述されたデータを基本情報としてイベント制御用の GUI を作成することで、余分なデータ通信を省くことが可能となる。GUI の自動生成にメタデータを利用するメリットはメタデータが XML で記述されているという点である。XML は多種のツールとの連携が可能であり、OS に依存しないだけでなく、プログラム言語にも依存しないため幅広い利用用途が有り、大きな拡張を行っても容易に対応させやすいという点がある。

3.2 XML を利用した GUI 生成

XML は内部データ SAX や DOM といった XML 解析用パーサを利用することで JAVA プログラム内で読み込んでデータとして利用することが可能である。SAX は XML ファイルを最初から順番に読み込んでいき、タグと記述されている内容を別々にイベントとして通知することが可能である。JAVA との親和性も高く、読み込み速度も DOM に比べて高速である。

それに対して DOM は XML の構造を木構造で格納し、必要なデータを取り出すことが可能である。複雑な構造を持つデータを読み込むのに適している。XML を利用して GUI を生成する技術は現在までに多く研究されている⁴⁾。XML を利用した GUI の開発は XUL,gui4j などで行われている。いずれも GUI 作成用の部品となるデータが書かれた xml ファイルを読み込み、解析した結果から GUI を自動生成するツールである。

Nanomon も XML を利用した GUI 作成ツールにも GUI を表示させるための専用のファイルが必要となる。それぞれ高度な GUI を作成可能であるが、センサデバイスの設定を行うだけの簡易な GUI を作成する場合、不必要に多くのファイルを通信で取得することが必要になってしまうことがある。

そのため本論文では低負荷である SAX パーサによるメタデータの解析を行い、GUI 専用のファイル無しでも GUI が生成できるツールを作成する。

3.3 イベント

センサデバイスがある指定された状態 (イベント)

```

<Device type="URI representing device type" id="デバイスID" name="ユーザー名">
<Specification>
<URLBase-base-URI for all relative URLs</URLBase>
<Manufacturer-manufacturer name</Manufacturer>
...
</Specification>
<StateVariableList>
...
<StateVariable name="ユーザー名" datatype="データの型" sendEvents="yes or no">
<DefaultValue>初期値</DefaultValue>
<AllowedValueRange>
<Minimum>最小値</Minimum>
<Maximum>最大値</Maximum>
<Step>主分割</Step>
</AllowedValueRange>
</StateVariable>
</StateVariableList>
<ServiceList>
<Service name="ユーザー名" type="URI service type"/>
</ServiceList>
<PrimitiveDeviceList>
<PrimitiveDevice type="URI device type" id="プリミティブデバイスID" name="ユーザー名">
Description of primitive devices (if any) go here
</PrimitiveDevice>
</PrimitiveDeviceList>
<EventConditionList>
<EventCondition>
<EventCondition id="生成条件ID">
<ConditionExpression format="条件式の種別" period="生成間隔"生成条件式</ConditionExpression>
<NotificationVariableList>
<StateVariable name="通知用状態変数のname属性値"/>
</NotificationVariableList>
</EventCondition>
...
</EventConditionList>
</Device>

```

図 2 メタデータ書式

を満たした場合、センサ GW 側でアクションを発生させるシステムがある。イベント例としては加速度センサデバイスに閾値を設定してその数値を超えた場合に警告を発生させるなどがある。

イベントが発生した場合センサ GW はクライアントに対してイベントが発生したことを通知する、あるいはイベントと共に設定していたサービスを起動させることができる。これはセンサを利用して家電機器などを自動制御等に利用される重要なシステムである⁵⁾。

あらかじめメタデータにイベントの生成条件を記述する場合には、各デバイスの持つ仕様を把握しておくことで、デバイスの持つ状態や得られるデータを知っておくことでデータの管理を容易にすることができる。この方式では、複数の状態変化を利用してイベントが定義されているサービスに有効である。例えば、人が外出する際に玄関を家の中から開けるといった情報をセンサによって検知し、さらに門から外に出ることで家の施錠を確認するようなサービスを想定する。このサービスの場合、各センサから与えられるデータも状態変数に限られ、イベントの発生条件は滅多に変更することは無いため、ユーザが独自にイベント条件を変更する必要性は無い。

しかし、室内外の気温や天気などの情報から内部の空調を調整するようなサービスの場合、温度センサから送られて来る温度のローデータをイベント発生条件に含み、ユーザ側が温度の閾値を変更することでイベント発生条件を変化させることができなければ使い勝手が良い。

3.3.1 条件式

イベント条件式には以下が利用される。

表 1 状態変数とイベント生成用に設定された値との比較

| 種類 | 例 | 使用する値 |
|---------------|---------------|---------|
| 大なり, 小なり | >, < | 数値 |
| 以下, 以上 | >=, <= | 数値 |
| 相当, 不当 | =, ≠ | 数値, 文字列 |
| 算術演算子 | +, -, *, /, % | 数値 |
| 括弧による演算順位 | () | 文字列 |
| 時刻指定 | 時, 分, 秒等 | 数値, 文字列 |
| 周期指定 | 時間間隔 | 数値, 文字列 |
| AND 条件, OR 条件 | &, | 文字列 |

これらの条件式を用いて、センサデバイスから送られてくる数値データ、文字列を比較し、条件式の結果が真であればイベントを生成し、偽であればイベントは生成しない。センサデバイスの制御において、文字列の部分一致、シフト演算、ビット演算を行う必用は無いと考えられ、イベント条件式からは省く。

3.3.2 イベントリスト

状態変数の変化によってイベントを発生させた場合、状態変数の定義がそのままイベント発生条件となり、複数の状態変数を組み合わせによるイベントの定義や、あるイベントの出力を別のイベントにするといったことが困難となる。イベント発生条件を独立させて定義することによって、状態変数の組み合わせ等によるイベントの定義が可能となる。

3.4 GUI の作成

動的 GUI 作成方法としてはまず JAVA プログラムの SAX パーサを用いて解析を行い、タグ毎にメタデータの内容を分類する。分類されたデータにはセンサが検知できる情報の種類や値の範囲などが含まれているため、使用するサービスと設定する値として利用する。センサ GW の持つ各種サービスを利用するためにサービスメタデータから URI を読み込み、設定したイベントと連携させるメッセージを生成し送信するモジュールを持つ。サービスの選択は各サービスの名前が定義されているため、その情報を読み込みラベルとして提示する。以上のプロセスを用いてイベント設定用の動的 GUI を作成させる。本論文のプロトタ

IPではデバイス選択部、イベント設定部、サービス選択部、イベント設定部の4項目を実装した。

メタデータの仕様として定義すべきセンサが増加するのに従って設定可能なセンサの種類を判別し直す必要がある。

4. メタデータ作成機能

4.1 メタデータの利用

PUCC の利用するセンサデバイス用メタデータにはセンサデバイスの基本情報を記述した必須項目以外にセンサ GW の管理者がオプションで付け加えることのできる項目も多い。オプションとして記述される項目には位置情報やイベント設定の細かさなどがあり、センサデバイスがどのように設置され、どのような情報を送るものなのかをより詳細に知るための項目である。センサデバイス自身の状態を知ることでセンサデバイスに対してより正確な設定をかけることが出来るため、サービスの正確性と位置情報自体を利用した新しいサービスを提供することが可能となる。例としては火災発生時の正確な火元の検知、センサによって読みとられた物品の移動経路探索などが挙げられる。

センサ GW には複数のセンサデバイスを管理している場合が多く、複数種類のセンサデバイスのデータをセンサ GW 毎のメタデータに記述しなければならない。センサ GW には他の GW と通信することも可能であり、他のセンサ GW のメタデータも同時に保持することになる。その際に複数のメタデータを統合させることもメタデータ作成機能で行われる。

4.2 利用アプリケーション

センサデバイスからは多種多様なサービスが構成されると考えられている¹⁾。

まず、基本的なサービスとして、センシングした情報を利用ユーザに直接提供するアプリケーションがある。これは実際に表面温度や音の大きさなどセンサが検知したデータを数値、あるいはその場の状態を示したデータで表示する。温度センサ、加速度センサなど一般的なセンサのほとんどで利用が可能となる。

次に、複数のセンサが連携することで情報を提供するアプリケーションがある。これはユーザの設定した閾値に対し複数のセンサが達した場合にユーザに通知、もしくは何らかのイベントの発生を行う。建物の警備センサなど複数のセンサの反応を見ることによって場所などより詳細なデータを得ることが出来るよう

| Name | Specification | Status |
|--------------------|---|--------|
| URL_Base | URL Base Base URL for all devices (URL) * | 必須 |
| Manufacturer | *Manufacturer name (Manufacturer) * | オプション |
| ManufacturerURL | *Manufacturer URL (URL) * | オプション |
| ManufacturerDate | *Manufacturer Date (Date) * | オプション |
| ManufacturerOption | *Manufacturer Option (Option) * | オプション |
| ModelBase | *Model Base (Model) * | オプション |
| ModelNumber | *Model Number (Model) * | オプション |
| ModelURL | *Model URL (URL) * | オプション |
| SerialNumber | *Serial Number (Serial) * | オプション |
| UDN | UDN (UDN) * | オプション |
| UPC | UPC (UPC) * | オプション |
| Vendor | Vendor (Vendor) * | オプション |
| Item | Item (Item) * | オプション |
| Model | Model (Model) * | 必須 |

図 3 メタデータ作成ツール

なサービスに用いられる。

また、家電などの機器と連携することにセンサデータをトリガーとして家電機器を動作させることが可能になるサービスがある。ドアに近づくことで電気が点くようなサービスを自分で構築することが可能となる。このセンサデバイスからの情報を連携させるシステムがサービスを多様化させる。

5. プロトタイプ

プロトタイプではイベント設定とサービス設定を行うための GUI を動的に生成するシステムを作成した。環境は WindowsXP, intel ATOM プロセッサ N270, メモリ 1G バイトの小型ノート PC を使用した。将来の携帯電話などのモバイル機器の性能を想定して現在普及が進んでいる小型ノート PC でプロトタイプを利用した。

5.1 イベント設定部

今回メタデータで利用する情報はセンサデバイス ID と状態変数、アイコン、単位、最大値、最小値、幅の情報を利用する。イベント設定ではメタデータからセンサ GW の保有するセンサを表示させ、使用するセンサを選択する。各センサの情報を基にイベントの条件式と閾値を指定し、イベント ID に振ってイベントの登録を行う。イベント設定では使用するセンサを複数選択することが可能で有り、それぞれのセンサに設定された条件を複合したイベントを設定することができる。

5.2 サービス設定部

サービス設定部ではセンサ GW 及び他の GW で利用可能なサービスの設定を行う。サービス用のメタデータに記述されたサービス名から利用するサービスの選択を行う。選択されたサービスと事前に登録しておいたイベントを関連付けることによってイベントと

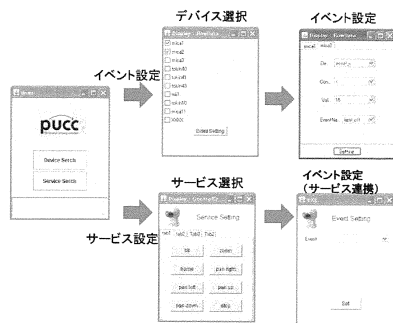


図 4 プロトタイプ GUI 図

サービスの連携が可能となる。プロトタイプを図 (4) に示す。

表 2 PC 性能別 GUI 生成時間の比較

| | PC1 | PC2 | PC3 |
|--------|----------|-----------|------------|
| CPU | Core2Duo | ATOM N270 | Turion x64 |
| クロック数 | 2.66GHz | ATOM | Turion |
| メモリ | 2G バイト | 1G バイト | 1G バイト |
| GUI 起動 | 220 ミリ秒 | 768 ミリ秒 | 820 ミリ秒 |
| ページ生成 | 42 ミリ秒 | 89 ミリ秒 | 55 ミリ秒 |

6. まとめ

本研究では異種センサネットワークを P2P を利用して統合的に管理するシステムを利用するための GUI について検討し、プロトタイプの作成を行った。本論文の GUI からはセンサ GW の保持するメタデータから JAVA プログラムで動的に生成され、メタデータを解析した情報から、センサデバイスに対してイベントを設定することが可能である。イベント設定を利用することによってセンサデバイスの提供する情報と家電機器など他のサービスの連携についての設定も行えるようになった。センサデバイスの効果的な利用についても検討し、メタデータをより詳細に記述するためのメタデータ作成ツールを作成した。

6.1 今後の課題

今後はセンサを利用したサービスの発展に伴い Web 上で利用される GoogleMap などのサービスとの連携するための機能を取りこんで行く予定である。また、センサ機器を新たに設置や排除した場合にメタデータに自動的に反映させるシステムを組み込むことで、センサネットワークをより扱いやすいモノへと変化させていくことができると考えられる。

参考文献

- 1) Magdalena Balazinska, Amol Deshpande, Michael J Franklin, Phillip B Gibbons, Jim Gray, Suman Nath, Mark Hansen, Michael Liebholt, Alexander Szalay, Vincent Tao, "Data Management in the Worldwide Sensor Web," IEEE Conference Society Pervasive computing, pp.30-40, April-june. 2007.
- 2) 石川憲洋, 角野宏光, 加藤剛志, "移動端末とセンサネットワーク連携サービスの実現にむけた研究開発," NTT DoCoMo テクニカルジャーナル, No 3, pp44-50, Oct. 2006.
- 3) Misun Yu, Heayong Kim, Pyeongsoo Mah, "NanoMon: An Adaptable Sensor Network Monitoring Software," IEEE International Symposium on Consumer Electronics2007(ISCE 2007), pp1-6 June. 2007
- 4) Marrthias Book, Volker Gruhn, Gerald Mucke, "An Instant Message-Driven User Interface Framework for Thin Client Applications," 21st IEEE International Conference on Automated Software Engineering (ASE'06), pp257-260, Sep. 2006
- 5) 小熊寿, 永田智弘, 山崎憲一, "柔軟なユビキタスサービス提供のためのセンサ情報処理方式," NTT DoCoMo テクニカルジャーナル, No 3, pp51-54, Oct. 2006.