

センサゲートウェイに用いるメタデータに関する検討

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。しかし、各センサが独自の仕様で動作しているために、統合して利用することが困難である。そこで本研究ではセンサGWをユーザが直接操作する端末との間に介して種類の異なるセンサデバイスを連携させた動作を実現させるために、センサデバイスが利用可能なサービスについて記述されるメタデータに関して考察すると共に、センサの普及によって増加するアプリケーションに対して、携帯電話でもスムーズに利用可能なように、GUIを動的に作成することシステムについて検討を行う。

Examination concerning Metadata used for SensorGateway

MASATOSHI OGURA,^{†1} HIROSHI MINENO,^{†4} TOMOYUKI ONOSE,^{†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 minding sensor GW with the terminals which the user operates directly it makes the operation actualize which cooperates the sensor device where type differs. As you consider in regard to the meta data which is described sensor device concerning the service which is available, in order smoothly to be available even by the portable telephone vis-a-vis the application which increases with the spread of the sensor, it examines concerning the thing system which draws up GUI dynamically.

1. 研究背景

今日のセンサ技術は急速に進化し、非常に多くの種類のセンサと共に多種のセンサネットワークが誕生している。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は動物の身体の内部的変化まで詳細に捉えることができる。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²⁾、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。

センサーネットワークは多くのデータを得ることができるが、単体で利用するよりも複数を組み合わせの方がより多くの情報を得ることが可能となる。

しかし現在のセンサネットワークの管理やサービスアプリケーションなどの研究は基礎的なものがほとんどである。現在開発されている多くのセンサは、通信性能や省電力化に応じて様々なセンサに最適化された独自の制御方式を採用しているため、異なるセンサネットワーク同士では通信が困難である。

そのため分散し、異なったセンサーネットワークを統合して管理するシステムについて盛んに研究が行われている。センサネットワークを統合して管理し、利用して行くためにはセンサネットワークを構成している個々のセンサデバイスに何が出来るのかを、利用する側で把握しなければならない。

さらにセンサデバイスの統合した際に各所に設置された複数のセンサデバイスを利用するためには専用のアプリケーション用インタフェースを使用する必要

^{†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} NTT ドコモ
NTT Docomo, inc
^{†4} 静岡大学情報学部
Shizuoka University Faculty of Informatics

がある。しかし、センサデバイスが変わる毎にインタフェースをダウンロードしていたのではサービスを利用するまでに時間がかかるという問題がある。

センサデバイスの情報を記述したメタデータからはセンサの可能なサービスを知ることが可能となる。そのため本研究ではセンサネットワークを統合管理する上で基礎となるセンサデバイスのサービス利用に関するメタデータと、そのメタデータを基にしてデバイスと目的によって動的に変化するG U Iについての検討を行う。

2. 関連研究

2.1 P U C C

P U C C は P 2 P 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研究が行っている標準化団体である。

P 2 P 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。しかし、超小型センサノードは処理能力が低く、P 2 P プラットフォームの実装が困難であるため P U C C ではセンサプロキシを用いて問題解決に取り組んでいる。

センサプロキシは、超小型のセンサノードとセンサネットワークにそれぞれ接続する通信インタフェースを持ち、仮想ノードを用いて他のセンサノードとの通信が可能となっている。このセンサプロキシを搭載したセンサG W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と考えられている。

実際にセンサプロキシを用いてイベントを生成するには、各センサデバイスが可能なサービスを記述したメタデータの定義を行う。その際、どのセンサデバイスから情報を取捨選択、あるいは統合ながら収集していくか適切に選択可能なようにセンサデバイスの種類を分類する必要がある。

2.2 G U I 生成

現在までにXMLを利用したG U Iの研究はいくつも行われており、S A XやD O Mといったパーサを利用することでXMLファイルをJ A V Aプログラム内で読み込んで、反映させることはすでに可能である。S A X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。J A V Aとの親和性も高く、単純なG U Iを作成する際の読み込み方式に

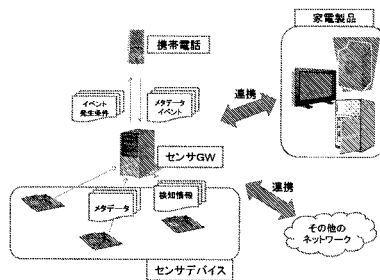


図1 センサGW図

適している。D O M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となる。複雑な構造を持つG U Iを作成する場合に適している。また、gui4jのようにウィンドウ生成時にG U Iの部品となるデータが書かれたxmlファイルを読み込み、G U Iを自動生成するツールも存在する。これらのツールを利用することによって目的に応じたG U I変化させることが可能となる。

3. メタデータ

G U Iを自動生成するためには基準となる情報が必要となる。メタデータにはセンサの可能なサービス内容が記述されているため、基準とするのに適している。

家電製品を個々のサービスとして認識するためにはサービス内容を、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、デバイスの提供する機能を冷凍、録画、再生など定義する。これ等の情報を得ることで必要なサービスを選択して制御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プロキシを利用してイベントを生成するためには、各センサデバイスのデータの内容が必要となる。本稿ではセンサデバイスを仕様、機能で分類することによって、探索や制御時に必要な分類を行った。

センサプロキシの動作についてはP U C Cの通信方式を利用し、提供するサービスの情報を記述したメタデータを利用する。

各センサデバイスではデバイスの情報が記載されたメタデータを保持しており、メタデータを基に適切なデバイスが選択される。そしてメタデータ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、あるいはセンサデバイスの制御を行う。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、

- センサデバイスの数
- 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。

二つ目はセンサプロキシが提供するサービス情報で、

- 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複数のローデータや状態変数を基に判断されるイベントを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。

3.1 データ形式

反応を検知したセンサデバイスが送ってくるデータの形式によってセンサを大きく二つに分類する。

一つ目はモノが接触しているかどうかを判断するマグネットセンサのように、状態変数をデータとして送信するタイプのセンサデバイスが分類される。状態変数が変化することでなんらかのイベントが発生するサービスを提供することになる。

もう一つは温度センサや距離センサなど、データを数値として送信するタイプのセンサデバイスが分類される。データを数値として得られるタイプのセンサデバイスの場合、そのまま数値をサービスとして提供することができ、データに閾値を設けることによって状態変数を細かく区切ることが可能となる。状態変数を細かく区切りトリガーとすることによってイベント連携を図ることも容易になる。

データ形式で分類する際のメリットはイベント定義を行う際に、閾値による柔軟な設定が可能かどうか知ることが可能となる。

デメリットは一つのデータとしては分類数が少ないという点がある。また、加速度センサなどは、普段の値は0で値が変化することで状態が変化したとみなすことができる。センサデバイスが起動している、いないなどの状態以外に、データを数値として扱うタイプのデバイスでも閾値の設定以外に状態変数を定義できるものが多い図(2)。

この分類は他の分類と組み合わせることで、提供できるサービスをより明確にする要素として利用する。

3.1.1 条件式

現在のセンサプロキシではデバイスの生成するイベントは、デバイスの管理する状態変数の値が変化したことを契機に生成される。しかし、センサデバイスは

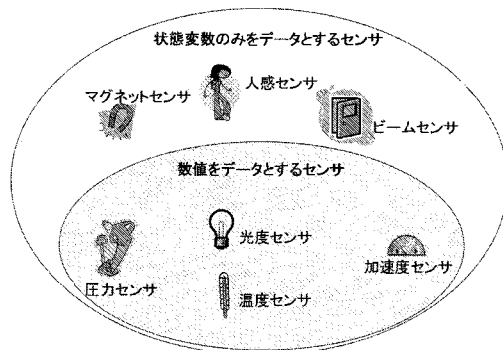


図2 センサの分類

環境の変化により状態変数が頻繁に変化してしまうため、状態変数の値が変化したことだけをイベント生成条件とすると、非常に多くのイベントが発生することになり、一つのセンサデバイスの状態変数が変化するという条件だけでは細かい制御ができないため、高度なイベント生成条件を設定するには不向きである。今後条件式による定義によってイベントの発生条件を向上させていく必要がある。

3.2 測定目的別分類

センサには測定する様々な値があるが、同じ温度センサにでも測定方法に違いがあり、測定対象も違う。明確にセンサを定義するためには、何を測定するかを記述しておく必要がある。

温度センサや加速度センサなど測定する内容が定義されているセンサデバイスもあるが、マグネットセンサなど、磁力よりも取り付けられているモノが重要な場合がある。例えば窓に取り付けられていた場合、窓に付いているという情報を加えることで、戸締りを監視するセンサデバイスとして認識が可能である。

この分類のデメリットは、センサに対する用途を限定して記述すると、他の用途で利用したい時に検出されない可能性がある。そのため、目的、程度、場所など分類を分けて記述することで適切なセンサデバイスを選択可能となる。

3.3 利用目的別分類

種類別の分類の分け方は多いが、今回はサービス提供を前提としており、使用用途による分類を考える。ヘルスセンサ、警備用センサ等、目的に応じたセンサデバイスの自動検出が行われるようにセンサデバイスを大きな使用目的に分類する図(3)。

この分類のメリットは目的に応じた一括制御が簡単になることである。デメリットはセンサ側で使用目的が決まっているセンサが少なく、初期段階で分類する

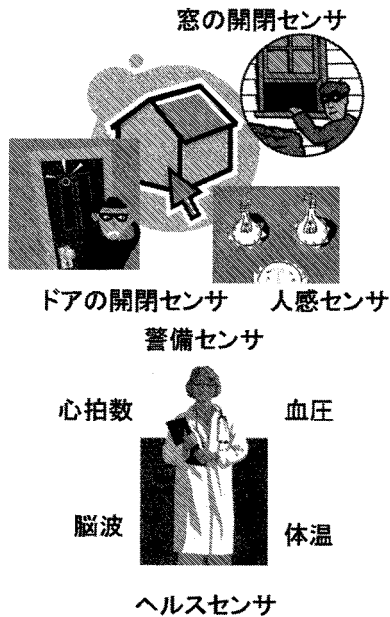


図3 センサの分類2

には曖昧な点である。

4. GUIの動的生成

4.1 メタデータの利用

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。高度なサービスでなおかつ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、単純なサービスでその場限りでしか利用しない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出てくる。

ユーザに利用しやすいGUIに関する研究は長年おこなわれており、いろいろな種類の定義が行われてきた。まず、システムを開発する上で設計初期ではGUIが無視されることが多く、全体のシステムの動作に集中する。

本論では細かい問題を省くために抽象的なシステムのモデルを利用することで情報をどのようにGUIに

利用していくかを検討する。

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。

現在携帯電話の高機能化に伴い、java環境が動作する携帯電話の普及が進み、それがさらに増加することを想定して、今回はjavaでGUIの作成を行う。

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²⁾。

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できるよう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。

携帯電話での利用を想定して実行するためにはタブブラウザ形式にすることで簡単なボタン操作で目的のページへ移動することが可能となる。

4.3 メタデータを利用した動的GUI

GUIの自動生成を行うには基となる情報が必要となる。本研究ではメタデータを基準としてGUIの作成を行う。各センサのメタデータからGUI作成のために得る情報は

- センサデバイスの名前
- デバイスの種類
- 状態変数
- 状態変数の取る値の範囲
- その他オプションで記述される情報

センサデバイスを利用して得られる情報は非常に幅広

いが、単体のセンサが送る情報の種類は少なく、提供方法も決まっている。そのため、これらの情報を使って基本的なサービスを構築することが可能となる。GUIの動的生成の方法には具体的に2種類の方法がある。一つ目は各センサデバイスの各サービスを1つのモジュールとして組み合わせてGUIを構成する。二つ目はあらかじめアプリケーションの基となるフレームを用意しておき、利用可能なセンサデバイスを基に内部の構造を変化させる方法がある。二つの方法について図(4)のようになる。

モジュールとして利用する場合にはメタデータから読み取れるサービス一つ一つに対し、表示、もしくは入力部分を作成し、メタデータに記述された順番にGUIとして表示していく。

センサの情報を閲覧するGUIの場合、まずメタデータの持つサービス名から得られる検知情報の種類を欲しい情報を選択できるように画面に表示する。センサGWには各センサから検知された情報が送信されるため、ユーザが選択した種類の情報をセンサGWと通信し、検知情報を表示する。これが検知情報を閲覧する流れとなる。図(4)上部。

複数のセンサデバイスを連携して動作させるサービスの場合、複数のデバイスから送られて来るメタデータから現在使用可能なデバイスをユーザに伝え、ユーザが指定可能な部分をデータ表示し、選択してもらう。その際、メタデータからはデータの値の上限、下限、単位の情報を読み取りユーザの入力できる値の範囲を示す。ライトON・OFFなど状態の選択や選択できる値の範囲が狭い場合にはリストによって提示し、ユーザ側で選んでもらう。

想定されるアプリケーションにはセンシングされた情報を研究機関などに自動的に送信し、その情報を分析した結果をユーザに通知するサービスなどが想定される。

アプリケーションの拡張については、動的生成時に利用されるパーツを拡張することで見た目の変化を行う。

次にあらかじめアプリケーションのフレームを用意する方法の場合は、基本的にはモジュールとして利用する場合とほとんど変わらない。異なるのはセンサデバイスのデータに特殊な行為をさせるためのアプリケーションが用意されていることである。センサデバイスが検知したデータを記録する際の記録ボタンの付加、家庭内の家電とセンサの連携サービスなど、特定の目的を持った高度なGUIを利用したい場合にはセンサデバイスの部分を既存のGUIに付加するという

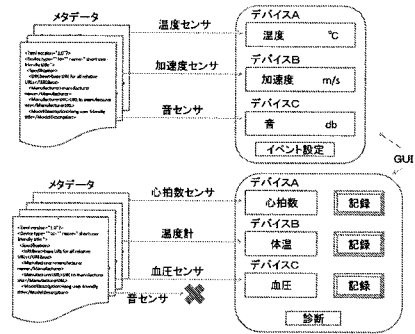


図 4 gui 分類図

形を取る。既存のアプリケーション側は必要なデータの種類だけを定義しておくだけで、センサからの情報を得ることができるようになり、不必要なセンサデバイスの情報は表示されなくなる。図(4)下部。

4.4 ユーザによる個人設定

動的生成でのGUIでは表示方法もユーザの指定によって変化させることが可能となる。表示されるセンサデバイスの順番を五十音順にしたり、自分が良く使用するサービス順に並べ替えるなど、ユーザの気に入った表示方法によってサービスの使い勝手を向上させることができる。ユーザが設定するのはその他背景など見た目に関する設定を行うことができるようになるだけでなく、将来的にセンサデバイスの検知情報を基に、web上のデータベースによってデータの解析を行ってから結果を届けるサービスが普及した際に何の情報を送信するかをあらかじめユーザが設定しておくことで、自動的にサービスの実行を行う。知られたくないような情報が勝手に流出してしまわないようにデータの種別による可否、閾値を設けて送る値の設定を行う。

5. 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構築する際に、センサプロキシで利用するメタデータについての考察し、動的GUIの作成方法にはXMLで記述されたメタデータの読み込みをjavaのプログラムで行い、解析されたデータに基づいてGUIを作成する方法について検討した。センサデバイスを利用するアプリケーションの種類は様々だが、センサ側が提供するデータの種類の提供方法は限られる。そのため、動的に生成する必要があるのは表示・入力部分が主となり、その配置の仕方の検討を行った。

5.1 今後の課題

センサデバイスから送られて来るデータはデータベースに格納されて保存されるため、それらの統計を基にしたイベント発生条件を利用できるようにすることで、さらにサービスの幅を広げることが出来るようになると思われる。その際にはGUIにも統計的データを取得する項目を動的に生成する仕組みの検討が必要となる。今後は検討したGUIについての実装を行い、ユーザの観点から使いやすい構造になるようシステム全体を見ながら改良を加えて行くように研究を進めて行く。また、現在センサデバイスが具体的にどの場所に存在して、何を目的としてその場所にセンサを設置したのかを知るための仕組みを考えることで、さらに複雑なアプリケーションを予想して、動的にGUIを作成する方式について考えていきたい。

参 考 文 献

- 1) 石川憲弘, 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, Oct2006.
- 2) Magdalena Balazinska etc., "Data Management in the Worldwide Sensor Web", IEEE Computer Society Pervasive computing, pp.30-40, April-june 2007.
- 3) Aii A.Alesheikh, Mehrnoosh Ghorbani, Hossein Mohammadi., "DESIGN AND IMPLEMENTATION OF SENSOR METADATA ON INTERNET".
- 4) 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, Oct2006.
- 5) Zunliang Yin, Chun Miao, Yuan Miao and Zhiqi Shen., "Actionable Knowledge Model for GUI Regression Testing", IEEE/WIC/ACM International Conference on Intelligent Agent Technology(IAT'05), pp165-168, Sep2005.
- 6) Judy Bowen, Steve Reeves., "Formal Refinement of Informal GUI Design Artefacts", Australian Software Engineering Conference(ASWEC'06), 10pp - , Apr2006.