

## 異種センサネットワークを連携させるゲートウェイ機能の OSGi 対応

伊藤崇洋<sup>†</sup> 峰野博史<sup>†</sup> 小佐野智之<sup>‡</sup> 石川憲洋<sup>‡</sup> 水野忠則<sup>†</sup><sup>†</sup> 静岡大学情報学部 <sup>‡</sup>(株) NTT ドコモ

## 1 はじめに

近年、センサネットワークに関する研究が環境調査やホームセキュリティ、医療、ヘルスケアなどさまざまな分野でおこなわれている。しかし、現時点で存在するセンサを利用したサービスや研究[1]は、そのほとんどが単一のセンサの管理や単一のセンサのデータを用いたサービスの提供など非常に基礎的かつ簡易的なものにとどまっているのが現状である。しかし、もし複数の異なるセンサのデータを簡単に利用することが可能となり、今までは繋がることのできなかったセンサや各種家電機器などが自由に繋がりがあうことができるようになれば、センサの持つ情報をさまざまな機器によって幅広く利用することが可能となる。こうした環境に、さらに、イベント発生条件やイベント発生時に実行されるサービスをユーザが独自に設定できる環境が加われば、まさに「ユビキタス」な世の中が開けるのではないかと考えられる。

そこで我々の研究室ではこれまで、各種センサや AV 機器、白物家電など複数の異なるデバイスをオーバーレイネットワークで接続(図 1)し、センサの値の変化によって各種デバイスを制御することが可能なアーキテクチャの提案をおこなってきた[2]。本アーキテクチャを用いることによって携帯端末から簡単にイベントやサービスの設定・削除をおこなうことが可能となったが、新規デバイスの追加時にはアーキテクチャ全体を再起動する必要があるなどの問題点があった。リアルタイムのデータを扱うセンサネットワークの分野においてシステムを停止することは出来る限り避けたい事柄である。そこで本論文では、アーキテクチャのゲートウェイ(GW)機能全体を OSGi 対応にすることによって、システムを再起動することなく新規デバイスや拡張機能を容易に追加・削除することが可能なアーキテクチャの提案をおこなう。

## 2 要素技術

## 2.1 PUCG

PUCG (P2P Universal Computing Consortium) [3] は、P2P ネットワーク技術を用いて既存のさまざまなデバイスを相互に接続・運用することを可能とする技術の開発を目的とした標準化団体であり、現在、各ワーキンググループ(WG)での活動を通して通信プロトコルとメタデータの標準化をおこなっている。我々の研究室は PUCG のセンサデバイス制御 WG に参加しており、本論文では現在開発中の PUCG の技術を用いることによって図 1 に示すような P2P のオーバーレイネットワークを形成し、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可能とする環境を実現する。

## Sensor gateway that applies OSGi

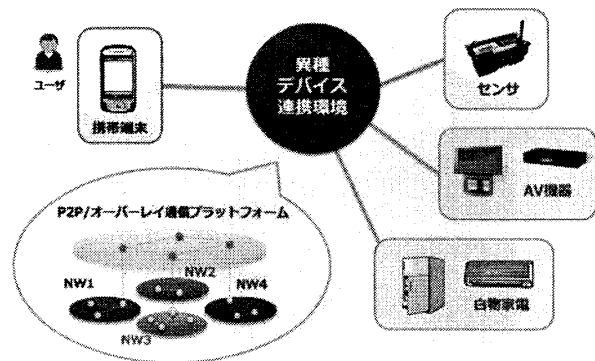
Takahiro Ito<sup>†</sup>, Hiroshi Mineno<sup>†</sup>, Tomoyuki Osano<sup>‡</sup>, Norihiro Ishikawa<sup>‡</sup>, and Tadanori Mizuno<sup>†</sup><sup>†</sup>Faculty of Informatics, Shizuoka University<sup>‡</sup>Dept. of development for Service&Solution, NTT DOCOMO

図 1: 提案する環境

## 2.2 OSGi

OSGi は OSGi Alliance[4] によって標準化がすすめられている Java ベースのサービスプラットフォームである。OSGi の仕様に基づいて作成されたフレームワークを用いることによって、バンドルと呼ばれる小さなアプリケーションを遠隔から動的にインストール・起動・停止・アンインストールすることができる。この際、個々のバンドルはそれぞれ孤立した 1 つのアプリケーションとして扱われ互いのサービスに影響を与えることはない。バンドル間の通信が必要であればフレームワークのサービスレジストリ機能を利用することで連携をとることも可能である。そのため、我々がこれまで提案してきた異種デバイス連携環境を OSGi に対応することによって、実行しているサービスを停止することなく新規デバイスへの対応や新たな機能の追加などが可能となり、より汎用的なアーキテクチャになると考えられる。

## 3 センサ GW

現在利用可能なセンサノードはどれもメモリ容量が少なく演算処理能力が低いため、PUCG で用いられている通信プラットフォームをそのまま実装することは困難である。そのため本論文では、一定の周期で常にセンシングをおこないシンクノードへとデータを送信しつづける単純なセンサノードの利用を想定し、これらのセンサをほかのデバイスと通信できるようにするために GW を設置する手法を用いる。

## 3.1 OSGi 非対応センサ GW

本論文にて新たに提案するセンサ GW を示す前に、これまで提案してきたセンサ GW について述べる。

図 2 に示すように、センサから送られてきたデータはセンサデータ格納用のアプリケーションを通してデータベースへと格納し、間接的にセンサ GW アプリケーションへとデータを流す。センサ GW アプリケーションはユーザに対し第 1 章で述べた機能を提供するための各種モジュールを有し、PUCG プラットフォームを通してほかの PUCG ノードと通信をおこなう。

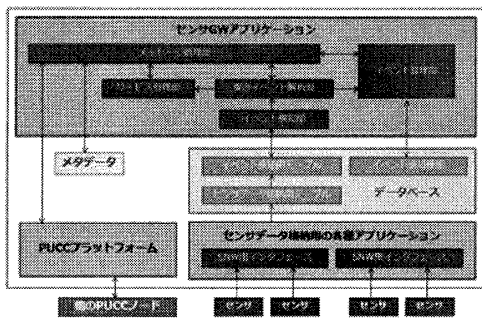


図 2: OSGi 非対応センサ GW のモジュール構成

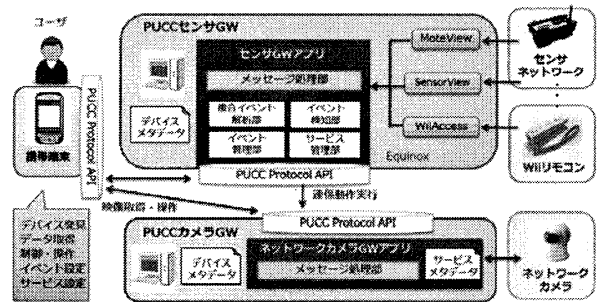


図 4: 実装環境

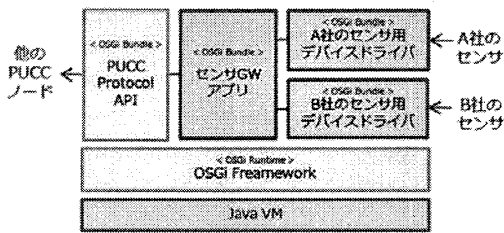


図 3: OSGi 対応センサ GW の構成

### 3.1.1 イベント設定

ユーザによって定義されたイベント発生条件は複合イベント解析部で解析がおこなわれ、イベント管理部を通してイベント検知用の設定がおこなわれる。もし、複合イベントの設定時等により自身のGW下でないセンサに対するイベント発生条件の定義が送られてきた場合は、メッセージ処理部からPuccプラットフォームを通してほかのPuccノードへと通知される。

### 3.1.2 イベント検知・サービス実行

イベント発生の検知は、データベースのトリガー機能を利用しておこなう。イベント発生条件を満たすセンサデータがセンサデータ格納用テーブルに格納された場合、データベースはそのデータをイベント検知用テーブルにコピーする。センサGWアプリケーションのイベント検知部は常にこのイベント検知用テーブルの監視をおこなっており、イベント検知用テーブルにデータが書き込まれたのを検知すると、そのデータを複合イベント解析部を通してサービス管理部へ通知し、設定されているサービスの実行をおこなう。

## 3.2 OSGi対応センサGW

本論文にて新たに提案するセンサGWの構成を図3に示す。

### 3.2.1 Puccバンドル

PuccバンドルはPuccプラットフォーム(Protocol API)をバンドル化したものである。事前にPuccバンドルのサービスをサービスレジストリに登録することで、ほかのバンドルがPuccバンドルを通してほかのPuccノードと通信することが可能となる。

### 3.2.2 センサデータ収集バンドル

センサネットワークのデータ収集インタフェースにはまだ統一的な規格が存在しないため、ネットワークごとにそれぞれデータを収集するためのバンドルを用意する。これらはセンサのデータを収集するとともに、データにある一定の統一性を持たせた上でセンサGWバンドルにデータを渡す役割を担う。

### 3.2.3 センサGWバンドル

センサGWバンドルは、センサGWアプリケーションの各種モジュールをまとめたバンドルである。基本的にはOSGi非対応のセンサGWアプリケーションをOSGiに対応するようカプセル化したものであるが、センサネットワークの増減やセンサGWアプリケーションの機能拡張などに対応できるように、イベント検知部などは適宜センサネットワークごとに別のバンドルとして設計する。

## 4 実装

第3.2節で提案した各種OSGiバンドルを開発し、実装をおこなった。全体的な構成図を図4に示す。

OSGiのフレームワークには現在Eclipseで利用されているEquinoxを用い、Windows XP SP3のPC上で実行した。動作評価用のセンサとして、Crossbow社のMOTE/Mica2、NECトーキン社製のセンサ、Wiiリモコンの3種類を用い、センサのデータを用いたサービス実行を評価するためにはSONY社製のネットワークカメラを用いた。センサGW以外のクライアント端末やネットワークカメラGWの実装については、[2]のものを利用する。

## 5 おわりに

本論文では、異種センサネットワーク連携環境を実現するセンサGWにおいて新規デバイスの追加容易化をはかる一手法としてOSGiへの対応を提案し、実装をおこなった。OSGiに対応したことで、システムを停止することなく動的に新規デバイスへ対応したり拡張機能をインストールすることなどが実現している。

今後は、OSGiフレームワークの搭載されたホームゲートウェイに実際に本アーキテクチャをインストールし、より現実に近い状態での評価をおこなうとともに、複合イベント判定における時間や順序などの考慮や、複数のユーザの利用によるイベント・サービス設定の競合の考慮など、より汎用的なアーキテクチャに向けた検討を進めていきたい。

## 参考文献

- [1] 幸島明男, 井上豊, 池田剛, 車谷浩一; "センサイベント指向のサービス連携ミドルウェア: SENSORD", 2006-UBI-12, pp.37-44, Nov.2006.
- [2] Y. Kato, T. Ito, H. Kamiya, M. Ogura, H. Mineno, N. Ishikawa, T. Mizuno, "Home Appliance Control Using Heterogeneous Sensor Networks", IEEE Consumer Communications and Networking Conference (CCNC2009).
- [3] Pucc <http://www.pucc.jp/>.
- [4] OSGi Alliance <http://www.osgi.org/>.