

# 動的な処理追加を可能とするセンサプラットフォームの提案

†三浦 明大 †金田重郎 †芳賀博英

†同志社大学工学部

## 1 はじめに

近年、家庭内におけるユビキタスコンピューティングの研究が盛んであり、種々のサービスが提案・開発されている。しかし、ユーザの求めるサービスは、家族構成・価値観によって変化する。実用ユビキタスシステムでは、「ユーザが自分の欲しいサービスを自由なタイミングで追加・削除」できなくてはならない。

ユビキタスサービスでは、ユーザやその環境情報であるコンテキスト情報が必要となる。このため、多数のセンサとそのセンサ情報を集約したセンサ DB を設けるのが通常である。この場合、環境の物理量を連続的に出力するアナログ型センサは、データ量が膨大であるため、ローデータのまま記憶すると、DB 容量不足やネットワーク負荷、と言った問題を起す。アナログ型センサでは、ローデータを信号処理して、シンボルとして DB に記憶せざるを得ない。

しかし、上記既存手法では、すでに設置されているアナログ型センサを新サービスに活用することは難しい。アナログ型センサのローデータには、新サービスはアクセスできず、別サービス用に加工されたシンボルしか利用できないからである。即ち、家庭内センサネットワークでは、ハードウェア・センサ・デバイスとソフトウェアであるサービスは完全に分離されている必要がある。センサデバイスを稼働させたまま、ローデータを処理するアルゴリズムを変更・追加できるスキームが必要とされる。本稿では、その実現への課題を示し、解決手法を提示する。

## 2 提案手法

本稿では、アナログ型のセンサに注目する。ローデータを加工した既存シンボルを利用するのではなく、新たなコンテキスト情報を DB に追加格納する。必要となる加工処理はサービスに依存するため、サービスが次々に追加される環境だと、あらかじめデバイスに加工処理を組み込むことは不可能である。そのため、加工処理のプログラムを次々にセンサデバイスに追加す

る。ユーザがサービスを追加すると、センサデバイス上にはコンテキスト情報を生み出すための処理（プログラム）が追加される。

加工処理を行うプログラムの開発には、センサデバイスの詳細仕様が必要となる。そこで、デバイス側とサービス側の双方に XML 表現の仕様ファイルを組み込み、参照することで、デバイスに適した加工処理プログラムを合成する。

## 3 提案するプラットフォーム

センサネットワークの基本構成要素は、センサデバイスを始めとして、デバイス間のネットワークを構成するソフトウェア、データを記録する DB、情報を加工する処理ソフトウェア、サービス・アプリケーションと多岐に渡る。

本稿で提案するセンサネットワークプラットフォームとは、前節で述べた家庭内ユビキタス環境を実現するために、センサネットワークに必要な要素をまとめたものであり、具体的には Sensor Device, Sensor Database, Sensor Server, Service Application からなる。構成概略を図 1 に示す。図中の黒線矢印は制御、破線矢印はデータ（ファイル）の流れを示す。

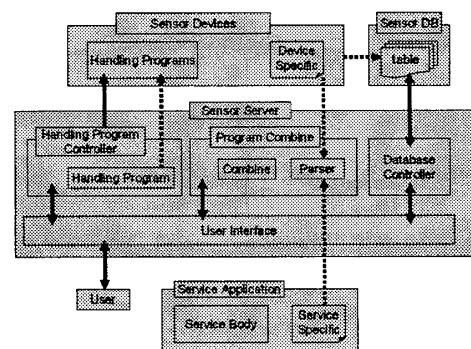


図 1: 提案プラットフォームの構成概略

### 3.1 Sensor Device (SD)

Sensor Device (SD) 上では、Handling Program (HP) が動作している。HP とはセンサのローデータから信号処理などを通して、コンテキスト情報を抽出するためのプログラムである。この HP は Service Application (SA) の追加に伴って、動的にデバイス上に追加される。SD は常に動作し続けるため、HP をデータとして受け取

Sensor Network Platform with dynamic modification of sensing data handling programs

†Akihiro MIURA, Shigeo KANEDA, Hirohide HAGA

‡Faculty of Engineering, Doshisha University

り、プログラムとして実行する動的プログラミングが必要である。SD は自らの詳細な仕様である Sensor Device Specification を持ち、HP の合成に利用される。

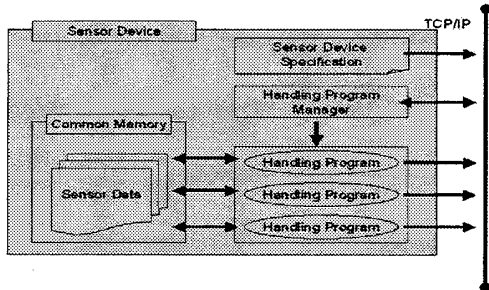


図 2: Sensor Device の構成

### 3.2 Sensor Database (SDB)

各 SD で、HP によって加工された情報は、ネットワーク上の一つの Sensor Database(SDB) に時系列として格納される。HP の出力結果であるコンテキスト情報を収めるカラムは、HP が新規に追加されるたびに、SDB のテーブルに追加される。

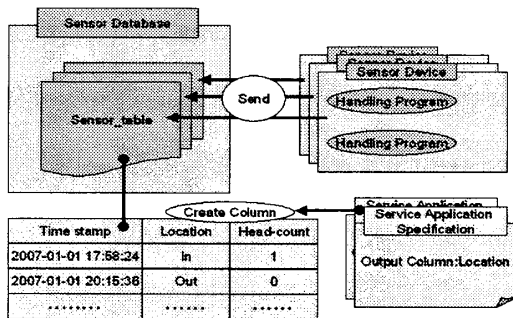


図 3: Sensor Database の構成

### 3.3 Service Application (SA)

Service Application(SA)は、ユーザに何らかの生活支援サービスを与えるソフトウェアである。Sensor Server(SS)を通して、センサネットワークに組み込まれ、SDB からの情報を入力として、サービスを提供する。SA は Service Application Specification という仕様を持ち、HP を合成するための情報が記述されている。

### 3.4 Sensor Server (SS)

Sensor Server(SS)は、SD と SA の持つ仕様を参照して、HP を合成し、それをネットワークを通して SD 上に追加する。サービスの追加や状態変化に伴って、SD や SDB を管理する機能も持つ。ユーザは、Sensor Server のユーザインタフェースを通して、センサネットワークに干渉できる。

## 4 Handling Program の合成

HP は SD の検出した値を加工するため、出力値やサンプリング周期などの SD の仕様に依存する部分が多い。サービス開発者が各家庭に設置された SD の仕様を見ながら、プログラムを作成することは不可能であるので、各家庭の SD の仕様に合わせて、自動的に HP をカスタマイズさせる。このカスタマイズを本稿では“合成”と呼ぶ。合成は、SD と SA の持つ仕様を元に行われる。

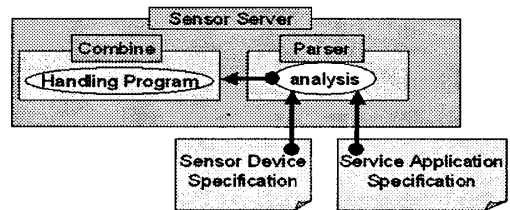


図 4: Handling Program の合成

## 5 プロトタイプの構築

焦電型赤外線センサデバイスを用い、提案プラットフォームのプロトタイプを構築した。プロトタイプでは、提案システムに必要な最低限の構成要素を実現した。実際に、焦電センサから人間の位置を検出するシステムを稼働させながら、焦電センサから身長を判断する別サービスを既存サービスの停止なし追加できた。

これに伴い、位置検出の HP (Ruby のソースで 90 行)、身長差検出の HP (Ruby のソースで 120 行)を合成できた。

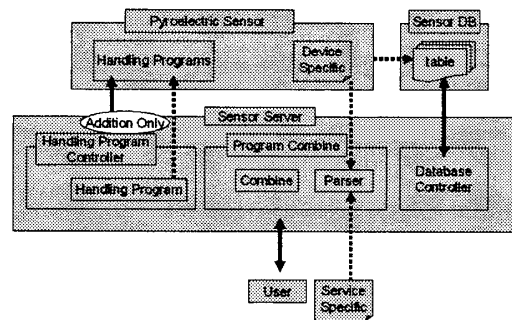


図 5: プロトタイプの構成概略

## 6 終りに

本稿の目的は「デバイスやサービスが固定されていない環境で、ユーザによる自由なサービス追加ができる」ユビキタス環境の構築である。その利用形態を満たすプラットフォームを構築し、プロトタイプを構築・動作させることで、提案手法の有効性を検証できた。