

推薦論文

センサノードのための加速度データに基づく ルール型動作制御方式

児玉賢治^{†1} 藤田直生^{†1} 柳沢豊^{†2}
塚本昌彦^{†1} 義久智樹^{†3}

近年、実世界中に存在する人間や物体の状態を、小型のセンサノードを用いてリアルタイムに取得し、様々なサービスを提供するシステムが提案されている。特に、加速度センサで取得したデータ（加速度データ）は、人間や物体の向き、動き、状態の変化などを細かく把握するために用いられるため、重要なセンサデータの1つである。この加速度データを扱うセンサノードでは、温度や湿度などの他のセンサデータと比べると、データ量が多く、高い応答性が求められる。また、通信量の削減や、設置状況による柔軟な設定の変更といった機能も求められる。しかし、既存のセンサノード上でのデータ処理方式は、加速度データを処理するための要件を十分に満たしていなかった。そこで本論文では、加速度データに基づくルール処理によりイベント駆動を行うセンサノードの動作制御方式を提案する。そして、加速度センサノード（MoCoMi-Chip）にこの処理方式を実装し、応答性や通信データ量を測定した評価実験の結果について報告する。

A Rule-based Acceleration Processing System for Small Sensor Nodes

KENJI KODAMA,^{†1} NAOTAKA FUJITA,^{†1}
YUTAKA YANAGISAWA,^{†2} MASAHICO TSUKAMOTO^{†1}
and TOMOKI YOSHIHISA^{†3}

In recent years, various small sensor nodes have been developed to obtain situations and events occurred in real world. Many application systems applied obtained sensor data to ubiquitous services. Especially, we consider the acceleration sensor is one of the most important sensors because a system can obtain motions, directions, and positions of objects accurately. In general, ap-

plication systems using acceleration data have the following three requirements: 1) a node can rapidly process data without using a large amount of memories since the amount of acceleration data is usually much more than that of other sensor data, 2) a node must reduce the amount of data to be sent to a server, 3) systems can easily reconfigure each node with moderately low cost. Existing sensor nodes, however, do not have enough functions to process such data. In this paper, we propose rule-based data processing system dedicated to process a acceleration data. Moreover, our system can rewrite rules on each node with few bytes data. To evaluate our proposing mechanism, we have done experiments using our rule-based system implemented on our developed small sensor node called MoCoMi-Chip.

1. はじめに

近年、小型のセンサノードを実世界中の様々な場所に配置し、実世界で起こった事象を情報化しようとする研究がさかんに行われている。これにより、多くの実世界の情報をコンピュータが扱えるようになり、今までにないサービスを提供することが可能になりつつある。

従来のセンサノードは、様々なアプリケーションに適用できるように、複数の種類のセンサを搭載している。たとえば、温度センサや湿度、加速度、光、気圧、磁気センサなどがある。これらの中でも特に加速度センサは、物体の動きや傾きがリアルタイムに検出できることから、実世界の人間や物体の動的な状況の変化を把握するために有効なセンサとして、多くのセンサノードに搭載されてきた¹⁾。加速度センサから得られるデータ（加速度データ）は、値の変化が他のセンサデータより速く、時間あたりのデータ量が多くなるといった他のセンサデータと異なる特徴を持っている。さらに、加速度データを扱うアプリケーションでは、その用途に応じてセンサノードの動作を個別に設定する必要がある。このため、センサノードには、加速度データに特化した動作制御方式を導入する必要がある。次の2つの要件があげられる。高速に大量のデータを扱う必要があるという加速度データの特徴から、

^{†1} 神戸大学大学院工学研究科
Graduate School of Engineering, Kobe University

^{†2} NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories

^{†3} 大阪大学サイバーメディアセンター
Cybermedia Center, Osaka University

本論文の内容は2007年7月のDICOMO2007にて報告され、UBI研究会主催により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である。

即応性と通信量の削減が求められる。また、多数のセンサノードを用いたアプリケーションを構築する場合の要件として、動作定義の動的な変更と記述の少量化の2つがあげられる。

一方で、従来のセンサノード上で動作するデータ処理システムでは、温度や湿度といった比較的サンプリング周期が低く、即応性が求められない他のセンサデータと同じ処理方法を用いて加速度データを扱っていることが多い。このように、従来手法では前述の要件が十分満たされていない。

このため本論文では、先にあげた4つの要件を満たすような、加速度データをセンサノード上で処理するためのルール型動作制御方式について述べる。本論文では、筆者らが開発した小型無線加速度センサノード(MoCoMi-Chip)に提案方式を実装し、評価実験を行った結果についても述べる。

以下、2章では関連研究について述べ、3章では提案する動作制御方式について説明する。4章では提案手法の設計、さらに5章ではその実装について示す。6章では提案手法の評価を行い、その有効性についての考察を述べる。最後に7章では本研究のまとめと今後の課題を示す。

2. 関連研究

ルール型の動作制御を用いたユビキタスコンピューティングの研究として、AhroD²⁾があげられる。AhroDでは、ユビキタスデバイスにおけるI/Oポートの入力と出力を結び付けるルール形式の動作定義を保有し、ルールを逐次処理して動作する方式を提案している。ルール記述にはイベント駆動型言語であるECAルールを採用している。ECAルールは、イベント(Event)、コンディション(Condition)、アクション(Action)の組合せで記述するもので、直感的にユーザが理解しやすい形で動作が定義できる点に特徴がある。AhroDは、このECAルールを用いて、入力や通信ポート、内部変数を制御することができる。しかし、AhroDは汎用のルールで処理を行うため、連続的に発生するセンサデータを取り扱うと処理が間に合わない可能性があり、センサノードとしての機能は期待できない。一方、このようなルール型の動作制御をセンサノードに適用した研究としてDYCOME^{3),4)}がある。DYCOMEでは、センサデータを条件とするルールを採用している。また、複数のルールセットを保有し、評価するルールセットを動的に切り替えることにより、異なる動作が要求される複数のアプリケーションへの対応を試みている。しかしながら、センサデータの平均値のみを用いてルールの条件を表現している点や、先述の加速度データを扱うための要件を満たしていない点などを考慮すると、加速度データを扱うセンサノードの動作方式として

は適していない。

一方、センサノードを人や様々な物に取り付け、それぞれのコンテキストを取得するシステムが多く研究されている。これらのシステムでも、加速度データをコンテキストの取得のためのデータとして用いている。たとえば、センサノードを日常生活の様々な物に取り付けるスマートオブジェクトサービスと呼ばれる研究として、コップに取り付けるMediaCup⁵⁾や紙の資料に取り付け管理を行うDigiClip⁶⁾などがあげられる。また、センサノードを人に取り付け、ユーザのコンテキストを取得する研究では、加速度データを解析することで人の行動推定する手法がある^{7),8)}。しかし、加速度データからコンテキストの推定を行う研究では、センサノード上で処理を行わず、ホスト側で処理を行っている事例が多い。このため、センサデータの通信量が場合によっては非常に大きくなる可能性がある。

こうした中で、センサノードがデータをすべてホスト側に送信するのではなく、必要なセンサデータのみを取り出す仕組みとして、TinyDB⁹⁾などが提案されている。TinyDBはSQLに似た書式を用いてクエリを記述し、センサノードが取り出すべきデータをサーバが直接指示することができる。しかし、高いサンプリング周期が必要な加速度データの処理については、処理方式の記述、即応性などの面で問題がある。

このほか、センサノードの動作を変更する研究としては、無線センサネットワークにおいてメンテナンスや管理を容易にするため、多数のセンサノードに対し無線通信を用いてセンサノードのプログラム変更を行うシステムが研究されている¹⁰⁾⁻¹²⁾。これらの研究では、大量のセンサノードに対して、少ない通信量で安全にプログラム変更を行うことを目的としている。たとえば、無線センサネットワークの研究開発に用いられている代表的なセンサノードであるMICA MOTE¹³⁾やSmart-Its Particle¹⁴⁾は無線通信によるプログラムの書き換えOTAP(Over-the-Air-Programming)に対応している。しかし、書き換えに必要な通信量や時間が大きいという問題がある。

3. アプローチ

筆者らの目標は、加速度データを取得して自律的に処理することができ、小型センサノード上で動作するデータ処理システムを開発することである。本章では、最初に筆者らが想定する典型的なアプリケーション例と、それを実現するためのセンサネットワークの構成について述べる。次に、想定する環境下で加速度データを用いてサービスを提供するときの、各センサノード上でデータを処理するシステムの要件について述べる。最後に、それぞれの要件を満たす方法についてその概略を述べる。

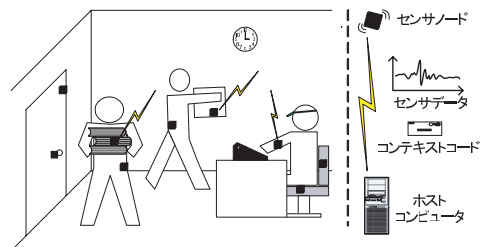


図1 アプリケーションのイメージ
Fig. 1 An application image.

3.1 想定するアプリケーション例

提案するシステムは、小型のセンサノードを屋内の様々な物に取り付け、発生する事象を情報化するようなアプリケーションへの適用を想定している。図1に想定するアプリケーションの一例を示す。この例では、オフィスや製造現場に存在する人や様々な物に取り付けている。物に取り付けられたセンサノードは、取得したセンサデータを処理し、処理したセンサデータや推定したコンテキストをホストコンピュータに無線通信で送信する。ホストコンピュータでは収集した多くのセンサデータやコンテキストを処理し、ユーザへの提示やアクチュエータへの出力を行う。これらより、モニタリングやコンテキストアウェアなサービスの提供を行うことができる。

このようなアプリケーションを実現するうえで、加速度データは重要な役割を担うセンサデータである。加速度データは物体の動きや向きなどの情報を含んでおり、人の行動や状況、物の使用状況や扱われ方などを把握することができる。これにより、たとえば人間の行動履歴を分析することによる業務の効率化や、物の状態の変化をトリガとした人間の日常行動の支援サービスといった、高度なアプリケーションを実現することが可能になる。

こうしたアプリケーションを実現するためには、センサノードには高度なデータ処理能力が求められる。一方で、物や人に取り付けるセンサノードは、小型かつワイヤレスである必要があるため、バッテリー駆動と無線通信での動作が必要となる。このため、センサノードの計算資源に限られるとともに、動作時間や通信時間をできるだけ短く抑える低消費電力設計が求められる。

3.2 加速度センサノードの動作要件

1章で述べた加速度データを扱うセンサノードにおける4つの要件について詳しく述べる。

3.2.1 加速度データの特徴による要件

- 即応性

加速度データの特徴としては、他のセンサデータと比べ変化の頻度が高いことや、それぞれのデータが固有の動きを示す独立したデータとなる場合が多いこと、事象の始まりを示すことが多いことなどがある。たとえば、ユーザの身体いくつかのセンサノードを取り付けて活動を取得する場合には、ユーザの身体各部は頻りに動き、各部のセンサデータ間の相関は小さくなると考えられる。これらの特徴から、加速度データを扱うセンサノードは、高速なセンシングと応答が求められる。人の動きをセンシングする場合には、数十Hz程度のサンプリング周期が必要である⁷⁾。また、物の衝突や落下などの動きをセンシングし危険を知らせるケースでは、人間がその状態を知ってからそれを回避するだけの時間的な余裕が必要であるため、可能な限り高速に状態を取得して通知する、高い即応性が求められる。

- 通信量の削減

加速度データは継続的に高いサンプリング周期で取得されることから、データ量が大きくなるという特徴もある。取得したデータをすべて無線通信で送信するためには、センサノードは大きな消費電力を必要とするため、可能な限り送信するデータ量を削減する必要がある。つまり、システムが必要とするデータのみを送信することが求められる。これを実現するためには、センサノード上で加速度データの処理を直接行い、外部に送信する最小限のデータのみを取り出す機能が必要である。ただし、センサノードでは計算に用いることができる資源が限られているため、可能な限りシンプルな計算量の少ない処理で行う必要がある。

3.2.2 アプリケーションからの要件

- 動作定義の動的な変更

センサネットワークを実際に運用する際には、システムの提供するサービスの内容やセンサノードが設置される環境が変化し、個々のセンサノードにおける動作定義の変更が求められることがある。このような変更を多数のセンサノードに対して頻りに行う場合には、ユーザのコストが大きくなると考えられ、センサノードを設置した後も、容易に動作定義を変更できる機能が必要である。また、システム全体の動作を中断することなく、個別のセンサノードに対して動的に動作定義を変更できる必要がある。

- 記述の少量化

想定するアプリケーションにおいて数多く用いられるセンサノードには、設置される場

所や状況に応じて個別の動作定義を設定する必要がある。よって、ユーザは多くの動作定義を設定することが求められるため、シンプルで分かりやすい少量の記述で動作定義を行う必要がある。また、前述の動作定義の動的な変更を行う際には、無線通信により行うことが前提となる。このため、変更を行うための通信量と時間を少なくすることが求められる、少量の動作記述が望まれる。

3.3 提案手法

本論文では、加速度を扱うセンサノードの動作制御方式として、加速度データを実行条件とする If-Then ルールを用いたセンサデータの処理方式を提案する。用いるルールは加速度データに基づく「実行条件」と、それが満たされるときに実行される「実行動作」の対で構成する。高速な動作を実現するため、ルールの実行条件の評価は加速度データのしきい値判定で行う。比較する値には、瞬時値のほか、平均値、偏差の合計、微分値、カウンタ値を用い、様々な動きや状態を設定する。この加速度データを用いたルール処理に基づきイベントトリガ動作を行うことで、センサノードは設定された動きが起こったときのみ送信を行い、送信する回数、データ量を削減できる。送信する内容は、ホスト側の処理に多様性を持たせるため、検知したコンテキストとセンサデータの両方とする。また、ルールにより処理動作を記述することで、動作定義を少量の記述で行え、個々に異なる多数のセンサノードの設定を容易に行える。加えて、ルールを追加、削除するだけで動作定義の動的な変更が容易に行え、変更に必要なデータ量や時間も少なくできる。

4. 加速度データのためのルール処理動作方式

4.1 ルール処理動作の設計

加速度データのセンシングとセンサデータの処理は、ノードの基本動作であり、これに基づきルール処理が行われる。このため、すべてのルール処理の前に毎回行うことにする。正確な周期でセンサノードの動作を行うため、ハードウェアタイマを用いてサンプリング周期の管理を行う。センシングやルール処理などを含むすべての処理をサンプリング周期内に終わらせ、ハードウェアタイマの割込みを待つことで正確な周期の動作が実現できる。

センサノードは、内部メモリ（ルールベース）に格納されているルールを逐次処理することで動作を行う。このルールを複数保有しておくことで、様々な状況に対応する動作を行える。さらに、複数の実行条件、または複数の実行動作を組み合わせることを可能にし、ルールの自由度の向上を図る。

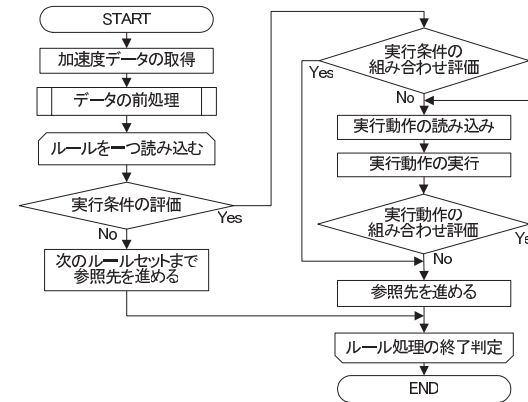


図 2 動作フローチャート
Fig.2 The system flowchart.

4.2 ルール処理動作の流れ

設計したルール処理動作による加速度センサノードの動作は、図 2 に示す流れに従い行われる。

まず、加速度センサから加速度データを取得し、センサデータの前処理を行う。前処理では、動きや状態の特徴を表すことのできる値をいくつか算出する。これらの値をもとに、実行条件をルールベースから 1 つずつ読み込み、評価を行う。実行条件が成立する場合、複数の実行条件の組合せであるか評価する。複数の実行条件の場合、次の実行条件をルールベースから読み込み、再び評価を行う。1 つまたは複数からなる実行条件がすべて成立するとき、次に格納されている実行動作を読み込み、実行する。実行動作も実行条件と同様に、複数の実行動作の組合せであるか評価を行い、1 つまたは複数からなる実行動作を実行する。また、実行条件が成立しない場合は、次の実行条件と実行動作の組合せ（ルールセット）になるまでとばしてから実行条件を読み込む。格納されているすべてのルールを評価し終わると、サンプリング周期になるまでハードウェアタイマにより待ち動作を行う。

以上の流れを繰り返し、ルールに基づいた加速度センサノードの動作制御を行う。

4.3 実行条件

加速度データから動きや状態を把握する研究では、平均値や分散値、偏差値、FFT、SVM (Support Vector Machine) などが用いられている^{7),8)}。しかし、計算資源が限られるセンサノード上で、FFT などの高度な処理は高速に行えない¹⁵⁾。また、SVM などの高度な判

別手法ではセンサデータを学習データとした場合に計算量が膨大となることが想像できるとともに、ルールによる判別を直感的な条件で設定できないと考えられる。そこで本研究では、単純な処理で高速に算出できる 8 ビット化した瞬時値や平均値、偏差の合計、微分値を前処理として算出し、実行条件の評価に用いる。平均値は、比較的簡単な処理で求められ、ノイズの平滑化などに用いることができる。分散値は、動きの有無を推定することができ有用であるが、処理がやや複雑であり低コストの MCU で高速に算出することは難しい。このため、分散値に似た意味の値として偏差の合計を用いることにした。なお、平均値と偏差の合計は過去 16 回分のデータを基に算出する。さらに、衝撃など急な値の変化を扱うために直前のデータとの差分を用いた微分値を採用する。

上記の前処理で算出する値に加え、繰り返される状況に対応できるよう 4 つのカウンタ値（条件カウンタ）を組み合わせて用いる。条件カウンタは実行動作によってインクリメントとリセットが行える。これにより、ある状態が特定の回数発生したときに条件を満たすというような実行条件や、状態の経過時間で評価する実行条件を記述できる。また、複数の周期にわたり条件を満たし続ける状態を 1 回の状態として扱えるようにするため、連続した条件の成立を認めない設定を用意する。これにより、ある期間連続して同一のイベント検知が行われる場合に、イベントの開始時のみ条件を満たすような実行条件の記述を可能にする。

実行条件の成立はしきい値との大小により評価し、データを比較する単純な処理で実現できる。

4.4 実行動作

ルール処理動作により実行する加速度センサノードの実行動作として次のものを考える。

- 加速度データの送信
- 加速度データの連続送信
- コンテキストの送信
- スリープモードへの移行
- ポート出力
- サンプリング周期の変更
- 加速度検知範囲の変更
- 条件カウンタの更新
- ホストへのルール編集の問合せ
- 格納しているルールの送信

加速度データの送信は、基本動作の 1 つであり、瞬時値のほか前処理で算出した特徴量の

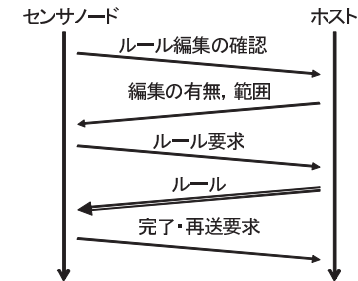


図 3 ルール編集の流れ
Fig. 3 The flow of rule edit.

送信が考えられる。ホスト側がストリームデータを必要とする場合には、一定時間加速度データを送信し続ける動作も必要である。加えて、数値データだけでなく、検知したイベントやセンサノードの状態をコンテキストとして送信する動作も用意する。さらに、センサノードの消費電力を抑えるためにスリープモードへの移行も必要な実行動作の 1 つである。そのほか、センサノードの LED などへの出力を行うためのポート出力や、使用条件や環境に応じてサンプリング周期、加速度センサの検知範囲の変更を行う実行動作を用意する。繰返しの状況に対応するための条件カウンタの更新は、インクリメントとリセットの実行動作を用意することで対応する。ホストに対するルール編集の問合せは、実行動作によっても実行できるようにし、ノードの格納しているルールを参照するためのルールを送信する実行動作も用意する。

4.5 ルールの編集機構

ルールの編集は図 3 の流れに従い行う。ルール編集の有無の確認はセンサノードがホスト側に対して、定期的もしくは実行動作として行う。これに対しホスト側は、編集の有無と範囲について通知する。変更がある場合、センサノードはルールの転送要求を行い、ホスト側から新たなルールを受信し指定された範囲のルールを更新する。ルールの転送終了後、センサノードはチェックサムにより検証を行い、ルール編集の完了か再送要求を行う。

5. ルール処理動作の実装

小型無線加速度センサノード (MoCoMi-Chip) を開発し、設計したルール型の動作制御方式の実装を行った。

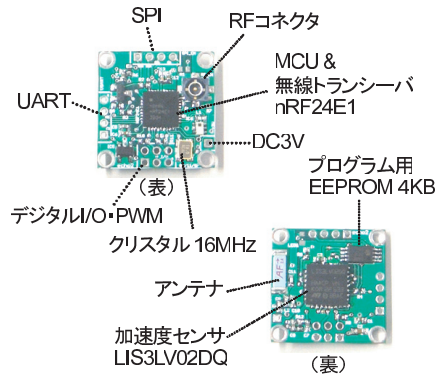


図 4 センサノードの実装基板

Fig. 4 The board layout of our proposed sensor node.

5.1 ハードウェア構成

提案手法の実装評価や想定するアプリケーションの実現には、人や物の動きのセンシングに適した無線加速度センサノードが必要である。このため、採用するセンサノードは、様々な場所に無理なく設置できるように小型かつ薄型で、精度良く加速度をセンシングできることが求められる。現在までにいくつかのセンサノードが開発、提供されているが、上記の要件を満たすものがなかったため、筆者らで新たなノードを開発した。今回開発した MoCoMi-Chip (Motion sensing and Communication Minimized Chip) の外観と主な仕様を図 4 と表 1 にそれぞれ示す。

MoCoMi-Chip では、Intel8051 互換 MCU と無線トランシーバを 1 つの LSI に搭載する Nordic 社の nRF24E1 を採用した。MCU は、内部 ROM として 512 バイト、内部 RAM として 256 バイトを搭載している。プログラム用のフラッシュメモリは搭載しておらず、起動時にブートローダが外部の EEPROM (4K バイト) からユーザプログラムを読み込む仕組みとなっている。無線トランシーバは 2.4 GHz 帯の周波帯で、Nordic 社の通信規格である ShockBurst により無線通信を行う。ShockBurst 通信は、最大 1 Mbps で 1 対 N の通信を行う機能を持ち、低消費電力を特長としている。加速度センサは、ST マイクロ社のデジタル 12 ビット出力の LIS3LV02DQ を採用した。LIS3LV02DQ は、 $\pm 2G$ と $\pm 6G$ の加速度検知範囲が選択可能であり、最大 640 Hz のデータ取得に対応する。また、LIS3LV02DQ は、外部に対して割り込み信号を発生させる機能を持ち、設定値以上の加速度が検出されたと

表 1 センサノードの主な仕様

Table 1 Specifications of our proposed sensor node.

MCU	Intel8051互換MCU (nRF24E1に内蔵)
MCU動作周波数	16MHz
プログラムメモリ	4KByte
データメモリ	256Byte
無線トランシーバ	nRF2401 (nRF24E1に内蔵)
通信規格	Nordic ShockBurst
無線通信速度	1Mbps/250Kbps
通信距離	25m程度
技術適合	2.4GHz帯高度化小電力データ通信システム
インタフェース	UART, SPI, デジタルI/O, PWM出力
加速度検知感度	$\pm 2g/6g$ 選択可能
加速度分解能	12bit
消費電流	送信時: Typ.20mA, スリープ時: Typ.4 μ A
電源	DC3~9V
外形寸法	20×20×3.9mm
質量	約1g

きに nRF24E1 をスリープモードから復帰させることができる。MoCoMi-Chip は、20 mm 四方の基板 1 枚で構成しており、既存のセンサノードの中でも最小クラスとなっている。

5.2 ルールの記述

ルールの記述は、実行条件記述と実行動作記述の組合せで構成する。ルールの記述として、ビット列の比較により容易にルールの識別処理が可能なバイナリルールを採用する。ルールのコーディングは図 5 と図 6 に示すフォーマットに沿って行う。実行条件記述、実行動作記述ともに 2 バイトの固定長で記述する。最上位ビットにフラグを設けることで、複数の実行条件、または複数の実行動作の組合せを可能にする。つまり、実行条件を連続して記述することにより条件を AND 判定することができる。また、OR 判定は別のルールとして用意することで可能となる。なお、ユーザによるルール作成には、入力フォームから項目を選択していくことでルールのコードを自動で出力するようなアプリケーションの利用を想定している。

5.2.1 実行条件記述

実行条件記述は、図 5 に示すように、比較するデータの種類、加速度センサの軸、比較符号、連続成立の許可、およびしきい値の記述により構成する。しきい値には 8 ビット化した加速度データや前処理で算出した特徴量、条件カウンタの値を記述する。

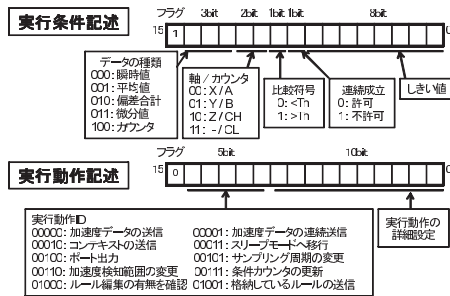


図 5 ルール記述のフォーマット

Fig. 5 The format of the rule description.

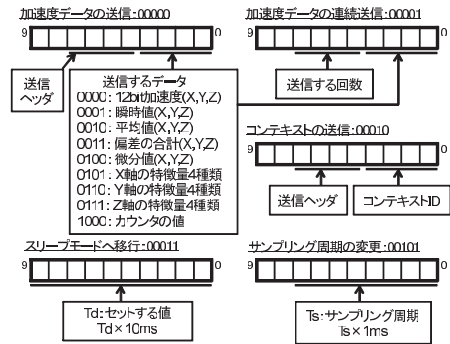


図 6 実行動作の詳細記述 (一部)

Fig. 6 A part of detail formats of the action description.

5.2.2 実行動作記述

実行動作記述は、図 5 に示すように 5 ビットの実行動作の ID と 10 ビットの実行動作の詳細により構成する。実行動作の ID により行う実行動作を識別し、下位 10 ビットでその詳細を設定する。実行動作の詳細は、図 6 に示すように、実行動作の内容によりフォーマットが異なる。たとえば、加速度データの送信する実行動作では、データの種類や軸ごとに送信するデータを選択する。

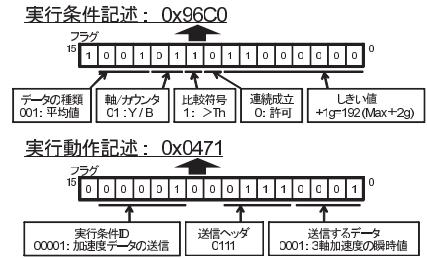


図 7 ルールの作成例

Fig. 7 An example of the rule description.

5.2.3 ルールの作成例

ルールの作成例を図 7 に示す。フォーマットに従い「Y 軸の加速度の平均値が +1g より大きいとき、3 軸加速度の瞬時値を送信する」というルールを作成した場合、「0x96C0, 0x0471」の 4 バイトで記述される。

6. 評価

提案手法の評価として、ルール処理動作を実装した MoCoMi-Chip を用いて実験やその考察を行い、先述の 4 つの要件を満たすかどうかを検証する。まず、即応性と通信量の削減を検証するために、MoCoMi-Chip を人や物に取り付けて表 2 に示す 9 つのイベントを検知するアプリケーションにそれぞれ適用した。イベント検出の例として、筆記動作の検知に適用した際の外観を図 8 に、その加速度データを図 9 に示す。また、これらのアプリケーションで用いたルールを一般的なプログラム記述と比較し、記述の少量化についても評価を行う。このほか、センサノード上のルールをホスト側との通信で動的に書き換える実験や、ルール処理動作に関する考察についても示す。

6.1 即応性

提案手法を 9 つのイベントを検出するアプリケーションにそれぞれ適用し、即応性について検証する。実験では、MoCoMi-Chip にイベントを検知する条件と、イベントの ID を送信する動作のルールを用意しておき、ルール処理を含むセンシングから通知送信までに要する時間を測定した。処理時間の測定は MCU のハードウェアタイマを用いて行った。各イベントを検知し通知送信するルール処理に要する時間を測定したところ、表 2 に示すように 2.5 ms 以下の処理時間を要することが分かった。実験結果から、単純なイベント検知と

表 2 測定結果
Table 2 Evaluation results.

検出イベント	送信量 [Byte]	データ量の比較	イベント検出回数	処理時間 [ms]	ルール数	C言語でのコード量 [Byte]
加速度データ常時送信	75000	100	-	-	-	-
自由落下の開始	150	0.2	10	2.47	7	138
床への落下	585	0.78	39	2.49	12	294
サイコロの出た目	270	0.36	18	2.58	18	396
ドアの開閉	180	0.24	12	2.33	6	130
引き出しの開閉	180	0.24	12	2.47	10	216
筆記動作の開始	1575	2.1	105	2.30	4	84
人の立ち座りの変化	435	0.58	29	2.38	8	169
人の移動の有無	2010	2.68	134	2.11	3	59
手の動きの停止	240	0.32	16	2.30	4	88

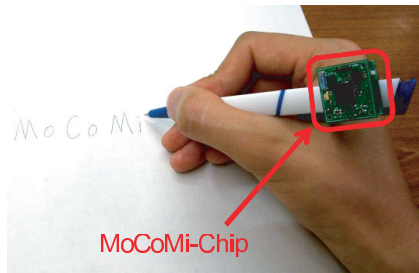


図 8 ボールペンに取り付けた MoCoMi-Chip
Fig. 8 A MoCoMi-Chip mounted on a pen.

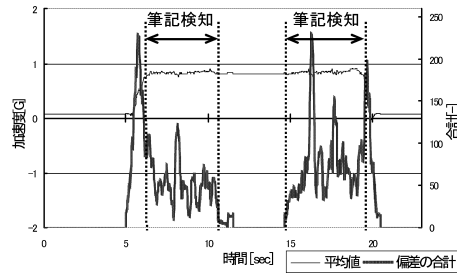


図 9 ボールペンの筆記動作における加速度データ
Fig. 9 Acceleration data for pen operation.

その通知送信を 400 Hz 程度で行えることが分かる。このことから、数十 Hz のサンプリング周期を必要とする人の動きのセンシングにも十分対応でき、リアルタイム性が求められるシステムに対しても十分に高速なイベント通知動作を提供できると考えられる。

また、ルール処理動作の課程をいくつかの部分に分けて測定したところ、送信動作が 1.0 ms と最も長い時間を要し、1 つの実行条件の評価には最大で 0.04 ms を要することが分かった。このことから、厳密に実行条件を設定して 1 サイクル内で成立するイベントを少なくすることにより、多くのイベントに対応するために保有するルール数を増やしても、処理時間は低下しないことが分かった。

6.2 通信量の削減

従来システムのセンサノードにおける通信の大部分であるセンサデータの送信と、提案手法によるノード上でイベントを抽出した場合の送信とでデータ量を比較し、通信量の削減について検証する。実験では、前述のイベントを検出するアプリケーションにおいて、ルール処理動作を用いてイベントを検知したときのみイベントの ID を送信する場合と、3 軸の加速度データをそのまま送信する場合の送信量を測定した。検知するイベントは、表 2 に示した 9 つの簡単なイベントで、100 秒間に 10 回発生させることにした。サンプリング周期は 50 Hz で行い、用いるルールは予備実験によりイベントの開始を精度良く検出できるようにあらかじめルールの実行条件の調整を行ってある。データの無線通信については、1 パケットが 15 バイトで、そのうちのペイロードが 8 バイトとなる構成を用いた。

表 2 に送信量とデータ量の比較、検知したイベントの回数の実験結果を示す。3 軸の加速度データを 100 秒間送信した場合、送信するデータ量は 75,000 バイトとなるのに対し、提案手法によりイベント通知動作を行った場合にはおよそ 2,000 バイト以下の通信量となった。イベント検出回数が 10 回を超えている理由としては、イベント発生時の遷移状態における多重検知と誤検知が考えられる。

以上の結果から、上記のアプリケーションではセンサノードが送信するデータを 100 分の 1 以下に削減できることが分かった。よって、提案手法により設定したイベントが発生したときのみ通知動作が行え、センサノードの省電力化を大幅に行える。

6.3 ルールの動的な変更

ルール編集処理に要する時間を測定し、ルールの動的な変更が行えることを検証した。実験では、MoCoMi-Chip に 32 個のルールを格納させておき、そのうち 8 個と 16 個のルールの書き換え処理に要する時間をそれぞれ測定した。

実験から、8 個のルールを書き換える場合には 12.4 ms、16 個の場合には 14.5 ms の処理時間を要することが分かった。この結果より、提案手法を用いて人や物の動きのセンシングに必要なサンプリング周期内に、センサノード内のルールが編集できることを確認した。すなわち、想定するアプリケーション環境に配置されたセンサノードに対して、動作を中断させることなく動作定義の動的な変更が可能である。

MNP や Impala, Stream などのセンサノードの動作定義を動的に書き換える研究では、ユーザプログラムのバイナリコードのすべて、もしくは一部を無線通信で配信して内部プログラムの更新を行ことができる。しかしながら、配信されるバイナリコードは数キロバイトから数百キロバイトのサイズであると報告されている。これに対し、提案手法では書き換え

ルール	C言語
Cx8b8e	if ((Accel_Z>0x8e)
Cxab8c	&&(Accel_Lif_Z>0x8c))
Cx088c	Send_Message(DrawerClosed);
Cx8a83	if (Accel_Z>0x83)
Cx1c00	IncrementCounter (CounterA);
Cx8884	if (Accel_Z<0x84)
Cx1d00	ResetCounter (CounterA);
Cxc208	if (Counter_A>Cx08) {
Cx088b	Send_Message(DrawerOpened);
Cx1d00	ResetCounter (CounterA); }

図 10 ルールと C 言語による記述例

Fig. 10 Description example by the rule and C-language.

に要する通信量はほぼ更新するルールのコード量であり、そのルールは数十バイト程度である。このように、提案手法ではルールによる短い動作記述を用いることで、非常に少ない通信量で動的にセンサノード上の動作定義を変更することができる。

6.4 記述の少量化

動作記述の少量化を評価するために、ルールによる動作記述と一般的にセンサノードのプログラミングに用いられる C 言語による記述とでコード量を比較した。この比較では、前述のイベントを検出し通知送信を行うアプリケーションで用いた 9 つのルール記述と、同様の条件判別と送信動作を行う C 言語による記述のコード量を用いた。表 2 にそれぞれのコード量を示し、図 10 に引き出しが開閉するイベントを検知する際に用いた記述例を示す。

提案するルールで記述する場合、1 ルールが 2 バイトのバイナリで構成されることから、16 進数表記を用いて 1 ルールにつき 4 バイトのテキストで記述することができる。したがって、提案するルール記述を用いることで C 言語で記述する場合より 5 分の 1 程度の少ないコード量で記述できることが分かる。C 言語で記述した場合には、演算や制御処理、関数を用いることができ、ノードの動作を詳しく設定できるという利点がある。しかし、決まった処理過程を持つセンサノードの動作を記述する場合には、ルールを用いて処理内容を記述する方が少量で記述できる。

6.5 ルール処理動作に関する考察

提案手法により人や物の姿勢や単純な動きのイベントは、それぞれが機能する際に維持される姿勢や動きの有無などを実行条件にすることで検知できたが、人の動作などの複雑な動きは提案手法で用いた加速度データの特徴量だけでは検知できない。この問題に対しては、動きがあるときのみ加速度データを送信するようにルールを設定し、従来システムと同様に計算資源の大きいホスト側で認識処理を行うことで対応できる。また、複雑な動きと

るイベントの遷移状態を除き、多数のノードから単純で静的なイベントを集約することで、複雑で動的なイベントを推測することも考えられる。このように、センサノードを提案手法で動作させ、センサノード上で行う検知処理とホスト側で行う高度な認識処理の組合せが、システム全体の省電力化とデータ量の削減に有効であると考えられる。

また、小型で低コストのセンサノードにおいてはシステムの実装に必要なプログラムサイズは小さいことが望まれる。関連研究の DYCOME では、Smart-Its Particle 上でルール処理動作に相当する動作スイッチング機構と呼ばれるサブシステムを 3,728 バイトで実装している。これに対し、本研究では、ルール処理動作のプログラムを 333 バイト、センサノードのプログラム全体を 4 キロバイト以下で実装しており、センサノードにおけるより小さいルール処理システムを実現している。そして、提案手法は多くの既存のセンサノードに適用が可能であるとともに、より低コストのセンサノードで様々なアプリケーションの実現を可能にする。

7. ま と め

本論文では、加速度データに基づくルール処理を用いたセンサノードの動作制御方式を提案し、設計を行った。また、人や物の動きのセンシングに適した加速度センサノード (MoCoMi-Chip) を開発し、提案するルール処理動作の実装と評価を行った。評価実験から、提案手法を用いることにより、加速度データを扱うセンサノードの動作要件としてあげた、即応性や通信量の削減、動作定義の動的な変更、記述の少量化が実現できることを確認した。

今後の課題としては、ルールの生成や再設定を自動で行えるシステムの開発、加速度以外のセンサデータも扱うルール処理動作の実現などがあげられる。

謝辞 本研究の一部は文部科学省特定領域研究「情報爆発のための装着型入出力デバイスを用いた情報操作方式」(19024056)、および、文部科学省科学研究費補助金 (18700085) の研究助成によるものである。ここに記して謝意を表す。

参 考 文 献

- 1) Beigl, M., Krohn, A., Zimmer, T., et al.: Typical Sensors needed in Ubiquitous and Pervasive Computing, *Proc. INSS 2004*, pp.153–158 (2004).
- 2) Terada, T., Tsukamoto, M., Hayakawa, K., et al.: Ubiquitous Chip: A Rule-based I/O Control Device for Ubiquitous Computing, *Proc. International Conference on Pervasive Computing (Pervasive 2004)*, pp.238–253 (2004).
- 3) Koizumi, K., Sakakibara, H., Iwai, M., et al.: A Context-Oriented Application

Switching Mechanism for Daily Life Supports, *7th International Conference on Ubiquitous Computing (UbiComp 2005)*, Poster Session, Tokyo, Japan (2005).

- 4) 小泉健吾, 中井彦一郎, 榎原寛ほか: 無線センサノードを用いた日常生活サービス構築機構の提案, 情報処理学会研究報告, コピキタスコンピューティングシステム研究会 (UBI-11-9), pp.59-66 (2006).
- 5) Beigl, M., Gellersen, H.-W. and Schmidt, A.: Mediacups: Experience with design and use of computer-augmented everyday artefacts, *Computer Networks*, Vol.35, No.4, pp.401-409 (2001).
- 6) Decker, C., Beigl, M., Eames, A., et al.: DigiClip: Activating Physical Documents, *Proc. 24th International Conference on Distributed Computing Systems Workshops (ICDCSW'04)*, pp.388-393 (2004).
- 7) Kawahara, Y., Kurasawa, H. and Morikawa, H.: Recognizing User Context Using Mobile Handsets with Acceleration Sensor, *Proc. IEEE International Conference on Portable Information Devices (IEEE Portable 2007)*, Orlando, FL U.S.A. (2007).
- 8) 田淵勝宏, 納谷 太, 大村 廉ほか: 加速度センサを用いた日常行動識別におけるデータ収集条件の識別性能への影響評価, 信学技報, Vol.106, No.PRMU2006-27, MI2006-27, pp.43-48 (2006).
- 9) Madden, S., Franklin, M., Hellerstein, J., et al.: TinyDB: An Acquisitional Query Processing System for Sensor Networks, *ACM Trans. Database Syst.*, Vol.30, No.1, pp.122-173 (2005).
- 10) Kulkarni, S. and Wang, L.: MNP: Multihop Network Reprogramming Service for Sensor Networks, *25th IEEE International Conference on Distributed Computing Systems*, pp.7-16 (2005).
- 11) Liu, T. and Martonosi, M.: Impala: A middleware system for managing autonomic, parallel sensor systems, *Proc. 9th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pp.107-118 (2003).
- 12) Panta, R.K., Khalil, I. and Bagchi, S.: Stream: Low Overhead Wireless Reprogramming for Sensor Networks, *Proc. IEEE Conference on Computer Communications (Infocom'07)* (2007).
- 13) Crossbow Technology, Inc. <http://www.xbow.com/>
- 14) Holmquist, L., Mattern, F., Schiele, B., et al.: Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts, *Proc. 3rd International Conference on Ubiquitous Computing (UbiComp 2001)*, pp.116-122 (2001).
- 15) 小澤政博, 川原圭博, 川西 直ほか: 資源制約デバイスにおける3軸加速度センサを用いた姿勢推定手法, 情報処理学会研究報告, コピキタスコンピューティングシステム研究会 (UBI-16-10), pp.65-70 (2007).

(平成 20 年 1 月 4 日受付)

(平成 20 年 9 月 10 日採録)

推 薦 文

センサネットワークにおいて小型, 汎用性, 利便性が求められるセンサノードに関して有望な実用技術研究である. 既存のセンサノードは, 設置後の機能変更が容易でない, 汎用性の高い設計となっているため小型でないなどの問題があった. 本論文ではルール型の動作制御を適用し, 小型, 省電力のセンサノードを開発した. コピキタスコンピューティングの実用化を目指した研究事例として高く評価し, 本学会技術の発展に寄与すると考えられ, 論文誌論文として推薦する.

(コピキタスコンピューティングシステム研究会主査 戸辺義人)



児玉 賢治

2005 年大阪府立工業高等専門学校システム制御工学科卒業. 2007 年神戸大学工学部電気電子工学科卒業. 現在, 同大学院工学研究科電気電子工学専攻博士前期課程に在籍. コピキタスコンピューティングに興味を持つ.



藤田 直生 (学生会員)

2002 年奈良工業高等専門学校卒業, 2004 年同専攻科修了. 2006 年大阪大学大学院工学研究科博士前期課程修了. 現在, 神戸大学工学研究科博士後期課程に在籍. コピキタス・ウェアラブルコンピュータに興味を持つ.



柳沢 豊 (正会員)

1994 年大阪大学工学部情報システム工学科卒業. 1998 年同大学院工学研究科博士後期課程修了. 同年日本電信電話株式会社入社, 基礎研究所にて空間情報処理の研究に従事. 2000 年より同社コミュニケーション科学基礎研究所に所属, 現在に至る. 工学 (博士). 空間情報処理, プログラミング, センサネットワークの研究に従事.



塚本 昌彦 (正会員)

1987年京都大学工学部卒業。1989年同大学院修士課程修了。同年シャープ(株)入社。1995年大阪大学大学院講師, 1996年同専攻助教授を経て, 2004年神戸大学工学部電気電子工学科教授となり, 現在に至る。特定非営利活動法人ウェアラブルコンピュータ研究開発機構理事長。工学博士。ウェアラブル・ユビキタスコンピューティングの研究に従事。ACM, IEEE

等8学会の会員。



義久 智樹 (正会員)

2002年大阪大学工学部電子情報エネルギー工学科卒業。2003年同大学院情報科学研究科マルチメディア工学専攻修士課程修了。2005年同専攻博士課程修了。博士(情報科学)。京都大学学術情報メディアセンター助教を経て, 2008年より大阪大学サイバーメディアセンター講師。この間, カリフォルニア大学アーバイン校客員研究員。ユビキタスコンピューティ

ング, 放送通信融合環境に興味を持つ。電子情報通信学会, 日本データベース学会, IEEE各会員。