

## センサ GW の連携による複合イベント検知の実現

神谷 英樹<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> 静岡大学大学院情報学研究科

<sup>††</sup> 静岡大学情報学部

<sup>†††</sup> NTT DoCoMo

**あらまし** センサネットワーク (SNW) の大きな役割の 1 つにイベント検知がある。キッチンに設置された専用センサによるガス漏れや火事などの災害検知やドアフォン付属の人感センサによる来客の検知など、SNW にはアプリケーションが要求する任意のイベントを検知し、通知する機能が求められる。イベントには単一のセンサデータから判断できる単一イベントと、複数の単一イベントにより判断される複合イベントが存在するが、各地に偏在する多種多様なセンサデバイスが移動し、混在する動的且つグローバルなネットワーク環境における複合イベントの検知は詳細に検討されていない。本稿ではセンサネットワークを管理するゲートウェイ (GW) を用いた動的且つグローバルなネットワークにおける複合イベントの検知可能なシステムについて述べる。

**キーワード** ユビキタス, センサネットワーク, イベント検知

## Composite Event Detection with cooperating between Sensor Gateways

Hideki KAMIYA<sup>†</sup>, Hiroshi MINENO<sup>††</sup>, Tomoyuki OSANO<sup>†††</sup>, Norihiro ISHIKAWA<sup>†††</sup>, and

Tadanori MIZUNO<sup>††</sup>

<sup>†</sup> Graduate School of Informatics, Shizuoka University, 3-5-1

Jyohoku, Naka-ku, Hamamatsu, Shizuoka, 432-8011, Japan

<sup>††</sup> Graduate School of Science and Technology, Shizuoka University, 3-5-1

Jyohoku, Naka-ku, Hamamatsu, Shizuoka, 432-8011, Japan

<sup>†††</sup> Service & Solution Development Department, NTT DoCoMo, Inc.

**Abstract** Event detection is one of important requests for sensor networks. Through ubiquitous environment that heterogeneous sensor devices exist, move and collaborate and network topology can change irregularly, it is not consider the way to detect composite event under such environment. We introduce composite event detection system under such environment using sensor gateways that is developed by authors.

**Key words**

### 1. はじめに

ユビキタス社会に向けてこれまで多くの議論や研究がされてきたが、「いつでも、どこでも、誰でも」サービスを受容できる環境の実現のために、センサネットワーク (以下 SNW) には非常に大きな期待が寄せられている。この SNW を普及させるために様々な取り組みが行われているが、中でもユーザのニーズの強い生活支援ツールとしてのセンサアプリケーションの開発が注目されている。これは Suica や Pismo など比較的活に近い SNW のサービス化が進んでいることや、低コストで簡単に構築できる小規模なネットワークのニーズが大きいという背景に起因すると考えられる。

将来的にインフラが整備され随所に SNW が配置された場合、多様なアプリケーション (図 1) の台頭が予想される。例えば、ユーザは SNW を家屋に設置することで、外出先でも空き巣の侵入や家族の帰宅などのイベント情報や電灯やガスなどの観測データなど我が家の実環境の状態を遠隔から把握するホームセキュリティ系アプリケーションが考えられる。また、スーパーなどの小売店に SNW が配置され駐車場の空車情報やタイムセールスなどの情報を提供していればユーザが付近のそれら小売店のセンサ情報を動的に発見・検索するような非常に汎用性の高いアプリケーションが開発されるかもしれない。

これらのセンサアプリケーションに利用される重要な技術にイベント検知がある。温度や湿度といった気象環境の変化の検

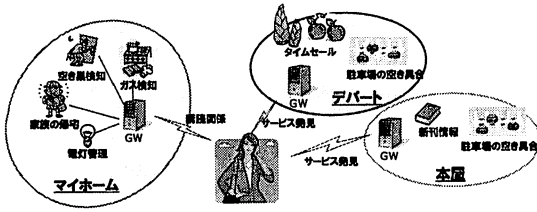


図1 アプリケーション例

知や地震・火災などの災害検知，来客や空き巣などの人検知などセンシングデータを元に判断される実環境でのイベントの検知はSNWに要求される大きな役割の1つである。単純に人に通知するようなものから，実環境の状態によって連係させた家電を制御を変える高度なものまで，どのようなセンサアプリケーションに対してもアクションを発生させるトリガーの役割を果たすためである。このイベント検知に関しても既にいくつもの既存研究が存在する。しかしながらそれらの研究は同種のデバイスが高密度に存在する単一のSNW内におけるイベント検知の通知時間の短縮や情報の精度，耐故障性などに関するものが多い。故にそれらが想定しているアプリケーションも火事や爆発の検知といった自治体などの公共機関により提供されるような極めて限定的なものとなっている。つまり，既存の研究はヘテロジーニアスなセンサデバイスが混在し，移動するようなユビキタスセンサネットワーク上でのイベント検知は検討されていない。

よって本稿ではセンサGW(ゲートウェイ)を利用して，ヘテロジーニアスなセンサデバイスが自由に移動し相互通信をする動的ネットワーク環境における，複数のSNWs上で検知されるイベントで構成される複合イベントの検知を実現するシステムの提案と開発について述べる。

本論文では，2章において関連研究とその課題について述べ，3章では，提案する複合イベント検知システムについて述べる。次に4章においてその実装について述べ，最後に，5章においてまとめと今後の課題について整理する。

## 2. 関連研究

筆者らが複合イベント検知システムに必要な要件や機能を検討するために，ここでは複合イベント検知に関する関連研究とその課題について整理する。

### 2.1 イベントの定義

イベントとは状態の変化であり，単一イベントと複合イベントの2種類が存在する。単一イベントは各1つずつのセンサ要素と閾値，関係演算子により定義される述語である。例えば室温が100℃以上であればスプリンクラーを起動するという火事検知アプリケーションならば必要なのは「100℃を超えた」というイベントの真・偽の情報なので，そのイベントを $P(x)$ とすると $P(x) = temperature > 100$ と表現できる。

一方で複合イベントは単一イベントよりも条件が複雑化したものであり， $E = P_1(x) \wedge P_2(x)$ のように単一イベントと論理

演算子で構成される。より複雑なイベントを定義する場合や，情報の精度を向上させる場合に用いられる。例えば，火事検知にしても温度のみを利用すると料理や暖房器具を誤検知するなど，情報の精度が低い可能性がある。

### 2.2 従来の研究とその課題

既存の研究として，ワイヤレスセンサネットワーク(以下WSN)内におけるイベントツリーを利用した複合イベント検知のフレームワークがある[1]。複合イベントを構成する単一イベントを木構造として管理し，データ収集時に単一イベントが検知されれば木に従って親デバイスにイベント情報を送り，親デバイスは要求された複合イベント条件にマッチするかを判定する。これを繰り返し最終的に根に到達すれば複合イベント検知となる。これを発展し，複合イベントの検知に必要なセンサデバイスをBFSライクな手法で検索して深さが小さくなるような木構造を作成し，更に送信する内容が1bitのフラグのみすることでイベント通知の時間短縮を実現した研究もある。[2]

従来の研究は単一WSN内での複合イベント検知しか想定していない点が問題として挙げられる。まず複合イベントに異種SNWの単一イベントを利用できないというのが課題である。SNWには標準的なプロトコルがなく，各ベンダが独自の通信方式を採用しており相互通信ができない。これにより様々な通信プロトコルで動作する異種SNW間での複合イベントを検知することができないため，検知可能な複合イベントの種類やその利用用途が限定的なる。将来的には各地に偏在する多種多様なセンサデバイスが移動し，混在する環境が生まれると考えられる。例えば，家屋という小さな空間にも空調管理器機用SNWやセキュリティ器機用SNW，物品管理SNWなど様々なSNWが混在することが予想される。このときこれらの情報を個別に管理・利用していたら非常に使い勝手の悪いものとなる。このように，複合イベント検知は単一だけではなく複数の異種SNW上でも機能するべきである。

また，単一WSNでは大規模化が難しいというスケーラビリティの課題もある。例えば，我が家と他県に別居中の老いた両親の家の両方にホームセキュリティ用SNWを完備し，イベント情報を一元管理したいというニーズがあるかもしれない。このように複合イベントの検知は特定エリアや特定ネットワーク内のみに限定されるべきではなく，規模に関わらず機能させるためにスケーラビリティが考慮される必要がある。

## 3. 複合イベント検知システム

先に挙げた従来の研究の課題の踏まえ，ここでは筆者らの提案する，ユーザや多種多様なセンサデバイスが移動し，混在する動的且つグローバルなネットワークにおける複合イベント検知システムについて述べる。まずこれを実現するために必要だと考えられる要求条件を整理した後に，システムの概要を説明する。

### 3.1 要求条件

**Mobility** ユーザとセンサデバイスの双方の移動を想定しているため，現存するどのSNWに対しても，ユーザは任意の場所からイベント検知サービスを実行することができる。また，

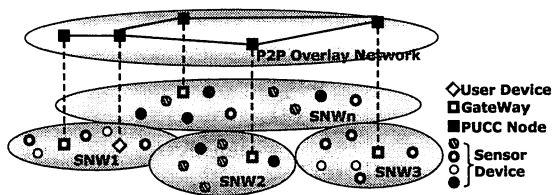


図 2 システム図

ユーザがイベント検知サービスを SNW に登録していた場合、イベント登録時の位置と異なる場所に移動しても、適切な経路を通りユーザにそのイベント内容を通知する仕組みが必要である。更に、センサデバイスの移動によりイベント検知が継続できなくなった場合には速やかにアプリケーションに通知する。

**Scalability** 本イベント検知システムは単一の SNW 内のみにて適用されるものではない。複合イベントが遠隔地にあるいくつかの SNW にて検知される単一イベントの組み合わせである場合も想定している。よって、センサネットワークの規模や数に関わらず、イベント検知が行われる必要がある。

**Network Independency** 現在 SNW の通信プロトコルには標準が存在しないため、zigbee 以外にも各ベンダーが独自の通信プロトコルを採用している。よって本イベント検知システムではヘテロジニアスな SNWs に対しても適応可能にする。

以上の要求に対し、本研究ではヘテロジニアスな SNWs を管理するセンサ GW を利用することで実現する。このセンサ GW はデータセントリックの概念に基づいて設計される必要がある。なぜなら、様々な通信プロトコルを利用するヘテロジニアスな SNWs やセンサデバイスを抽象化することでセンサデバイスの移動や故障によるネットワークトポロジの変化や、イベント検知に利用されるセンサデバイスの存在を隠蔽できる。これによりイベント検知サービスの継続性を向上させるとともに、Network Independency を満たすことができるからである。

任意の場所に存在するユーザから複合イベント検知の要求を受けたセンサ GW は複合イベントを構成する単一イベントを検知する SNW を管理するセンサ GW と連携し、必要な単一イベント情報を収集し複合イベントを判断する。なお、この動作は SNW の数や規模に関わらず行われるため Scalability を保証する。そして複合イベントだと判断された場合は適切な通信プロトコルを利用し遠隔ユーザにその情報を通知することができる。これにより Mobility の要求を満たすことができる。

### 3.2 システムモデル

ここでは、本複合イベント検知システムを構成する要素と通信について述べる。

#### A. 通信

本システムでは GW がユーザにイベントを通知する手段とし

て Pucc プロトコルを採用している。ここで Pucc について簡単に紹介する。標準化団体 Pucc(P2P Universal Computing Consortium) とは既存の様々な器機やネットワークをオーバーレイネットワークにより相互接続し、シームレスに利用可能とするデジタルコンバージェンス技術の開発・標準化を行っている団体である。オーバーレイネットワークを構築する手段として、Pucc は JAVA でコーディングされたミドルウェアを提供しているため、このミドルウェアを搭載したノードは Pucc プロトコルを用いてオーバーレイネットワークに参加できる。また、Pucc プロトコルはデバイス発見やサービス実行などの各種 API も提供している。Pucc プロトコルを用いることでユーザデバイスは IP ネットワークなどの下位リンクを左右されず、オーバーレイネットワークに参加する限りどこに移動しても継続して GW からのイベント検知サービスが利用可能である。

#### B. 構成要素

**ユーザデバイス** Pucc プロトコルを採用しているため、Pucc ノードとなる。このため、ユーザアプリケーションは Pucc プロトコルのミドルウェア API の提供する各種メソッドを利用して任意の場所からセンサデバイス発見、イベント検知サービスを実行可能である。

**センサ GW** 役割は SNW の管理とサービスの提供である。自身の担当する SNW についてデータをデータベースに蓄積するとともに、複合イベント検知などのセンササービスを提供するのが役割である。また、Pucc プロトコルを採用し、User Device と同様 Pucc ノードとなる。このため、ユーザデバイスと同様 Pucc プロトコルのミドルウェア API の提供する各種メソッドを利用可能である。

**センサデバイス** 温度や湿度などの実環境を定量化するセンシング機能と他のセンサノードや GW との通信機能を備えた小型機器である。

**Pucc ノード** Pucc プロトコルを採用したデバイスで、隣接ノードと相互接続し自律的に P2P オーバーレイネットワークを構築する。また、自分のプロパティを記載したメタデータを保持する。

**メタデータ** Pucc ノードに関する各種情報を記載した XML ファイルであり以下の 3 つの要素が記載されている。specification 要素にはデバイスの名前や位置、製造番号などのデバイス自体に関する諸情報が記載される。次に、電源や動作モードなどデバイスの状態情報を記した StateVariable 要素がある。そして、イベント検知などのデバイスが提供可能なサービスの内容がリスト上に記載される serviceList が存在する。また、デバイスがサブデバイスを保持していた場合、デバイスのメタデータはサブデバイスのメタデータを含む。GW が管理する SNW のデバイスはサブデバイスにあたるため、GW のメタデータに SNW の情報が記載される。もしデバイスに何らかの変更が生

### Algorithm : Composite Event Detection

Here E = composite event (like  $P1(x) \cap P2(x) \cup \dots, Pn(x)$ )  
Here  $Pn(x)$  = atomic event that forms E (like temperature > 30°C)  
Here T = event Tree  
Here  $\supset$  is a set operation  
Let D be the composite device  
Let n be the total number of P(x)

#### Registry Phase

Require:Arrival of invoke Message to create composite device  
Create URI and stateVariable for newD  
Send URI for newD  
Require:Arrival of Subscribe Message to D  
execute parse and create T(E)  
for i=1 to n do  
If myManagementDevice(Pi(x)) == true then  
Registry(Pi(x))  
Start sensing  
else  
Send subscribe(Pi(x)) to otherGateWay( $\supset$ Pi(x))  
end if  
end for

#### Aggregate Phase

Require:receive data from Sensor  
for i=1 to n do  
if match ( Pi(data) ) == true then  
Pi(x)'s flag = true  
if detection(E) == true then  
send Notify to user  
end if  
end if  
end for  
Require:Notify Pj(x) from otherGW  
if detection(E) == true then  
send Notify to user  
end if

図3 イベント検知システムの検知アルゴリズム

じた場合はそれに従いメタデータを更新する。

### C. システム概要

本システムを図2に示す。本システムでは PUCG ノード間で P2P オーバレイネットワークが構築され、デバイス発見やサービス実行メッセージがやりとりされる。

単一イベントを検知する SNW の情報を発見するために、ユーザはセンシング要素の検索条件を指定しクエリメッセージをブロードキャストで送信する。なお、このメッセージにはセンシング要素だけでなく、状態やサービス内容などの検索条件を含むことが可能である。GW はこのメッセージを受信したら、ユーザが指定した検索条件とメタデータに記載される自身の情報をマッチングし、該当したらメタデータをユーザに返信する。

ユーザは受け取ったメタデータを元に任意の SNW の情報を得ることができる。また、これら SNW のメタデータの情報から各 SNW で検知できる単一イベントとそれらを組み合わせた複合イベントの定義が可能となり、サービス実行メッセージを利用して任意のデバイスに対する複合イベント検知が実現できる。

## 4. 実装

### 4.1 イベント検知アルゴリズム

イベント検知システムのアルゴリズム (図3) は非常にシンプルである。

まず、登録フェーズについて述べる。ユーザアプリケーションはまず任意のノード (ここではセンサ GW を指す) を選択し、

### 例)複合イベント条件式

((単一イベント1@GW1) & (単一イベント2@GW2)) || (単一イベント3@GW3)

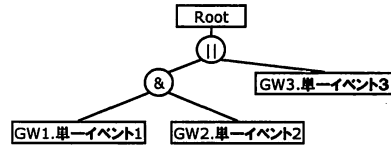


図4 イベントツリー構築例

Create Composite Device サービスを実行する。このサービスは仮想的なデバイスを作成し、ユーザにその URI を提示する。次にあらかじめ得た任意の SNW のメタデータの情報を元に単一イベントを組み合わせて複合イベント条件式を作成し、先に選択したデバイスに対して複合イベントの条件式を含めたサブスクライブメッセージを送信する。これにより先ほどの仮想デバイスはそのアプリケーションより提示された独自の複合イベントを検知できるデバイスとなると共に、イベントが発生したときにはアプリケーションに対してノーティファイを行うことができる。

メッセージを受信したノードは筆者らの開発したセンサ GW 用ライブラリの提供する API を利用して、構文解析とイベントツリー (図4参照) の作成を行う。イベントツリーは複合イベント、単一イベントと論理演算子で構成され、それぞれ根、葉、節に対応する。イベントツリーの全ての葉を調べ、単一イベントが自身の管理する SNW で検知されるものであれば、PUCG センサ GW 用ライブラリに単一イベントを登録する。もし他のセンサ GW で管理する SNW で検知されるべきものであれば、そのセンサ GW にサブスクライブメッセージを送信する。(ちなみにサブスクライブメッセージを送信したノードが受信したノードの親となる)

次にイベント収集フェーズについて述べる。まず、親ノードは自身の管理する SNW においてデータが通知された場合はまず、各単一イベントの条件を満たすかどうかを判断する。もし満たしていれば、次に複合イベントを満たすかを判定する。子ノードからのノーティファイを受信した場合はそのつど複合イベントを満たすかを判定する。なお、これらの複合イベント判定は作成されたイベントツリーに沿って行われる。そして複合イベントを満たすと判断された場合は、購読先のアプリケーションにノーティファイを行う。

### 4.2 通信

ここでは、システムの実現に必要な以下の4つの通信に関するメソッドについて述べる。

**Device Discovery** ユーザがイベントを定義するためにはデバイスの種類や搭載センサなどの SNW に関する情報が必要となる。これらの SNW の情報を得るために Device Discovery メソッドを利用する。このメソッドによりユーザは他の PUCG ノードに対してセンシング要素や位置座標などの検索条件を含めたメッセージをブロードキャストし任意の SNW の情報を得

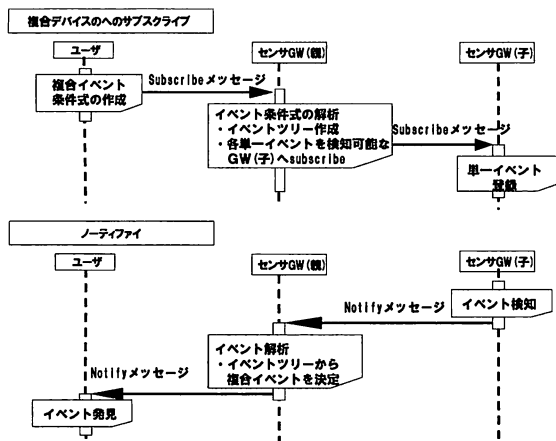


図 5 複合デバイスに対するサブスクライブとノーティファイ

ることができる。

このメソッドは Discovery メッセージと Discovery Response メッセージで構成される。Discovery メッセージ内に検索条件を指定することができ、P2P オーバレイネットワーク上の PUCG ノードにブロードキャストすることができる。Discovery メッセージを受け取り、条件にマッチしたノードは自身のメタデータを含めた Discovery Response メッセージを返信する仕組みになっている。

**Service Invoke** PUCG プロトコルの Invoke メソッドは、複合イベントを検知できる仮想的なデバイスである複合デバイスを新たに作成するときに用いられる。このメッセージは Invoke メッセージと Invoke Response メッセージにより構成される。予め得たメタデータを元に対象ノード (センサ GW) 名とサービス URI を指定し、Invoke メッセージを P2P オーバレイネットワーク上の対象 PUCG ノードにユニキャストで送信する。上記メッセージを受け取ったノードは内部でデバイス情報や状態変数、サービスを含めた新規複合イベントの定義・作成を行い、その複合デバイスの URI を Invoke Response メッセージに含めて返信する。

**Subscribe & Unsubscribe** 新規作成した複合デバイスや他のイベントにサブスクライブを掛ける場合は PUCG プロトコルの Subscribe メソッドを利用する (図 5 参照)。このメソッドは Subscribe メソッドと SubscribeResponse メソッドで定義される。予め得たメタデータを元に実行ノード名と対象となるデバイスの URI を指定し、Invoke メッセージを P2P オーバレイネットワーク上の対象 PUCG ノードにユニキャストで送信する。Unsubscribe も同様に定義され、実行ノード名と subscribeID を指定することで実行できる。

**Notify** PUCG プロトコルの Notify メソッドを利用することで、Notify メッセージを作成しイベントの発生をユーザに通知することができる (図 5 参照)。Notify が行われるのは以

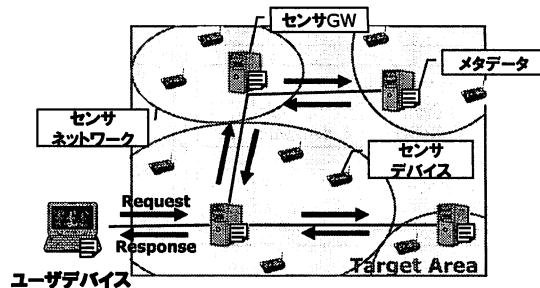


図 6 プロトタイプアーキテクチャ

下の 2 つのケースを想定している。まず、ユーザの指定したデバイスの状態が変化してイベントが発生した場合である。もうひとつは、ユーザの指定した複合デバイスを構成するデバイスが故障あるいは移動などによりイベント検知ができなくなった場合である。この 2 つのいずれのケースもアプリケーションは PUCG センサ用ライブラリのメソッドを利用して知ることができる。そして変更内容を Notify メッセージを用いてユーザに連絡する。これにより、センサデバイスが移動又は故障しイベントを検知できなかった場合でもこの情報を元にアプリケーションは代替センサの選出など速やかにその対処をすることができる。

### 4.3 プロトタイプ

前章までに述べた本システムを構成するアプリケーションやライブラリは PUCG プロトコルを含め全て JAVA を用いて実装している。このためプラットフォームに非依存のシステムとなっている。まず、システムの概要図を図 6 に示し内容を説明する。

本システムはユーザデバイス、センサ GW、センサデバイスで構成されている。ユーザデバイスは PUCG プロトコルを搭載しており、プロキシを介さず直接 PUCG ノードとして P2P オーバレイネットワーク上で動作する。SNW は MOTE/Mica2 と NEC Tokin 社のセンサデバイスを採用しており、センサ GW はこれらのデバイスが通信に用いる独自のプロトコルをサポートしている。なお、これらのデバイスは PUCG プロトコルをサポートしない。代わりに PUCG プロトコルを用いているセンサ GW が、センサデバイスの仮想 PUCG ノードとして P2P オーバレイネットワークに参加する。これはメモリや演算処理能力などの制限により SNW を構成するセンサデバイスは PUCG プロトコルを搭載できないためである。また、これによりセンサネットワークのプロトコルは隠匿され、P2P オーバレイネットワークを介して多様な SNW の相互接続を実現する。なお、GW デバイスは他にもセンサアプリケーションのために多様な機能を有しており、今回は複合イベント検知に関するものを次に紹介する。

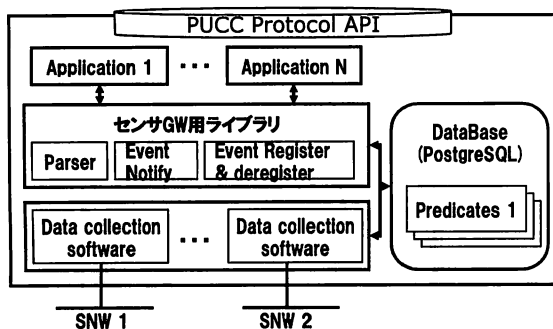


図 7 センサ GateWay

#### 4.4 センサ GateWay

センサ GW の概要を図 7 に示し各機能を説明する。この GW は PUCG プロトコル、アプリケーション、センサ GW 用ライブラリ、データベース、情報収集ソフトウェアにより構成される。情報収集ソフトウェアはベンダの提供するものを使用している。ここでは、複合イベントの検知をサポートをするセンサ GW 用ライブラリについて述べる。

**Parser** この機能は GW デバイスが受け取った複合イベント条件を解析するためのメソッドである。これによりイベントツリーが構築され、葉に相当する単一イベントのフラグの管理や、複合イベントの判定に用いることができる。

**Event Notify** この機能はイベントだと判断された場合と SNW の構成に変化があった場合にアプリケーションにその情報を提供するメソッドである。

**Event Register & Deregister** これはローデータ (センシングされた値そのもの) がイベントかどうかを判断するメソッドである。本 GW では DataBase として PostgreSQL を採用しており、指定したアプリケーションから渡されたイベント条件を pl/pgsql 言語で書かれたストアードプロシージャに変換して DataBase に登録することで収集したローデータのイベント検知を実現する。また、各ストアードプロシージャは各イベントに割り振られるイベント ID と対応しており、このイベント ID を指定することでストアードプロシージャを解除できる。

### 5. おわりに

本稿ではセンサ GW を利用して、ユーザやヘテロジニアスなセンサデバイスが自由に移動し相互通信をする動的ネットワーク環境下において複数の SWNs の単一イベントを利用した複合イベントの検知を実現するシステムについて提案し、その開発について述べた。今後は、位置情報を考慮した複合イベント検知の研究を検討している。

#### 文 献

[1] A.V.U.P. Kumar, A.M. Reddy, V.D. Janakiram, "Distributed collaboration for event detection in wireless sensor networks," The 3rd international workshop

on Middleware for pervasive and ad-hoc computing, no11, Grenoble, France, November. 2005.

- [2] C.T. Vu, R.A. Beyah, and Y Li, "Composite event detection in wireless sensor networks," Performance, Computing, and Communications Conference, no9528783, pp264-271, New Orleans, USA, Apr. 2007.
- [3] E.M. Ould-Ahmed-Vall, G.F. Riley, and B.S. Heck, "A distributed fault-tolerant algorithm for event detection using heterogeneous wireless sensor networks", Decision and Control, 2006 45th IEEE Conference on, noThB18.5, pp3634-3639, San Diego, USA, December. 2006.
- [4] H. Sumino, Y. Uchida, N. Ishikawa, H. Tsutsui, H. Ochi, and Y. Nakamura, "Home application control from mobile phones", Consumer Communications and Networking Conference, pp793-797, Las Vegas, USA, January. 2007.
- [5] N. Ishikawa, T. Kato, H. Sumino, and J. Hjelm, "PUCC architecture, protocols and applications", Consumer Communications and Networking Conference, pp788-792, Las Vegas, USA, January. 2007.