

ホームネットワークにおいてセンサによる高度な イベント通知を実現するフレームワークの提案

松浦 知子[†] 北須賀 輝明^{††} 田頭 茂明^{†††}
中西 恒夫^{†††} 福田 晃^{†††}

通信機能を有する家電製品の登場により、ホームネットワークを利用することで様々な家電やセンサを有機的に連携させることが可能になっている。このようなホームネットワークにおいて、実世界の情報に応じた処理を行うコンテキストウェアネス技術の実現が望まれている。本稿では、ホームネットワーク上に分散しているセンサを利用して、コンテキストウェアネス実現に必要なコンテキストの変化を効率的に検出するフレームワークを提案する。具体的には、多様なセンサを抽象化し、センサを利用するアプリケーションに対して統一的なインタフェースを提供する。また、分散する複数のセンサ間で連携して、それらのセンサに関係する複合したイベントを効率よく検出する仕組みも提供する。

A Proposal of a New Framework for Advanced Event Notifications Provided by Sensors in Home Networks

TOMOKO MATSUURA,[†] TERUAKI KITASUKA,^{††}
SHIGEAKI TAGASHIRA,^{†††} TSUNEO NAKANISHI^{†††}
and AKIRA FUKUDA^{†††}

Many home electric appliances and sensors can be organically linked to each other through a home network due to the increasing communication capability of these devices. In such a home network, one of the important techniques is context awareness in which predetermined actions can be taken according to situations on real world called context. In this paper, we propose a new framework which can help to detect transitions of context based on information obtained by sensors distributed over the home network. More concretely, it provides a sensor abstraction for handling sensors in a uniform fashion and an efficient mechanism that can detect complex events concerned with sensing data on multiple sensors.

1. はじめに

コンテキストウェアネスアプリケーションは、センサを利用して、実環境の状況すなわちコンテキストを判断する。特に、コンテキストの変化に対応して、提供するサービスの開始・内容の変更・終了といった振る舞いを決定することが多い。したがって、コンテキストウェアネスアプリケーションではコンテキストの変化

を検出することが重要となる。コンテキストの変化を検出するためにセンサの測定データをどのように利用するかは、アプリケーションに依存するが、多くのアプリケーションが共通して必要とする処理も含んでいる。例えば、センサの測定データがある条件を満たしたというイベントを検出する処理などがある。

本稿で想定している環境は、様々なセンサと家電とからなるホームネットワークである。ホームネットワーク上でのセンサの利用形態としては、ネットワーク上に単体で存在するセンサをネットワーク内の他の機器から利用するという形態と、ネットワーク内の家電に内蔵されたセンサを他の機器からも共有して利用するという形態とが考えられる。本稿では、これらふたつの利用形態を考慮し、多様なセンサと家電とがネットワーク上に分散して存在し、必要に応じて連携する分散型のホームネットワークを想定する。

[†] 九州大学大学院システム情報科学府
Graduate School of Information Science and Electrical
Engineering, Kyushu University

^{††} 熊本大学大学院自然科学研究科
Graduate School of Science and Technology, Ku-
mamoto University

^{†††} 九州大学大学院システム情報科学研究院
Faculty of Information Science and Electrical Engineer-
ing, Kyushu University

多くのセンサが分散して存在する環境において、アプリケーションがセンサから実環境の情報を得るための仕組みとして、センサネットワークをデータベースとみなし、センサの測定データを要求するクエリをSQLのような形式で記述するという手法が提案されている¹⁾²⁾。これらの文献ではセンサの測定データに現れるイベントの検出も考慮されている。しかし、自然環境のモニタリングシステムのような数種類のセンサノードからなる大規模なセンサネットワークを想定し、多種多様なセンサからなるネットワークを意識したものではない。一方で、ホームネットワーク上でのコンテキストウェアネス実現を目指し、イベントとそれに伴って起動されるコンテキストウェアサービスの関係を表したルールを規定しておき、そのルールに則ってサービスを起動するという手法も提案されている³⁾⁴⁾。これらの文献は、分散したセンサデータがあるノードに集める集中管理型を想定しており、ネットワークのトラフィックや負荷分散の観点から効率的であるとは言えない。

本稿では、ホームネットワーク上に分散しているセンサを利用して、コンテキストの変化を効率的に検出するフレームワークを提案する。具体的には、多様なセンサを抽象化し、センサを利用するアプリケーションに対して統一的なインタフェースを提供する。加えて、分散する複数のセンサ間で連携して、それらのセンサに関係する複合したイベントを効率よく検出する仕組みも提供する。また、提案フレームワークのUPnP上での実装方針についても記述する。

本稿の構成は、以下の通りである。まず、第2章でコンテキストウェアアプリケーションとイベントの検出について述べる。第3章で本稿で提案するイベント通知フレームワークについて述べ、第4章で本フレームワークに則ったイベント通知機能を提供するセンサの設計を述べる。最後に第5章でまとめと今後の課題を述べる。

2. コンテキスト変化の検出

コンテキストウェアアプリケーションは、センサを利用してユーザの置かれている実環境の情報を取得し、コンテキストを推定する。しかし、センサにより得られた情報がそのままコンテキスト情報として利用できることは少なく、センサの測定データをコンテキスト情報に変換する処理が必要である。また、コンテキストウェアアプリケーションは、変換処理を施して得られたコンテキスト情報の変化に応じて、サービスの開始、内容の変更、終了を行う。よって、コンテ

キストの変化を検出する必要がある。本章では、センサの測定データとコンテキストの関係、コンテキストの変化を検出する処理、そして、本稿で想定する分散ホームネットワーク環境でコンテキストの変化を検出する際の問題点について述べる。

2.1 センサ測定データとコンテキスト情報の関係

センサの測定データとコンテキストの関係について考察する。センサの測定データからのコンテキストへの変換処理には、様々なものが考えられるが、代表的ないくつかの処理について説明する。

- (1) 測定データが取り得る値をいくつかの範囲に分け、現在の値がどの範囲内にあるかにより異なるコンテキストと判断する。
- (2) 現在の測定データだけではなく、ある期間の測定データの時系列変化をもとにコンテキストを判断する。例えば、一定時間よりも長くある範囲内の値が続くことがある特定のコンテキストを示すということが考えられる。また、少し前から現在までの測定値を利用して測定値の傾きを求め、これにより、測定値の急激な変化を捉えることも可能である。
- (3) 複数のセンサを利用し、各センサにより得られた情報を複合的に判断して、コンテキストを推定する。

これらの関係を具体的な状況に当てはめて説明する。机の上に設置された焦電赤外線センサ⁵⁾を利用して、その机に人が着席しているか否かという機のコンテキスト情報を得る場合を考える。このコンテキスト情報は、仮にその机を使う人が決まっている場合は、その機の持ち主のコンテキストとみなすこともできる。焦電赤外線センサは、自動ドアなどに利用されている人感センサである。著者らの行った実験により、人が着席しているか否かは、机上の焦電赤外線センサが人を検知しているか否かと、必ずしも一致しないということが確認された⁶⁾。これは、机に座り続けている人が検出されなくなる場合があるためである。焦電赤外線センサは、ある一定の距離以内にある周囲と温度差がある物体が移動したことを検出する。よって、人が焦電赤外線センサの前にも、大きな動作を伴わない作業を続けていると、検出されなくなることがある。このような状況では、焦電赤外線センサが人を検出している状態を着席、検出していない状態を席をはずしていると捉えるのではなく、例えば、(2)のケースとして、人を検出していない状態が1分以上続けば人は席についていないと判断したり、(3)のケースとして、椅子に装着した加速度センサなど他のセンサの測定データ

も合わせて考慮したりすることで正確な着席状況が得られると考えられる。

2.2 コンテキスト変化の検出

前節で考察したように、センサ測定データとコンテキストとの関係は多種多様であるが、コンテキストウェアアプリケーションでは、コンテキストの変化を検出する必要がある。すなわち、コンテキストの変化は、コンテキストに関連する測定データが何らかの条件を満たしたことを検出することになる。本稿では、この条件を満たしたことをイベントと定義する。また、前節で述べた(3)のケースのように複数のセンサの測定データを利用してコンテキストを推定する場合は、構成される測定データに関する複数のイベントを組み合わせて単一のイベントとして取り扱う。例えば、温度センサ、人感センサ、およびエアコンがホームネットワークにより接続された環境を考える。エアコンは、温度センサの測定値が27度以上であれば、その部屋のコンテキストを「暑い」と判断する。また、人感センサの測定値が人を検知していれば、その部屋のコンテキストを「人がいる」と判断する。そして、その部屋のコンテキストが、人がいてかつ暑いというコンテキストに変化すると、冷房サービスを開始するものとする。このコンテキストの変化は、温度センサの測定値が27度以上という条件を満たしたというイベントと、人感センサの測定値が人を検知したというイベントを組み合わせたイベントにより検出できる。

2.3 イベントの検出方法

イベントを検出する方法としては、アプリケーションがネットワークを介してポーリングにより定期的にセンサの測定データを取得し、アプリケーション側でイベントを検出する方法と、センサ側で検出したイベントをアプリケーションに対して通知する方法の2つが考えられる。前者では、センサはアプリケーションからの要求に対して現在の測定データを返すような単純な機能のみをもっていけばよいが、イベントを検出するためには、アプリケーションは定期的に現在の測定データを取得し、アプリケーション側で目的のイベントを検出しなければならない。ポーリングをする場合は、ホームネットワーク上で多くの通信を要する。後者では、センサはアプリケーションからのイベント通知要求を受け付け、自身の測定データに対して条件判定を行い、目的のイベントを検出し、要求元であるアプリケーションに対して通知する機能を持っている必要がある。しかし、センサ内部でイベントの発生を検出しイベントが発生したときのみセンサからアプリケーションへの通知が発生するので、通信量を削減で

き効率がよい。

センサ内部でイベントの発生を検出し、アプリケーションに通知するためにはセンサが高機能である必要があるが、今後発展が期待されるホームネットワークにおいては、多くのコンテキストウェアアプリケーションが、ネットワーク上の多種多様なセンサから情報を得て、サービスを提供することが予想される。アプリケーションに依存しない処理は、アプリケーションが各々行うのではなく、センサ側で行う方が開発効率の点から考えても有利である。上記の理由から本研究では、ポーリングによりイベントを検出するのではなく、センサ側でイベントを検出してアプリケーションに対して通知する手法に着目する。

2.4 イベント検出処理の課題

多種多様なセンサと家電が分散するホームネットワーク上で、イベントを検出するためには、いくつかの課題が存在する。以下に、分散ホームネットワーク環境でイベントを検出する際の問題点をまとめる。

- ホームネットワーク上に存在するセンサは、その種類や精度、提供ベンダなどが様々で、それぞれ扱い方が異なる。アプリケーションはセンサごとのインタフェースの違いを考慮して、センサを利用しなければならない。
- 複数のイベントを組み合わせたイベントを検出する際、アプリケーションは、それぞれのセンサに対して個別にイベントの検出を依頼し、得られた情報をもとにアプリケーション側で目的のイベントを検出しなければならない。効率的でない。また、アプリケーションにより検出したいイベントが異なるため、様々なイベントに柔軟に対応する必要がある。
- センサとアプリケーションがネットワークを介して分散していることから、イベントの検出に多くの通信量と計算量がかかる。

3. 提案フレームワーク

3.1 概要

コンテキストの変化は、センサの測定データが何らかの条件を満たしたというイベントを検出することにより検出できる。提案フレームワークでは、イベントの条件を測定データがある閾値を超えたという閾値超過条件に限定する。測定データとコンテキスト情報の関係を考えるとこの閾値超過条件のみでも十分にコンテキストの変化を検出することができる。センサの測定データがある閾値を越えたというイベントを、センサ側で検出し、通知する機能をセンサが提供

するものとする。

本稿では、このセンサによるイベント通知機能をフレームワークにより規定することで、ホームネットワーク上でのコンテキストウェアネス実現を目指す。すなわち、ホームネットワーク上のセンサがこのフレームワークに則ったイベント通知機能を提供すれば、コンテキストの変化を検出する処理の一部をセンサ側で行うことができ、アプリケーションの開発効率が上がる。また、すべてのセンサがフレームワークに則っていることにより、センサ間での連携が可能となり、アプリケーションに対して複数のセンサからなるネットワーク全体を抽象化したインタフェースを提供することができる。これにより、複数センサを利用するアプリケーションの開発効率も上がると考える。

提案フレームワークで提供する機能を以下に概説する。

- (1) **多様なセンサの抽象化** ネットワーク上のすべてのセンサが、アプリケーションに対して、共通のインタフェースを提供することで、アプリケーションは、センサの多様性を意識することなくセンサを利用できる。本フレームワークは、センサがアプリケーションに対してセンサ固有情報を開示する統一された仕組みと、センサがアプリケーションからのイベント通知要求を受け付ける統一された仕組みを提供する。
- (2) **分散するセンサの連携** アプリケーションが複数のセンサを利用する場合、アプリケーションが複数のセンサに個別にイベント通知を要求するのではなく、複数のセンサが連携してアプリケーションの要求に応える。
- (3) **実行効率** これらの仕組みは、センサとアプリケーション間の通信量をなるべく少なく抑えながら、また、センサやアプリケーションが動作する機器の消費電力をなるべく低く抑えながら実現されなければならない。

3.2 イベント通知フレームワークの設計

前節で述べた提案フレームワークの機能の詳細について記述する。

3.2.1 多様なセンサの抽象化

ホームネットワーク上のセンサは、単体で存在することも、家電に内蔵された形で存在することもある。また、その種類や精度、提供ベンダなども様々で、それぞれ扱い方が異なる。このような多種多様なセンサを同一のインタフェースで扱える仕組みが必要である。センサのコンテキストウェアアプリケーションに対するインタフェースとしては、センサ固有情報の開示

とイベント通知要求の受け付けがある。

センサ固有情報の開示

コンテキストウェアアプリケーションは、自身が知りたい情報を得るために、適切な位置に設置された、十分な精度とサンプリング周期をもったセンサを選ぶ必要がある。そのために、センサはアプリケーションに対して自身の固有情報を開示する。本フレームワークに準拠するセンサは少なくとも、設置場所、精度、サンプリング周期を開示するものとする。

イベント通知要求の受け付け

コンテキストウェアアプリケーションは、センサに対して、興味のあるイベントを何らかの形で表現し、イベントの通知を要求する。本フレームワークに準拠するセンサはすべて、以下の統一された方法で表現されたイベント通知要求を受け付ける機能を提供しなければならない。

まず、イベントの条件は、アプリケーションが指定した閾値を超えたか否かという条件を組み合わせて表現するものとする。

閾値を設定する測定データとしては、センサの測定値に加えて、測定値の傾きも考える。この傾きの値についても、指定された閾値を超えたか否かを判断することにより、測定値の急激な変化、すなわち、コンテキストの変化を検出することができる。測定値の傾きは最小二乗法により求めるものとする。また、閾値の設定は時刻に対しても行えるものとする。時刻を指定した閾値を超えた場合の通知では、測定値または測定値の傾きを通知する。

値を指定して閾値を設定する場合も、時刻を指定して閾値を設定する場合も、値または時刻を直接指定して設定する方法と、始点・終点・間隔を指定して等間隔で設定する方法とが考えられる。また、測定値や測定値の傾きを閾値とする場合は、その閾値ををどちら向きに越えたことを条件とするかも指定する。時刻の閾値の場合は、閾値を越える向きの指定は必要ない。

さらに、測定値、測定値の傾き、時刻が閾値を超えたか否かという条件を単位条件とし、複数の単位条件の論理積や論理和も条件としてを扱うことで複雑な条件も表現できる。各単位条件はネットワーク上の異なるセンサの測定データに関するものを指定することができる。例えば、「湿度センサの測定値が40パーセント以上 かつ 温度センサの測定値の傾きが5度/分」といったイベント通知条件が表現できる。

また、アプリケーションは、イベント通知要求の際に、どの程度の即応性を求めるかも指定する。センサは、アプリケーションからのイベント通知要求を受け

付けると、自身の測定データを監視し、要求に合致するイベントが検出された場合は、要求元のアプリケーションに対して通知する。しかしこのとき、アプリケーションによって、即応性を求める度合いが異なることがある。ログを必要とするアプリケーションでは、センサが目的のイベントを検出してもすぐに通知するのではなく、後からまとめて通知する方が通知の回数を削減でき効率がよい。リアルタイム性を要するアプリケーションでは、センサは目的のイベントを検出したらすぐに通知することが求められる。

アプリケーションの即応性を求める度合いの指定は、まとめて通知するイベントの数によって行う。すぐに通知してほしい場合は、イベントを1つ検出したら通知するよう要求すればよい。即応性を求めない場合は、いくつかのイベントを検出したらまとめて通知するよう要求する。複数の単位条件からなるイベント通知条件の場合、このイベントの個数は、単位条件の論理積からなる条件の場合は全体で検出されたイベントの数を、単位条件の論理和からなる条件の場合はそれぞれの単位条件ごとに検出されたイベントの数を表すものとする。

単一の単位条件からなるイベント通知要求の場合、アプリケーションは、利用するひとつのセンサに対して要求を出せばよい。複数の単位条件からなるイベント通知要求の場合は、その条件中に出現するセンサのうち任意のひとつに対して要求を出せるものとする。イベント通知要求を受け付けたセンサは、要求内容を解析し、複数の単位条件からなるイベント通知条件の場合は、他の関係するセンサに要求があったことを通知する。このとき、要求されたイベント通知条件とまとめて通知するイベントの数も同時に通知する。

アプリケーションが要求をキャンセルする場合は、初めに要求を出したセンサに対して、キャンセル要求を出すものとする。複数の単位条件からなるイベント通知要求に対するキャンセルを受け付けたセンサは、他の関係するセンサに要求がキャンセルされたことを通知する。

3.2.2 分散するセンサの連携

センサは、アプリケーションからの要求を受け付けた後、自身の測定データに対して条件判定を行い、要求されたイベントを検出する。複数の単位条件からなるイベント通知条件の場合は、効率よく目的のイベントを検出するために、センサ間での連携の仕方を工夫する必要がある。

センサは、各単位条件ごとにその条件を判定するモジュールを用意するものとする。このモジュールが互

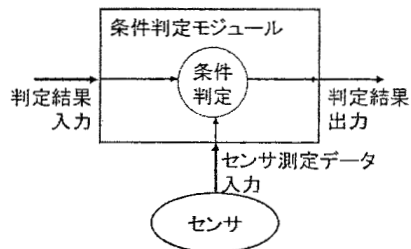


図1 条件判定モジュールの概念図

いに連携し、目的のイベントを検出する。条件判定モジュールの概念を図1に示す。各モジュールは他のモジュールの判定結果を受け取る入力と、センサから測定データを受け取る入力をもつ。また、判定結果を出力する。各モジュールは測定が行われる度に条件判定を行い、条件を満たしたら、その旨を出力する。この出力先は、アプリケーションから要求された条件が、単一の単位条件からなるか、複数の単位条件の論理積からなるか、複数の単位条件の論理和からなるかにより異なる。

単一の単位条件からなるイベント通知条件の場合、この出力が、要求元であるアプリケーションに対して行われ、イベント通知が実現される。

複数の単位条件の論理和からなるイベント通知条件の場合は、単一の単位条件からなるイベント通知条件の場合と同様に、それぞれのモジュールが個別に条件を満たしたことをアプリケーションに対して出力することでイベント通知を実現できる。

複数の単位条件の論理積からなるイベント通知条件の場合は、以下の手順でイベント通知を実現する。まず初めに、各単位条件の成立する頻度を過去の履歴から予想し、予想頻度の高いモジュールほどアプリケーション側になるように直列につなぐ。最も予想頻度の低いモジュールから条件判定を開始する。他のモジュールは休止している（図2(a)）。もっとも予想頻度の低いモジュールは、条件を満たせば、ひとつアプリケーション側のモジュールに対して、その旨を出力する。この出力を受けたモジュールは起きて、条件判定を開始する（図2(b)）。このように最も予想頻度の低いものから条件を満たしたときにはひとつアプリケーション側のモジュールに対して通知し、隣のモジュールを起こしていく。条件を判定中のモジュールよりアプリケーション側のモジュールは、条件判定中のモジュールの条件が成立するまで休止状態でいられる。最もアプリケーション側のモジュールは隣のモジュールからの通知により休止状態から起こされ、条件判定を開始

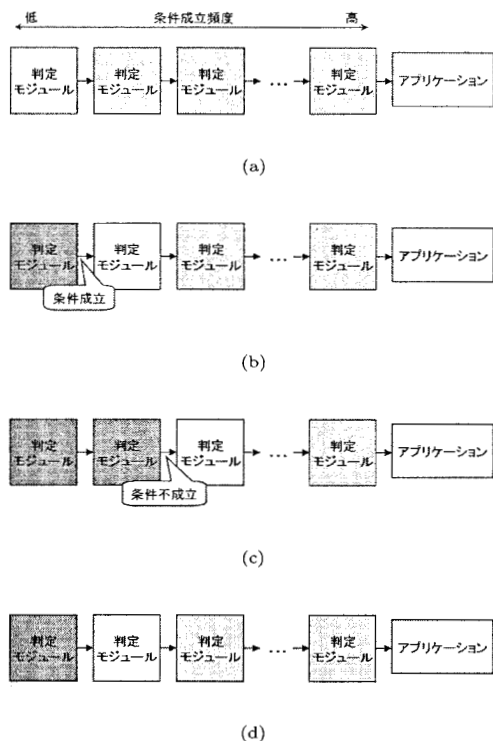


図2 複数の単位条件の論理積からなる条件でのイベント検出

し、条件を満たしたときには、アプリケーションに対して、その旨を出力する。これにより要求元のアプリケーションへのイベント通知が実現される。各モジュールは条件成立を通知した後は、条件が満たされなくなるのを判定する。実環境の状態が変化し、条件を満たさなくなった場合は、条件成立の通知を取り下げる必要があるためである。通知を取り下げたモジュールは再び条件判定を続け、それよりアプリケーション側のモジュールは再び休止状態に戻ることができる(図2(c)、図2(d))。このようなセンサ間の連携を可能とする枠組みにより、複数の単位条件の論理積からなるイベント通知条件の場合も、効率よくイベントの検出ができる。

4. 実装方針

本フレームワークに則ったイベント通知機能をもつセンサの設計について述べる。まず、一般的な設計について述べ、次に分散型ホームネットワークの構成を目指す技術である UPnP (Universal Plug and Play)⁷⁾

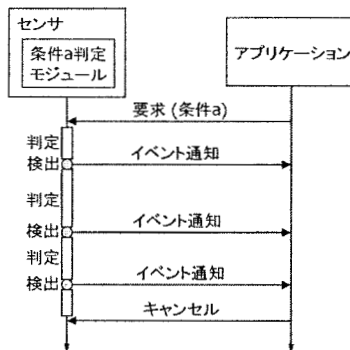


図3 単一の単位条件からなるイベントの検出

を使った設計について述べる。

4.1 イベント通知センサの設計

各センサは、自身に関する情報をネットワーク内のアプリケーションに開示する仕組みを提供する。各アプリケーションはこの情報をもとに利用するセンサを決定する。開示する情報は、センサの設置場所、精度、サンプリング周期とする。

アプリケーションからの要求は、センサの ID と、閾値の設定対象(測定値、測定値の傾き、時刻のいずれか)、閾値の値または始点・終点・間隔と閾値を越える向きをひとつの組とし、この組を AND または OR でつないで表現される。各センサは、要求を受け付けると、その内容を解析し、その内容に応じて必要な処理を行う。

単一の単位条件からなるイベント通知要求の場合の処理を図3に示す。要求を受け付けると、単位条件判定モジュールが条件判定を開始し、イベントが検出されるとアプリケーションに対して通知する。この条件判定は、イベント通知要求がキャンセルされるまで続けられる。

複数の単位条件の論理和からなるイベント通知要求の場合の処理を図4に示す。あるセンサが要求を受け付けると、そのセンサは、関係のある他のセンサに対して要求があったことを通知する。その後は、それぞれの条件判定モジュールが条件判定を行い、イベントを検出した際には、それぞれアプリケーションに対して通知する。

複数の単位条件の論理積からなるイベント通知要求の場合の処理を図5に示す。あるセンサが要求を受け付けると、そのセンサは、関係のある他のセンサに対して要求があったことを通知し、連携を要求する。そこで、それぞれの測定データの過去の履歴から条件が成立する頻度を予想し、予想成立頻度の高いものほど

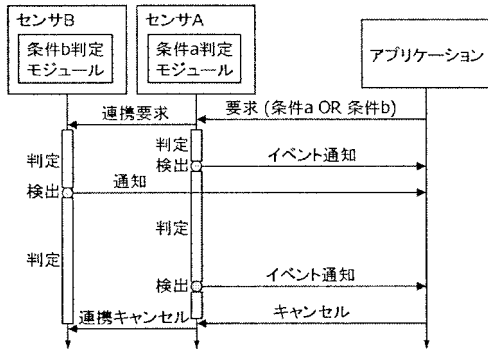


図 4 複数の単位条件の論理和からなるイベントの検出

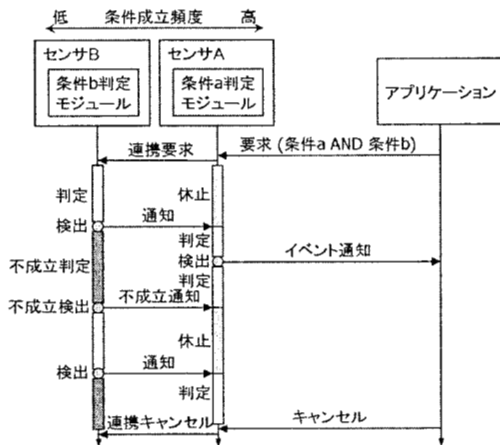


図 5 複数の単位条件の論理積からなるイベントの検出

アプリケーション側にくるように直列に連結する。図 5 では、条件 a の方が成立頻度が高いと予想されたとする。そこで、まず条件 b の判定モジュールから条件判定を開始し、条件 a の判定モジュールは休止している。条件 b が成立し、条件 a 判定モジュールに通知されると、条件 a 判定モジュールは条件判定を開始する。条件 a も成立すれば、条件がすべて満たされイベントが検出されるので、条件 a 判定モジュールからアプリケーションに対してイベントが通知される。条件 b 判定モジュールは、条件 b が成立し、条件 a 判定モジュールにその旨を通知した後、条件不成立の判定を開始する。条件が満たされなくなった場合は、条件 a 判定モジュールに対して通知する。これにより、条件 a 判定モジュールは再び休止状態に戻り、条件 b 判定モジュールは再び条件成立を判定する。

4.2 UPnP を使った連携

ここでは、前節で述べたイベント通知センサを

UPnP (Universal Plug and Play)⁷⁾ を使って実現する設計について述べる。

各センサは UPnP のサービスとして動作する。

各センササービスは、センサの設置場所、精度、サンプリング周期をネットワーク内のアプリケーションに開示する。UPnP により実装する場合、これらの情報はデバイスディスクリプション内に記述する。

アプリケーションからのイベント通知要求は、各サービスに対して行われる。複数の単位条件を含むイベント通知条件の場合は、その条件に含まれるセンササービスのうちのいずれかに対して要求する。要求を受け付けたセンササービスは、条件に含まれる他のセンササービスに対して要求があったことを知らせる。

各センササービスは、アプリケーションから要求を受け付けるか、他のセンササービスから要求があったことを通知された際には、各単位条件ごとにその条件を判定するモジュールを新たなサービスとして生成するものとする。もとのサービスを親サービス、新たに生成されたサービスを子サービスと呼ぶ。この子サービスは自身が判定する条件が成立しているか否かを表す状態変数をもつ。この状態変数は、イベント対応とする。また、AND を含むイベント通知条件のため、自身の担当する単位条件が他の単位条件と AND でつながっているか否かを表す変数と、サービスが直列につながれた際に自分より条件成立頻度が低い方の隣のサービスの条件が成立しているか否かを表すフラグを変数としてもつ。

単一の単位条件からなるイベント通知要求の場合は、その単位条件を判定する子センササービスを動的に生成し、アプリケーションに対しては、この子センササービスにサブスクライブするよう通知する。各単位条件ごとに新たにサービスを生成して対応することで、1 つのセンササービスを複数のアプリケーションから異なる条件で利用することが可能となる。

複数の単位条件の論理和からなるイベント通知要求があるセンササービスが受け付けると、そのセンササービスは、関係のある他のセンササービスに対して要求があったことを、各センササービスへの単位条件とともに、通知する。通知を受けたセンササービスは、この単位条件の判定を行う子サービスの ID を応答する。要求を受け付けたセンササービスも同様に子センササービスを生成し、要求元のアプリケーションにすべての子センササービスに対してサブスクライブするよう通知する。

複数の単位条件の論理積からなるイベント通知要求があるセンササービスが受け付けると、そのセンサ

サービスは、関係のある他のセンササービスに対して要求があったことを、各センササービスへの単位条件とともに通知する。通知を受けたセンササービスは、自身の測定データが通知された単位条件を満たす頻度を数値化して応答する。今回は、過去30分以内に当該の単位条件を満たした回数を応答することにする。また、この応答の際に新たに生成した子サービスのIDも通知する。各センササービスから応答を受けたもとのセンササービスは、単位条件の成立頻度をもとに、各子センササービスの連結の順番を決定する。決定した順番を各子センササービスに通知し、各子センササービスは自分のアプリケーション側ではない方の隣の子センササービスにサブスクライブして、その子センササービスの条件が成立した際に通知を受ける。この通知があるまでは、休止状態である。さらにアプリケーションからの要求を受け付けた親サービスはもともとアプリケーション側になる子センササービスにサブスクライブするよう、アプリケーションに通知する。

アプリケーションが要求をキャンセルする場合は、初めに要求を出した親センササービスに対して、キャンセルを要求する。複数の単位条件からなるイベント通知条件での要求をキャンセルする場合は、キャンセル要求を受け付けた親センササービスが、他の関係する親センササービスにキャンセルを通知する。キャンセルを受け付けた親センササービスおよびキャンセルを通知された親センササービスは、生成した子センササービスを削除する。

5. おわりに

コンテキストウェアアプリケーションは、センサを利用して、実環境の状況すなわちコンテキストを判断する。特に、コンテキストの変化に対応して、提供するサービスの開始・内容の変更・終了といった振る舞いを決定することが多い。よって、コンテキストの変化を検出することが重要となる。本稿では、コンテキストの変化を検出する処理の一部をセンサが行うためのフレームワークを提案した。本フレームワークは、多様なセンサを抽象化し、センサを利用するアプリケーションに対して統一的なインタフェースを提供する。また、分散する複数のセンサ間で連携して、複数のセンサの測定データに対する条件で表されるイベントも効率よく検出する仕組みを提供する。これにより、コンテキストウェアアプリケーション開発者の負担を削減できる。また、提案フレームワークのUPnP上での実装方針を示した。

今後は、本稿で述べたUPnPを利用したセンサの

設計に基づいて、UDSS⁶⁾を用いて提案フレームワークの実装を行う予定である。そして、簡単な測定値を提供する機能しかもたないセンサに対してポーリングにより目的のイベントを検出する場合と比較して本フレームワークの評価を行う予定である。

謝 辞

本研究の一部は、科学技術研究費および次世代研究スーパースター養成プログラム（九州大学総長裁量経費）による助成を受けている。

参 考 文 献

- 1) S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong: "The design of an acquisitional query processor for sensor networks", *In Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pp.491-502 (Jun. 2003).
- 2) P.Bonnet, J.Gehrke, and P.Seshadri: "Towards Sensor Database Systems", *In Proceedings of the Second International Conference on Mobile Data Management*, pp.3-14 (Jan. 2001).
- 3) 西垣 弘二, 安本 慶一, 柴田 直樹, 伊藤 実: コンテキストに基づいた情報家電の連携を実現するためのフレームワークおよびルールベース言語の提案, 情報処理学会研究報告 UBI, Vol.2004, No.112, pp.21-27 (Nov. 2004).
- 4) 川原 圭博, 司 化, 猪鹿倉 知広, 登内 敏夫, 森川 博之, 青山 友紀: 行動履歴と制約条件を考慮した情報家電制御機構, 情報処理学会研究報告 UBI, Vol.2006, No.14, pp.55-60 (Feb. 2006).
- 5) 松下電工株式会社 焦電型 MP モーションセンサ NaPiOn; <http://www.mew.co.jp/ac/control/sensor/human/napion/index.jsp>
- 6) Tomoko Matsuura, Kenji Hisazumi, Teruaki Kitasuka, Tsuneo Nakanishi and Akira Fukuda: "UDSS: Sensor Device for Context Awareness in Home Network", *In Proceedings of Fourth International Workshop on Networked Sensing Systems (INSS) 2007*, pp.196-200 (Jun. 2007).
- 7) UPnP Forum; <http://www.upnp.org/>