無線センサノード群による監視動作を実現するノードプログラムの自動生成

森 駿介† 梅津 高朗†‡ 廣森 聡仁†‡ 山口 弘純†‡ 東野 輝夫†‡
†大阪大学 大学院情報科学研究科 ‡独立行政法人科学技術振興機構, CREST
{ s-mori, umedu, ,hiromori, h-yamagu, higashino }@ist.osaka-u.ac.jp

1 はじめに

ワイヤレスセンサネットワーク(WSN)におけるアプリケーションの多くは，センサにより観測した周辺環境の変化をイベントとして補足し，それに対するデータ処理（モニタリング）を行う．モニタリングの実現のためには，WSN全体の振舞いやトポロジを想定しながら，各ノードのデータの送受信や演算などの振舞いをセンサノードプログラムとしてセンサ端末に合わせて（多くの場合）イベントドリブン形式で記述する方法が一般的である．しかし，このような個々のノードプログラムはあくまでノード単体の動作を規定するものであり，全体の振舞いを規定したい設計者には直観的でないため記述が困難で誤りを伴うことも多い．そこで本研究では，WSNのモニタリングプログラムの設計支援手法について述べる．提案手法では，センサノードの無線接続関係や位置関係を用いてセンサノード集合を指定できる関数群を提供し，設計者はそれらを用いて，WSN全体としてどのノード集合がどのような動作をどの順で行うかを指定する．これに対し，その動作を実現するセンサノードプログラムを自動で決定するアルゴリズムを述べる．

2 関連研究

WSNのノードプログラミングを支援する手法はこれまでにいくつか提案されている．

例えば，Abstract Regions [1] は通信の接続性や地理的条件などにより抽象的なノード集合定義を提供しており，設計者は実装レベルの通信，データ共有，収集などの詳細を抽象化したアプリケーション設計を行うことができる．また，Kairos[2] は個々のノードIDの管理，隣接ノードのグループ構成，任意ノードからのデータ取得といった機能を提供し，プログラムからこれらの処理を隠蔽することにより，ノード単位の形ではなくセンサネットワーク全体の振る舞いを記述することを可能とする手法である．

本稿で提案する手法は，WSNをノード集合としてモデル化している点でこれらの手法に近いアプローチであるが，一方で例えば「検知温度の平均値がある値以上となる10ノード以上の近接センサノード集合があれば，その周囲2ホップ以内のセンサノードの平均温度も検出する」といったように，センサノード集合の構成条件やその集合によるデータ計算，計算の逐次的実行など，プログラム言語やサービス記述言語に近く高い記述能力を持つ言語によるモニタリング要求記述が可能である．

3 モニタリング要求記述

まず本稿では，全てのノードが単一のプログラムにより動作し，自身の周辺ノードの情報しか持たない分散制御型のセンサネットワークを想定している．また，モニタリングとは個々のセンサノードの状態をネットワークを介して監視し，監視対象のうちの一定の範囲が特定の状態になったことを検知した場合に対応した処理を行うものであるとする．例えば，火災などによる温度の上昇を検知した場合，その位置などの情報を基地局へ送信する，といった処理が考えられる．本稿では，このようなモニタリング処理に対する要求を論理的かつ簡潔に記述するための記述言語について述べる．この言語では，モニタリングの動作仕様を，個々のノードではなくノード集合の状態や動作を元に処理の記述でき，具体的な通信やノード間協調の処理の詳細を隠蔽することでモニタリング機構の構築に伴う実装の労力や煩雑さを抑制することができる．

$$
\begin{aligned}
&S1\{\forall s1 \in S1\ s1.isActive \wedge (s1.battery < 0.4) \wedge setSize(S1, 1)\\
&S2\{\forall s2 \in S2\ s2.isNeighbor(s1, 2) \wedge (average(S2, battery) < 0.5)\\
&\qquad => s2.powerSaving()\}\\
&\}catch\{interval(100sec)\}
\end{aligned}
$$

図1: 記述例

モニタリング要求記述の例を図1に示す．これは，「動作がアクティブであるノードの電力残量が40%未満となった場合，周囲2ホップ以内のノードの平均電力残量も調べ，50%未満であればそのエリアの各ノードは省電力モードへ移行する」という動作を表す記述例である．このモニタリング動作により，電力残量が少なくなったエリアのパフォーマンスを下げ，消費電力抑制によるネットワークを延命を目的としている．モニタリング要求は，特定の条件を満たすノード集合が存在した場合に，そのノード集合に含まれるノードで実行される動作を規定すると言う形で記述する．まず1行目ではセンサノード集合（以下，単に集合とよぶ）S1とその存在条件が記述されている．具体的には，動作がアクティブ(変数isActiveが真)かつ電力残量(battery)が0.4未満のノード（そのそれぞれをs1で表す）からなり，含まれるノード数が1(setSizeにより指定)，となるノードが存在すればこれを集合S1とすることを規定している．2行目は集合S2とその条件である．これは，ノードs1から2ホップ以内の隣接ノード（isNeighbor(s1,2)が真）からなり，集合の各ノードの電力残量の平均値（average(S2,battery)）が50%未満という条件を持つ集合である．3行目はこの条件に該当するノード集合が存在した場合に行われる処理であり，集合S2に含まれるノードは省電力モードへ移行する．また，4行目はこの条件が真とならなかった場合の例外処理であり，100秒の間は集合S1の条件をチェックした各ノードが再チェックを行わない(関数intervalによる)ことが記述されている．

このように，モニタリング要求記述では所属するノードが満たすべき条件を定義することによりノードの集合を記述できる．条件としては，ノードの持つ変数やノードの位置に関する条件，集合に所属する各ノードの持つデータの平均値や最大値などのような集合全体に関する条件が記述できる．要求記述で用いることの出来る個々の条件

Automated Generation of Wireless Sensor Nodes' Programs for Monitoring Applications
†Shunsuke Mori ††Takaaki Umedu ††Akihito Hiromori ††Hirozumi Yamaguchi ††Teruo Higashino
†Graduate School of Information Science and Technology, Osaka University
‡Japan Science Technology and Agency, CREST

（リテラル）は，集合に属する各ノードに関するノード条件，および集合に関する集合条件の 2 タイプに大きく分けられる．ノード条件の場合は，ノードの持つ変数を用いた真偽値式，及びノードを引数とする領域関数などが利用可能である．集合条件は，集合を構築する際にサイズなどの制限を与える集合規定関数，構築した集合について所属ノード数や値の平均値などを得る集合判定関数がリテラルとして記述できる．例えば，集合規定関数を用いた"setMaxSize(S1, 10)"という記述を行えば，集合 S1 構築時にノード数を 10 以下に制限することができる．また，集合判定関数を用いれば，"average(S3, temperature)"という記述により，集合 S3 の各ノードが持つ変数 temperature の値の平均値を条件式に用いることができる．以上のように，対象となるノード集合の条件による絞り込みやネットワークトポロジや地理的な隣接関係に基づく周辺ノードへの拡張といった逐次的な処理を規定可能とすることで，WSN のモニタリングとして要求される様々な処理を容易に記述することができる．

4 分散環境における実行プログラムの導出

本章では，3 章で述べたようなモニタリング要求記述を元に，実際の WSN におけるモニタリング動作を実現する実行方針について述べる．なお，提案手法では各ノードが自分，隣接ノード，基地局の位置情報を持ち，他ノードの位置やノード総数は未知である WSN を想定している．また，相手の位置情報を持っている場合はノード間通信が可能とする．ノード集合は，制御や情報の集約を行うリーダノードとそれに付随するメンバノードを持ち，ツリー構造やクラスタ構造などからなるものとする．

モニタリング要求記述は集合とその存在条件の記述の連続から成り立っている．提案手法では，この要求記述から個々のノードの具体的なプログラムを自動的に生成する．各ノードは周辺の監視と必要に応じた周辺ノードと通信を行い，要求記述の先頭に規定された集合の構築を試みる．指定された集合の存在条件をもとにノード集合を構築し，条件の判定が真となった場合は，その集合からの通知に従って次集合の構築および判定を開始する．このような処理の繰り返しにより，要求記述で規定された集合を順に構築する．判定を行うべき存在条件は判定処理に伴う動作や実行すべきタイミングなどにより，内部変数の式など各ノードがそれぞれ単体で判定する局所条件，集合のサイズなどノード集合の構築時に与えるパラメータである集合構成条件，複数センサ値の平均など集合を構成してから集合全体で判定する大域条件へと分類される．この分類に従って集合の存在条件の判定を行う動作プログラムを導出する．

図 1 の記述例を元に出力したプログラム例（ただし詳細は割愛）が図 2 である．このプログラムは個々のノードで動作するものであるため，ノード間の通信処理やメッセージ受信などのイベント時の処理など，WSN 全体を対象とした記述であるモニタリング要求記述では隠蔽されている処理も含んでいる．プログラム中において，まず，各ノードが関数 checkLiteral，および関数 checkLocalCondition において局所条件の判定を行う．例えば"s1.isActive"などのノード毎に判断可能な局所条件がここに反映される．関数 checkLocalCondition において局所条件が真となったノードにより，関数 buildGroup を呼び出しノード集合を構築する．この際，所属ノード数の条件"setSize(S1,1)"などの集合構成条件をパラメータとして与える．集合の構築が完了したときにイベント関数 buildGroup.done が呼び出される．

```
checkLiteral(){
   lit[1] = isActive;
   lit[2] = (battery<0.4) ;
}
event Notification.receive.(msg){
   lit[4]= isNeighbor(msg.hop,2);
}
checkLocalCondition(){
   local[S1]= lit[1] && lit[2];
   local[S2]= lit[3];
   if (local[S1]) buildGroup(S1,1,1)else exception();
   if (local[S2]) buildGroup(S2, MAX, MIN);
}
event buildGroup.done(){
   if (role[S1]==LEADER) lit[3]=true;
   if (role[S2]==MEMBER)
      GlobalData.send( id(LEADER), battery );
}
event GlobalData.receive(msg){
   if (msg.type==S2){
   average=calcAverage(average, msg.id,
                       msg.battery,++num);
   if(num=num[S2]||timeout[S2])lit[5]=(average < 0.5);
   }
}
checkAllCondition(){
   Global[S1]=lit[1] && lit[2] && lit[3];
   Global[S2]=lit[4] && lit[5];
   if (Global[S1]) Notification.khopBroadcast(S1,2);
   if (Global[S2]) Execute.sendMember(S2);
}
exception{
   interval(100);
}
event MonitoringTimer.fired(){
   checkLiteral();
   checkLocalCondition();
   checkGlobalCondition();
}
```

図 2: プログラム例

ここで，"average(S2,average)"のような大域条件の判定を行うため，集合に属するメンバノードは集合の統率を行うリーダノードへ判定に必要な"battery"の値を集約し，リーダノードは値の平均値が 50%未満かという大域条件の判定を行う．最終的に，条件式全体を関数 checkAllCondition で判定し，真であればこの集合の成立が真となる．集合 S1 の成立時には集合 S2 を判定するため，S1 から 2 ホップ圏内へ通知を送信する．集合 S2 の成立時はリーダノードがアクション実行を関数 Execute により通知し，受信した各ノードが実行する．

5 まとめと今後の課題

本稿ではノード協調によるモニタリング機構の導出手法について述べた．現在，その記述を実現する各センサノードの振舞いを通信量や遅延の考慮に基づき適切なノード集合の構築や集合間の通信の手法を自動で決定し，モニタリングに伴うコストを削減する手法について検討している．また，我々が開発している WSN 開発支援システム D-sense[3] へ本手法を導入することも検討している．

参考文献

[1] Matt Welsh et al. Programming sensor networks using abstract regions. In *Proc. of NSDI*, pp. 29–42, 2004.

[2] Ramakrishna Gummadi et al. Macro-programming Wireless Sensor Networks using Kairos. In *Proc. of DCOSS*, pp. 126–140, 2005.

[3] 森 駿介 他. ワイヤレスセンサネットワークの設計開発支援環境 D-sense. 情報処理学会論文誌, Vol. 50, No. 10, pp. 2556–2567, 2009.