論文

異機種間接続センサネットワークにおける安全なビヘイビア定義システム

大林真人*1)　高野哲寿*1)　横田裕史*1)　大畑敏美*1)

Design of Secured Behavior Description System for Heterogeneous Sensor Network

Makoto OBAYASHI, Norihisa TAKANO, Hiroshi YOKOTA and Toshimi OOHATA

Abstract　In this paper, we develop a multi-agent language and its execution environment that enables us to easily define sensor network behavior. Using sensor network and various devices, it is easy to form a ubiquitous computing environment. However, it requires bidirectional communication between nodes autonomous control by each sensor node to determine the intercommunication suitable for its situation, and recognition and secure cooperation between unspecified numbers of sensor nodes in the heterogeneous network to provide the various system services. To solve these problems, we developed a multi-agent system capable of execution in a sensor node with limited computer resources. We then demonstrate the applicability of our system through the obtained experimental results.

Keywords　Ubiquitous computing, Sensor network, Multi-agent, Security

１．はじめに

近年,センサネットワークの可能性や重要性が注目されている。センサネットワークとは，センサと無線通信機能を持った自律動作する多数のノードによって構成されるシステムである[2)]。一般的に，ノードとなるハードウェアは低消費電力性能を重視して設計されており,非常に限定されたリソースで動作する。このとき，無線通信においてアドホックネットワークによるルーティングプロトコルを使用することにより,多種多様なセンサを搭載したノードを簡単にネットワークに参加させ,システムを構築することが可能となる。個々のノードは小型かつ安価であるため，屋外や危険地域において，それらを適用することも可能である。また，環境および人体への取り付けも違和感無く実現することが容易であり,多様なセンサ技術とセンサネットワークを併用することによって,多様な情報を収集し，様々なサービスを実現することが可能となる[4) 5)]。ここで，システムの使用者が，センサ情報を取得するだけであるならば,センサネットワーク内の特定のステーションへデータをルーティングするだけでよい。しかしながら，使用者が能動的にセンサネットワーク内の特定ノードに対してサービスを要求する場合には双方向性が必要とされる。また，異なる機能を持ったノードによって構成されるヘテロジニアスな環境では,実世界のイベントに応じたノード間の相互処理を実現するために,動的に増減する不特定多数のノードを正しく認識させ,協調や競合の解消を行うためのフレームワークが必要となる。さらに，環境情報や個人情報が無線によるアドホックネットワークで送受信される場合,そのデータは不特定多数のノードを経由して伝達される。このとき，悪意のある攻撃者によるデータの盗聴や改ざんの危険にさらされることとなる[1)]。

これに対して，我々は，上述した問題を解決することを可能とするセンサネットワークのビヘイビア定義システムを開発する。本研究によって開発されるシステムは，自律ロボットや分散デバイスの動的協調システムとして開発されたマルチエージェント言語[6)]におけるフレームワークを用いることによって実現される。本システムは動作記述言語および処理系として実装され,センサノードの簡便な動作定義を可能とする。また，自律分散型無線ネットワークによって構築されるセンサネットワーク上の通信パケットに対して,暗号化および認証の機構を導入することにより，各種同期処理および分散サービスの解決，安全な相互通信を実現する。そして，評価実験を通じて，本システムの有効性を検討する。

２．設計方針

2.1　動作記述言語

本研究において開発されるマルチエージェント記述言語 TinyMRL(Tiny Multi-agent Robot Language)では，各セ

*1) 情報科学グループ

ンサノードがエージェントとして定義され，その動作は述語の集合によって構築される。各述語は，GHC (Guarded Horn Clause)による記述形式と同様に，述語名，条件節，実行節から構成される。述語は，述語名および引数の数によって分類される。すなわち，同じ述語名と同じ数の引数を持つ全ての述語は，同一のグループに分類され，同じグループの述語が同時に実行されることは無い。呼び出される述語は，条件節における定義式が，呼び出し元による引数の値およびセンサノードの内部状態が一致するものだけが呼び出される。ここで，条件に適合する述語が複数存在する場合には，条件節内における定義式の数によって優先度が設定され，最も多い条件を持つ述語が最高優先度として処理される。また，最高優先度となる述語が複数存在する場合には，その中から一つの述語が無作為に選択される。これにより，各述語の実行節の最後に自身の述語を再帰的に呼び出すことによって，その同じグループに属する述語集合で構成される一連の処理が継続される。すなわち，TinyMRL による述語構成により，容易に状態遷移を構築することが可能となる。同じ述語グループに属するものは，同一の状態を表現し，再帰的に自己を呼び出すことによって，同じ状態での動作を保持することができる。また，他のノードからの通信や各種のイベントにしたがって，自己を他の状態へ遷移させるときには，条件節に特定のイベントに適合するルールを記述した述語を定義し，実行節の最後に他のグループに属する述語の呼び出しを行う。状態の動作の終了は，実行節内部での述語呼出を行わずに述語定義を記述することによって実現できる。このとき，述語の実行の終了と同時に一連の状態遷移動作は完全に終了する。図 1 は，他のノードからのメッセージ受信待ち動作を行う記述例であり，start から呼び出される run および action の各述語は並列に実行される。また，TinyMRL 処理系は，分散サービス解決におけるファシリテータを介した様々なエージェント動作をシステムコールとして実装しているため，自律的な協調動作を簡便に記述することが可能である。

2.2 セキュリティシステム

TinyMRL 処理系が提供するセキュリティ機能は，データの暗号化による傍受の防止，セマンティックセキュリティの実現，データ認証である。前提として，開発者自身が設置した全てのセンサノードは信頼することが可能であり，悪意のある攻撃者によるクラッキングの試みは，他のノードによるアドホックネットワークへの参加を通じて行われるものとする。データの暗号化および復号化，データ認証に使用される全ての鍵は，信頼できる全てのノードが共通して所持する秘密鍵から生成される。TinyMRL によって記述されたエージェント $Agent_a$ が，他の特定のエージェント $Agent_b$ への通信を行うための以下のシステムコールを実行すると，エージェントの動作は TinyMRL 処理系へと移行する。

```
00 : start() :- true {
01 :   sys:advertise(property(...));
02 :   run(@wait);
03 :   action(@reply);
04 : }

05 : run(atom state) :- state == @wait {
06 :   /*  do Actions */
07 :   run(@wait);
08 : }

09 : action(@reply) :- sys:recvMsg() {
10 :   /* do Actions */
11 : }
```

- Predicate 1
  It is the predicate for start up.

- Predicate 2
  The predicate of the loop for waiting.

- Predicate 3
  The predicate is launched when the message is received from other agents.

図 1 センサノード動作の記述例

図 2　本研究で使用されるセンサノード

$$\text{syscall:sendMsg}(Agent_b, \langle\text{Message}\rangle) \quad (1)$$

このとき，TinyMRL 処理系によって実行される $Agent_a$ および $Agent_b$ における相互通信の内容は以下のように表現することが可能である。

$$Agent_a \rightarrow Agent_b : N_a, \langle\text{Message}\rangle \quad (2)$$

$$Agent_b \rightarrow Agent_a : \{\langle\text{Message}\rangle\}\langle K_{ba}\rangle, \text{MAC}(K_{ba}, N_a \| E_b) \quad (3)$$

ただし，$E_b = \{\langle\text{Message}\rangle\}\langle K_{ba}\rangle$ (4)

(2)式および(3)式における N_a は，$Agent_a$ として定義されたセンサノードが出力するノンスを意味する。本研究におけるシステムは，エージェント間の通信を実行する毎にノンスを生成する。ノンスとは，無作為のビット列であり，メッセージの順序性やデータの新規性を保障するために用いられる。(1)式によるデータ通信システムコールの呼び出しと共にノンスが生成されると，TinyMRL 処理系は，内部に存在するデータ領域に使用されたノンスを格納する。この後，他のエージェントから受信した返答メッセージおよび MAC(Message Authentication Code)内に，自身が生成したノンスが含まれていることを確認することによってメッセージの順序性を保障し，エージェント間協調プロセスを起動させる。

3. 実装

3.1 使用デバイス

次に本研究によって開発するマルチエージェント言語

および言語処理系の実装について述べる。本研究において，我々は CrossBow 社によって開発されたセンサーネットワークデバイス MOTE をセンサノードとして使用する。図 2 は，本研究で使用するために，様々なセンサによって MOTE[3)] を拡張したデバイスである。このデバイスは ATMEL 社製の 8bit の CPU と 4kbyte の RAM を持ち，315MHz 帯での無線通信を行う。また，組込み OS として TinyOS を用いている。システムの構成は，大きく 3 階層構造として構築される。最上位に位置するのは，ルールベースによる述語の集合によって定義された TinyMRL アプリケーションである。中間に位置するのは，TinyMRL の処理系であり，最下層の組込み OS と最上位アプリケーションとの仲介を行うミドルウェアである。処理系では，エージェントの動作実行処理およびエージェント協調の様々なフレームワークとセキュリティ機能を実行する。また，タイマの使用や AD コンバータを介したセンサからの測定値の取得も，TinyMRL 処理系へのシステムコールを通じて利用することが可能である。本研究による我々のシステムは，最小セットの実装イメージにとして，必要 ROM サイズは 12kbyte，必要 RAM サイズは 2.8kbyte となっている。

3.2 述語の実行スケジューラ

エージェントの動作は，複数の述語の定義と実行によって決定されるが，ノードの内部状態および実環境によって複数の述語を並列に動作させることも必要となる。このとき，動作する述語は，エージェントの内部状態と外部からの入力によって動的に決定されることとなる。各ノードの動作を決定する述語の実行処理の流れを図 3 に示す。これは，各タスク毎にスタック領域を確保しないシングルスタックのシステムにおける実装であり，非常に小さなリソースにおける計算機システムに特化した手法である。述語は，述語名および引数の数によってグループ化される。他の述語からの呼び出しによって，実行可能状態に遷移した述語グループは，実行待ちキューに投入される。このとき，周期的に動作するタスク切替器が，実行待ちキューの先頭に位置する述語グループを取り出す。次に，取り出された述

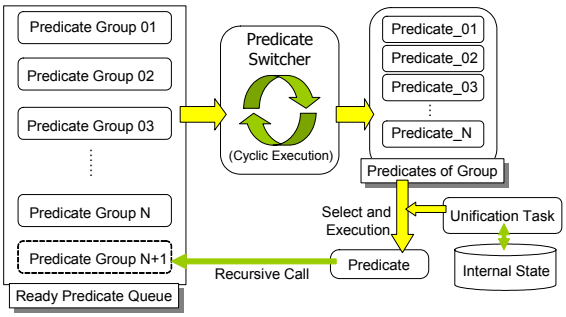

図 3 述語の実行におけるスケジューリング処理の流れ

語グループ内部に格納された述語群と，エージェントの内部状態および外部からの入力による状態変数を参照して，各述語に定義されている実行条件に適合する述語が選択され，実行に移る。このとき，グループ内の全ての述語が実行条件に適合しない場合には，再び述語グループが実行待ちキューの最後に投入されることとなる。また，実行する述語の最後に，自身の述語を再帰的に呼び出すことによって，自身の属する述語グループを実行待ちキューの最後に投入することが可能となる。これは，エージェントの状態遷移における一状態の持続を意味する。

４．評価実験および考察

4.1 処理速度評価

本研究によって実装される TinyMRL の言語処理系およびセキュリティシステムの有効性を検討するために以下の実験を行う。まず，センサノード間におけるセキュアな通信・協調動作を実行し，一連の処理の完了に至るまでの所要時間を計測する。そして，通常の手法による通信結果との比較を示す。図 4 に本システムにおける通信所要時間を示す。通信所要時間の比較として，(A)暗号化有り・認証有り，(B)暗号化無し・認証有り，(C)暗号化無し・認証無し（通常の手法）の 3 つの条件について，それぞれ計測を行った。図 4 において，Y 軸は通信所要時間，X 軸はホップ数を示している。また，データはホップ数 2，4，6 の場合においてそれぞれ計測している。図より，ホップ数が増大するにしたがって，通信所要時間が単調増加することが確認できる。同様に，暗号および認証処理を実行するにしたがって，処理時間が増加することが確認できる。実験結果より，我々のシステムにおいて，ホップ数が 2 以下における所要時間は 60msec（(C)の場合），暗号化や認証等の処理を付加して 110msec 弱（(A)の場合）で実行することが可能である。しかし，ホップ数が増加するにしたがって通信に必要とされる時間も増加し，ホップ数が 6 の実験結果では，所要時間が 300msec に達し（(A)の場合），ホップ数が 10 になると，530msec（(A)の場合）となることが確認できる。これは，即応的かつ緊急性を要するネットワークに対して，適しているとは考えにくい。これは，暗号化・複合化，認証の処理に要する時間だけでなく，パケットのルーティング処理に多くの時間が必要となるためであり，実験装置が持つ CPU(8bit, 8MHz)の低い処理能力が影響していると考えられる。また，無線を使用した通信における不確実性も影響の一要素として考えられる。

しかしながら，実際のユビキタスコンピューティング環境の構築では，10 回のホップによって，十分に遠距離に存在するノードまで到達することが可能であり（本実験環境および機器では約 150m），その間に十分な数のステーション（インターネットへのルータ）が設置されているこ

とが予想できる。これにより，実環境状態のセンシングや，歩行者のトラッキングおよび状態監視に対しては十分な能力を持っていると考えられる。これに併せて，アドホックネットワークによるセンサネットワーク構築の簡易さと，屋外環境への導入の容易さを考慮すると，本研究によるシステムは，セキュアなユビキタス環境を容易に構築するツールとして非常に有益であると思われる。

4.2 記述性評価

次に，本研究における TinyMRL を用いたセンサノードの動作構築の簡易性を実験を通じて評価する。実験には複数の被験者を用意し，達成すべき課題を提示する。課題には，TinyMRL を用いて定義すべきエージェントの動作内容が示される。被験者に課題を基にしてセンサノードの動作をプログラミングさせ，課題達成に要した時間を計測することによって，動作構築の簡易性評価とする。

実験の結果を図 5 に示す。被験者として，コンピュータサイエンスの豊富な経験を持つ人物（被験者 1 および被験者 2）と，手続き型言語の初歩のみを知識として持つ人物（被験者 3 および被験者 4）で行う。また，各被験者に提示する課題は共通しており，それぞれ，デバイス制御，イベント処理，通信処理，ファシリテータを介したサービス解決処理の 4 つを提示した。図より，各被験者の全ての計測結果が 1.0 を大きく下回っていることが確認できる。これは，被験者に提示された全ての実験課題について，TinyMRL による動作記述が，ネイティブな言語を使用した場合と比較して，大幅な簡易性を持っていることを示している。これにより，プログラマのスキルに関わらず，センサノードの簡便な動作記述が可能になることが確認できる。また，全ての被験者が 4 つの実験課題に同じ傾向の結果を出している点に注意されたい。特に実験課題 3 および 4 の結果は，各被験者共に 0.2 前後の低い値が計測されている。これは，本研究による TinyMRL が，センサネットワーク内の通信を伴う不特定多数の協調動作記述において，非常に優れた簡易性を持つことを示している。

５．まとめ

本研究において，我々は，センサネットワークのノード間によって必要とされる動的な協調問題を実現することを目的としたマルチエージェント記述言語 TinyMRL および処理系を開発した。これは，センサノードの動作をルール形式による単位動作の列挙として記述することが可能であるだけでなく，マルチエージェントの機能である分散サービスの動的な発見・提供を行う動作をシステムコールとして実装している。これにより，自律動作するセンサノード間の協調動作を簡潔に記述することを可能とするものである。また，無線アドホックネットワークにおける安全な通信を実現するために，限られたリソース上で動作するセキュリティシステムを構築し，処理系と融合させることによって，セキュリティを意識せずにセンサノードの動作設定を行うことを可能とした。そして，我々の開発したシステムの有効性を実証するために，その実行速度についての性能を測定したほか，動作定義の簡易性を異なるスキルを持つ複数の被験者によって計測し，その有効性を示した。

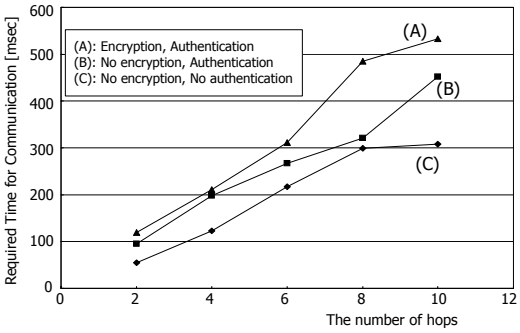

図 4 TinyMRL を使用したノード間における通信速度

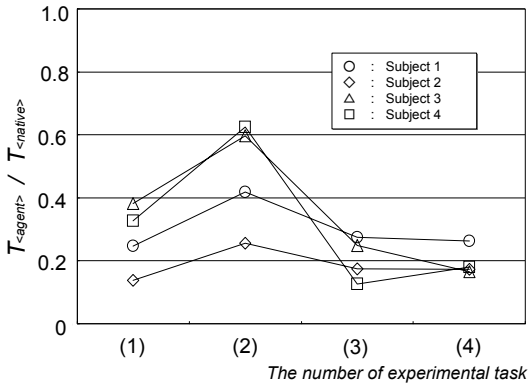

図 5 TinyMRL による動作記述簡易性評価

参考文献

1) A. Perrig , R. Szewczyk , V. Wen , D. Culler , J. D. Tygar : Mobile computing and networking, pp.189-199, July 2001, Rome, Italy.
2) B.Warneke, M.Last, B.Liebowitz, and K.Pister : IEEE Computer, pages 44-51, January 2001.
3) Crossbow Inc. http://www.xbow.com
4) F.Mizoguchi, H.Nishiyama, H.Ohwada and H.Hiraishi : Artificial Intelligence, Vol.114(1999).
5) Masuoka, R., Parsia, B. and Labrou, Y.: In Proceedings of the 2nd International Semantic Web Conference 2003, October 20-23(2003).
6) 西山裕之，大林真人，大和田勇人，溝口文雄: ロボット学会誌，vol.19，No.5，pp.620-631(2001).

（原稿受付　平成 17 年 8 月 3 日）