知的移動エージェントによるマルチパーパスワイヤレスセンサネットワークアプリケーション

長 健太[†] 大須賀 昭彦[†] 本位田 真一[††]

多数の小型センサから構成されるワイヤレスセンサネットワーク（WSN）を用いたビルオートメーションシステム，環境モニタリングシステム，サプライチェーンモニタリングシステムなどが実現されつつある．WSN 内の大量のセンサ群は環境に広範に分散配置され，かつそれらに搭載されているバッテリはきわめて限られているため，WSN アプリケーションにおいては電力消費量を抑えることが重要になる．WSN アプリケーションを実現するためのフレームワークは，消費電力と応答性能の間で発生するトレードオフを正しく扱える必要がある．我々は，複数の知的移動エージェントを用いてマルチパーパスな WSN アプリケーションを構築するというアプローチをとった．我々のフレームワークはエージェント群の動作方式を変化させることで，アプリケーションごとに異なる応答性能の要求に適応することができる．また，ビルディングオートメーションにおける移動エージェント群の動作をシミュレートすることで，消費電力と応答性能に関する評価を行った．

Intelligent Mobile Agent Framework for Multi-purpose Wireless Sensor Network Applications

Kenta Cho,[†] Akihiko Ohsuga[†] and Shinichi Honiden[††]

A wireless sensor network (WSN) that is a network of many small sensors is being used for many applications such as structure and equipment monitoring, environmental monitoring and supply chain monitoring. Since these many sensors are widely spread in the environment and a battery of sensors is very limited, main concern for WSN applications is reducing battery consumption. Framework for these WSN applications should handle the trade-off between battery consumption and response time. Our approach for this problem is using a multi intelligent mobile agent framework to implement multi-purpose WSN applications. By changing the movement pattern of agents, our framework can adapt to the requirements of response time that differ from each application. We evaluate battery consumption and response time with simulating mobile agents for the building automation WSN application.

1. はじめに

近年，複数の小型センサ機器をワイヤレスでネットワーク接続し，それらセンサ機器を連携させることで様々なアプリケーションを実現する，ワイヤレスセンサネットワーク（Wireless Sensor Network，WSN）が注目されている．スマートダストと呼ばれる小型センサおよび WSN 用ミドルウェアを用いて，野外環境のモニタリングシステムを実現する例など，WSN の実用化に向けた様々な取り組みがなされてきている．

WSN を利用したアプリケーションは，ビルやホーム，プラントのオートメーションを実現する事例，野外環境において農業のサポートを行う事例や環境モニタを行う事例，商品の流通をトレースする事例など多岐にわたる．WSN 上でアプリケーションを構築する場合，各センサ上で動作するプログラムをアドホックに記述し，多数のセンサを適切に連携させることは難しい．そのため，WSN 上で動作するミドルウェアを用いることが多い．WSN 上のミドルウェアは，いくつかのアーキテクチャに分類される．具体的なアーキテクチャとしては，センサで発生したイベントをトリガにその情報を，センサ群を管理，制御するセンタに通知する Event Based なアーキテクチャ，センタから必要とするデータのクエリを WSN に向けて通知し，センサが必要な情報を返信する Data Driven なアーキテクチャ，WSN 上を各センサで動作するコードを移動させて処理を行う Mobile Agent Based なアーキテクチャなどがあげられる．

† 株式会社東芝
Toshiba Corporation
†† 国立情報学研究所
National Institute of Informatics

文献 19) によれば，Event Based，Data Driven，Mobile Agent Based それぞれのアーキテクチャによるミドルウェアには，それぞれの特性に応じた適切なアプリケーションが存在する．Event Based なアーキテクチャは発生頻度の低い緊急事態を感知し，警報発令を行うアプリケーション，具体的には頻度の低い火災監視などに適している．Data Driven なアーキテクチャはセンサ網からのオンデマンドな情報収集アプリケーション，具体的には野外の広範な範囲での環境調査などに適している．Mobile Agent Based なアーキテクチャはスマートハウスなどにおけるサービス指向アプリケーションに適している．様々なサービスを行う必要があり，サービスごとのイベントの発生頻度が様々であるビル，ホーム，プラントオートメーションにおいては，サービス指向である Mobile Agent Based なアーキテクチャが適していると考えられる．

WSN を用いた実用アプリケーションとして，ビル，プラントオートメーションやセキュリティ用途のシステムが存在する☆．また文献 14) にあるような，野外環境での生態系調査などにも用いられている．これらの事例の多くは，単一のアプリケーションを WSN を用いて実現するものであるが，将来的には単機能なセンサを様々な用途に利用する，マルチパーパスな WSN アプリケーションが求められると考えられる[17),18)]．ビルや野外に設置されたセンサ群を様々な用途で活用することにより，対費用効果に優れたシステム設計が可能になる．

WSN アプリケーションで最も問題となる点は，センサを動作させる際の消費電力である．多くのセンサはそれに付加されたバッテリで動作するため，バッテリの電力が尽きた場合は人手で交換を行う必要がある．WSN アプリケーションでは多数のセンサが用いられかつそれらが広範な範囲に設置されるため，センサ交換の手間を極力減らすことがメンテナンスコストの減少につながる．そのため，アプリケーションを動作させる際の電力消費を抑えることが重要である．

消費電力はミドルウェアのアーキテクチャによっても大きく異なってくる．センサが値を一定時間ごとにセンタへ送信し続けるようなアーキテクチャにおいては，センサが値を受け取ってからセンタに通知されるまでの反応時間は短くなるが消費電力は多くなる．逆にモバイルエージェントが必要なセンサを巡回し，センサを駆動させるようなアーキテクチャでは，反応時間は長くなるが，エージェントが必要最低限のセンサを動作させるようにすることで消費電力を抑えることができる．

WSN アプリケーションでは，多くの場合その応答性能と消費電力の間でトレードオフが発生する．短い応答時間を要求する場合は，WSN 全体での消費電力が増し，逆に応答性能を低く抑えた場合は消費電力が少なくなる．そのため，単純に消費電力を抑えた動作を実現するだけではなく，アプリケーションで必要とされる応答性能，ネットワークトポロジ，各センサにおけるイベント発生頻度などを総合的に判断したうえで適切な動作をアプリケーションが行えるように支援する，WSN ミドルウェア，フレームワークが必要とされる．この判断は，複数のパーパスを WSN 上で行うマルチパーパスな WSN アプリケーションではよりいっそう複雑になる．それぞれのパーパスごとに要求される応答性能が異なる，同一のセンサが異なる用途に使われるなどの状況が発生するため，それらの条件，状況に応じた動作を実現する必要がある．

我々は知的移動エージェントを用いてマルチパーパスな WSN アプリケーションを実現することで，アプリケーションの要求する応答性能やネットワークトポロジを加味しつつ，消費電力を抑えた動作を実現する手法を提案する．動作させるエージェントとしては，我々の開発した小型知的移動エージェントである picoPlangent を用いる．

以下，2 章において移動エージェントを用いた WSN アプリケーションの構築，知的移動エージェント picoPlangent の概要，picoPlangent エージェントの複数の動作方式について述べ，3 章で picoPlangent によるビルオートメーション WSN アプリケーションの実現方式，エージェントに与えるルール群およびエージェントの動作について述べる．4 章において，ビルオートメーションにおけるエージェントの動作をシミュレートし，各動作方式における応答性能と消費電力を評価する．5 章で関連研究について述べる．

2. 移動エージェントによる WSN アプリケーション

WSN アプリケーションを構築する際に，各センサで動作するコードを移動させて処理を行う Mobile Agent Based なアーキテクチャを用いるアプローチがある．本章では WSN 上で動作する既存の移動エージェント，今回の提案で用いる小型知的移動エージェント picoPlangent の概要，および picoPlangent エージェントが備えるプランを変更することで複数のエージェント動作方式を実現し，それぞれの動作方式における

☆ DUST NETWORKS 社[15)]，Millennial Net 社[16)]

応答性能と消費電力の特性から，状況に応じた適切な方式を選択する手法について述べる．

2.1 WSN 上で動作する移動エージェント

従来から WSN 上で動作する移動エージェントが存在する．Agilla[7),8)] はカリフォルニア大学バークレー校で開発された小型センサ Mote 上で動作する移動エージェントであり，Mote 上に実現される WSN に特化した OS である TinyOS[1)] 上のバーチャルマシン Mate[6)] 上で動作する．Agilla はコードおよびその実行状態を保持したストロングマイグレーションが可能なモバイルエージェントである．Agilla エージェントは Mate 上に最大 4 つまで存在することができ，タプルスペースを介したエージェント間通信が可能である．WSN 上で動作するモバイルエージェントとしては，ほかに文献 9)〜12) が知られている．

2.2 知的移動エージェント picoPlangent

本節では，我々が開発した知的移動エージェント picoPlangent[26)] について説明する．picoPlangent は，PC 上で動作する知的移動エージェントである Plangent[27),28)] をベースとしている．Plangent は知識ベースをもとにしたプログラミングモデル，動的なプランの再生成（再プランニング）機能を備える．picoPlangent はリソースの豊富な PC 上から小型のモバイル機器上まで，様々な環境で動作することを特長としている．

picoPlangent エージェントは，Prolog の述語で記述された，ユーザの要求を表すゴールを保持する．ゴールはコンポーネントと呼ばれる，エージェントの動作するプラットフォーム上に配置されるソフトウェアによって解決される．コンポーネントがゴールを解決する際，コンポーネントがそのゴールを解決する動作を直接実行する場合と，そのゴールをより具体的で粒度の細かいゴール列に分解し，エージェントにそれら新たなゴール列を通知する場合がある．コンポーネントはエージェントからゴールを受け取るための特定のインタフェースを備え，エージェントからゴール解決の依頼を受ける．コンポーネントが直接解決可能なゴールを特にアクションと呼び，移動コンポーネントによってプラットフォーム間のエージェントの移動をさせる goto アクションなどが用意されている．

ゴールの列はプランと呼ばれる．picoPlangent はゴールツリーと呼ばれるデータ構造を用いてプランの管理を行う．プランの生成はプランニングコンポーネント内のプラナがルールに従ってゴールを分解することによって行われる．ゴールツリー内のプランは Prolog のリテラルの列で表記される全順序プランである[29)]．我々は半順序プランを扱うことができるエージェントの開発も行った[30)]．半順序プランは並行に動作する動作の記述を行うことや，必要のない動作の再実行を防ぐことができる．文献 31) では，picoPlangent エージェントのマルチエージェントシステムにおける投機的な動作における有用性を述べている．

WSN アプリケーション，特にビルオートメーションなどのアプリケーションにおいては，実現すべきサービスが多岐にわたるとともに，扱うべきセンサ，機器の備えるリソースも様々である．picoPlangent エージェントは，様々なサイズのコンポーネントをプラットフォーム上に配置し，それらを組み合わせてサービスを実現するプランをプラナを用いて生成，実行することができ，このような環境での動作に適している．

2.3 エージェントの動作方式

picoPlangent エージェントに与えるプランを変更することで，エージェントがどのセンサ上を巡回し，どのように情報を収集するかという，エージェントの動作方式を変更することができる．それらエージェントの動作方式を複数用意し，要求される応答性能などの状況に従って適切な動作方式を選択することで，応答性能と消費電力間でのトレードオフを加味した適切なアプリケーション動作が実現できる．

エージェントの動作方式としては以下を想定する．アプリケーションの応答性能を向上させるために，WSN 上で複数のエージェントを動作させる場合も想定する．

- シングルエージェント
  単一のエージェントが，すべてのパーパスを処理するプランを実行する．
- ロケーションベースマルチエージェント
  ロケーションは，ビル内の部屋などの特定の位置を表す．特定の位置に配置されているセンサ群ごとに，それらセンサ群を巡回するエージェントを配置し，複数のエージェントでアプリケーションのカバーすべきセンサ群を網羅する．
- パーパスベースマルチエージェント
  パーパスは，エージェントが WSN を用いて提供する単機能なサービスを表す．アプリケーションが実現すべき複数のパーパス 1 つずつにエージェントを割り当て，単機能のエージェントを複数動作させることでアプリケーションを実現する．

これらそれぞれの方式について，応答性能，消費電力を評価することで，ある状況においてどの方式を選択することが最適かを判断することができるようになる．シングルエージェントはセンサの動作回数や通信回数を抑えることができるため，消費電力の面では

有利だが，情報収集に必要なセンサを網羅的に巡回する必要があるため，応答性能は低いと想定される．複数エージェントを動作させることにより，応答性能の向上や，ロケーション，パーパスごとの応答性能の調整などが実現できるが，センサの動作回数やエージェントの移動回数などが増すため消費電力は増加する．複数エージェントを動作させる場合でも，同一センサ上にいる複数のエージェントに対して，センサから得られた情報を共用したり，移動の際の情報の送信を共用したりすることで，消費電力を抑えることが可能になる．

各方式における応答性能および消費電力は，エージェントがセンサ読み取りや移動などの動作を行う時間間隔によっても左右される．各方式の特性については，3 章において，ビルオートメーションを例にとったエージェント動作を説明し，4 章において，そのシミュレーションによって評価を行う．

3. picoPlangent によるビルオートメーション WSN アプリケーション

本章では，具体的な WSN アプリケーションとしてビルオートメーションを想定し，その上で上記の各方式に従って picoPlangent エージェントを動作させるためのエージェントやルールに関して説明する．

3.1 パーパス，ネットワークトポロジ

ビル内に WSN を構築し，空調や照明などを管理するビルオートメーションを実現するアプリケーションは，WSN を活用する事例として広く知られ，実用化されている．今回は，ビル内に温度を感知する温度センサ，および人の有無を感知する人感センサを配置し，それらセンサから得られる情報から，空調機器と照明を操作するとともに，温度センサが高温を感知した場合にその警告を行うという 3 つのパーパスを持つアプリケーションを，picoPlangent を用いて実現する方法を説明する．

具体的には，以下のパーパスを実現する．

- 空調
  温度センサから得られる温度と設定温度を比較し，空調を制御する．
- 照明
  人感センサから人の有無を判断し，部屋に人がいない場合は照明を切る．
- 防災
  温度センサから高温が感知された場合は，警告を表示する．

ビル内の WSN のネットワークトポロジとしては，**図 1** を想定する．あるフロアに 4 つの部屋が存在し，それぞれに空調，照明，防災制御機器（Control）が存在する．各部屋には 2 つの人感センサ（Occupancy Sensor），4 つの温度センサ（Temperature Sensor）が存在し，線で結ばれた 2 つのセンサ，機器間で通信が可能とする．WSN を構築する場合，スター型，メッシュ型，ハイブリッド型の 3 つのトポロジが考えられるが，特にエネルギー管理アプリケーションにおいて適切だとされている[16] メッシュ型を本アプリケーションでは採用した．

3.2 ル ー ル

picoPlangent エージェントを動作させるには，プラナがプランを生成するためのルール群を作成する必要がある．ビルオートメーションアプリケーションにおけるルール群は，エージェントの動作方式に従って，それぞれ**図 2**，**図 3**，**図 4** のように記述できる．action はコンポーネントの動作によって解決可能なゴール，つまりアクションを指定するための構文であ

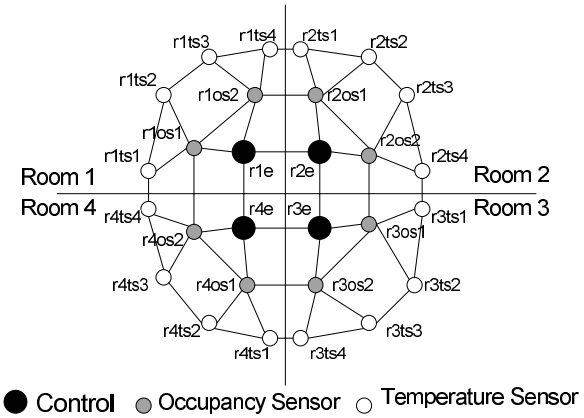

図 1 ネットワークトポロジ
Fig. 1 Network topology.

```
action(goto(_)).
action(newgoal(_)).
action(senseTemperature).
action(senseOccupancy).
action(controlEquipment).
axiom(automateBuilding,
 [automateRoom1,
  ...
  automateRoom4,
  newgoal(automateBuilding)]).
axiom(automateRoom1,
 [goto(r1e),
  goto(r1os1), senseOccupancy,
  goto(r1ts1), senseTemperature,
  ...
  goto(r1ts4), senseTemperature,
  goto(r1os2), senseOccupancy,
  goto(r1e), controlEquipment]).
...
```

図 2 シングルエージェントルール
Fig. 2 Single agent rules.

```
axiom(automateRoom1Repeat,
 [automateRoom1,
  newgoal(automateRoom1Repeat)]).
...
```

図 3 ロケーションベースマルチエージェントルール
Fig. 3 Location based multi agent rules.

る．axiom はゴールをサブゴールに分解するルールを表している．また goto アクションの引数に与えられる引数 r1e〜r4e は，各部屋の空調，照明，防災制御機器を，r1os1，r1os2 が Room1 に存在する 2 つの人感センサ上に構築されたプラットフォームを，r1ts1〜r1ts4 が Room1 に存在する 4 つの温度センサ上に構築されたプラットフォームを表す．Room2〜4 にも同様にプラットフォームが設置される．

以下では，エージェントが動作する各方式を，プラナに与えるルールを変化させることによって実現する．プラナは，各部屋の空調，照明，防災制御機器上に配置され，制御機器上にエージェントが生成された後，エージェントはプラナを用いてプランを生成する．

3.2.1 シングルエージェント

単一のエージェントがすべてのパーパスを実現する場合，まずエージェントに automateBuilding というゴールがユーザから与えられ，r1e プラットフォーム上に生成される．このゴールは図 2 の axiom によって 4 つのゴール，automateRoom1〜4，newgoal(automateBuilding) をシーケンシャルに実行するプランに分解される．automateRoom1 は以下の動作を行うプランを生成する．

```
action(clone(_)).
action(checkTemperature).
action(checkOccupancy).
axiom(adjustTemperature,
  [goto(r1os1), adjustTemperatureStart]).
axiom(adjustTemperatureStart,
  [goto(r1ts1), senseTemperature,
   ...
   goto(r1ts4), senseTemperature,
   checkTemperature,
   clone(controlAirConditioningR1),
   goto(r2ts1), senseTemperature,
   ...
   goto(r4ts4), senseTemperature,
   checkTemperature,
   clone(controlAirConditioningR4),
   newgoal(adjustTemperatureStart)]).
axiom(controlAirConditioningR1,
  [goto(r1os2), goto(r1e), controlEquipment]).
...
axiom(adjustLighting,
  [goto(r1os1), senseOccupancy,
   goto(r1os2), senseOccupancy,
   checkOcuupancy, clone(controlLighting),
   ...
   newgoal(adjustLighting)]).
axiom(warnTemperature,
  [goto(r1os1), warnTemperatureStart]).
axiom(warnTemperatureStart,
  [goto(r1ts1), senseTemperature,
   ...
   goto(r1ts4), senseTemperature,
   checkWarnTemperature, clone(giveWarning),
   ...
   newgoal(warnTemperatureStart)]).
...
```

図 4 パーパスベースマルチエージェントルール
Fig. 4 Purpose based multi agent rules.

(1) Room1 の人感センサおよび温度センサを巡回する（goto アクション）．
(2) 人感センサ上では人の有無を感知（senseOccupancy アクション），温度センサ上では温度を読み取り（senseTemperature アクション），エージェント内部にデータを蓄積する．エージェント内部では，部屋の平均温度，部屋の異常高温感知有無，部屋の人の有無，各部屋の空調の状態，各部屋の照明の状態，クローン生成を行うか否かのフラグなどのデータを管理している．
(3) Room1 の制御機器上に移動し，エージェント内部に保持してある平均温度，異常高温感知有無，人の有無の情報を参照し，空調や照明の制御や防災のための警告の表示などを行う（controlEquipment アクション）．

以上の動作を，Room1〜4 に対して行うことで，エージェントはすべての部屋のすべてのセンサを巡

回し，センサから得られた情報に従って機器の制御を行う．その後，newgoal アクションによって再び automateBuilding ゴールがエージェントに与えられ，巡回を継続する．

3.2.2 ロケーションベースマルチエージェント

より速い応答性能が要求される，各部屋ごとに要求される応答性能が異なるなどの場合，エージェントを各部屋ごとに配置する方法が考えられる．その場合は，図 3 のルールにおける，automateRoom1Repeat〜automateRoom4Repeat のゴールを持ったエージェントを r1e〜r4e プラットフォーム上に生成し，それらエージェントを並行動作させる．

3.2.3 パーパスベースマルチエージェント

上記 2 つの動作方式では 1 つのエージェントが，空調，照明，防災すべてのパーパスを受け持って動作している．そのため，特定のパーパスにより速い応答性能が要求されるなどの制約がある場合に対処することができない．パーパスごとの細かな制御が必要とされる場合は，パーパスごとにエージェントを生成し，それらのエージェントを並行して動作させることが必要となる．

パーパスごとのエージェントを生成するためのルールは図 4 のようになる．この場合，空調を制御する adjustTemperature ゴールが与えられたエージェント，照明を制御する adjustLighting ゴールが与えられたエージェント，防災を行う warnTemperature ゴールが与えられたエージェントの 3 つのエージェントが r1e プラットフォーム上に生成される．それぞれのエージェントは以下のように動作する．

- ajustTemperature ゴールはまずエージェントを温度センサに隣接する rtos1 プラットフォームに移動させ，その後 adjustTemperatureStart ゴールの解決を試みる．
- adjustTemperatureStart ゴールはまず Room1 の温度センサを巡回した後，それらの温度センサから得られた平均温度を計算し，Room1 の空調の状態と照らしあわせ，空調の動作を変更する必要があるかどうかを判断する（checkTemperature アクション）．変更する必要がある場合は，clone アクションで引数に与えられたゴールを実行する子エージェントを別途生成する．生成された子エージェントは部屋の制御機器まで移動し，空調の動作の変更を行い，消滅する．
- adjustLighting ゴールで生成されたエージェントは人感センサを巡回し，照明機器を変更する必要がある場合に子エージェントを生成する．子エージェントは部屋の制御機器まで移動し，照明の操作を行い，消滅する．
- warnTemperature ゴールで生成されたエージェントは温度センサを巡回し，高温を感知した場合に子エージェントを生成する．子エージェントは部屋の制御機器まで移動し，防災機器の操作を行い，消滅する．

4. 評　　価

本章では，3.2 節で示した各動作方式において，その応答性能および消費電力を評価する．評価にあたっては，前記ネットワークトポロジ上でのエージェント動作のシミュレートを行う．

4.1 応答性能の計測方法

ビルオートメーションアプリケーションを再現するにあたって，各部屋におけるセンサから得られる情報に関してもシミュレートを行う必要がある．また，エージェントの応答性能を測るには，そのセンサ情報の変化が発生してから，エージェントが制御機器の操作を行うまでの時間を測定する必要がある．今回は，以下の方式でシミュレートおよび応答性能の計測を行った．

- 空調機器の動作を変更しなければならない温度の発生，照明機器の動作を変更しなければならない人の有無の状態の変化，防災機器を動作させなければならない温度の発生などのイベントが，各部屋においてそれぞれ**表 1** の間隔で発生するとする．発生間隔のぶれを発生させる乱数は，各試行ごとに一定であるとする．
- 各イベントの発生を感知したセンサは，イベントが発生したという状態をバッファリングする．エージェントのセンサ上での読み取りは，センサがバッファリングしているイベントに対して行われ，上記のイベントが発生していた場合はその内容がエージェント内部のデータに記録される．そのデータを保持したエージェントが制御機器上で制御を行った場合，その時間を記録し，上記のイベント発生時間から経過した時間を，エージェントの応答時間とする．
- センサ上でイベントが発生した後，その次のイベントが発生するまでの間にエージェントがセンサ

表 1 イベント発生間隔
Table 1 Event interval.

イベント	発生間隔 (sec.)
空調機器変更	300〜600
照明機器変更	900〜1,800
防災機器動作	3,000〜6,000

上に移動，読み取りを行った場合，エージェントがイベントの読み取りに成功したとする．エージェントがセンサ上で読み取りを行う前に次のイベントが発生した場合，そのイベントの読み取りは失敗である．

- イベントの発生間隔は，発生頻度の異なる複数種類のイベントが発生する，ビルオートメーションアプリケーションの特性をシミュレートするよう設定を行った．アプリケーションにおける消費電力，応答性能，エージェントがイベントの読み取りに成功する確率は，後述のエージェントの動作間隔とイベント発生間隔の比率に影響を受ける．イベントが発生する頻度が低く，イベント発生間隔が長い状況の場合，イベント発生間隔に合わせてエージェントの動作間隔をより長い間隔に調整する必要がある．
- 空調機器変更と防災機器動作は両方とも温度センサによって発生するイベントであるため，それらは厳密には独立に発生するイベントではないが，今回のシミュレートではこれらイベントも独立なものとして扱った．イベント間でその発生が独立でないもの，特定のイベント発生に別のイベント発生が必要であるもの，などの条件を考慮したシミュレートに関しては，今後の課題である．

4.2 消費電力の計測方法

センサ上での消費電力を正確にシミュレートすることは難しいが，既存の研究において消費電力量を測定した事例を見ることができる．文献 13) では，Mote を用いた生態環境調査アプリケーションを例に，センサの各動作における消費電力を測定している．この例では，タイマ動作に 0.0034 mJ，外部からパケットの感知に 0.465 mJ などの微量の電力を消費する動作と，ロングプリアンブルを持つパケットの送信に 39.2 mJ，天候センサの動作に 36.4 mJ，人感センサの動作に 35.3 mJ などの大量の電力を消費する動作についての測定結果があげられている．このことから，パケットの送信，およびセンサの駆動に対して大量の電力が消費されているということが分かる．そこで今回は，エージェントの移動，およびエージェントがセンサ上で読み取りを行った回数を記録，それらを比較することで各動作方式における消費電力を比較する．

複数エージェントが動作する場合，同一時間に同一センサ上で複数エージェントがセンサの読み取りを行う場合，それらエージェント間で読み取り結果の共用が行えるので，読み取り回数は 1 回と数える．また同一時間に同一プラットフォームから複数のエージェントが移動する場合，パケットが周辺プラットフォームにブロードキャストされる特性を持つ WSN の場合，それら複数のエージェントの情報をパケットに含めることで，パケットを送信する回数を抑えることができる．そこでエージェント移動回数とは別に，パケット送信回数を記録し，複数エージェントが同時に移動する場合，パケット送信回数は 1 回と数える．複数エージェントを送信することにより，伝送データ量は増加するが，センサがパケット送信開始時に消費する電力を抑えることができ，またパケット内の共通なヘッダ部分，複数エージェント間で共通するデータ部分などのデータ量を削減することができるため，一定の消費電力削減効果があると考えられる．

4.3 エージェントの動作間隔

Event Based アーキテクチャの WSN アプリケーションでは，センサを一定間隔ごとに動作させ，読み取った値を監視し，異常な値を検知した場合にその情報をセンタに通知する動作を行う．こういった動作の場合，応答性能と消費電力は，センサの動作間隔に強く依存する．Mobile Agent Based アーキテクチャにおいても，エージェントがどの程度の間隔で移動，センサの読み取りを行うかによって応答性能と消費電力が影響を受ける．

今回の評価においては，エージェントの各動作方式において，エージェントが移動もしくはセンサの読み取りのどちらかのアクションを行う間隔を，エージェントの動作間隔と定義し，エージェントの動作間隔を変化させることにより，応答性能および消費電力がどのように変化するかを測定した．

4.4 ネットワークトポロジ

ネットワークトポロジは前述の図 1 を想定する．線でつながったプラットフォーム間は，エージェントが 1 回のパケット送信，1 回の動作間隔で移動できる．

4.5 シミュレーション結果

これらの設定において，エージェントを 24 時間動作させた場合の結果を，3.2 節にあげたそれぞれの動作方式についてシミュレートした．

4.5.1 シングルエージェント

図 2 のルールを用いるシングルエージェントを，その動作間隔を 5 秒，10 秒，20 秒，30 秒と変化させた場合の温度センサ動作回数，人感センサ動作回数，エージェントの移動回数，パケット送信回数は**図 5** に，空調，照明，防災の各パーパス別応答性能，イベント読み取り成功率は**図 6** になる．

シングルエージェントにおいては，パケット送信回数やセンサ動作回数などを他の方式と比較して少なく

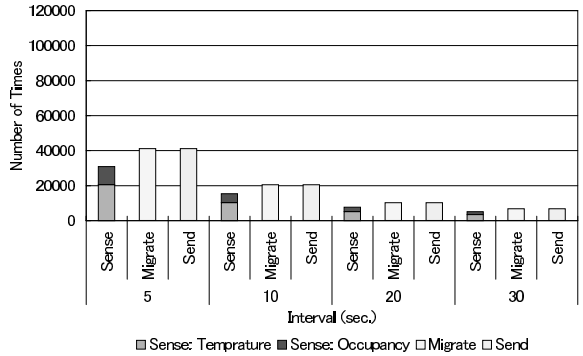

図 5 動作回数（シングルエージェント）
Fig. 5 Number of operations (Single agent).

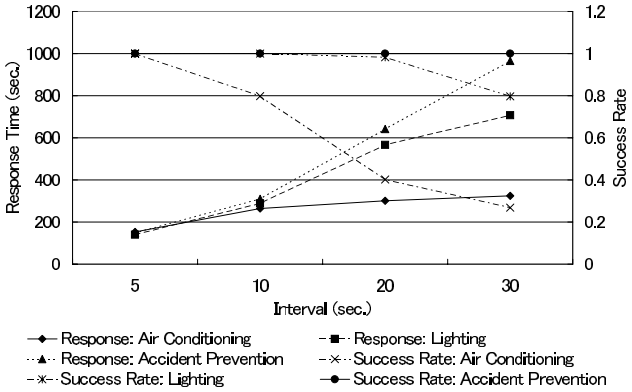

図 6 応答性能（シングルエージェント）
Fig. 6 Response time (Single agent).

抑えることができ，消費電力面では有利である．しかし動作間隔を広くとると応答性能やイベント読み取り成功率が著しく低下する．また応答性能と消費電力間のトレードオフを，単一エージェントの動作間隔で調整する以外の方法がなく，動作間隔を広げるとすべてのパーパスの応答性能が一様に低下してしまうという問題がある．

4.5.2 ロケーションベースマルチエージェント

図 3 のルールを用いたロケーションベースマルチエージェントにおいても，生成される 4 つのエージェントの動作間隔を，シングルエージェントの場合の 4 倍の 20 秒，40 秒，80 秒，120 秒に設定した場合，応答性能と消費電力間のトレードオフはシングルエージェントとそれほど変化はない．しかし，ロケーションごとに特性が異なる，たとえば特定の部屋において高い応答性能が要求されない場合に，対応するエージェントの動作間隔を広げる，といった調整を行うことができる．

4.5.3 パーパスベースマルチエージェント

図 4 のルールを用いるパーパスベースマルチエージェントにおいては，各パーパスを実行するエージェントごとの異なる動作間隔を設定することができる．今回のアプリケーションでは，特に防災に関しては他のパーパスよりも速い応答性能が求められることが考えられるため，今回の評価では防災パーパスを実現するエージェントの動作間隔を短くする．また照明の制御に関しては，それほど高い応答性能は必要ないので，エージェントの動作間隔を長くする．空調，照明，防災それぞれのパーパスを実現するエージェントの動作間隔を**表 2** のように設定した場合の結果は**図 7** および**図 8** になる．

表 2 エージェント動作間隔（マルチパーパス）
Table 2 Agent operation interval (Multi-purpose).

エージェントパーパス	動作間隔 (sec.)			
防災	5	10	20	30
空調	10	20	40	60
照明	50	100	200	300

防災パーパスに関する応答性能が大幅に改善されたが，それにともないパケット送信回数やセンサ動作回数も増加している．また照明パーパスは動作間隔を広

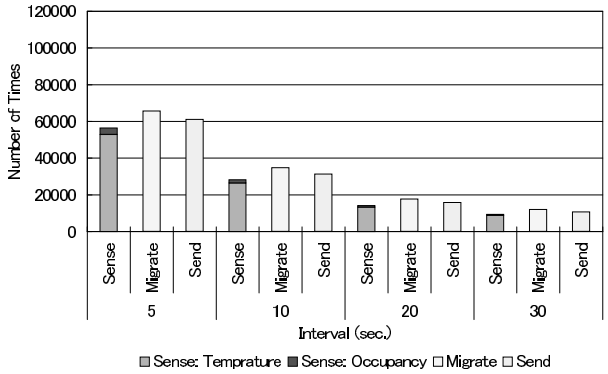

図 7 動作回数（パーパスベースマルチエージェント）
Fig. 7 Number of operations (Purpose based multi agent).

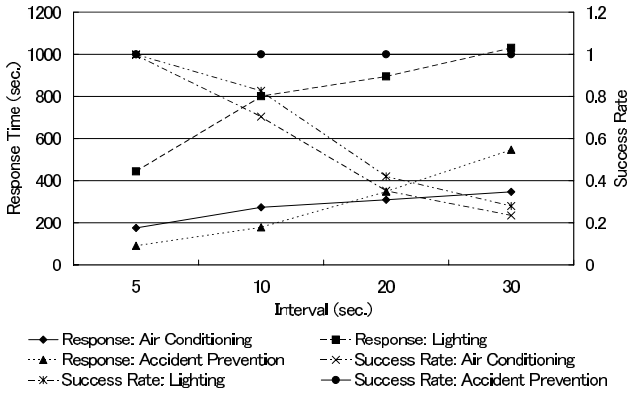

図 8 応答性能（パーパスベースマルチエージェント）
Fig. 8 Response time (Purpose based multi agent).

げた分応答性能が悪化しているが，他のパーパスの応答性能に対する影響はない．

4.5.4 ハイブリッドマルチエージェント

ロケーションベースマルチエージェントを用いることで，ビル内の部屋など，特定の場所に関しての応答性能を調整することができる．また，パーパスベースマルチエージェントを用いることで，要求する応答性能が異なる複数のパーパスに対応することができる．

これらの方式を組み合わせることで，場所，パーパスそれぞれでイベントの発生頻度が異なったり，要求される応答性能が異なったりする WSN アプリケーションにおいて，応答性能を満たしつつ消費電力を抑える動作を，マルチエージェントに行わせることができる．具体的には，パーパスベースマルチエージェントで動作しているエージェント群において，特定パーパスを受け持つエージェントにロケーションベースの動作を加える，もしくはロケーションベースマルチエージェントで動作しているエージェント群において，特定ロケーションを受け持つエージェントにパーパスベースの動作を加える，というように 2 つの動作方式を組み合わせる．この方式をハイブリッドマルチエージェントと呼ぶこととする．

ビルオートメーションアプリケーションにおいては，ビル内で実現すべき様々なパーパス，フロアや部屋などの場所ごとに異なるイベント発生頻度，要求される応答性能に対応するため，ハイブリッドマルチエージェントを用いて対応を行うことが有効である．今回の評価で用いたアプリケーションの場合，頻繁に機器を制御する必要のある空調パーパスに関してのみ，各部屋ごとに別エージェントを動作させることが考えられる．その場合のルールは**図 9** になる．adjustLighting，warnTemperature ゴールを持つエージェント以外に，adjustTemperatureRoom1〜adjustTemperatureRoom4 のゴールが与えられたエージェントを r1e〜r4e プラットフォーム上に生成する．

図 9 のルールを用いるハイブリッドマルチエージェントを，**表 3** に設定した動作間隔で動作させた場合

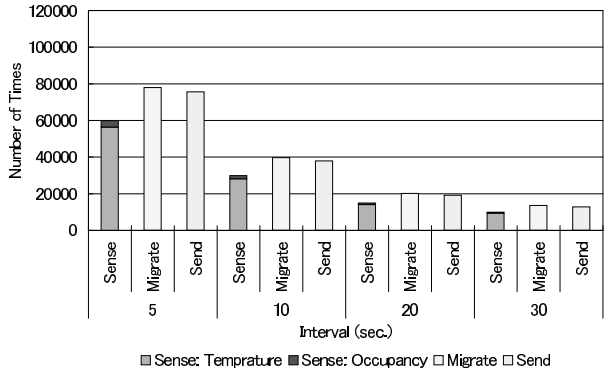

図 10 動作回数（ハイブリッドマルチエージェント）
Fig. 10 Number of operations (Hybrid multi agent).

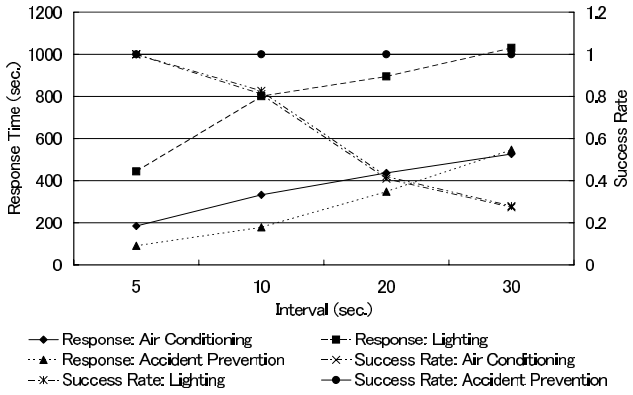

図 11 応答性能（ハイブリッドマルチエージェント）
Fig. 11 Response time (Hybrid multi agent).

```
axiom(adjustTemperatureRoom1,
 [goto(r1e),
  goto(r1os1),
  goto(r1ts1), senseTemperature,
  ...
  goto(r1ts4), senseTemperature,
  goto(r1os2),
  goto(r1e), controlEquipment,
  newgoal(adjustTemperatureRoom1)]).
...
```

図 9 ハイブリッドマルチエージェントルール
Fig. 9 Hybrid multi agent rules.

の結果は**図 10** および**図 11** になる．

図 7 および図 10 双方において，パケット送信回数やセンサ動作回数が増加しているが，これらは複数エージェントが互いに協調せずに移動，読み取りを行っていることが原因でもある．これらの方式のエージェントは，ロケーションごと，パーパスごとに柔軟な応答性能設定ができるが，消費電力の面で無駄が発生する．

エージェントが移動もしくはセンサ読み取りを行う際に，他のエージェントが同様の動作を行うかどうかを監視し，同様の動作を行うエージェントがいた場合は，同時に移動を行ったり，その読み取り結果を共用したりすることで，消費電力を抑えることができる．

表 3 エージェント動作間隔（ハイブリッド）
Table 3 Agent operation interval (Hybrid).

エージェントパーパス	動作間隔 (sec.)			
防災	5	10	20	30
空調（各部屋ごと）	25	50	100	150
照明	50	100	200	300

エージェントが動作間隔で設定された時間分，移動前に待機しているときに他のエージェントが移動を行った場合，そのエージェントは待機をやめすぐに移動する動作，およびセンサの読み取りを行おうと待機しているときに，他のエージェントが読み取りを行った場合に待機をやめその読み取り結果を共用する動作をハイブリッドマルチエージェントに組み込んだ場合の結果は**図 12** および**図 13** になる．

移動および読み取りの共用を行わなかったときに比

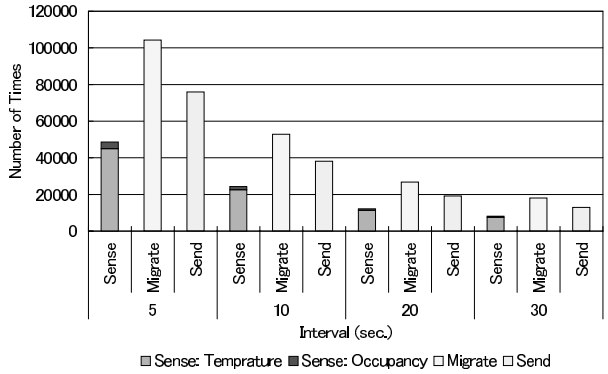

図 12 動作回数（ハイブリッドマルチエージェント，移動センシング共用）
Fig. 12 Number of operations (Hybrid multi agent, migration sensing sharing).

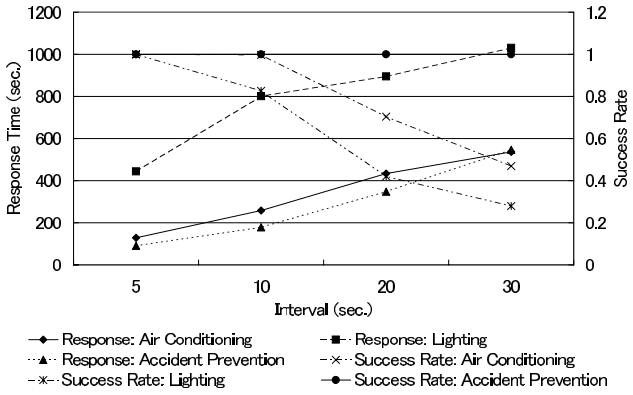

図 13 応答性能（ハイブリッドマルチエージェント，移動センシング共用）
Fig. 13 Response time (Hybrid multi agent, migration sensing sharing).

べ，センサ動作回数が減少し，シングルエージェントの場合と比較した増加量も少なく抑えることができる．エージェント移動回数自体は図 10 から増加しているが，パケット送信回数自体は図 10 と同程度に抑えられ，各パーパスの応答性能やイベント読み取り成功率は改善されており，アプリケーション全体での消費電力増加を抑えつつ，応答性能を改善することができることが分かる．

パーパスベースマルチエージェントに対して同様の処理を行った場合は，空調パーパスを担当するエージェントと防災パーパスを担当するエージェントの移動経路が同じであり，かつそれぞれの動作間隔が異なるため，移動や読み取りを共用できる場合が少なく，得られる効果は少なくなる．

このように，複数パーパスを持つ WSN アプリケーションを実現する際に，複数の動作方式を備えたエージェント群を動作させることで，ロケーションごとの特性，パーパスごとの特性に合わせて，アプリケーションの動作を柔軟に変更し，応答性能と消費電力の間で発生するトレードオフをふまえ，適切な動作を選択することが可能になる．またマルチエージェントにおいては，エージェントの移動，センサ読み取りを共用することで，消費電力を抑えることができる．

特定のアプリケーションに対して，エージェントの動作方式のうち，どの方式を用いるのかが適切であるかの選択に関しては，本評価で行ったようなシミュレーションを行うことで判断することが可能である．ただし，より正確な選択を行うために，エージェントの移動による伝送データ量，複数エージェントを移動させる場合の伝送データの削減方式などを考慮した消費電力の評価を行うことが今後の課題である．また，ネットワークトポロジをロケーションに分割する場合の適切なロケーションの割当て方法，各エージェントにどのパーパスを割り当てるかといった判断を，シミュレーションによる評価と，その結果を受けたロケーション，パーパスの割当ての改善，というサイクルによって効率良く行うための手法を整備する必要がある．

5. 関連研究

WSN 上で消費電力を抑えたアプリケーションを実現するためのフレームワークやパラダイムが従来から存在する．Directed Diffusion [20] は，data-centric な WSN 上の情報伝播において，情報を収集するタスクにどういった情報を収集したいかを表す interests を付加することで，必要な範囲のセンサのみを駆動し，少ない消費電力で情報を収集するパラダイムである．文献 21) では，各センサを駆動することによる情報増加量，エネルギーコスト，通信距離を定式化したうえで，各センサの持つ belief state を更新することで適切な範囲のセンサを駆動するアルゴリズムを提案している．文献 22) では，センサから得られる情報にメタデータを付加し，情報を送信する前にメタデータを送信することで，送信先のセンサがその情報を必要とするかを判断し，必要な場合だけ情報を送信し，消費電力を抑える．これらの挙動は，picoPlangent エージェントにおいても，各センサ上でそのセンサおよび隣接するセンサに関する属性をルールとして配置しておき，そのルールを参照してエージェントのプランを構築することで実現可能と思われる．ただしセンサ上でも動作する，軽量なプラナを実現する必要がある．

Impala [23] も消費電力を抑える機構を備えた，Event Based なミドルウェアである．Impala では，センサと通信を行ったピア数をカウントし，必要に応じて通信プロトコルを切り替える adapter を備えることで消費電力を抑えるなどの動作を実現することができる．ただし adapter 自体は，アプリケーションごとに作りこむ必要があり，消費電力などのコストをアプリケーション非依存に扱う仕組みなどは備わっていない．TinyDB は WSN に対して送信されたクエリに対し，適切な範囲のセンサ間で情報をやりとりすることで，少ない消費電力で適切な結果を得ることを目標としたミドルウェアである [2)〜5)]．クエリに対して複数のセンサが反応する場合に，隣接した他のセンサがクエリに対して送信した値を受信し，その値を用いて自身が値を送信する必要があるかどうかを判断する．picoPlangent エージェントにおいては，同じセンサ上および隣接するセンサから移動する他エージェントの保持するデータに従ってプランを変更するなどの動作で同様の機構が実現できると考えられる．

Semantic Streams [17] は，semantic service programming model を用いて複数のセンサから得られる情報を様々なパーパスにマッピングしている．picoPlangent と同様に Prolog を用いたルール表記を行い，センサやサービスを記述することができる．Semantic Streams では消費電力などコストに関する考慮は行っていない．文献 18) では，各センサのノードにその種別や位置情報などの属性を付加し，属性を用いてノード群のスコープを定義，スコープごとにアクセス，QoS コントロールを行うことで，複数パーパスを持つ WSN アプリケーションの開発を容易にしている．スコープを用いてノード群を分類する方法は，picoPlangent エージェントの移動経路を生成するルールの記述方法などにも応用可能と考えられる．

文献 24) は，時間とコストのトレードオフを加味したプランニング方式を提案している．この方式を picoPlangent のプラナに組み込むことで，エージェント動作中にプランの消費電力が変化するなどのより複雑な状況に対応できると思われる．

WSN 上で動作する移動エージェントとして MADSN [25] がある．MADSN では各センサの読み取り値の解像度を調整することで消費電力を抑える工夫をしている．また文献 9) では MADSN でのデータ収集におけるエージェントの経路を Genetic Algorithm を用いて最適化している．MADSN では複数エージェントの連携による動作については特に注意を払っていない．またプラン生成などの機能は備わっていない．

6. おわりに

知的移動エージェント picoPlangent により WSN 上のビルオートメションアプリケーションを実現する事例を用いて，シングルエージェント，ロケーションベースマルチエージェント，パーパスベースマルチエージェント，ハイブリッドマルチエージェントの 4 つの動作方式のエージェントを動作させる手法に関して提案した．エージェントの動作方式を変化させることで，応答性能と消費電力間のトレードオフを加味し，要求される応答性能に応じたアプリケーション動作を行わせることが可能になる．

今後の課題としてはルールの記述方式の洗練があげられる．今回の例では，各動作方式別にほぼ独立したルールを記述したが，動作方式に依存しないネットワークトポロジや各センサ上でのエージェント動作などは共通のルールを記述したうえで，エージェントの移動経路に関してのみ各動作方式別にルールで記述するなどの工夫を行うことで，ルールの記述がより容易になると考えられる．

アプリケーションを構築するアーキテクチャ，およびエージェント動作方式の適切な選択手法の整備も課題である．そのためには，本論文における提案方式に

加え，関連研究で触れた各方式に対する性能評価を，様々な WSN アプリケーションに対して行い，それぞれの方式の特性を測定する必要がある．また，複数の方式を組み合わせ，シミュレーションを繰り返すことにより，対象のアプリケーションに対して最適なアーキテクチャを構築する手法も必要とされる．

参 考 文 献

1) Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D.E., Kristofer, S. and Pister, J.: System Architecture Directions for Networked Sensors, *Architectural Support for Programming Languages and Operating Systems*, pp.93–104 (2000).
2) Madden, S., Szewczyk, R., Franklin, M. and Culler, D.: Supporting Aggregate Queries Over Ad-Hoc Wireless Sensor Networks, *Proc. 4th IEEE Workshop on Mobile Computing and Systems Applications* (2002).
3) Madden, S., Franklin, M., Hellerstein, J. and Hong, W.: The design of an acquisitional query processor for sensor networks, *Proc. 2003 ACM SIGMOD international conference on Management of data*, pp.491–502 (2002).
4) Madden, S., Franklin, M.J., Hellerstein, J. and Hong, W.: TAG: a Tiny Aggregation Service for Ad-Hoc Sensor Networks, *Proc. OSDI* (2002).
5) Hellerstein, J., Hong, W., Madden, S. and Stanek, K.: Beyond Average: Towards Sophisticated Sensing with Queries, *Workshop on Information Processing In Sensor Networks* (*IPSN*) (2003).
6) Levis, P. and Culler, D.: Mate: A tiny virtual machine for sensor networks, *International Conference on Architectural Support for Programming Languages and Operating Systems*, San Jose, CA, USA (2002).
7) Fok, C.-L., Roman, G.-C. and Lu, C.: Rapid Development and Flexible Deployment of Adaptive Wireless Sensor Network Applications, *Proc. 25th IEEE International Conference on Distributed Computing Systems* (*ICDCS'05*), Vol.00, pp.653–662 (2005).
8) Fok, C.-L., Roman, G.-C. and Lu, C.: Mobile Agent Middleware for Sensor Networks: An Application Case Study, *International Conference on Information Processing in Sensor Networks* (*IPSN'05*), Special track on Platform, Tools and Design Methods for Network Embedded Sensors (SPOTS), Los Angeles, CA (2005).
9) Wu, Q., Nageswara, S., Rao, V., Barhen, J., Iyengar, S.S., Vaishnavi, V.K., Qi, H. and Chakrabarty, K.: On Computing Mobile Agent Routes for Data Fusion in Distributed Sensor Networks, *IEEE Trans. Knowledge and Data Engineering*, Vol.16, No.6, pp.740–753 (2004).
10) Qi, H., Xu, Y. and Kuruganti, P.T.: The Mobile Agent Framework for Collaborative Processing in Sensor Networks, *Frontiers in Distributed Sensor Networks*, pp.783–800, CRC Press (2004).
11) Szumel, L., LeBrun, J. and Owens, J.D.: Towards a Mobile Agent Framework for Sensor Networks, *2nd IEEE Workshop on Embedded Networked Sensors* (*EmNetS-TT*), Sydney, Australia (2005).
12) Tseng, Y.C., Kuo, S.P., Lee, H.W. and Huang, C.F.: A Mobile-Agent Approach for Location Tracking in a Wireless Sensor Network, *Intl Computer Symp.* (2002).
13) Szewczyk, R., Mainwaring, A., Polastre, J., Anderson, J. and Culler, D.: An analysis of a large scale habitat monitoring application, *Proc. 2nd International Conference on Embedded Networked Sensor Systems* (*SenSys 2004*), pp.214–226, ACM Press (2004).
14) Szewczyk, R., Osterweil, E., Polastre, J., Hamilton, M., Mainwaring, A. and Estrin, D.: Habitat monitoring with sensor networks, *Comm. ACM*, Vol.47, No.6, pp.34–40 (2004).
15) DUST NETWORKS. http://www.dustnetworks.com/
16) Millennial Net: wireless sensor mesh networking system software and modules. http://www.millennial.net/technology/topologies.asp
17) Whitehouse, K., Zhao, F. and Liu, J.: Semantic Streams: A Framework for Composable Semantic Interpretation of Sensor Data, *EWSN 2006*, pp.5–20 (2006).
18) Steffan, J., Fiege, L., Cilia, M. and Buchmann, A.: Towards Multi-Purpose Wireless Sensor Networks, *International Conference on Sensor Networks* (*SENET'05*), IEEE, Montreal, Canada (2005).
19) Yoneki, E. and Bacon, J.: A Survey of Wireless Sensor Network Technologies: Research Trends and Middleware's Role, Technical Report 646, UCAM-CL-TR-646, ISSN 1476-2986, University of Cambridge (2005).
20) Intanagonwiwat, C., Govindan, R. and Estrin, D.: Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks, *Proc. 6th Annual International Conference on Mobile Computing and Networking* (*MobiCOM '00*), Boston, Massachussetts (2000).

21) Chu, M., Haussecker, H. and Zhao, F.: Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks, *International Journal of High Performance Computing Applications* (2002).
22) Kulik, J., Heinzelman, W.R. and Balakrishnan, H.: Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks, *ACM/IEEE Int. Conf. on Mobile Computing and Networking*, Seattle, WA (1999).
23) Liu, T. and Martonosi, M.: Impala: A middleware system for managing autonomic, parallel sensor systems, *Proc. 9th ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming*, pp.107–118 (2003).
24) Do, M. and Kambhampati, S.: Planning graph-based heuristics for cost-sensitive temporal planning, *Proc. 6th International Conference on Artificial Intelligence Planning and Scheduling* (2002).
25) Qi, H., Iyengar, S.S. and Chakrabarty, K.: Distributed multiresolution data integration using mobile agents, *Proc. IEEE Aerospace Conference* (2001).
26) Cho, K., Hayashi, H., Hattori, M., Ohsuga, A. and Honiden, S.: picoPlangent: An Intelligent Mobile Agent System for Ubiquitous Computing, *7th Pacific Rim International Workshop on Multi-Agents*, pp.45–56 (2004).
27) Ohsuga, A., Nagai, Y., Irie, Y., Hattori, M. and Honiden, S.: PLANGENT: An Approach to Making Mobile Agents Intelligent, *IEEE Internet Computing*, Vol.1, No.4, pp.50–57 (1997).
28) 長 健太，入江 豊，大須賀昭彦，関口勝彦，本位田真一：組み込み機器向け知的移動エージェント μPlangent を用いた電力系統巡視システム，電子情報通信学会論文誌，Vol.J85-D-1, No.5, pp.465–475 (2002).
29) Hayashi, H., Cho, K. and Ohsuga, A.: Mobile Agents and Logic Programming, *IEEE International Conference on Mobile Agents*, pp.32–46 (2002).
30) Hayashi, H., Cho, K. and Ohsuga, A.: A New HTN Planning Framework for Agents in Dynamic Environments, *International Workshop on Computational Logic and Multi-Agent Systems* (2004).
31) Hayashi, H., Cho, K. and Ohsuga, A.: Speculative Computation and Action Execution in Multi-Agent Systems. ICLP Workshop on Computational Logic in Multi-Agent Systems, *Electronic Notes in Theoretical Computer Science*, Vol.70, No.5 (2002).

(平成 18 年 3 月 30 日受付)
(平成 18 年 10 月 3 日採録)

長 健太（正会員）

1995 年早稲田大学理工学部情報学科卒業，1997 年同大学大学院理工学研究科修士課程修了．同年（株）東芝入社．現在，同社研究開発センター知識メディアラボラトリー所属．エージェント技術，コンテクストアウェア技術の研究開発に従事．日本ソフトウェア科学会，電気学会各会員．

大須賀昭彦（正会員）

1981 年上智大学理工学部数学科卒業．同年（株）東芝入社．1985〜1989 年（財）新世代コンピュータ技術開発機構（ICOT）に出向．現在，（株）東芝ソフトウェア技術センター先端ソフトウェア開発担当グループ長．工学博士（早稲田大学）．2002 年より電気通信大学大学院客員助教授，2003 年より同大学院客員教授兼任．2002 年より大阪大学大学院非常勤講師兼任．主としてソフトウェアのためのフォーマルメソッド，エージェント技術の研究に従事．1986 年度情報処理学会論文賞受賞．電子情報通信学会，日本ソフトウェア科学会，IEEE CS 各会員．

本位田真一（正会員）

1978 年早稲田大学大学院理工学研究科修士課程修了．（株）東芝を経て 2000 年より国立情報学研究所教授，2004 年より同研究所アーキテクチャ科学研究系研究主幹を併任，現在に至る．2001 年より東京大学大学院情報理工学系研究科教授を兼任，現在に至る．2002 年 5 月〜2003 年 1 月英国 UCL ならびに Imperial College 客員研究員（文部科学省在外研究員）．2005 年度パリ第 6 大学招聘教授．早稲田大学客員教授．工学博士（早稲田大学）．1986 年度情報処理学会論文賞受賞．ソフトウェア工学，エージェント技術，ユビキタスコンピューティングの研究に従事．IEEE，ACM 等各会員，日本ソフトウェア科学会理事，情報処理学会理事を歴任．日本学術会議連携会員．