

GRACE TECHNICAL REPORTS

XACプロジェクト

オープン無線センサーネットワークミドルウェア研究開発プロジェクト

鄭顕志，末永俊一郎，中村善行，清雄一，吉岡信和，
深澤良彰，本位田真一

GRACE-TR 2009-02

2009年5月



CENTER FOR GLOBAL RESEARCH IN
ADVANCED SOFTWARE SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF INFORMATICS
2-1-2 HITOTSUBASHI, CHIYODA-KU, TOKYO, JAPAN

WWW page: <http://grace-center.jp/>

テクニカル・レポートは、国内外の論文誌、Proceedings 等への投稿原稿、マニュアル、資料、研究の中間報告です。著作権は、全て著者に属します。ただし、同一あるいは類似の論文が外部の論文誌等で発行される場合はホームページへの掲載等中止することがあります。その場合、著作権者が学会等に変更される場合もあります。

The GRACE technical reports are published as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, notwithstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

XAC プロジェクト

オープン無線センサーネットワークミドルウェア研究開発プロジェクト

鄭 顕志 *† 末永 俊一郎 * 中村 善行 * 清雄一 ◇

吉岡 信和 † 深澤 良彰 * 本位田真一 †‡

* 早稲田大学 † 国立情報学研究所 ‡ 東京大学

* 日本ユニシス株式会社 ◇ 株式会社三菱総合研究所

2009 年 5 月 27 日

概要

本稿では無線センサーネットワークに関する研究プロジェクトである XAC プロジェクトを、研究、開発の観点から紹介する。XAC プロジェクトでは、無線センサーネットワークをオープンインフラとして利用可能にすることを目指し、無線センサーネットワークのミドルウェアに関する研究、開発を行っている。本稿では、XAC プロジェクトが取り組む研究課題の研究概要や XAC プロジェクトの取り組みを示す。また、XAC プロジェクトが開発するミドルウェアの現状を示す。

1 はじめに

環境中に埋め込まれた情報通信機器を制御し、ソフトウェアによって人間の生活空間を支援するパーベイシブコンピューティング [1](または、ユビキタスコンピューティング [2]) への期待が高まっている。パーベイシブコンピューティング環境では、環境の状態や利用者の意図に応じて適応的に振る舞う状況依存型アプリケーションが複数稼働する。状況依存型アプリケーションは自身の振る舞いに関連する環境情報を必要とするため、個々の状況依存型アプリケーションを効率的に開発するためには、環境情報を収集する共通基盤が必要となる。

環境情報を収集する共通基盤は、実環境の様々な箇所へ設置可能で、かつ、様々な状況依存型アプリケーションが必要とする環境情報を収集可能でなければならない。無線センサーネットワーク (Wireless Sensor Network : WSN) は、パーベイシブコンピューティングのための環境情報を収集する共通基盤として期待されている。WSN とは、センサーを備えた小型コンピュータで構成される無線アドホックネットワークである。以降、WSN

を構成するコンピュータをノードと呼ぶ。ノードは無線通信デバイスを備え、ノード間で無線アドホックネットワークを構築し、構築したネットワークを転送経路として観測データを報告することができる。従って、WSNは設置する際に新規の有線ネットワークや電源を確保する必要がなく、屋内、屋外への設置が容易である。また、ノードは複数種類のセンサーを備え、観測に使用するセンサーや観測間隔等を変更することができる。従って、ノードを制御するプログラムを変更することによって、状況依存型アプリケーション毎に異なる環境情報を1つのWSNから取得することができる。パーベイシブコンピューティング環境を構築するためには、WSNをオープン環境でのインフラとして利用可能にすることが必要となる。

WSNをオープン環境でのインフラとして利用可能にするためには、WSNを管理するミドルウェアが必要となる。WSNミドルウェアは次のような要求に答えなければならない。

- 多様なタスクの扱い
- 運用時のタスク管理
- WSNの限られた資源の有効活用
- 環境変化への対応

まず、必要となる環境情報や、求める精度によっては変わる。期待する環境情報を取得するための処理を以降タスクと呼ぶ。WSNは状況依存アプリケーション毎に必要とされる多様なタスクを扱えなければならない。また、オープン環境で動作するためには、システム運用時にタスクの追加・削除が行えなければならない。さらに、WSNを構成するノードは、計算資源、メモリ、通信帯域、バッテリー容量が乏しい。多数のタスクを長時間実行可能にするためには、タスクの消費資源量を抑える必要がある。加えて、ノードの追加・消失などによるWSN環境変化や、タスクが対象とする実環境の変化によってWSN内での最適な処理は変わる。従って、環境変化に対応してWSN内の構成、振る舞いを変更する必要がある。

XACプロジェクト¹では、パーベイシブコンピューティング環境構築のために、オープン環境下で動作するWSN管理ミドルウェアの研究、開発を行っている。本論文では、オープン環境におけるWSN管理ミドルウェアの研究課題を示し、XACプロジェクトの取り組みを紹介する。

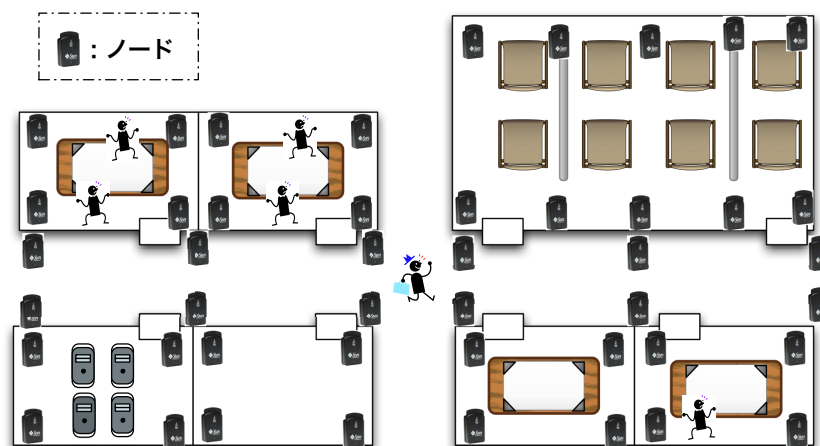


図 1: オフィスビルでの環境例

2 シナリオ

本章では、オープン無線センサーネットワークの課題を明確にするために、WSN タスクの具体例を示す。あるオフィスビルで業務効率化のためにパーベイシブコンピューティング環境を構築し、状況依存型アプリケーションを実行させようとしている。オフィスビル各階の部屋、廊下、各階を繋ぐ階段には、図 1 のようにノードが設置され、オフィスビルの環境情報を観測、取得できる WSN が稼働しているものとする。WSN 内のノードで観測された環境情報は、ノード間の無線マルチホップ通信を経由して、WSN を管理するサーバ（以後、ベースステーションと呼称する）に集められる。

ベースステーションに集められたオフィスビルの環境情報を用いて状況依存型アプリケーションが動作しているシナリオ S1～S4 を表 1 に示す S1 では滞在者が希望する温度に会議室の室温を調整するアプリケーションが、S2 では会議室の空き状況を予約データや現在の滞在者の有無から判定するアプリケーションが、S3 では各社員のプレゼンスを管理するアプリケーションが、S4 では不審者検知アプリケーションが動作している。

状況依存型アプリケーションはそれぞれ自身の動作に関連する環境情報を必要とする。S1 では室温、S2 では会議室内の滞在者の有無、S3 では各社員の現在地、S4 では部屋周辺の滞在者の有無といった環境情報を必要とする。

S1～S4 はそれぞれに必要とする環境情報と要求する精度が異なる。各ノードが備えるセンサーが観測したデータには誤差が含まれる。高い精度

¹<http://xac-project.jp>

表 1: 状況依存型アプリケーションシナリオ例

番号	アプリケーション名	動作内容	環境情報	要求精度
S1	室温管理	滞在者の希望温度に室温を調整	指定された部屋の温度	誤差 2 度以内
S2	会議室空き状況管理	会議室の予約状況や滞在者の有無から、実際に空いている会議室を表示	指定された部屋内の人物の有無	誤検知率 1% 以内
S3	プレゼンス管理	各社員がどの部屋にいるかを管理	指定された人物の位置	デスク室では 1m 以内の誤差、会議室では部屋単位で
S4	不審者検知	重要な物品、情報を管理する部屋の前で認証動作をせずに長時間滞在している人物がいた場合、警備員に通知	指定された部屋周辺の人物の位置	誤差 2m 以内

の環境情報を得るためには、一般に、複数のノードによって観測を行い、その観測結果を統合することによって精度を向上させる必要がある。例えば、S3 では対象人物が会議室にいる場合、会議室中にいるかどうかはわかればよいので、高い精度は要求されず使用するノードは少数でよい。一方で、S4 では誤差 2m 以内で人物の位置を観測しなければならず、多数のノードからの観測結果を統合する必要がある。

一方で、WSN 管理者の視点に立つと、同時に実行できるタスク数や WSN の稼働時間を向上させるために個々のタスクが消費する資源量を抑える必要がある。WSN を構成するノードの CPU、メモリ、ネットワークデバイスは貧弱で、計算資源、保存容量、通信帯域は乏しい。また、ノードがバッテリー駆動している場合、個々のノードの負荷が高まると、短時間でノードがバッテリー切れになってしまう。例えば、広く使用されているノードである Crossbow 社の Mica2 は、1 秒に 1 回計測した温度を送信す

る場合、ノードのバッテリーは約7日で枯渇してしまう [3]。WSN の生存時間を向上させるためには、観測データを複数まとめて送信したり、精度を落とさない程度に観測頻度を落としデューティサイクルを調整するなど、様々な最適化手法を使用することでノードの生存時間を向上させる必要がある。

3 オープン WSN ミドルウェアの研究課題

XAC プロジェクトでは、オープン WSN のためのミドルウェアに関連して、特に次の4点の研究課題の解決に取り組んでいる。

- タスク記述言語
- タスクの配置管理
- 自己適応性
- セキュリティ

本章では、上記の4つの観点から、既存研究の取り組みや課題を示し、XAC プロジェクトの取り組みを紹介する。

3.1 タスク記述言語

WSN ミドルウェアを用いて開発を行う場合、ミドルウェアで提供されているタスク記述言語を用いてタスクを記述する。タスク記述言語は、ミドルウェアを通じて WSN を利用するためのインタフェースとなる。

3.1.1 既存の取り組み

タスク記述言語によってプログラマに対して提供する API の抽象度が異なる。抽象度が高ければ、WSN 内部の複雑さをプログラマから隠蔽し容易にタスクを記述できるようになるが、一方で、記述可能な処理の自由度が減るため、観測結果の精度を向上し資源消費量を低減させるために WSN 内部特有の性質に基づいた最適化の余地は少なくなる。これまでに様々なタスク記述言語が提案されているが、それぞれ抽象度が異なる。本節では、既存のタスク記述言語を抽象度の違いによってデータレベル、グループレベル、ノードレベルに分類し、紹介する。

データレベル TinyDB[4], Cougar[5], TinyLIME[6], TeenyLIME[7] などのミドルウェアでは、データレベルのタスク記述言語を提供している。データレベルのタスク記述言語では、期待する環境情報を取得するために必要なデータを記述する。

これらのミドルウェアでは抽象化の方法はそれぞれ異なる。例えば、TinyDB, Cougar は WSN を関係データベースとして抽象化し、SQL ベースのタスク記述言語での観測データ取得を、TinyLIME, TeenyLIME は WSN をタブルスペースとして抽象化し、タブルの問い合わせ言語ベースのタスク記述言語での実現している。

データレベルのタスク記述言語では、取得したいデータの性質のみを記述し、通信経路の構築や、ネットワーク内集約など WSN 内部の最適化はプログラマに対して透過的に実行される。例えば、TinyDB では、WSN 内部でツリー構造のネットワークトポロジを構築し、構築したトポロジに沿って観測したデータを収集する。また、ツリー構造に沿って多数のデータを集める際に、転送途中のノード上で観測結果を集約することで通信量を減らす最適化を行っている。従って、処理は簡易に記述できるが、例えばツリー構造からクラスタリングへのトポロジに切り替えなどはプログラマは記述することができない。

グループレベル EnviroTrack[8], Hood[9], AbstractRegion[10], Generic Role Assignment[11], DFuse[12] といったミドルウェアでは、グループレベルのタスク記述言語を提供している。グループレベルのタスク記述言語では、必要なデータを計測するタスクを、複数のノード上で実行されるマクロ動作として記述する。

EnviroTrack, Hood, AbstractRegion では、観測対象近隣のノード群を 1 つのグループとして抽象化し、グループによるマクロな動作をタスクとして記述する。ホップ数などでグループを構成するノードを設定し、グループ内のノード群上で行う、観測、転送、集約処理を集中的に記述できる。例えば AbstractRegion では、グループ内での通信経路もプログラマがカスタマイズすることができる。Generic Role Assignment のタスク記述言語では、ノードの動作を役割として記述し、ノードは指定された条件に応じて役割を切り替えることで動作を変更することができる。また、DFuse では、観測、転送、集約、データ保存などのタスク全体を構成するサブタスクグラフを記述し、サブタスクが配置されるノードの条件をグラフの制約として記述することができる。

グループレベルのタスク記述言語では、データの性質ではなく、データの観測方法を記述する。ただし、個々のノードの動作ではなく、タスク全体を記述し、ミドルウェアが個々のノードに分割、実行する。グループレベルのタスク記述言語では、さらに、タスクを構築する要素間でのネット

ワークトポロジや，ネットワーク内集約手法などをカスタマイズ可能となっている．

ノードレベル TinyOS[13] , Squawk[14] , Agilla[15] , SensorWare[16] , ActorNet[17] といったミドルウェアでは，ノードレベルのタスク記述言語を提供している．ノードレベルのタスク記述言語ではグループベースよりさらに抽象度を下げ，個々のノードの動作を記述する．

TinyOS や Squawk のタスク記述言語では，個々のノードの動作を記述する．Agilla , SensorWare , ActorNet のタスク記述言語では，モバイルエージェントベースの処理を記述し，個々のノードを渡り歩いて実行する処理を記述することができる．

ノードレベルのタスク記述言語では，個々のノードのデューティサイクルなど，詳細な動作のカスタマイズを可能にしている．一方で，タスクを構築するサブタスク毎も個々のノードの動作として記述しなければならず，トポロジ構築やネットワーク内集約などの処理を分散プログラミングとして記述しなければならず，開発の難易度は高くなってしまふ．

3.1.2 XAC プロジェクトによる取り組み

前節で示した既存のデータレベル，グループレベル，ノードレベルのタスク記述言語は抽象度がそれぞれ異なり，記述容易性とプログラマによる最適化の余地のトレードオフとなっている．アプリケーションが要求する精度，消費資源量でタスクを記述することができる，最も高い抽象度の言語を使用するのが望ましいが，求められる精度や消費資源量はアプリケーションによって異なるため，アプリケーションによって適切な抽象度は異なる．

しかしながら，既存のミドルウェアでは抽象度が固定であるタスク記述言語しか提供しておらず，多様なアプリケーション要求に応えることができない．例えば，データレベルのタスク記述言語では，サブタスク間のネットワークトポロジを記述することができない．ネットワークトポロジをスパニングツリー型にするのか，クラスタ型にするのか，階層型クラスタにするのかでデータ収集効率が異なる．適切なトポロジによる処理の非機能特性は，計測対象の範囲，計測データの種類などのタスクの特徴に依存する．従って，タスク毎に要求精度，使用資源量を満たすようにネットワークトポロジを選択する必要がある．ネットワークトポロジはデータレベルのタスク記述言語での抽象度では隠蔽されているため，グループレベル以下の抽象度のタスク記述言語でのみ最適化できる．

XAC プロジェクトでは，複数の抽象度でタスクを記述可能なタスク記述言語に関する研究を行っている．データレベル，グループレベル，ノー

ドレベルでそれぞれタスク記述言語を提供し、それらのレベルでのモデル間で変換してタスクを開発するモデル駆動開発を行う。データレベルで記述したタスクからグループレベルのタスクへ、グループレベルで記述したタスクからノードレベルのタスクへとモデル変換することでタスクを段階的に詳細化して開発する XAC プロジェクトでは、これまでに、既存の WSN ミドルウェアやタスク記述言語を調査し [18, 19, 20]、タスク記述言語に必要な記述能力の指標を導出する分析手法を提案している [21]。現在は、各レベルでの記述モデル、モデル間の変換ルールの定義に取り組んでいる。

モデル変換によって抽象度の低いレベルにモデルを変換後、非機能的要求を満たすようタスクの最適化を行う必要がある。データレベルではデータ解析に関する、グループレベルではグループ構成やグループ内通信に関する、ノードレベルでは個々のノードの振る舞いに関して、適切な実現手段を選択、適用することで必要 n な精度と消費資源量の調整を行うことができる。例えば、2章で示した S2, S4 のタスクは人物の現在地を観測するが、それぞれ要求する精度が異なる。S2 のように要求精度が低い場合は、単純に対象からのビーコン受信方式で十分だが、S4 のように要求精度が高い場合は、電波強度を用いる RSSI (Radio Signal Strength Indicatin) 方式や電波の到達時間差を用いる TDoA (Time Difference of Arrival) 方式を用いて複数のデータを統合、解析することで精度を向上させる必要がある。従って、S2 と S4 では同じ環境情報を必要とするが、それぞれのタスクのデータレベルの処理は異なる。また、データレベルで RSSI を用いる場合、グループレベルではどのノードが観測するのか、観測したデータをどのノードに集めるのか、どのような転送路を用いるのかによって精度や消費資源量が変化するため、適切に選択、決定する必要がある。このように、データレベル、グループレベル、ノードレベルでそれぞれ、精度と消費資源量を調整するための最適化を支援する手法が必要となる。

XAC プロジェクトでは、各レベルの抽象度におけるプログラマによる処理の最適化を支援するための研究にも取り組んでいる。XAC プロジェクトでは、これまでにプログラマによる最適化を支援するためのパターンカタログの整備 [22] を行ってきた。今後は、パターン適用によって向上する非機能特性を明らかにし、要求する精度、消費資源量を達成するために必要なパターン選択手法を扱う。また、下位レベルでの最適化によって処理ロジックを変更すると、上位レベルで記述したタスクの内容を逸脱してしまう危険性がある。従って、最適化後のタスクが上位レベルでの記述内容で記述した性質を保持しているかを検証する必要がある。XAC プロジェクトでは、モデル間の整合性検証についても取り扱う。

3.2 タスクの配置管理

WSN を用いてタスクを運用する際、タスクをノードに配置する必要が生じ、タスクの配置管理が必要になる。一般に WSN では、タスクをノードに配置する場合、ノードとベースステーションを物理的に接続する必要が生じる。WSN を敷設した後にタスクの追加・変更・削除等でタスクを再配置する必要が生じた場合、ノードを一度回収し、物理的な接続を実施すると運用に支障を来す。そこで、ベースステーションからノードにタスクを無線経路で配置するリプログラミング [23] が提案され、タスクの運用時に一般的に用いられるようになった。

3.2.1 既存の取り組み

リプログラミングは以下の 2 つの手法に分類できる。

ベースステーションによる配置管理

ベースステーションから、タスクの配置を集中管理する手法。

タスク自身による配置管理

タスク自身がタスクの配置を決定し、移動を行って配置管理を実現する手法²。

ベースステーションによる配置管理 タスクの配置管理を実施する場合、タスクを全ノードに配置するか、タスクを実行するノードのみに配置するかを考慮する必要がある。例えば、表 1 で挙げた S1 は室温を観測する部屋のみに配置され、S3 は全ノードに配置されることが妥当となる。仮に、観測条件の変更等で S1, S3 のタスクの変更が必要になった場合、S1 についてはタスクを実行するノードのみタスクを更新し、S3 については全ノードのタスクを更新する必要があり、タスク更新の対象範囲を考慮する必要が生じる。

タスク更新の対象範囲を指定し、ベースステーションから配置管理を行う代表的な研究例に Deluge[24] がある。Deluge は、更新対象となるタスクをベースステーションから WSN の全ノードに送信する際に、更新対象となるノード ID をオプションとして指定することで、タスク更新の対象範囲を指定することを可能にしている。WSN を構成するノードは、電力・通信帯域が限られているために、Deluge では以下 3 点の工夫を行っている。

²ベースステーションから集中管理することもできるが、タスク自身が配置管理を実施することができる。

- 通信送信間隔の動的な変更により，ノード密集地域での通信集中を削減する．
- 安定したノードとの通信を動的に実行することによって，ノード間の非対称リンクへの対処を行う．
- ラウンドを導入することによる，複数ノードによるブロードキャスト集中を抑制する．

Deluge は，タスクに必要な全コードをベースステーションから送信する手法を取る．この手法では，コード容量が大きくなり，コードの通信・転送コストが大きく，ノードの電力を消費する課題がある．そこで，タスクに共通するコードを予め各ノードに配置しておき，タスク固有のコードのみ配置する手法が研究された．代表的な研究例に Maté[25] がある．Maté[25] は，各ノードに Virtual Machine を予め配置しておき，VM が解釈可能なタスク固有のコードを送信することでコードの軽量化を実現している．Maté の VM は，bytecode のインタプリタであり，各 bytecode は 1byte で表現される．この結果，100 行のスクリプトであっても 100byte で記述することを可能にしている．なお，Maté[25] は，タスク更新の対象範囲を指定することができないが，改訂版である Trickle[26] によってタスク更新の対象範囲の指定を可能としている．

タスク自身による配置管理 ベースステーションによる配置管理を行う場合，ベースステーションから常にタスクを配置する必要が生じる．この手法は，全てのタスクを全てのノードに予め配置しておくことが出来ない場合にタスクの稼働場所が変更すると，ベースステーションからタスクを再配置することが必要になる．例えば，表 1 で挙げた S4 では，重要な物品が移動する毎にタスクの再配置を行うことが必要になる．この結果，ベースステーション周辺のノードに通信トラフィックが集中する課題が生じる．タスク自身が物品の移動に対応し，移動を行ってタスク自身の配置を管理できれば，ベースステーション周辺の通信トラフィックの集中を削減することができる．³ こうした背景から，タスク自身がタスクの配置管理を実施する研究が行われた．代表的な研究例に Agilla[15]，ActorNet[17]，SensorWare[27] がある．

Agilla は，Maté で提供された VM を拡張し，タスクの移動を可能にして，タスク自身による配置管理を実現した研究例である．Agilla におけるタスク

³ベースステーションとタスクが配置されたノードを結ぶ線の垂直二等分線で WSN を区切った場合，タスクが配置されたノードに近い領域に移動する場合はタスクが配置されたノードから移動した方が移動距離が短い．空間的な平均をとればベースステーションから常に配備するより平均移動距離は短くなり，WSN 全体での通信コストを削減することができる．

は、VMで解釈可能な1byte~2byteのISA(Instruction Set Architecture)で記述され、Matéと同様にタスクのコード容量を削減している。Agillaは、タスク間通信を分散タブルスペース [28] を用いて実現することも可能としている。

ActorNetは、actorモデルをWSNに導入した研究例であり、各actorはActorNetで提供されるSchemeインタプリタ上で動作する。ActorNetでは、仮想メモリの提供、SchemeインタプリタとTinyOS間のIOスケジューリングの同期、ガーベッジコレクションの提供という3つの工夫を行っている。

SensorWareは、タスクの移動とタスク自身による配置管理を実現する点で、AgillaおよびActorNetと共通している。ただし、標準的なプラットフォームとしてiPAQを想定しており、1Mbyte程度のメモリと128Kbyte程度のメモリが稼動環境として必要になる。WSNのノードとしては潤沢な計算資源(数百MHzのCPU及び1Mbyteのメモリ)を必要とすることから、一般的に想定されるWSNの計算資源とは乖離がある。

3.2.2 XACプロジェクトによる取り組み

XACプロジェクトでは、オープンな環境下でWSNが利用され、複数のシステム利用者が存在することを想定している。本想定環境下では、既存のWSNに対し、利用者がタスクを追加・変更・削除することが通常となる⁴。また、表1で挙げたS4のように、稼動場所を変更するタスクも想定している。以上の経緯から、XACプロジェクトでは、3.2.1節で述べたタスク自身による配置管理に着目して来た。3.2.1節で挙げた既存研究では、タスクを複数のノードに存在するコンポーネントで構成することは可能である。例えば、表1で挙げたS1(室温管理)の場合、部屋に存在する複数のノードに、本タスクを構成するコンポーネントを配置させることができる。ただし、複数コンポーネントの移動や、複数コンポーネントの動的再構成の観点欠缺しており、タスク運用時の課題となっていた。そこで、XACプロジェクトでは、タスクが複数のコンポーネントで構成される場合に着目し研究を行ってきた。

複数コンポーネントの移動 GDD(Generative Dynamic Deployment)[29],[30]は、“タスクを構成する複数のコンポーネント”を移動させるミドルウェアである。GDDでは、複数のコンポーネントで構成されるアーキテクチャと移動手法を提案している。既存研究では、配置されるコンポーネントに適するアーキテクチャが存在せず、移動の信頼性が低下する課題がある⁵。

⁴予め全利用者の全タスクを配置しておくことはできない

⁵WSNの通信はパケットロスがあることが知られている [31]。

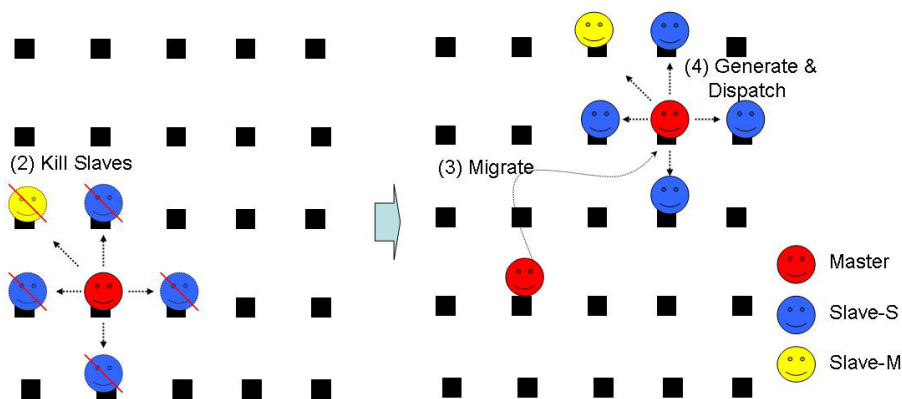


図 2: GDD の動作例

GDD はこの課題に対し、*Master*、*Slave-S*、*Slave-M* と呼ばれる 3 つのコンポーネントから構成されるアーキテクチャを提案し、移動の信頼性を向上させている。加えて、複数のコンポーネントが移動を実施する際、コンポーネント間通信が頻発することで移動の信頼性が低下する課題がある。GDD は、この課題に対し、*Master* が他コンポーネント (*Slave-S, M*) を生成する手法を提案し、移動時に *Master* だけが移動を実行し、必要に応じて他コンポーネントを生成することで、移動の信頼性を向上している。なお、GDD では移動前に *Master* が *Slave-S, M* を消去する手法を採用している。

図 2 に GDD の動作例をしめす。

1. *Master*、*Slave-S, M* が配置される。
2. *Master* は移動が必要になった際、配置された *Slave-S, M* を消去する。
3. *Master* は次の配置場所に移動する。
4. *Master* が移動後、*Slave-S, M* を生成し、*Slave-S, M* を周辺ノードに配置する。

これを繰り返すことで、継続したタスクの実行を可能にしている。GDD を用いることで、複数のコンポーネントで構成されるタスクの実現の現実性が増す。

コンポーネントの動的再構成 Platon ら [32] は、観測を実行する複数のコンポーネントで構成されるタスクを配置し、配置されたタスクがノードの状態に応じ再構成されるミドルウェア Ragilla を提案している。既存研究では、ノードのバッテリーが枯渇した場合や、ノードの故障が生じた場合

に、タスクを構成するコンポーネントがノード上で活動を停止し、プログラマの要求を満たさない課題がある。Ragilla では、タスクを構成するコンポーネントの稼動条件（コンポーネント数、距離範囲、稼動するバッテリーレベル）を定義し、各コンポーネントが稼動条件を満たしながら移動や複製を行うことで、この課題を解決している。

例えば、図3に示すように、ハードウェアの故障が生じた場合、次のような処理を行うことで、一定の距離範囲内でコンポーネントが動的に再構成され、継続して対象物を計測できる。

1. ハードウェア障害によってリーダーが稼動しているノードの活動が停止する。
2. ミドルウェアがハードウェアの故障を検知し、コピーの中から次のリーダー候補を選定する。
3. ミドルウェアがリーダー候補を指定された範囲内にコピーする。

本手法を用いることで、利用者のタスクに対する要求を満たしながらタスクを構成し、タスクの現実性を向上することができる。

GDD, Ragilla だけでなく、現在 XAC プロジェクトでは、タスクを構成する複数のコンポーネントの役割分担を詳細化し、各コンポーネントの役割に基づき適切な配置条件を定め、WSN の状態に応じて動的に再配置を行う再配置管理機構に取り組んでいる。

3.3 自己適応性

WSN では、ノードが利用できる計算資源やバッテリー残量などのノード内の状況は時間と共に変化することが予想される。また、障害物の移動による通信環境の変化や、観測対象の移動など、ノード外の状況も変化することが予想される。このようにノード内外の状況が変化した場合、3.1 節で述べたようにプログラマの意思を反映させることで状況に応じて最適

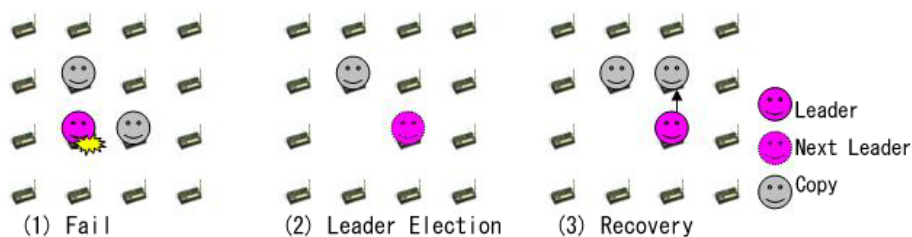


図 3: Ragilla の動作例

化させることが出来る。しかし、状況が変化する度にプログラムの判断を仰いでいると、プログラムの負担が大きくなってしまふ。そこで、アプリケーションに依らずミドルウェアで判断可能な部分に関しては、ミドルウェア側で自己適応を行うことでプログラムの負担を軽減することが出来る。

3.3.1 既存手法の取り組み

既存研究では以下の点について自己適応の研究を行っている。

ノード内部の変化に対する自己適応 ノードの持つ計算資源は乏しく、同時に複数のタスクを実行する場合には処理能力が不足することが考えられる。このため、WSNに複数のタスクを配備する場合、各タスクが独自に配備ノードを決定すると、ノード上で資源競合が発生し、タスクの要求する処理が実行できないことが考えられる。また、3.2節で述べたように観測の品質を高めるためには、1つのタスクに対して複数のノードを使用して観測を行うことが求められる。このため、オープン環境でのWSNではノードにおける資源競合が頻繁に発生すると考えられる。

タスクの資源競合を回避するためには、タスクを配備する際に配備先のノードの決定や、時間によるセンサー使用のスケジューリングを決定に関する研究が必要となる。これらの研究では、配備するタスクの要求を満たすために必要なノードグループの条件を考慮した上で、配備先のノードやセンサーのスケジューリングを決定することが必要となる。アプリケーションの配備に関する研究として、[33, 34]などがある。

ノード外部の変化に対する自己適応 パーベイシブな環境におけるWSNでは、タスクを配備するノードを状況に応じて変更する必要がある。そのため、ノード間の通信リンクの強さや周囲のノードの密度など、タスクが動作するノードの周囲の環境は時間により変化することが考えられる。また、同一ノード上でも障害物の移動による通信環境の変化や観測対象の移動などにより、ノード外部の状況が変化することが考えられる。このような場合、タスクに要求される品質を得るためには、タスクで用いるアルゴリズムや内部パラメータを環境に適したものに変更することが必要となる。

既存研究ではこのような問題に対して、通信アルゴリズムのパラメータ調整により環境に適応させている [35, 36, 37]。例えば、ルーティングツリー上のノードと通信リンクが切れてしまった場合、別の通信可能なノードとツリーを再構築することで適応することが出来る。

3.3.2 XAC プロジェクトの取り組み

XAC プロジェクトでは上記の研究分野に対して以下のような研究を行っている。

ノード内部の変化に対する自己適応 XAC プロジェクトではノードにおける資源競合を回避するために、タスクの要求を満たした必要最小限のノードグループの構成 [38] や、資源競合発生時のタスクの退避機構に関する研究を行っている [39]。

2章のシナリオのように、WSN では求められる精度・優先度の異なるタスクが共存している。また、これらのタスクは全ての部屋に配備する (T2)、人のいる部屋にのみ配備する (T1,T3)、部屋の中であればどのノードでも良い (T2)、などの条件が存在する。タスクの退避機構では、このような条件に基づいてタスクを配備し、競合が避けられない場合にはタスク毎の優先度がより低いものを他ノードへ移動させる。

本手法を用いることによりタスクの資源競合の可能性を抑え、資源競合が発生した場合にはタスクが別ノード上で処理を実行できるかを判定し、タスクを退避することが可能となる。

ノード外部の変化に対する自己適応 XAC プロジェクトでは環境の変化に対する自己適応として、環境に応じて通信アルゴリズムそのものを変更する研究を行っている。タスクの動作するノードの変更やモノの移動などにより、タスクが動作するノード周辺の環境が変化した場合、それぞれの環境に適したアルゴリズムを選択することで通信の信頼性・通信量を向上させることが出来る。2章のシナリオのように社員の現在地を報告させるタスクを考えると、社員が別々の場所に居た場合には様々な部屋のノードから報告が来るのに対し、会議中のようにほとんどの社員が同一の部屋に居た場合には同じノードから頻繁に報告が来ると思われる。このように報告頻度が異なる場合にはそれぞれ有効となるルーティングやMAC プロトコルに変更することで、より効率の良い通信を行うことが出来る。

また、XAC プロジェクトではアルゴリズム内のパラメータ調整に関する研究も行っている [40, 41]。ノードグループから報告を受ける場合、グループ内で一度データを集約することで通信量を抑えることが出来る。そこで、グループ内のどこで・どのようにデータ集約することで効率よく通信が行えるのかを調べることで、ルーティングアルゴリズムが変化した際にも適切なデータ集約を行うことが可能となる。

これらの研究を用いることで、タスクが動作するノードが変更され、ノード周辺の環境が変化した場合にも、環境に適したアルゴリズムに変更・修正することが出来、タスクの品質を向上させることが出来る。

3.4 セキュリティ

WSNの実運用を行うためには、セキュリティの問題に取り組む必要がある。たとえば不審者検知を行う場合、攻撃者が、不審な行動を検知できなくなるような攻撃をWSNに与えることが想定される。本章では、WSNならではのセキュリティ課題を中心に挙げ、現在の研究動向と今後の課題を述べる。

3.4.1 WSNにおけるセキュリティ課題

WSNにおいては、攻撃者がセンサーノードを物理的に取得する脅威が存在する。コストの制約から、ノードは耐タンパハードウェアを備えていないことが多い。したがって、攻撃者は物理的に取得したノードから、ノードの秘密鍵等の全てのデータを取得することができる。このノード不正取得攻撃が、WSNにおけるセキュリティ課題の解決を困難なものにしている。

また、ノードの特徴としてバッテリー駆動であることが挙げられる。したがって、できる限りメッセージ通信量や計算量を削減する必要がある。また、計算量の削減を行うためには次のような課題が存在する。従来のネットワークでは公開鍵暗号化方式が広く用いられているが、公開鍵暗号化方式は共通鍵暗号化方式に比べて計算量が大幅に増加してしまう。そのためWSNでの利用はできるだけ避けることが必要である [42, 43]。

また、ノードは故障しやすいという性質を持つ。多数のノードが故障しても機能するセキュリティ手法が必要である。

3.4.2 WSNにおける攻撃

セキュリティの三要素である機密性・完全性・可用性について、主に次のような攻撃が存在する。

機密性: ノード間通信の盗聴 [44]

完全性: メッセージの改ざん [45], 不正メッセージの作成 [46]

可用性: 不正メッセージの大量作成による DoS 攻撃, メッセージの破棄 [47], 誤ったルーティング情報の挿入 [48]

また、これらの攻撃の元となる攻撃として、複製ノード攻撃(一つのノードを不正取得し、そのデータを用いて、多数の複製を作成する [49, 50]) や誤った位置情報を与える攻撃(複数ノードが協力して位置情報を取得する際、誤った情報を与える [51]) 等の攻撃も存在する。

3.4.3 既存研究における取り組み

機密性: ノード間通信の盗聴に関しては, 隣接ノード間において共通の鍵を保持して暗号化通信を行うことにより, 悪意のあるノードを通らない限りメッセージの機密性を保証できる. 鍵をノードにどう振り分けるかについてはすでに多くの研究が行われている [44, 52, 53]. 中継ノードがメッセージの中身を知る必要が無いのであれば, メッセージ作成元ノードがベースステーションと共有する鍵でメッセージの暗号化を行えば良い.

完全性: メッセージの改ざんに関しては, 複数のメッセージを作成していくつかのルートで送信することである程度対応可能である [54, 47]. メッセージがベースステーションへ到達する確率を高いレベルで維持しながら, できるだけメッセージ数を減らすことが求められる. また, メッセージ作成元ノードがベースステーションと共有する鍵でメッセージ認証コードを付加することにより, メッセージの改ざん検知を保証できる. また, WSN の特徴として近接するノードは似た環境情報を取得することが挙げられる. この特徴を用いて, 不審者等のイベントを検知したノード群がメッセージ認証コードをメッセージに付加し, ベースステーションがそれらを検証することにより, 不正メッセージの検知をある程度行うことが可能である.

可用性: 不正メッセージの大量作成による DoS 攻撃に対応するためには, 不正メッセージを中継ノードが検知する必要がある. イベントを検知したノード群がメッセージ認証コードをメッセージに付加し, それらの中継ノードが検証する手法が多く提案されている [55, 46]. メッセージの破棄検知に関しては, メッセージの改ざん検知と同様, 複数メッセージをいくつかのルートで送信することによりある程度対応可能である. 誤ったルーティング情報の挿入に関しては, 既存の無線ネットワークの技術を応用可能であるが, より低コストで検知できる手法が数多く提案されている [48].

3.4.4 XAC の取り組み

これまでの WSN におけるセキュリティ手法では, 個々の問題に別々に対応しているものが多い. たとえば, 不審者検知を目的とした場合, 不審者を検知したノードが作成したメッセージをできるだけ早く確実にベースステーションへ届ける必要がある. 不正ノードによるメッセージの改ざんや破棄に対応するためには, 前節で述べたように複数メッセージをいくつ

かのルートで送信する必要がある。だがこの手法は、不正メッセージによる DoS 攻撃を容易にしてしまうし、それ自体が DoS 攻撃の一因となる。

XAC プロジェクトでは第一ステップとして、DoS 攻撃を行ったノードを既存研究よりも少ないトラフィック量で検知可能な手法を提案している [56]。だがこの手法は、いくつかの不正メッセージがベースステーションへ届いた後に利用できる手法であり、発生した DoS 攻撃をただちに制御するものではない。

また、不正メッセージを中継ノードが検知する既存手法では以下の課題が存在する。ある閾値（5 程度）のノードが不正取得されるとそれ以降の検知が不可能になってしまう [55, 46]、または、メッセージ作成元ノードからベースステーションまでの経路を限定する必要がある [57]。後者の場合、経路上のノードが故障ノードまたは不正ノードである場合はメッセージがベースステーションまで到達しない。また、複数メッセージをいくつかの経路で送信する手法の効果が限定されてしまう。XAC プロジェクトでは、これらの課題を解決する手法を提案している [58]。だが、不正メッセージをより早い段階で検知できる仕組みが必要である。

このように、XAC プロジェクトでは、正しいメッセージがベースステーションまで早く確実に届くことを目標として研究を行っている。現在は、正しいメッセージを複数ルートで送信しながら、同時に DoS 攻撃を防ぐという相反した事象をいかに扱うかについて主に取り組んでいる。

4 XAC ミドルウェアの紹介

XAC プロジェクトでは、3 章で紹介した研究成果を反映した、無線センサーネットワーク向けのミドルウェアを開発している。2009 年 5 月にはプロトタイプ版を公開し、以降段階的に各研究成果を適用していく予定である。プロトタイプ版は SunMicrosystems 社の開発したセンサーノードである SunSPOT 上に実装している。プロトタイプ版では、グループレベルのタスク記述言語と運用時のタスク配置機能を備えている。複数レベルのタスク記述言語、運用時のタスク再配置機能、自己適応性、セキュリティ機能は段階的に追加していく予定である。

4.1 プロトタイプ実装の概要

プロトタイプ版の XAC ミドルウェアの構成を図 4 に示す。XAC ミドルウェアは、ベースステーション上で動作するミドルウェア (XACMiddleware-B) と、センサーノード上で動作するミドルウェア (XACMiddleware-S) で構成される。

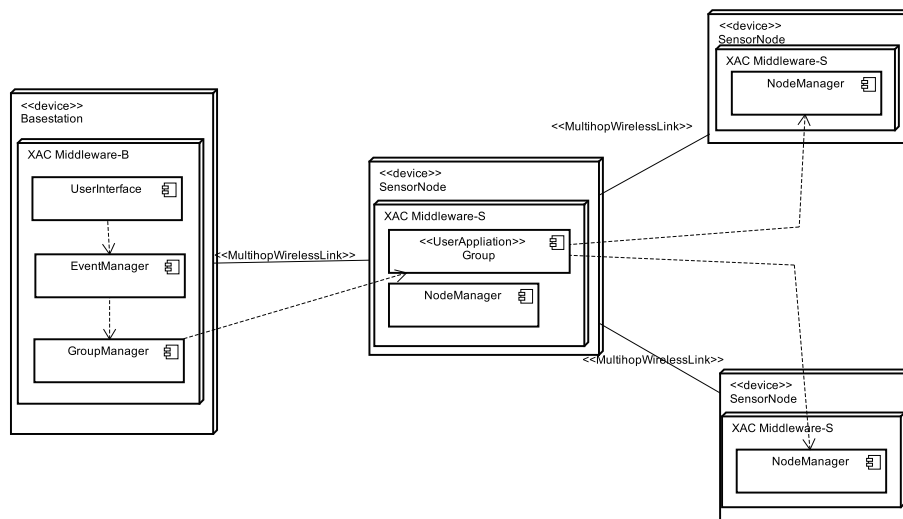


図 4: XAC ミドルウェアの構成

ベースステーション側のミドルウェアは、UserInterface、EventManager、GroupManager コンポーネントから構成される。プログラマは UserInterface を通じてタスクの登録、削除や、タスクが計測した結果であるイベントに対するハンドラの登録、削除を行う。イベントハンドラは EventManager に、タスクは GroupManager によって管理される。GroupManager はタスクに対応した Group コンポーネントを生成し、センサーノード上に配置し、計測処理を開始させる。Group コンポーネントは計測したデータが指定された条件を満たすとイベントが発生したと見なし、GroupManager を経由して EventManager に通知する。EventManager はイベントの種類に対応したハンドラを起動する。

センサーノード側のミドルウェアは、Group、NodeManager コンポーネントから構成される。Group には、Group コンポーネント自体を配備するノードの条件、データを計測するノードの条件、計測データの種類、データ統合処理を記載する。Group は GroupManager によって生成され、プログラマが指定した条件を満たすノードに配備される。Group は、計測すべきノードの条件に該当するノードを検索、選択し、該当するノードの NodeManager に、必要となるデータの計測依頼を出す。Group は NodeManager からの計測結果を集め、データ統合処理を行い、指定された条件を満たしていた場合はイベントが発生したと見なし、GroupManager に通知する。

5 まとめ

本稿では，XAC プロジェクトの紹介をした．XAC プロジェクトはオープン無線センサーネットワーク向けミドルウェアに関する研究，開発を行うプロジェクトである．XAC プロジェクトではタスク記述言語，タスクの配置管理，自己適応性，セキュリティの観点から研究を行っている．また，研究成果を反映したミドルウェアの開発を行い，オープンソースとして公開する予定である．2009年4月の時点ではミドルウェアのプロトタイプ実装に取り組んでおり，今後段階的に機能追加していく予定である．

参考文献

- [1] Mahadev Satyanarayanan. Pervasive computing: Vision and challenges. *IEEE Personal Communications*, Vol. 8, No. 4, pp. 10–17, 2001.
- [2] Mark Weiser. Some computer science issues in ubiquitous computing. *Commun. ACM*, Vol. 36, No. 7, pp. 75–84, 1993.
- [3] Victor Shnayder, Mark Hempstead, Bor-rong Chen, Geoff Werner Allen, and Matt Welsh. Simulating the power consumption of large-scale sensor network applications. In *the 2nd International Conference on Embedded Networked Sensor Systems*, pp. 188–200. ACM, 2004.
- [4] Samuel R. Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. Tinydb: an acquisitional query processing system for sensor networks. *ACM Trans. Database Syst.*, Vol. 30, No. 1, pp. 122–173, 2005.
- [5] Yong Yao and Johannes Gehrke. The cougar approach to in-network query processing in sensor networks. *SIGMOD Rec.*, Vol. 31, No. 3, pp. 9–18, 2002.
- [6] Carlo Curino, Matteo Giani, Marco Giorgetta, Alessandro Giusti, Amy L. Murphy, and Gian Pietro Picco. TINYLIME: Bridging mobile and sensor networks through middleware. In *the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 61–72. IEEE Computer Society, mar 2005.

- [7] Paolo Costa, Luca Mottola, Amy L. Murphy, and Gian Pietro Picco. Programming wireless sensor networks with the teenylime middleware. In *the ACM/IFIP/USENIX 2007 International Conference on Middleware (Middleware)*, pp. 429–449. Springer-Verlag New York, Inc., 2007.
- [8] T. Abdelzaher, B. Blum, Q. Cao, Y. Chen, D. Evans, J. George, S. George, L. Gu, T. He, S. Krishnamurthy, L. Luo, S. Son, J. Stankovic, R. Stoleru, and A. Wood. Envirotrack: Towards an environmental computing paradigm for distributed sensor networks. In *the 24th International Conference on Distributed Computing Systems (ICDCS)*, pp. 582–589. IEEE Computer Society, 2004.
- [9] Kamin Whitehouse, Cory Sharp, Eric Brewer, and David Culler. Hood: a neighborhood abstraction for sensor networks. In *the 2nd international conference on Mobile systems, applications, and services (MobiSys)*, pp. 99–110. ACM Press, 2004.
- [10] Matt Welsh and Geoff Mainland. Programming sensor networks using abstract regions. In *the 1st conference on Symposium on Networked Systems Design and Implementation (NSDI)*, pp. 29–42, 2004.
- [11] Christian Frank and Kay Romer. Algorithms for generic role assignment in wireless sensor networks. In *the 3rd international conference on Embedded networked sensor systems (SenSys)*, pp. 230–242. ACM Press, 2005.
- [12] Rajnish Kumar, Matthew Wolenetz, Bikash Agarwalla, JunSuk Shin, Phillip Hutto, Arnab Paul, and Umakishore Ramachandran. Dfuse: a framework for distributed data fusion. In *the 1st international conference on Embedded networked sensor systems (SenSys)*, pp. 114–125. ACM, 2003.
- [13] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. System architecture directions for networked sensors. *SIGPLAN Not.*, Vol. 35, No. 11, pp. 93–104, 2000.
- [14] Doug Simon and Cristina Cifuentes. The squawk virtual machine: Java on the bare metal. In *Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, sys-*

- tems, languages, and applications (OOPSLA)*, pp. 150–151. ACM, 2005.
- [15] Chien-Liang Fok, Gruia-Catalin Roman, and Chenyang Lu. Rapid development and flexible deployment of adaptive wireless sensor network applications. In *the 25th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp. 653–662. IEEE Computer Society, 2005.
- [16] Athanassios Boulis, Chih-Chieh Han, Roy Shea, and Mani B. Srivastava. Sensorware: Programming sensor networks beyond code update and querying. *Pervasive Mob. Comput.*, Vol. 3, No. 4, pp. 386–412, 2007.
- [17] YoungMin Kwon, Sameer Sundresh, Kirill Mechitov, and Gul Agha. Actornet: an actor platform for wireless sensor networks. In *the fifth international joint conference on Autonomous agents and multiagent systems (AAMAS)*, pp. 1297–1300. ACM Press, 2006.
- [18] 鄭顕志, 深澤良彰, 本位田真一. 無線センサーネットワークミドルウェアの研究動向とエージェント技術の適用可能性. 合同エージェントワークショップ&シンポジウム 2006 (JAWS2006), 2006.
- [19] 末永俊一郎, 松崎和賢, 本位田真一. 無線センサネットワークにおけるプログラミングモデル - プログラミングサポートとプログラミングアブストラクション. 合同エージェントワークショップ&シンポジウム 2006 (JAWS2006), 2006.
- [20] 松崎和賢, 末永俊一郎, 本位田真一. 無線センサネットワークにおけるプログラミングモデル - マクロプログラミングとトラディショナルアプローチ. 合同エージェントワークショップ&シンポジウム 2006 (JAWS2006), 2006.
- [21] 鄭顕志, 深澤良彰, 本位田真一. 無線センサネットワークにおける計測記述言語のための規範モデル導出手法の構築. ソフトウェア工学の基礎ワークショップ (FOSE2008), pp. 69–75, 2008.
- [22] Kenji Tei, Yoshiaki Fukazawa, and Shinichi Honiden. An efficient node selection metric for in-network process deployment. In *the Fourth International Wireless Internet Conference (WICON 2008)*, 2008.

- [23] Qiang Wang, Yaoyao Zhu, and Liang Cheng. Reprogramming wireless sensor networks: challenges and approaches. *IEEE Network*, Vol. 20, No. 3, pp. 48–55, 2006.
- [24] Jonathan W. Hui and David Culler. The dynamic behavior of a data dissemination protocol for network programming at scale. In *the 2nd international conference on Embedded networked sensor systems (SenSys)*, pp. 81–94, 2004.
- [25] Philip Levis and David Culler. Maté: a tiny virtual machine for sensor networks. In *the 10th international conference on Architectural support for programming languages and operating systems (ASPLOS)*, pp. 85–95. ACM Press, 2002.
- [26] Philip Levis, Neil Patel, David Culler, and Scott Shenker. Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *the 1st conference on Symposium on Networked Systems Design and Implementation (NSDI)*, pp. 15–28. USENIX Association, 2004.
- [27] Athanassios Boulis, Chih-Chieh Han, and Mani B. Srivastava. Design and implementation of a framework for efficient and programmable sensor networks. In *the 1st international conference on Mobile systems, applications and services (MobiSys)*, pp. 187–200. ACM Press, 2003.
- [28] David Gelernter. Generative communication in linda. *ACM Trans. Program. Lang. Syst.*, Vol. 7, No. 1, pp. 80–112, 1985.
- [29] Shunichiro Suenaga, Nobukazu Yoshioka, and Shinichi Honiden. Generative dynamic deployment of multiple components in wireless sensor networks. In *the Sixth International Conference on Wireless On-demand Network Systems and Services (WONS)*, pp. 197–204, 2009.
- [30] 末永俊一郎, 吉岡信和, 本位田真一. 無線センサネットワークにおける複数プログラムの動的配備. 「新しい時代を切り拓くモバイル通信と高度交通システム」特集号, pp. 14–30. 情報処理学会, 2009.
- [31] Philip Levis, Nelson Lee, Matt Welsh, and David Culler. Tossim: accurate and scalable simulation of entire tinyos applications. In *the 1st international conference on Embedded networked sensor systems (SenSys)*, pp. 126–137. ACM Press, 2003.

- [32] Eric Platon, Shunichiro Suenaga, Nobukazu Yoshioka, and Shinichi Honiden. Transparent application lifetime management in wireless sensor networks. In *Demo Track of the 10th International Conference on Ubiquitous Computing (UbiComp)*, 2008.
- [33] Wendi B. Heinzelman, Amy L. Murphy, Hervaldo S. Carvalho, and Mark A. Perillo. Middleware to support sensor network applications. *IEEE Network*, Vol. 18, No. 1, pp. 6–14, 2004.
- [34] Adam Dunkels, Niclas Finne, Joakim Eriksson, and Thiemo Voigt. Run-time dynamic linking for reprogramming wireless sensor networks. In *the 4th international conference on Embedded networked sensor systems (SenSys)*, pp. 15–28. ACM, 2006.
- [35] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie. Protocols for self-organization of a wireless sensor network. *Personal Communications, IEEE [see also IEEE Wireless Communications]*, Vol. 7, No. 5, pp. 16–27, 2000.
- [36] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient mac protocol for wireless sensor networks. In *the 21st Conference on Computer Communications (INFOCOM)*, pp. 1567–1576. IEEE, 2002.
- [37] Venkatesh Rajendran, Katia Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. *Wirel. Netw.*, Vol. 12, No. 1, pp. 63–78, 2006.
- [38] Yoshiyuki Nakamura, Kenji Tei, Yoshiaki Fukazawa, and Shinichi Honiden. Region-based sensor selection for wireless sensor networks. In *the 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC)*, pp. 326–331. IEEE Computer Society, 2008.
- [39] Makoto ISHIGURO, Kenji TEI, Yoshiaki FUKAZAWA, and Shinichi HONIDEN. A sensor middleware for lightweight relocatable sensing programs. In *International Conference on Computational Intelligence for Modelling, Control and Automation (CIMCA)*, p. 195. IEEE Computer Society, 2006.

- [40] 中里彦俊, 中村善行, 鄭顕志, 深澤良彰, 本位田真一. 通信コストと計測精度を考慮したトラッキングのためのセンサモデルの提案. マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO2008), 2008.
- [41] 金木陽一, 清雄一, 本位田真一. センサーネットワークにおける耐故障性を考慮したデータ集約. 合同エージェントワークショップ&シンポジウム (JAWS2008), 2008.
- [42] Adrian Perrig, John Stankovic, and David Wagner. Security in wireless sensor networks. *Commun. ACM*, Vol. 47, No. 6, pp. 53–57, 2004.
- [43] Adrian Perrig, Robert Szewczyk, Victor Wen, David E. Culler, and J. D. Tygar. SPINS: Security protocols for sensor networks. In *the seventh annual international conference on Mobile computing and networking (MOBICOM)*, pp. 189–199, 2001.
- [44] Haowen Chan, Adrian Perrig, and Dawn Song. Random key pre-distribution schemes for sensor networks. In *IEEE Symposium on Security and Privacy*, pp. 197–213, 2003.
- [45] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *the seventh annual international conference on Mobile computing and networking (MOBICOM)*, pp. 255–265, 2000.
- [46] Zhen Yu and Yong Guan. A dynamic en-route scheme for filtering false data injection in wireless sensor networks. In *the 25th Conference on Computer Communications (INFOCOM)*, pp. 1–12, 2006.
- [47] Deepak Ganesan, Ramesh Govindan, Scott Shenker, and Deborah Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, Vol. 5, No. 4, pp. 11–25, 2001.
- [48] Issa Khalil, Saurabh Bagchi, and Ness B. Shroff. Liteworp: Detection and isolation of the wormhole attack in static multihop wireless networks. *Comput. Netw.*, Vol. 51, No. 13, pp. 3750–3772, 2007.
- [49] M. Conti, R. Di Pietro, L.V. Mancini, and A. Mei. A randomized, efficient, and distributed protocol for the detection of node replication attacks in wireless sensor networks. In *the Eight ACM*

International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), pp. 80–89, 2007.

- [50] Bryan Parno, Adrian Perrig, and Virgil Gligor. Distributed detection of node replication attacks in sensor networks. In *IEEE Symposium on Security and Privacy*, pp. 49–63, 2005.
- [51] Eylem Ekici, Janise McNair, and Dawood Al-Abri. Secure probabilistic location verification in randomly deployed wireless sensor networks. *Ad Hoc Netw.*, Vol. 6, No. 2, pp. 195–209, 2008.
- [52] Wenliang Du, Jing Deng, Yungshiang S. Han, Shigang Chen, and Pramod K. Varshney. A key management scheme for wireless sensor networks using deployment knowledge. In *the 23rd Conference on Computer Communications (INFOCOM)*, 2004.
- [53] Wenliang Du, Jing Deng, Yungshiang S. Han, and Pramod K. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks. In *the 10th ACM conference on Computer and communications security*, pp. 42–51, 2003.
- [54] Budhaditya Deb, Sudeept Bhatnagar, and Badri Nath. Reinform: Reliable information forwarding using multiple paths in sensor networks. In *28th Annual IEEE International Conference on Local Computer Networks (LCN)*, pp. 406–415, 2003.
- [55] Fan Ye, Haiyun Luo, Songwu Lu, and Lixia Zhang. Statistical en-route filtering of injected false data in sensor networks. *IEEE Journal on Selected Areas in Communications*, Vol. 23, No. 4, pp. 839–850, April 2005.
- [56] 清雄一, 本位田真一. 無線センサーネットワークにおける不正メッセージ作成元ノードの検知. *情報処理学会論文誌*, Vol. 50, No. 2, pp. 787–797, 2009.
- [57] Hao Yang, Fan Ye, Yuan Yuan, Songwu Lu, and William Arbaugh. Toward resilient security in wireless sensor networks. In *the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 34–45, 2005.
- [58] 清雄一, 本位田真一. 無線センサーネットワークにおける false event の検知. *情報処理学会論文誌*, Vol. 49, No. 2, pp. 628–638, 2009.