

アプリケーション適応型センサノード配置ロボットの提案

今枝 卓也¹ 大澤 亮² 高汐 一紀¹ 徳田 英幸^{1,2}

¹慶應義塾大学環境情報学部 ²慶應義塾大学大学院政策・メディア研究科

仮想空間から現実空間の情報を取得する方法として、センサノードで構成されたセンサネットワークを用いる方法がある。センサネットワークにより、現実空間の状態に応じたサービスを行うアプリケーションの開発が可能になる。しかし、センサネットワークの構築はセンサノードの専門家でなければ難しい。また、センサノードの配置条件はアプリケーションによって違うため、アプリケーションに関する詳細な知識も必要になる。本研究の目的は、専門家でなくても利用するアプリケーションに適応したセンサネットワークの構築を可能にすることである。そこで、本研究はセンサノードを環境に配置するロボット、及びアプリケーション毎にロボットの挙動を変更するミドルウェアから成るシステム ROASEN を提案する。アプリケーション開発者は ROASEN ミドルウェアを利用する事で、センサノード配置条件を指定できる。ROASEN ロボットはこのセンサノード配置条件に沿うようにセンサノードを配置する。

A Robot for Application Adoptive Sensor Node Deployment

Takuya Imaeda¹ Ryo Ohsawa² Kazunori Takashio¹ Hideyuki Tokuda^{1,2}

¹Faculty of Environmental Information, Keio University

²Graduate School of Media and Governance, Keio University

It is possible to acquire information of real space as digital data using sensor networks that consist sensor nodes. Sensor networks enable applications to provide service according to the state of real space. It is too difficult for persons who are not specialists to construct sensor networks. Besides, the condition of deployment of sensor nodes is different by the application. The purpose of this research is to enable to construct sensor networks and make the sensor networks adjust to the application without specialists. In this research we propose a ROASEN, a system that consists of a robot that deploys sensor nodes. Also the system provides a middleware that change behavior of the robot based on each application. Application developers can specify the condition that deploy sensor nodes by using the ROASEN middleware. The ROASEN robot satisfies this condition when the robot deploys sensor nodes.

1 はじめに

近年、ユビキタスコンピューティング環境の実現を目指す研究が盛んになりつつある。ユビキタスコンピューティング環境では、あらゆる物がネットワークにつながり、仮想空間から現実空間の情報が参照できるようになる。現実空間の情報を仮想空間上で取得できるようになると、ユーザや現実空間上の物の状況に応じて様々なサービスやアプリケーションが提供されるようになる。現実空間の情報を取得する方法として、環境にセンサや計算機を埋め込む手法がある。しかし、この手法はコストがかかり、広い空間に適用しにくいという問題がある。そこで、複数のセンサノードを用いてセンサネットワークを構築する手法が注目されている。センサノードとは、センサを搭載した小型の計算機で、無線によりネットワークを構築できる。比較的安価に入手でき、広い範囲にも展開できる。センサノードには、カリフォルニア大学を中心に開発が進んでいる Mote[1] や、カールスルーエ大学 TecO 研究所の Smart-Its[2] 等がある。センサノードは光センサや温度センサ、加速度センサ等のセンサを備えている。センサネットワークから得られる情報を利用すること

で、室内の温度や光量を自動調節するといった、現実空間の状態に応じたサービス提供ができるアプリケーションが可能になる。

センサネットワークを構築するには、センサノードを環境内に配置する必要がある。しかし、センサノードの配置には人手がかかるという問題がある。システム構築者は環境内に多くのセンサノードを様々な場所に配置するが、これは面倒な作業になりがちである。また、センサネットワークの構築にはセンサノードの専門的な知識が必要になる。システム構築者は使用するセンサノードの特性に合わせて設置場所を選ぶ必要がある。そのため、システム構築者は事前にセンサノードに関する知識を入手しなくてはならず、専門家以外の人間がセンサネットワークを構築するのは困難である。そして、提供したいサービスやアプリケーションによってセンサノードの配置条件を変える必要がある。提供するサービスやアプリケーションの種類によって、センサノードの設置場所はランダムになったり特定の場所になったりする。例えば、植物を監視したい場合はセンサノードを植物の近くに置く必要がある。そのため、アプリケーションに関して詳細な知識を持って

いない人間は、センサネットワークを構築しても有効に活用できない。本稿では、アプリケーションのセンサノード配置条件に合わせて動的にセンサノード配置方法を変更する事を、アプリケーション適応的であるとする。

本研究は、上記の問題の解決を目的としている。この目的を達成するために、センサノードを配置するロボット、及びアプリケーション毎にロボットの挙動を変更するミドルウェアから成る ROASEN システムを提案する。これらを利用する事により、人手によらずアプリケーションに適応したセンサノードの配置が可能になる。

2章では関連研究を紹介する。3章では本研究の目的を達成するためのアプローチとして、アプリケーション毎にセンサノード配置条件を指定する手法を説明する。4章で ROASEN システムのプロトタイプの実装を述べる。5章で人手によるセンサノード配置に対するロボットの優位性についての議論について述べる。6章で今後の予定について述べ、7章でまとめを述べる。

2 関連研究

ロボット技術の発展に伴い、工場や災害地等の様々な場所でロボットの導入が進められている。人間の生活空間にロボットが使われるケースも増えてきており、SONY の AIBO[3] を初めとするペットロボットは多くの家庭で飼われている。今後は環境内に埋め込まれたセンサや計算機から情報を取得し、より環境に適合した行動が可能なロボットが増えていくと考えられる。また、本来は動けないセンサノードに移動機能を持たせロボットにするモバイルセンサノードを提案する研究もある。モバイルセンサノードの効率的な配置を目的としたアルゴリズムも提案されている。

2.1 ユビキタスコンピューティング環境におけるロボット

従来のロボットは周辺環境の情報取得方法としてロボット自身に取り付けられたセンサに頼っていた。しかし、近年のユビキタスコンピューティング環境に関する研究の結果、環境内に遍在するセンサから環境情報を取得するロボットが提案されている。例えば、Shiomi らは、科学館において環境に埋め込まれたセンサ群と協調動作して展示紹介を行うロボット Robovie の実験を紹介している [4]。

ロボットとセンサノードの組み合わせとしては、Intel Research Laboratory の Anthony らがセンサネットワークのメンテナンスを行うロボットを提案している [5]。このロボットは、PlantCare プロジェクトで植物を監視するセンサネットワークにおいて、センサノードのメンテナンスを行う。実験ではセンサノードに搭載されたセンサの誤差調整をロボットを使って行い、その有効性を確かめている。この研究は人手のか

かるセンサネットワークのメンテナンスをロボットに行わせるという点で本研究の先行研究と言える。しかし、センサノードの配置問題に関しては詳しく述べられておらず、どのようにアプリケーション毎にセンサネットワークの適応を行うのかという問題が残されている。

2.2 モバイルセンサノード

センサノードの配置に関する困難を解決するために、センサノード自体に移動性能を持たせる研究がある。Southern California 大学の RoboMote は、現在センサノードとしては最も普及している Mote に移動性能を持たせる研究である [6]。同様の研究として、Notre Dame 大学の MICAbot がある [7]。MIT の Mathew らはセンサノードには移動性能を持たせず、人間や乗り物等の他の移動体にセンサノードを運ばせる手法を提案している [8]。本研究の目的を達成するための手法として、RoboMote や MICAbot 等のモバイルセンサノードの使用も可能である。しかし、モバイルセンサノードは通常のセンサノードに比べサイズが大きくなるという欠点を持つ。また、電力消費も大きくなり、長期的な使用には向かない。Mathew らの手法にはこれらの欠点は無いが、目的地を細かく設定できない。一方で、ロボットを用いた手法は、通常のセンサノードが使用でき、長期的な使用も可能である。また、従来の手法ではセンサノードの正確な位置を取得するには、各センサノードに位置取得のためのハードウェアを搭載する必要があった。センサノードに搭載するハードウェアの拡張は電力消費の増大を招く。しかし、ロボットを使用すると、ロボット自身に位置情報センサを搭載しておけばセンサノードの位置の把握が可能である。これらの理由から、本研究ではロボットを用いたセンサノードの配置を行う。

2.3 センサノード配置アルゴリズム

モバイルセンサノード研究の発展に伴って、センサノードの自動配置アルゴリズムが考案されるようになった。Howard らの提案するアルゴリズムはモバイルセンサノードを障害物のある未知の空間に配置する [9]。このアルゴリズムでは、直前に移動したモバイルセンサノードの情報を元に、各モバイルセンサノードの配置場所を逐次的に決定していく。各センサノードの監視範囲が重複しないように配置していくため、少ないセンサノードでもセンサネットワーク全体の監視範囲を大きくできる。しかし、この手法ではアプリケーションに適応したセンサネットワークの構築ができない。村瀬氏はモバイルセンサノードの自動配置に際して、配置場所の偏りを無くしつつ、センシング対象を複数のモバイルセンサノードで監視する手法を提案している [10]。この手法により、センシング対象を監視できない時間を短くし、誤検知やノイズが起き

てもセンシング精度を維持できる。また、アプリケーションによってセンシング対象が異なる事を考慮している点が注目できる。しかし、センサノード配置条件としてはセンシング対象しか指定できず、十分にアプリケーションに適応できるとは言えない。

3 アプリケーション適応的なセンサノード配置

本研究では、人手をかけずにアプリケーション適応的なセンサノードの配置を可能にする事を目的としている。本章ではこの目的を達成するための手法について説明する。

3.1 想定環境

本研究では、リビングやオフィスの会議室等の室内環境を想定環境とする。このような人間が生活する空間には、既にペットロボットや家事ロボット等が登場し、今後も様々なロボットが導入されると考えられる。これらのロボットがセンサネットワーク構築機能を持てば、ユビキタスコンピューティング環境の構築に役立つ。また、情報家電や位置取得センサ等のロボットから利用可能なデバイスの増加に伴い、将来的には人間の生活空間にロボットが増えていくことと予想される。そこで、本研究ではロボットを用いてセンサノードの配置を行う。

人間の生活空間におけるセンサノードを利用したアプリケーション例として、室内環境調節、植物監視、地震対策、災害救助が挙げられる。室内環境調節とは、センサノードから得られた環境内の情報を元に、光量や室温の調整を行うアプリケーションである。植物監視とは、室内観葉植物の周辺環境情報から、植物の成長を促進するように環境の調整をするアプリケーションを指す。地震対策とは、センサノードの加速度センサで地震を検知すると、稼働中の機器を止めるアプリケーションである。災害救助とは、ガス漏れや地震等の災害で室内に人が入れなくなった時に、ロボットを送り込みセンサネットワークを構築し、室内の状況を把握するアプリケーションである。本研究ではこれらのようなアプリケーションに適応したセンサノード配置を可能にする手法を提案する。

3.2 アプリケーションへの適応

アプリケーションによってセンサノードの配置条件は違う。このため、人手でセンサノードを配置する時には、システム構築者はアプリケーションがどのような情報を期待し、どのように動作するか詳しく知っていなければならない。本研究ではアプリケーションがセンサノードの配置条件を指定する手法を提案する。条件指定の方法として、アプリケーションから利用可能なミドルウェアを開発する。アプリケーション開発者はこのミドルウェアを利用してセンサノードの配置条件を指定し、ロボットは条件に沿うように配置する。

3.2.1 アプリケーションによる配置条件の違い

センサノード配置条件の要素として、配置する場所、センサノードの密度、最低限必要なノード数、センサの種類が挙げられる。センサノードを配置する場所はアプリケーションの監視対象によって違う。部屋全体を監視する場合には部屋全体に均一になるように配置場所を設定する。ただし、部屋の中での人間の移動が多い場合は部屋の中央付近にはセンサノードを置きたくない時がある。その場合は壁際に沿ってセンサノードを配置するよう設定する。特定の監視対象がある場合にはその対象の近くにセンサノードを置く。

センサノードの密度や最低限必要なセンサノード数は監視対象がどの程度重要かによって異なる。センサノードに搭載されるセンサは性能に個体差があり、単一のセンサノードでは信頼性に欠ける場合がある。またネットワークの安定度によってはパケットのロスが起こり環境情報が取得できない場合がある。そこで重要な監視対象に対しては環境情報取得の信頼性を上げるために、一つの対象を複数のセンサノードで監視する手法を取る。また、監視したい現象によってもセンサノードの密度や最低限必要なセンサノード数は変化する。地震のような環境全体に影響を与えるイベントを監視するアプリケーションでは、センサノードは最低1個で監視可能である。一方、光量や温度の変化のような現象は直射日光の当たる場所と影になっている場所で著しく変わる場合がある。単一のセンサノードではある現象に対して偏った値しか観測できない可能性があり、複数のセンサノードで監視する必要がある。

センサの種類は監視したい現象によって異なる。光量の変化を監視したい場合には光センサを用いる。室温を監視したい場合は温度センサを使う。使用するセンサによっては信頼性が劣る場合があるので、信頼性の低いセンサを使う時には、センサノードの数を増やして対応する。表1にセンサノードを利用するアプリケーションのセンサノード配置条件の違いを示す。

表 1: アプリケーションによるセンサノード配置条件の違い

	場所	密度, 最低限必要な数	センサの種類
室内環境調節	特定の場所 (人間の近く)	中 (~高)	光, 温度, 音響
植物監視	特定の場所 (植物の近く)	低 (~中)	光, 温度
地震対策	任意	低	加速度
災害救助	万遍なく	高	光, 温度, 音響, その他

3.2.2 センサノード配置条件の書式

配置条件の設定はXML形式でアプリケーション開発者が記述する。

図 1 にセンサノード配置条件の書式例を示す。
また図 2 にセンサノード配置条件 XML の DTD を示す。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE roasenXML SYSTEM "roasenxml.dtd">
<roasenXML applicationName="officeObserver">
  <sensorCondition>
    <type>light</type>
    <object typeId="10001">
      <minimum>3</minimum>
      <distanceFromObject>0.1</distanceFromObject>
    </object>
  </sensorCondition>
  <sensorCondition>
    <type>temperature</type>
    <space>
      <density>2</density>
      <distanceFromWall>1</distanceFromWall>
    </space>
  </sensorCondition>
  <sensorCondition>
    <type>oscillation</type>
    <wallside>
      <density>3</density>
      <distanceFromWall>0.5</distanceFromWall>
    </wallside>
  </sensorCondition>
</roasenXML>
```

図 1: センサノード配置条件の書式例

```
<!ELEMENT minimum (#PCDATA)>
<!ELEMENT distanceFromObject (#PCDATA)>
<!ELEMENT density (#PCDATA)>
<!ELEMENT distanceFromWall (#PCDATA)>
<!ELEMENT type (#PCDATA)>
<!ELEMENT object (minimum,distanceFromObject?)>
<!ATTLIST object typeId CDATA #REQUIRED>
<!ELEMENT space (density,distanceFromWall?)>
<!ELEMENT wallside (density,distanceFromWall?)>
<!ELEMENT sensorCondition (type,(object | space | wallside))>
<!ELEMENT roasenXML (sensorCondition)*>
<!ATTLIST roasenXML applicationName CDATA #REQUIRED>
```

図 2: センサノード配置条件指定 XML の DTD

センサノードを配置する際、搭載センサの異なるセンサノードを混ぜて配置する場合がある。例えば、光センサを搭載したセンサノードを部屋全体に配置し、加速度センサを持つセンサノードは部屋に 1 つだけで良い、といった場合である。このため条件の設定はセンサノードに搭載されたセンサ毎に行う。XML 内では sensorCondition 要素で条件を設定する。type 要素でセンサの種類を指定する。場所の指定は object, space, wallside のいずれかの要素を用いることで指定する。object 要素はセンサノードを物の近くに配置したい場合に記述する。属性として物の種類を表す typeId を指定する。監視対象一つに対して最低何個のセンサノードが必要か minimum 要素で指定する。また、物とセンサノードの距離を distanceFromObject 要素で

メートル単位で指定できる。space 要素はセンサノードを部屋全体に分散して配置したい場合に記述する。space 要素内では、density 要素と distanceFromWall 要素を記述できる。density 要素ではセンサノードの密度を記述し、メートル単位でセンサノード間の間隔を指定する。distanceFromWall 要素ではセンサノードと壁の距離を記述できる。wallside 要素はセンサノードを壁際に沿って配置したい場合に記述し、space 要素と同様に density 要素と distanceFromWall 要素を記述できる。

3.2.3 アプリに適応した配置アルゴリズム

本システムでは、センサノードの配置方法の際にオブジェクト注目型、空間網羅型、壁際型の 3 つの方法を取れる。また、使用するセンサによってセンサノードの数を補正する。

オブジェクト注目型はセンサノードを特定の場所に置く場合の配置方法である。システムは XML 内に object 要素があった場合にこの配置方法を取る。typeID 属性で指定された物に近い所にセンサノードを置く。センサノードの設置数は minimum 要素に基づき、センサノードと物との距離は distanceFromWall 要素に基づいて決定する。図 3 にオブジェクト注目型配置の様子を示す。

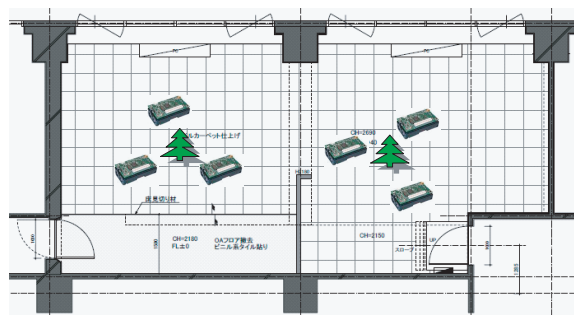


図 3: オブジェクト注目型

空間網羅型はセンサノードを部屋全体に置く場合の配置方法である。XML 内に space 要素があった場合、システムはこの方法でセンサノードを配置する。センサノードを部屋全体に均一になるように格子状に配置していく。センサノードの設置数は density 要素で決定する。また、distanceFromWall 要素があった場合には、センサノードと壁との距離は指定された値に沿うようにする。図 4 に空間網羅型配置の様子を示す。

壁際はセンサノードを壁際に配置する場合の配置方法である。XML に wallside 要素が記述されていた場合、システムはこの配置方法を取る。センサノードの設置数、センサノードと壁との距離は (b) と同様である。図 5 に壁際型配置の様子を示す。

信頼性に欠けるセンサを使用する場合には、システムは信頼性を向上するためにセンサノードの数を増や

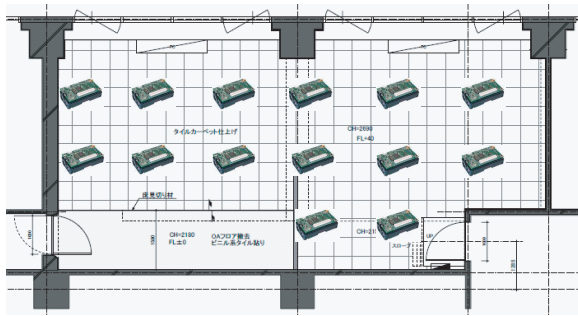


図 4: 空間網羅型配置の様子

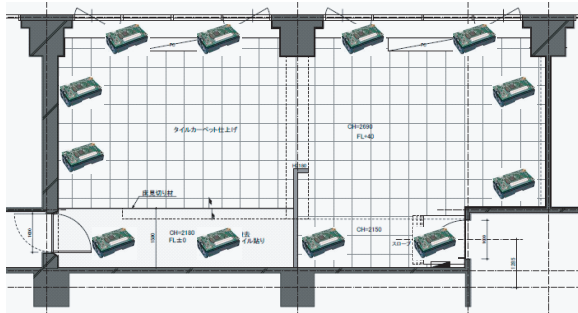


図 5: 壁際型配置の様子

して対応する．type タグに信頼性に欠けるセンサがあった場合、この補正をかけてセンサノードを配置する．図 6 に使用センサによって配置ノード数に補正をかける様子を示す．

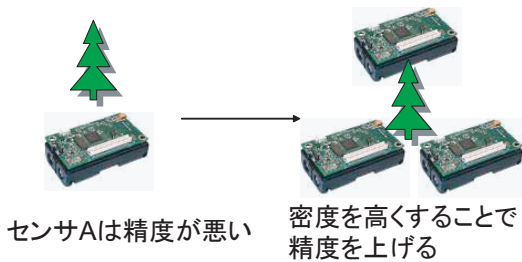


図 6: 配置ノード数の補正

3.2.4 室内図

室内図は部屋の形状や内装、部屋に置かれた物の情報を格納したバイナリファイルで、システム構築者が作成する．部屋の形状や内装に関する情報としては、部屋の大きさや壁と柱の位置等がある．また、机や植物等の部屋に置かれた物に関する情報として、物の typeID や物の大きさ、位置等がある．システムはバイナリデータを解析し、どこにセンサノードを配置するか決定する．本システムでは、システム構築者が容易に室内図を作成できるように室内図作成ツールを用意した．

4 実装

本章では ROASEN システムのプロトタイプの実装について述べる．

4.1 ハードウェア

図 7 にプロトタイプ ROASEN ロボットのハードウェア構成図を示す．

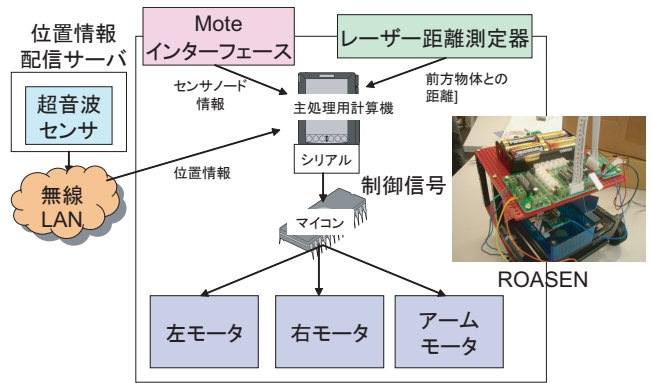


図 7: ハードウェア構成図

ロボットの基本ハードウェアは JAPAN ROBOTTECH の RoboDesigner[11] を利用して製作した．RoboDesigner は簡単に自作ロボットの製作が可能な学習用キットである．マイコンを搭載したコントローラボードから各種センサからの入力やモータの操作等を行える．本システムでは、ロボットの車体を 2 層構造とし、上層にはコントローラボードと電池を備え付けた．この部分は主にモータ制御の役割を持つ．車体下層にはモータ、センサノード配置機構、主処理用計算機を備えた．2 個のモータで車輪を回して移動し、前進、後進、回転が可能である．また、ロボットはセンサノード配置のための機構を持つ．モータによって板を前後に動かし、センサノードを車体から押し出して環境内に配置していく．また、ロボットが室内での相対位置を取得できるように、環境に超音波位置測定センサと位置情報配信サーバが備わっていると想定し、車体に超音波タグを付ける．超音波位置測定センサは InterSense の IS-600 を用いた [12]．位置情報の取得やセンサノード配置場所の決定等の主となる処理を行う計算機として SONY の Vaio typeU を搭載する．主処理用計算機は 802.11b の無線 LAN を介して部屋の位置情報配信サーバと通信し、ロボットの現在位置を取得する．針路変更やセンサノード配置のタイミングになると、モータ制御用のマイコンに制御信号を出力してロボットを操作する．また、針路決定の際の前方確認用ハードウェアとしてレーザー距離測定器を Vaio typeU に接続する．センサノードには Mote を採用した．Mote はマルチホップアドホックネットワークが構築可能なセンサノードである．無線帯域と

しては 315MHz または 2.4GHz 帯を使用する．センサとして光，温度，湿度，気圧，加速度センサ等を選択できる．Vaio typeU に Mote 用 PC インターフェイス基板を接続し，ロボット自身がセンサネットワークの基地局として動作できるようにした．

4.2 ソフトウェア

ソフトウェア構成図を図 8 に示す．

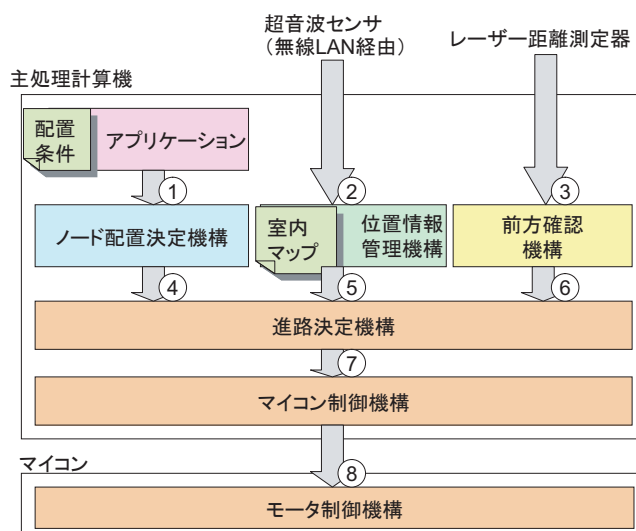


図 8: ソフトウェア構成図

本システムのソフトウェアは、マイコン上で動くモータ制御機構と、主処理用計算機で動作する ROASEN ミドルウェアからなる．アプリケーションは主処理用計算機上で動作する．アプリケーション開発者は、アプリケーションと一緒にセンサノード配置条件を記述した XML を作成し配布する．ネットワーク構築者は、利用したいアプリケーションと室内図作成ツールで作った室内図ファイルを主処理用計算機にインストールする．主処理用計算機上のアプリケーションを起動すると、ロボットは実際にセンサノードの配置を始める．ソフトウェア構成図の番号は処理の順番を示している．まずノード配置機構はアプリケーションのセンサノード配置条件 XML を読み込み (1)、センサノードの配置場所を決定する．次に位置情報管理機構が超音波位置測定センサから位置情報を取得し (2)、室内図を元にロボットの部屋内での相対位置を求める．レーザー距離測定器で前方物体との距離を確認する (3)．センサノード配置場所、ロボットの位置、前方物体との距離からロボットの進路を決定する (4, 5, 6)．進路決定機構は決定した進路をマイコン制御機構に伝え (7)、マイコン制御機構がマイコンに命令信号を送る (8)．マイコン内のモータ制御機構は、送られてきた信号を元にモータを制御する．センサノードを配置し終わるとロボットは活動を停止する．モータ制御を行うマイコン上で動くプログラムは C 言語で

実装した．主処理用計算機で動作するのミドルウェアは C++ で実装した．また、室内図作成ツールは Java で実装した．

5 議論

センサノードにはマルチホップアドホックネットワークの構築が可能な種類がある．広い範囲にセンサネットワークを構築する場合、ネットワークの途中で無線接続が切れないように中継ノードを配置する．無線通信の電波到達範囲は理論的には球形に広がるはずである．しかし、実際には壁や柱等の障害物や電波反射の影響で歪な形に広がり、人間が実際の電波到達範囲を把握する事は困難である．そのため人間がセンサノードを配置する際には無線到達距離の理論値に頼るしかないが、電波の届かない場所に中継ノードを配置してネットワークが寸断されてしまったり、逆に必要のない場所に中継ノードを配置してセンサノードに無駄ができてしまったりする (図 9)．また、人手によるセンサネットワークの配置は専門家の知識や多大な労力が必要となり、簡単ではない．

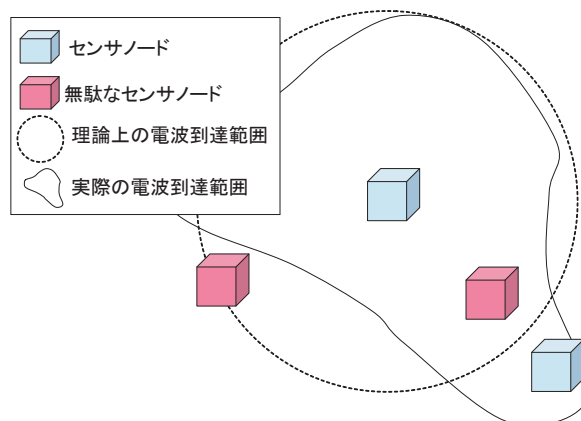


図 9: 人間によるセンサノード配置

一方、ロボットは実際に電波が受信できるかどうかを確認しながらセンサノードを配置できる．そのため電波の届く場所に限定したセンサノードの配置が可能である．また、電波到達範囲の外縁付近にセンサノードを配置していくことにより、少ないセンサノード数でセンサネットワークの無線到達範囲を最大にできる (図 10)．ロボットの導入には初期コストが必要になるが、以降は繰り返し使え、専門家がいなくてもセンサネットワークの構築が可能である．

6 今後の予定

今回のロボットはセンサノードの配置のみを行うロボットとしたが、今後の展開としてセンサネットワークの展開から維持、回収までを行うロボットシステムを実装する予定である．センサノードは電池残量の低下や故障等の理由で動作を停止する場合がある．停

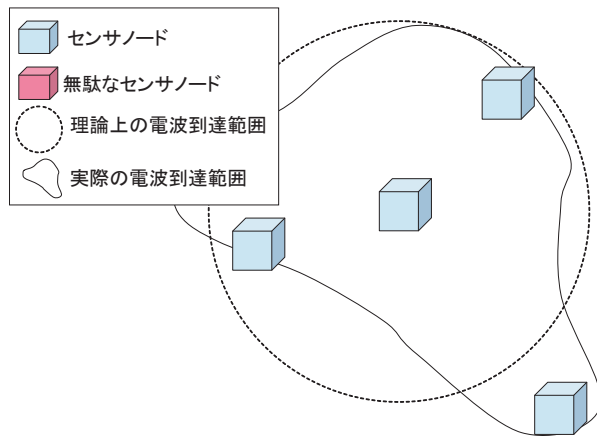


図 10: ロボットによるセンサーノード配置

止したセンサーノードがマルチホップアドホックネットワークでの中継ノードとして機能していた場合、ネットワーク全体に影響を与える。そのためセンサネットワーク維持のために機能していないセンサーノードを正常動作している物と交換する必要がある。また、学会や実験等で一時的に構築したセンサネットワークの場合、役割の終了とともにセンサーノードを回収する必要がある。これらの作業は専門的な知識や多大な労力がかかり、人間が行うには負担が大きい。そこでロボットにこれらの作業を行わせることで、簡単にセンサネットワークの運営が行えるようになる。

7 おわりに

本稿では、アプリケーション適応的なセンサーノードの配置を行うロボットとミドルウェアについて提案した。これまでの人手によるセンサーノードの配置は専門家の知識を必要とし、アプリケーションに適応した配置を行う事も難しかった。ロボットを使ってセンサーノードの配置を行う事で、専門家でなくてもセンサネットワークの構築が可能になる。また、アプリケーションによってロボットの動作を変えるミドルウェアにより、アプリケーション適応的なセンサーノードの配置を可能にする。アプリケーション開発者はXMLでセンサーノード配置条件を記述し、ロボットはこの条件に合うようにセンサーノードを配置する。アプリケーションの目的によって、センサーノードの配置方法はオブジェクト監視型、空間網羅型、壁際型の3種類の方法を取れる。監視対象がどの程度重要かによってセンサーノードの配置密度を指定できる。また、使用するセンサによってセンサーノードの配置数を補正する。

8 謝辞

本研究は総務省「ユビキタスネットワーク制御・管理技術の研究開発 (ubila プロジェクト)」の一部として支援頂いた。

参考文献

- [1] Crossbow. Mote. <http://www.xbow.jp/motemica.html>.
- [2] Hans Gellersen, Gerd Kortuem, Albrecht Schmidt, and Michael Beigl. Physical prototyping with smart-its. *IEEE Pervasive Computing*, Vol. 03, No. 3, pp. 74–82, 2004.
- [3] SONY. Aibo. <http://www.jp.aibo.com/>.
- [4] 塩見昌裕, 神田崇行, Daniel Eaton, 石黒浩. 引力・斥力モデルに基づいたセンサーノードの動的再配置手法. ユビキタスセンサネットワークと連動したコミュニケーションロボットによる科学館での展示案内, pp. 127–134, 2005.
- [5] Anthony LaMarca, Waylon Brunette, David Koizumi, Matthew Lease, Stefan B. Sigurdsson, Kevin Sikorski, Dieter Fox, and Gaetano Borriello. Making sensor networks practical with robots. In *First International Conference Pervasive 2002*, 2002.
- [6] G.T. Sibley, M.H. Rahimi, and G.S. Sukhatme. Robomote: a tiny mobile robot platform for large-scale ad-hocsensor networks. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, 2002.
- [7] M.B. McMickell and L.A. Goodwine, B. and Montestruque. Micabot: a robotic platform for large-scale distributed robotics. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, 2003.
- [8] Mathew Laibowitz and Joseph A. Paradiso. Parasitic mobility for pervasive sensor networks. In *Pervasive*, pp. 255–278, 2005.
- [9] Andrew Howard, Maja J. Matarić, and Gaurav S. Sukhatme. An incremental self-deployment algorithm for mobile sensor networks. *Auton. Robots*, Vol. 13, No. 2, pp. 113–126, 2002.
- [10] 村瀬正名, 西尾信彦, 徳田英幸. 引力・斥力モデルに基づいたセンサーノードの動的再配置手法. 情報処理学会システムソフトウェアとオペレーティングシステム研究会論文集, pp. 31–38, 2003.
- [11] JAPAN ROBOTTECH LTD. Robodesigner. <http://www.japan-robottech.com/index.html>.
- [12] InterSense. Is-600 mark 2 precision motion tracker. <http://www.isense.com/products/prec/is600/>.