情報処理学会論文誌 Vol. 49 No. 6 1778–1793 (June 2008)

時空間モデルを考慮したロボティックセンサノード機構の実現

青木崇行[†1] 中澤仁[†1] 徳田英幸[†1,†2]

センサをモータやロボット等のアクチュエータに装着したロボティックセンサノード機構を実現するために，Spinning Sensors ツールキットと呼ぶプログラミングツールキットを提案する．ロボティックセンサノードを実現するには，センサとアクチュエータの協調処理を実現するだけでなく，協調動作によるセンシング範囲変化を計算する必要がある．これらの問題を即興的協調実現問題と検知範囲把握問題として提示する．本論文では，ロボティックセンサノードのセンシングにおける時空間モデルを定義する．そしてロボティックセンサノード機構構築のためのプログラミングツールキットを提案する．時空間モデルにおいては，センサノードとオブジェクトが動くことを想定し，センシング空間やセンシング時間とセンサノードの動作の関係についてモデリングした．また，ロボティックセンサノード構築に必要な協調処理，データ処理，検知範囲解析処理を実現したツールキットを用いて，2 つのプロトタイプアプリケーションを実装した．さらに評価実験を行い，ロボティックセンサノードの有効性を示した．Spinning Sensors ツールキットを用いてロボティックセンサノード機構を構築することにより，即興的協調実現問題と検知範囲把握問題を解決し，既存のセンサとアクチュエータを用いてより検知範囲が広く精度の高いセンサシステムを構築可能になる．

A Robotic Sensor Node with Spatiotemporal Model

Soko Aoki,[†1] Jin Nakazawa[†1]
and Hideyuki Tokuda[†1,†2]

This paper proposes a Spinning Sensors programming toolkit which realizes a robotic sensor node mechanism comprised of a sensor node and a robotic actuator node such as a motor or a mobile robot. To realize a robotic sensor node, we need to realize not only the collaborative utilization of arbitrary sensors and actuators but also analysis of sensing area and time. We stated these problems as spontaneous coordination realization problem and sensing area analysis problem. The Spinning Sensors toolkit provides the mechanism of device coordination, data processing, and management of tempo-spatial model of robotic sensor nodes. By using this toolkit, the programmers are able to create application software using the robotic sensor nodes. In this paper, we discuss a robotic sensor node model and design and implementation of the toolkit. We introduce two kinds of applications using the toolkit: environment monitoring and sensor controlled robot. The experiments using the robotic sensor node and the toolkit are conducted to evaluate and measure the possibility, performance, and practicality of a robotic sensor node mechanism. The Spinning Sensors toolkit overcomes both spontaneous coordination realization problem and sensing area analysis problem and increases sensing space and accuracy of a sensor node.

1. はじめに

センサネットワーク技術は，都市や農場の監視システムといった広域なものから屋内での住環境監視システムといった狭域なものまで，様々な場面で活用され始めている．センサネットワークシステムを構築するにあたり，センサノードの設置管理コスト，センシング範囲，ノードの総数等は重要な検討要素になる．センサノード単体のコストが低い場合には多量の静的なセンサを設置するのが効率的になる．一方で高価な人間感知センサを使用する場合等，センサノード単体のコストが高い場合には多量のセンサを配置するのが難しいため，少量の移動可能なセンサにより環境内をセンシングするほうがトータルのコストが低くなる．本研究の目的は，センサに対して移動機能を持つロボティックアクチュエータを付加することにより，ロボティックセンサノード機構を実現し，ホームユビキタスコンピューティング環境でのセンサシステムのセンシング領域やセンシング精度の向上を図ることである．

センサネットワークやモバイルセンサノードに関する研究は，アプリケーションに特化し専用のハードウェアと専用のソフトウェアを構築する垂直統合型ノードの開発研究[1)] や，ソフトウェアシミュレータをベースとしたセンシング範囲の評価研究[2),3)] 等，いくつか行われている．しかし，既存の技術ではハードウェアの組合せは固定的になっており，センサネットワークシステムを構築するにあたり任意のセンサとアクチュエータを組み合わせられない．また，組合せの変更を想定していないために，センサとアクチュエータを組み合わせた際のセンシング特性について考慮できていない．センシング特性としては，静的なセンサ

†1 慶應義塾大学大学院政策・メディア研究科
Graduate School of Media and Governance, Keio University
†2 慶應義塾大学環境情報学部
Faculty of Environment and Information Studies, Keio University

© 2008 Information Processing Society of Japan

が動的になることによるセンシング空間拡大や，ある領域に対するセンシング時間の延長，またセンサのセンシング周波数とノードの動作の関係の考察が必要になる．本論文では前者を即興的協調実現問題，後者を検知範囲把握問題として提示する．

本論文では，これらの問題を解決するツールキットとして，Spinning Sensors ツールキットを提案し，センサとロボティックアクチュエータを用いてロボティックセンサノードを実現可能にした．Spinning Sensors ツールキットは，ハードウェアの抽象化を行うノード抽象化層，協調処理を実現するフュージョン層，そしてロボティックセンサノードに有用な各種機能を実現する機能層の 3 層から構成される．また，その設計過程でロボティックセンサノードに特有の時空間モデルについて考察し，ロボティックセンサノードの有効性を示す．ツールキットを用いたアプリケーションについては，環境モニタリング，ラジコンロボットの 2 種類を構築した．評価として実際に構築したロボティックセンサノードの動作実験と定性的な評価を行った．

本論文ではまず 2 章においてロボティックセンサノードモデルについて述べ，3 章において，本ツールキットの設計と実装について説明する．4 章では，本ツールキットの評価を行う．そして 5 章で関連研究との比較を行い，最後に 6 章で本論文をまとめる．

2. ロボティックセンサノードモデル

センサネットワークを構成するセンサノードを，物理的に動作可能なモータやスライダ等のロボティックアクチュエータに装着したものを本論文ではロボティックセンサノードと呼ぶ．ロボティックセンサを用いたアプリケーション例を**表 1** に示す．本章では，ロボティックセンサノード機構を構築するにあたっての既存の技術の問題点を説明する．また，センサとロボティックアクチュエータの分類を行うことよりロボティックセンサノードの検知範囲拡大の有効性について検討し，それらを基にロボティックセンサノードに特有な時空間モデルの構築を行った．さらにロボティックセンサノード機構構築に必要なソフトウェア技術の整理を行った．

表 1　ロボティックセンサを用いたアプリケーション例
Table 1　Examples of robotic sensor application.

アプリケーション	使用デバイス
環境モニタリング	気象系センサとモータやスライダ
センサ制御ラジコン	制御系センサとロボット
コンテキストアウェアサービス	位置認識センサとロボット

2.1　即興的協調実現問題と検知範囲把握問題

本論文では，センサネットワークを用いた環境監視アプリケーションの例として，以下に示すシナリオ「オフィス内環境モニタリング」を扱う．

> ある会社のオフィスにおいて，良質な職場環境の維持や業務の円滑化のためにセンサネットワークでのオフィス内環境モニタリングを導入することにした．オフィス管理者は，オフィス内の温度，湿度，照度等の分布とともに，人間の居場所や向いている方向等をセンシングするシステムの導入を決めた．導入コストや管理コストを考え，一部の高価なセンサについてはオフィス内の天井や柱上を移動可能なロボティックアクチュエータに装着し，ロボティックセンサノードとして実現した．ロボティックセンサノードでは，たとえば温度が高いところに自動的に移動し，そこを重点的にセンシングする等，センサとアクチュエータの協調が実現されている．オフィスの配置換え時には，そのオフィスの役割に応じて，ロボティックアクチュエータ上のセンサを変更することになる．

このシナリオを実現するには，センサの出力をアクチュエータの入力として利用する場合等にセンサとアクチュエータ間の通信が必要になる．また，センサとアクチュエータの組合せを自由に変更可能である必要がある．さらに，センサとアクチュエータにより実現されるロボティックセンサノードのセンシング範囲をオフィス管理者が把握することにより，より厳密なオフィス環境監視を実現できる．

ロボティックアクチュエータにおいて，自由なセンサとアクチュエータの組合せを実現するには，センサとアクチュエータ間のデータ通信の実現やそれぞれの制御が必要になる．既存の分散コンポーネント技術[4),5)] ではネットワーク上のサービスを統合するフレームワークを提供しているものの，センサやアクチュエータに特有なデータ処理や，センサに特有なポーリングモデルとイベントドリブンモデルを考慮したデータ通信機能がない．またセンサネットワークを対象としたデータフュージョンミドルウェア[6),7)] においても，複数の同種のセンサノードを対象としており，異種のセンサとアクチュエータの組合せは考慮されていない．また，ハードウェアとソフトウェアを垂直統合的に開発し，モバイルセンサノードを実現している研究[1)] においては，ハードウェア間やハードウェアとソフトウェアの関係が固定的になっており，センサとアクチュエータの組合せの変更が困難である．本論文ではこれらの問題を即興的協調実現問題とし，ツールキットによるハードウェアの抽象化機能，協調処理，データ処理等の機能を提供することにより解決する．

また，センサとアクチュエータを統合することにより，センサのセンシング範囲は拡大，

© 2008 Information Processing Society of Japan

表 2　センサの特性の分類
Table 2　Classification of sensors characteristics.

特性	種類
指向性あり/検知範囲小	パッシブ RFID，加速度
指向性あり/検知範囲大	アクティブ RFID，音量，超音波，照度
指向性なし/検知範囲小	温度，湿度
指向性なし/検知範囲大	人間感知

変更される．従来の研究では，センシング範囲のシミュレーションをしたもの[2),3)] は多いが，センサとアクチュエータの組合せの変更による検知範囲の拡大や変化に関する考察がされていない．また，センサノードを構築し，実際に稼動しないとその検知範囲を把握できなくなっていた．本論文ではこれらの問題を検知範囲把握問題とし，センサとアクチュエータの組合せによる時空間モデルの検討を行い，センサとアクチュエータを組み合わせた時点でセンシング範囲が算出される仕組みを導入し解決する．

2.2　センサとロボティックアクチュエータの分類

センサデバイスは，**表 2** に示すように，指向性の有無と検知範囲の大小により 4 種類に分類できる．センサの指向性とは，そのセンサが向いている方向が観測データに与える影響を意味する．温度センサは無指向性センサである．逆に照度センサはそれが光源の方向に向いていれば強い照度を観測結果として出力するため，指向性センサである．センサにおける検知範囲とは，そのセンサが自身からどのくらいの距離までのデータを取得するかであり，たとえば温度センサはそのセンサが触れているきわめて小さい範囲からデータを取得するので検知範囲は小さく，アクティブ RFID リーダはある距離内から RFID タグの有無を検知するので，検知範囲が大きい．

ロボティックアクチュエータデバイスは，**表 3** に示すように，その運動特性により 3 種類に分類できる．1 次元的な直線移動や回転運動をするモータやスライダ，2 次元的な移動，平面上の移動をする車輪つきロボット等，3 次元空間を移動するロボットアームや飛行可能なロボティックアクチュエータ等である．モータやスライダについては，シナリオで述べたような既存の生活環境に対して人間の行動を邪魔することなく配置できるが，平面上や空間内を移動するロボット等は生活環境での導入に際して，障害物との衝突に関する検討や位置認識等，技術的かつ運用的な各種問題が発生する．そこで本論文では，システム構築に用いるアクチュエータをスライダやモータといった 1 次元的運動をするものに限定し，モデル化やツールキットにおいても 1 次元運動を対象とする．

表 3　アクチュエータの特性の分類
Table 3　Classification of actuators characteristics.

特性	運動例	具体例
1 次元的運動	直線，回転運動	スライダ，モータ
2 次元的運動	平面移動	車両型ロボット
3 次元的運動	空間移動	水中ロボット，飛行ロボット

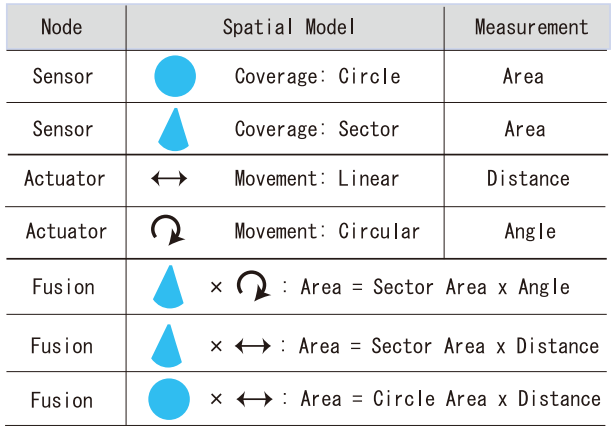


図 1　センサとロボティックアクチュエータの組合せによる検知範囲拡大
Fig. 1　Examples of enlargement of space model.

これらのセンサとアクチュエータを組み合わせると，そのセンサの検知領域を拡大できる．たとえば，指向性センサを回転するモータに載せた場合に，従来は一方向のみ検知可能だったセンサも，360 度全方位の検知が可能になる．また，無指向性センサにおいても，そのセンサを移動可能にした場合に，検知領域を格段に増やせる．**図 1** にセンサとアクチュエータの組合せによる検知領域拡大例を示した．なお，図 1 中のフュージョンが，センサとアクチュエータを組み合わせたロボティックセンサノードである．

2.3　時空間モデル

任意のセンサとロボティックアクチュエータを組み合わせると，センサのセンシング特性が変化する．本章ではその組合せ結果として発生するセンシングの時空間モデルについて考察を行う．まず空間モデルではセンサとアクチュエータの組合せによる，センシング空間の変化をモデル化した．次にある空間がセンシングされる時間と，センシング周波数とアク

© 2008 Information Processing Society of Japan

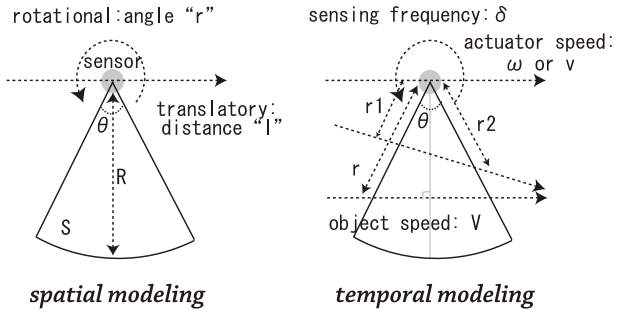


図 2　時空間モデルとパラメータ
Fig. 2　Parameters for space time model.

チュエータの移動速度の関係を時間モデルとして提示する．また，ロボティックセンサノードを使ったシステムの要件として，センシング対象が動くオブジェクトの場合を想定し，実時間モデルを提示する．各モデルで使用されるパラメータについては**図 2** に示す．

2.3.1　空間モデル

センサが移動可能になることによるセンシング空間の変化を検討する．センサの特性として，本来はそのセンシング範囲は 3 次元的な広がりを持つものが多いが，ここではモデリングをシンプルにするために，指向性のないセンサのセンシング空間は円形として示し，指向性のあるセンサのセンシング空間は扇形として示す．なお，ツールキットにおける対応についても，指向性のないセンサのセンシング空間は円形とし，指向性のあるセンサのセンシング空間は扇形としている．

- **センサ＋回転系アクチュエータ**
  センサと回転系アクチュエータを組み合わせた例を示す．ここでは，S：センシング範囲の面積，θ：センシング幅（角度），r：回転系アクチュエータの回転角度，R：センサのセンシング半径とすると，式 (1) が成り立つ．ただし $\theta + r$ は 360 以下であり，また指向性のないセンサについては $\theta + r = 360$ となる．

$$S = \frac{\theta + r}{360} R^2 \pi \tag{1}$$

- **センサ＋直動系アクチュエータ**
  センサと直動系アクチュエータを組み合わせた例を**図 3** に示す．ここではセンシング範囲の拡大を狙っていることから，センサのセンシング方向（以下，センシング方向とはセンシング領域を 2 等分する線の向く方向とする）と移動方向は直交するとし，センシング範囲の最大化を図る．また，移動中にセンシングする領域の重複は計算せず，最終的なセンシング範囲の面積を求めることにする．l：直動系アクチュエータの移動距離とし，θ は 180 以下とすると，式 (2) が成り立つ．

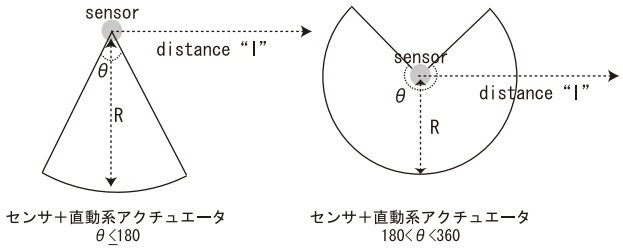


図 3　直動系：空間モデルとパラメータ
Fig. 3　Translatory: parameters for space model.

$$S = \frac{\theta}{360} R^2 \pi + lR \tag{2}$$

さらに θ が 180 以上 360 未満の場合は，センシング方向と移動方向が直交すると仮定すると，式 (3) が成り立つ．

$$S = \frac{\theta}{360} R^2 \pi + \left(R + R \sin \frac{\theta - 180}{2} \right) l + R \cos \frac{360 - \theta}{2} R \sin \frac{360 - \theta}{2} \tag{3}$$

また，θ が 360 の場合は式 (4) が成り立つ．

$$S = \frac{\theta}{360} R^2 \pi + 2lR \tag{4}$$

これらのモデルより，静的な指向性のあるセンサを回転系，直動系アクチュエータと統合した際の面積の拡大率を計算できる．これらはセンサシステムを構築する際のセンシング空間の見積り時や，他のセンサシステムとのセンシング範囲比較等に用いられる．

2.3.2　時間モデル

センサが移動可能になると，領域やオブジェクトがセンシングされている時間が動的に変化する．ここでは，ロボティックセンサノード実現により発生する時間モデルについて整理する．なお，モデリングをシンプルにするためにセンシング対象は面ではなく点として考察する．

- **静的オブジェクト＋動的アクチュエータ**
  オブジェクトが静止しており，センサノードが動いている場合において，T：センサが

© 2008 Information Processing Society of Japan

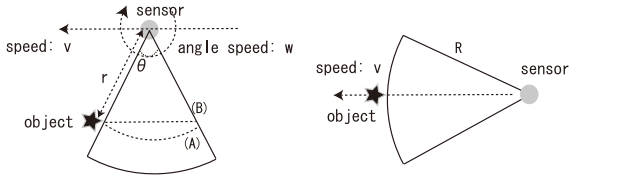


図 4　時間モデルとパラメータ
Fig. 4　Parameters for time model.

オブジェクトをセンシングする時間，θ：センサのセンシング角度，ω：回転系アクチュエータの角速度とすると，回転系アクチュエータを使用した場合には，式 (5) が成り立つ．なお θ については，以降では明記がない限り 180 以下とする．本時間モデルについて**図 4** 左側 (A) に示す．

$$T = \frac{\theta}{\omega} \tag{5}$$

また直動系アクチュエータを使用した時間モデルではオブジェクトをセンシングしている時間を最大化するために，センサのセンシング方向と直動系アクチュエータの移動方向の角度については，90 度の場合と 0 度の場合を取り扱う．

まず，センシング方向と移動方向の角度が 90 度の場合は，r：オブジェクトがセンシング領域内に入った時点でのセンサとオブジェクトの距離，v：直動系アクチュエータの動作速度とすると，式 (6) が成り立つ．本時間モデルについて図 4 左側 (B) に示す．

$$T = \frac{2r \sin 0.5\theta}{v} \tag{6}$$

また，センシング方向と移動方向の角度が 0 度の場合は，式 (7) が成り立つ．本時間モデルについて図 4 右側に示す．

$$T = \frac{r}{v} \tag{7}$$

- **動的オブジェクト＋静的アクチュエータ**

次にセンサノードは動かずにオブジェクトが動く場合について述べる．図 2 右側に動的オブジェクトの移動軌跡を示した．まず，オブジェクトが扇形のセンシングエリアに入り，出てゆくまでの時間モデルを求めた．ここでは，オブジェクトがセンシング領域の扇形を 2 等分する線に直交する角度で入ってきた場合について述べる．T：センシング範囲をオブジェクトが通過する時間，V：オブジェクトの通過速度，θ：センサのセンシング角度，r：オブジェクトがセンシング領域内に入った時点でのセンサとオブジェクトの距離とすると式 (8) が成り立つ．

$$T = \frac{2r \sin 0.5\theta}{V} \tag{8}$$

オブジェクトがセンシング領域に斜めに入ってきた場合には余弦定理を用いた式 (9) が成り立つ．なお式 (9) において，r_1 はセンシング領域に入ってきた際のセンサからオブジェクトへの距離，r_2 はセンシング領域から出る際のセンサからオブジェクトへの距離である．

$$T = \frac{\sqrt{r_1^2 + r_2^2 - 2r_1 r_2 \cos\theta}}{V} \tag{9}$$

- **動的オブジェクト＋動的アクチュエータ**

オブジェクトとセンサノードの両方が動く場合について述べる．オブジェクトをセンシングしている時間を最大化するために，センサのセンシング方向と直動系アクチュエータの移動方向の角度については，90 度，0 度，180 度の場合を取り扱う．さらに，オブジェクトとセンサの移動方向については，センサとオブジェクトが斜めに動く場合もありうるが，ここでは並行的に動く同方向と逆方向の 2 種を取り扱うとする．斜めに動く場合は次式右辺の分子をオブジェクトのセンシング範囲内での移動軌跡距離にするとよい．**図 5** に動的オブジェクトと動的アクチュエータを用いた時間モデルについて示した．まず最初に直動系アクチュエータの場合，V：オブジェクトの通過速度，v：アクチュエータの動作速度，r：オブジェクトがセンシング領域内に入った時点でのセンサとオブジェクトの距離とし，センシング方向と移動方向の角度を 90 度，オブジェクトとセンサが逆方向に動くとすると，式 (10) が成り立つ．

$$T = \frac{2r \sin 0.5\theta}{V + v} \tag{10}$$

また回転系アクチュエータの場合，その角速度を ω とすると式 (11) が成り立つ．

$$T = \theta \div \left(\omega + \frac{360V}{2\pi r}\right) \tag{11}$$

また，オブジェクトとセンサが同方向に動くとすると，直動系アクチュエータの場合は式 (12) が成り立つ．

© 2008 Information Processing Society of Japan

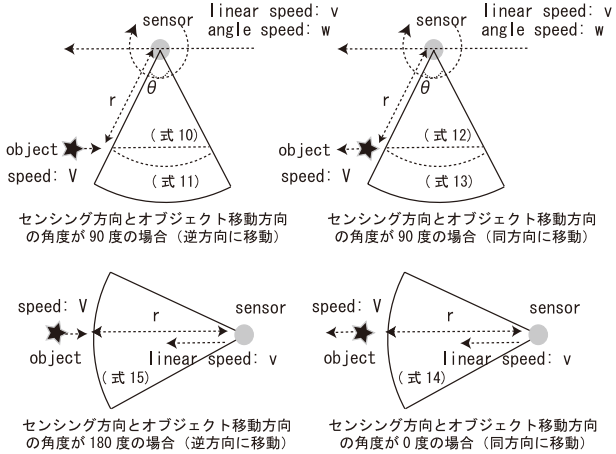


図 5　時間モデル：センサとオブジェクトが移動する場合
Fig. 5　Time model: mobile sensor and object.

$$T = \frac{2r \sin 0.5\theta}{|V - v|} \tag{12}$$

同様に回転系アクチュエータの場合は式 (13) が成り立つ．

$$T = \theta \div \left| \omega - \frac{360V}{2\pi r} \right| \tag{13}$$

センシング方向とオブジェクトの移動方向の角度を 180 度（逆方向）とすると式 (14) が成り立つ．

$$T = \frac{r}{V + v} \tag{14}$$

そして，センシング方向とオブジェクトの移動方向の角度を 0 度（同方向）とすると式 (15) が成り立つ．

$$T = \frac{r}{|V - v|} \tag{15}$$

これらの式により，ロボティックノードの動作速度とオブジェクトの移動速度の関係を算出でき，結果としてオブジェクトのセンシング時間を求められた．オブジェクトによっては，ある一定時間センシングすることにより，センシングの精度を上げられる場合も多く，そのような時間的制約の検討において，これらの時間モデルを活用できる．

上記の例では，センサのセンシング周波数が十分に高い場合を想定しており，周波数の低さによるセンシングもれを考慮していない．しかしセンシング周波数がある周波数（単位：Hz）以上でない場合は，回転や移動にともないセンシングできない領域も出てくる．以下でセンシング周波数とアクチュエータの動作速度の関係について考察する．

- **センシング周波数＋動的アクチュエータ**

  モータが回転する時間内に，不足なくすべての領域をセンシングするには式 (16) を満たす必要がある．ここで ω：モータの角速度で，θ：センサが 1 回のセンシングでカバーする角度，δ：センサのセンシング周波数である．本式において，センシング周波数を高くするか，モータの速度を遅くする分には，センシングもれは発生しない．

  $$\omega \leq \delta\theta \tag{16}$$

  同様に直動系アクチュエータ上でセンサが動作する場合には式 (17) が成り立つ．ここで，v：アクチュエータの直線速度，δ：センサのセンシング周波数，W：センサが 1 回のセンシングでカバーするセンシング幅とする．

  $$v \leq \delta W \tag{17}$$

これらの式により，ロボティックセンサノードの動作速度とオブジェクトの動作速度の関係が計算され，あるオブジェクトが一定領域内にいる間に，複数回センシングする必要がある場合等，センサのセンシング周波数やアクチュエータの動作速度の設定に役立てられる．

2.3.3　実時間モデル

ロボティックセンサノードもしくはオブジェクトのどちらかもしくは両方が動いており，そのオブジェクトがセンシング空間内にいる間にある処理を完遂する必要がある場合には，そのセンサシステムは実時間性を有する必要がある．

ここではオブジェクトが動いている場合について考察する．式 (8) と式 (9) の右辺を基に，α：センサがオブジェクトをセンスし，データをコンピュータに送信する時間，PT：コンピュータがデータを処理し，動作ポリシを決定する時間，β：コンピュータがアクチュエータと通信し，アクチュエータを動作させる時間，とすると式 (18) が成り立ち，システムはこの時間的制約を満たす必要が出てくる．

$$\frac{2r \sin 0.5\theta}{V} \geq \alpha + PT + \beta \tag{18}$$

同様に，オブジェクトがセンシング領域である扇形に斜めに入ってくる場合にはシステムは式 (19) を満たす必要が出てくる．

© 2008 Information Processing Society of Japan

$$\frac{\sqrt{r_1^2 + r_2^2 - 2r_1 r_2 \cos\theta}}{V} \geq \alpha + PT + \beta \tag{19}$$

本式により，センサやアクチュエータの動作性能，オブジェクトの移動性能と，システムの動作速度の関係を算出でき，ロボティックセンサノードの仕様により，データ処理を行うコンピュータ・ソフトウェア側の性能を決定する際に役立つ．

2.4　ロボティックセンサノード対応ソフトウェア技術

ロボティックセンサノードの構築に必要な機能である，協調処理，データ処理，検知範囲解析処理について説明する．

(1)　協調処理
センサとアクチュエータを統合的に利用するには，それらをソフトウェアにより協調させる必要がある．センサからの情報取得方法にはソフトウェアから能動的に問合せをするポーリングモデルと，センサの変化に応じてソフトウェアが受動的にメッセージを受け取るイベントドリブンモデルの 2 種類に大別できる．これらのセンサからの情報に基づきアクチュエータを制御する．

(2)　データ処理
センサをロボティックアクチュエータのコントローラとするような場合に，センサから出力されるデータとアクチュエータに入力されるデータのフォーマットや，ダイナミックレンジ等が一致しない場合がある．数値データの場合は正規化・単位変換等の処理を行い，マルチメディアデータの場合はフォーマット変換の処理が必要になる．これらの処理により任意のセンサとアクチュエータを接続できるようになる．

(3)　検知範囲解析処理
センサを用いたシステムにおいて，センサの検知範囲，検知時間等の時空間モデルが重要になってくる．たとえば環境モニタリングアプリケーションの場合にはそのシステムが部屋のどの範囲をどのくらいの時間センシングしているか等である．ロボティックセンサノードのプログラムを書く際に，パラメータとして検知範囲，移動距離等を設定することによりその検知範囲，検知時間が算出される機能が必要になる．

3.　Spinning Sensors ツールキット

Spinning Sensors ツールキットの設計と実装について説明する．またツールキットを用いて実装したセンサとロボティックアクチュエータによるアプリケーションについて，環境モニタリング，ラジコンロボットの 2 点を紹介する．

3.1　設計と実装

Spinning Sensors ツールキットは，多種多様なセンサ，アクチュエータとアプリケーションに対して必要な機能を提供するために，システム構成をノード抽象化層，フュージョン層，機能層の 3 構成に分けた．**図 6** に，センサやアクチュエータ等のハードウェア，アプリケーションソフトウェアと Spinning Sensors ツールキットの関係，そしてツールキットの階層構造について示した．

実装は Java 言語[8] を用いて行った．ユーザは本ツールキットをライブラリとしてインストールし，パスの設定をするだけでアプリケーション開発に本ツールキットを用いられる．今回本ツールキットを用いて制御対象としたセンサとアクチュエータは**表 4** のとおりである．今回はアプリケーションでの利用を考慮し，モータやロボットを扱ったロボティックアクチュエータだけでなく，PC 上で実現されるサービスコンポーネントも実装した．構築したアプリケーションは**表 5** のとおりである．センサネットワーク研究の多くが無線センサノードのみを対象としたものが多い[9),10)] が，本論文では有線と無線両方のセンサノードを活用している．

3.1.1　ノード抽象化層

ユビキタスコンピューティング[14] 環境を構成するセンサアクチュエータについては，図 6 右側にも示すようにその開発元，機能，制御方法が多岐にわたる．本ツールキットでは多様なハードウェアに対応した汎用性と，ツールキットの高機能性を両立させるために，ハードウェア制御に関するプログラムを，ツールキットが提供する抽象クラスを継承して実装する．この抽象クラスではセンサとアクチュエータをそれぞれノードとして抽象化し，ツールキットからハードウェアの種別にかかわらず透過的に扱えるようにした．

センサ抽象化クラスを継承したセンサ実装クラスを**図 7** に示す．センサを実装する際には，センサ抽象化クラスを継承し，センサの具体的な挙動については action メソッド内に記述する．アクチュエータ実装クラスについてもほぼ同様の構成になっている．

3.1.2　フュージョン層

複数のセンサやアクチュエータを協調させるには，フュージョン抽象クラスを継承してフュージョンクラスを実装する．フュージョンクラスでは任意の 2 つ以上のセンサやアクチュエータの実装クラスをメンバとして保持し，フュージョンとして登録された場合には，機能層で実現する協調処理メカニズムを利用して，メンバのセンサやアクチュエータに変化があった際に，登録されている他のメンバに変更を通知する．

フュージョン抽象化クラスを継承したサンプルアプリケーションクラスを**図 8** に示す．

© 2008 Information Processing Society of Japan

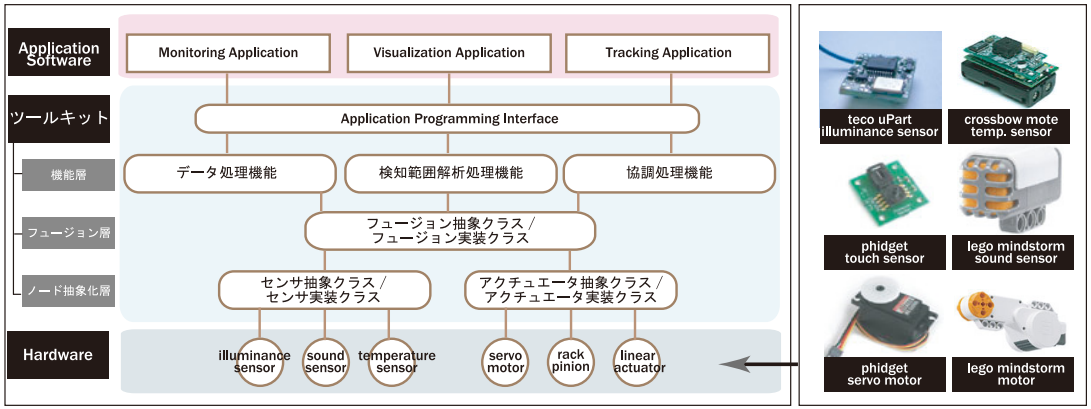


図 6　ソフトウェア構成
Fig. 6　Spinning sensors toolkit software architecture.

表 4　サポートされているセンサとアクチュエータの例
Table 4　List of supported sensors and actuators.

分類	機器名	内容	実装クラス名	行数	サイズ
Sensor	TECO uPart [11]	light, temperature, movement	UpartSensorImpl	117 行	2.83KB
Sensor	LEGO Mindstorms [12]	light, sound, ultrasonic, touch	MindstormsSensorImpl	82 行	2.18KB
Sensor	Phidgets [13]	temp, light, rotation, slider, etc.	PhidgetsSensorImpl	98 行	2.22KB
Sensor	Phidgets RFID [13]	RFID reader and tags	PhidgetsRFIDImpl	135 行	2.70 KB
Robotic Actuator	LEGO Mindstorms	motor	MindstormsActuatorImpl	107 行	2.81 KB
Robotic Actuator	Phidgets	motor	PhidgetsActuatorImpl	126 行	2.94 KB
Service Component	Aviosys IPPower	power control	IpPowerImpl	62 行	1.68 KB
Service Component	JFreeChart	graph viewer	DataViewerImpl	194 行	5.95 KB
Service Component	Apple Quicktime	movie player	VideoControllerImpl	57 行	1.46 KB

表 5　開発したアプリケーションの例
Table 5　Examples of implemented applications.

アプリケーション名	内容	実装クラス名	使用デバイス	行数	サイズ
環境モニタリング	ロボティックセンサによる環境監視	EnvMonitoring	Sensor と Motor と Chart	150 行	4.89 KB
ラジコンロボット	センサによるロボット制御	SensorControlRobot	Sensors と Robot	183 行	5.23 KB

© 2008 Information Processing Society of Japan

```
public class UpartSensorImpl extends
                          SensorManager{
  public uPartImpl(){
    //API 利用の為のインスタンス取得
    api = SpinningSensorsAPI.getInstance();
   }
  public void setCurrentParam(){
    //センサデータの登録（照度データ）
    this.setParam(SpinConstants.LIGHT, light);
   }
  public void dataParse(byte[] data){
    //センサデータの処理（uPart センサの照度データのみ取得）
     if((data[52]) == Integer.parseInt
                (this.getId().getName()) ) {
       tempe = (data[62]&0xFF);
     }
   }
  public void action(ActionState state) {
    //センサの挙動を記述する
    while(true){
      socket.receive(packet);//uPart センサからのデータを取得
      received_buf = packet.getData() ;
      dataParse(received_buf);
      this.setCurrentParam();
      api.updateNode(this);//関連ノードにデータ変更通知
    }
  }
}
```

図 7　センサ実装コード一部抜粋
Fig. 7　Sample code of sensor implementation.

```
public class EnvMonitoring extends
                        FusionManager{
 public static void main(String[] args) {
  SpinningSensorsAPI api =
   SpinningSensorsAPI.getInstance();
  //使用センサの宣言（例えば照度センサ）
  SensorManager us =
   api.getNodeInstance("UpartSensorImpl");
  us.action(new ActionState());
  api.registerNode(us);
  //使用アクチュエータの宣言（例えばモータ）
  ActuatorManager pa =
  api.getNodeInstance("PhidgetsActuatorImpl");
  api.registerNode(pa);
  //フュージョン宣言とセンサとアクチュエータの登録
  FusionManager em =
   api.getNodeInstance("EnvMonitoring");
  em.setId("realtime measurement");
  em.addRelatedNode(us.getId());
  em.addRelatedNode(pa.getId());
  api.registerNode(em);
 }
 public void update(NotifyMessage message) {
   //フュージョンに登録されているセンサかアクチュエータの取る値に変更があった場合に呼
ばれる
 }
}
```

図 8　フュージョンコード一部抜粋
Fig. 8　Sample code of fusion implementation.

フュージョンクラスでは，そのアプリケーションで使用するセンサとアクチュエータを宣言し，それらが取る値が変化した場合には関連ノードに変化を通知する update メソッドを用意した．

3.1.3 機　能　層

2 章で述べた協調処理，データ処理，検知範囲解析処理については，機能層として実装した．

協調処理では，イベントドリブン方式を採用した．各センサやアクチュエータの実装の際にイベントリスナを付加することにより，そのセンサやアクチュエータの変化を監視し，変更があった際には関連ノードに通知する．協調処理の実装はフュージョン層での関連ノードの登録と，機能層でのイベント配送メカニズムにより実現されている．

データ処理は，任意のセンサの出力をアクチュエータの入力とするような協調処理が必要な場合にそのデータの正規化・変換を行う．**図 9** 上部にデータ処理の一例を示した．ここ

© 2008 Information Processing Society of Japan

表 6　検知範囲解析処理例
Table 6　Implementation of space time model.

センサ（入力）	アクチュエータ（入力）	フュージョン（出力）	関連モデル
検知半径，検知角度	回転角度	検知面積	空間モデル：式 (1)
検知半径，検知角度	移動距離	検知面積	空間モデル：式 (2)，(3)，(4)
検知角度，オブジェクトとの距離	角速度	検知時間	時間モデル：式 (5)
検知角度，オブジェクトとの距離	移動速度	検知時間	時間モデル：式 (6)，(7)
検知角度，周波数	角速度	角度ごとの検知の可否	時間モデル：式 (16)
検知角度，周波数	移動速度	角度ごとの検知の可否	時間モデル：式 (17)

```
//データ処理：照度センサの最大値と最小値の登録例
sensor.setParam(SpinConstants.L-MAX, 220);
sensor.setParam(SpinConstants.L-MIN, 0);

//検知範囲解析処理：指向性センサの検知角度の登録例
sensor.setParam(SpinConstants.ANGLE, 140);
//検知範囲解析処理：スライダアクチュエータの移動距離の登録例
actuator.setParam(SpinConstants.DISTANCE, 50);
//検知範囲解析処理：フュージョンでの検知面積取得例
fusion.getParam(SpinConstants.SPACE);
```

図 9　機能層コード一部抜粋
Fig. 9　Sample code of function implementation.

ではあるセンサのとるべき最大値と最小値を事前に登録することにより，そのセンサの値をアクチュエータの入力として利用する際に，センサデータを正規化して利用できる．

検知範囲解析処理については，センサやアクチュエータを実装する際に，パラメータとして検知半径，移動速度，移動距離等を入力するようにし，フュージョンノードにおいてその組合せ計算を行う．図 9 下部にパラメータ登録例を示すとともに，**表 6** には検知範囲解析処理でのセンサ，アクチュエータ，フュージョンの各クラスでの入出力パラメータについて例示する．2.3 節で示した空間モデルと時間モデルについては，この検知範囲解析処理内で実装した．時間モデルや実時間モデルでのオブジェクトの移動がともなうケースについては，センサとアクチュエータの情報だけでは計算ができないために，現在のところ未対応になっている．

3.2　プロトタイプアプリケーション

Spinning Sensors ツールキットを用いてアプリケーションプログラムを作成した．これらのアプリケーションでは前章で述べたハードウェアに特化した実装クラスを利用して，表 5 に示すようにアプリケーション用にフュージョンクラスを作成した．

3.2.1　環境モニタリングアプリケーション

環境モニタリングではロボティックセンサノードの取得値をグラフ化した．センサノードにはワイヤレスセンサノードである uPart [11] を Phidgets [13] のモータ上に設置し回転可能にした．本アプリケーションのために作成したフュージョンクラスは EnvMonitoring クラスであり，その内部で UpartSensorImpl クラス，PhidgetsActuatorImpl クラス，DataViewerImpl クラスを呼んでいる．

uPart の照度センサは Taos Inc. の TSL13T という光ダイオードにトランスインピーダンス増幅器を合わせたもので，センサに対して光の入射角 0 度のときが最も感度が良い．入射角が −70 度や 70 度の角度では，感度は 50 パーセントに落ちる．入射角に対応して取得照度値が変わることは，照度センサを回転することに意味があることを示している．また，Phidgets モータの稼動範囲は 20 度から 220 度である．

3.2.2　ラジコンロボットアプリケーション

ラジコンロボットでは，Phidgets のローテーションセンサをハンドル，またスライダセンサを前後移動のアクセルとし，Mindstorm [12] で作成されたロボットを Bluetooth 経由で制御対象とした．任意のセンサをラジコンのコントローラとして利用するユニバーサルラジコンの一例となるアプリケーションである．本アプリケーションのために作成したフュージョンクラスは SensorControlRobot クラスであり，その内部で PhidgetsSensorImpl クラスと MindstormsActuatorImpl クラスを呼んでいる．

本アプリケーションではセンサの制御とアクチュエータでの実際の動作の時間差がラジコ

© 2008 Information Processing Society of Japan

ンの使用感に影響するため，実時間性が重要になってくる．本アプリケーションを動作させたところ，センサを制御してから実際にロボットが動作するまでにタイムラグがあり，ロボットを指定した場所で止める等の操作で精度が悪くなった．動作遅延の原因は，PC 上で動作しているロボットの実装クラスから Bluetooth を経由してロボットを実際に制御している通信部分にあった．

4. 評　　価

Spinning Sensors ツールキットとその時空間モデルの評価を行う．まず，ツールキットを用いて各種ロボティックセンサノードシステムを実現し，それらのシステムにより空間モデル，時間モデル，実時間モデルについて評価実験と考察を行う．また，Spinning Sensors ツールキットにより，本論文であげた即興的協調実現問題と検知範囲把握問題がどのように解決されたかを説明する．

4.1 空間モデル評価実験

空間モデル評価実験では，センサと回転可能なアクチュエータを用いたロボティックセンサノードにより，センシング範囲増加とセンシング特性について，実験 1 と実験 2 の 2 種類の測定を行った．実験 1 では環境モニタリングアプリケーションで実装したロボティックセンサノードを使用し，実験 2 では LEGO Mindstorm NXT ロボットと，それに搭載されている照度・音量・超音波センサを使用した．

実験 1 では，センサの角度とその際の照度の関係について測定を行った．ロボティックセンサノードを机上において，スポットライトをノードから水平方向に 8 cm（Position 1），16 cm（Position 2，4），24 cm（Position 3），垂直方向（高さ）0 cm の位置に設置して，ノードを回転させた．実験結果は**図 10** のとおりである．図 10 中の Position 1，2，3 を見ると，センサとスポットライトの距離や関係が照度値として現れている．Position 1 では，向きにかかわらず光がセンサまでに十分に届いてしまい，多くの角度で照度値が最大値になった．Position 2 では，センサとスポットライトの距離が近く，光の入射角度が照度値として現れている．Position 3 でも，距離が離れている分，照度値は小さいが，入射角度が照度値結果に反映されている．また Position 4 では，スポットライトを Position 2 と同じ場所に上向きに設置した．上向きに設置したために光がセンサまで届かなかった．

実験 1 で用いた照度センサを，発光物捜索センサとして使用すると，Position 3 で照度を検知できていることから検知半径を 24 cm とし，検知角度を 10 度と設定できる．それらを式 (1) に代入すると，センサをモータに乗せない場合は，$S = \frac{10}{360} \times 24 \times 24 \times \pi$ となり，

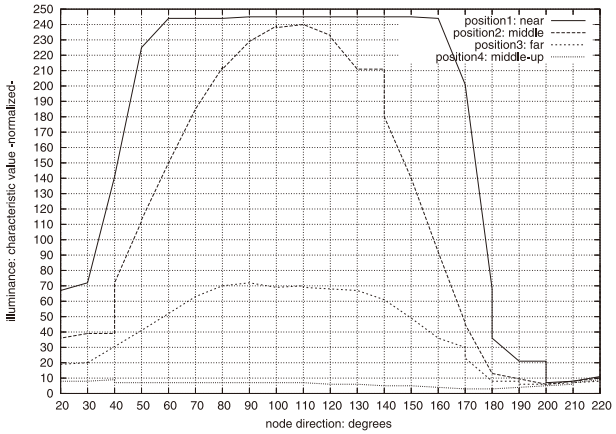


図 10　空間モデル評価実験：実験 1 の結果
Fig. 10　Space model experiment 1.

センサをモータに乗せた場合は $S = \frac{10+210}{360} \times 24 \times 24 \times \pi$ となり，センシング面積が 22 倍拡大されたことが分かる．

実験 2 では，LEGO Mindstorm NXT をロボット状に構築し，モータを使用してロボットを 1 回転させて，Mindstorm NXT 搭載のサウンドセンサ，超音波センサ，照度センサの値を計測した．実験環境は，ロボットの周辺円状の 90 度ごとにピンクノイズを発生するスピーカ，超音波センサにより認知される紙袋，そして実験 1 で用いたスポットライトを設置した．実験結果を**図 11** に示す．実験 1 と同様に超音波センサと照度センサについては，ロボットが向いている角度によって取得されるデータが変わった．サウンドセンサについては，スピーカから発生されている音がセンサに届く前に空間内に拡散されているせいか，角度による違いはほとんど出ていない．ここでは 1 回転を約 22 度ずつセンシングを行ったので，センシング領域は 16 倍に拡大した．

ロボティックセンサノードにより，1 方向の照度や物体情報しか検知できなかったセンサが，多方向の照度や物体情報を検知できるようになった．それらの情報を集計することにより，スポットライトや物体の位置・角度を推定できる．また，物体の位置を時系列的に取得し，物体の移動速度を推定することにより，たとえば，物体の角度と移動速度からデッドレコニング（推測航法）を用いて，物体の次に移動する位置を推測しそれに備えてアクチュエータを動作することも可能になる．

© 2008 Information Processing Society of Japan

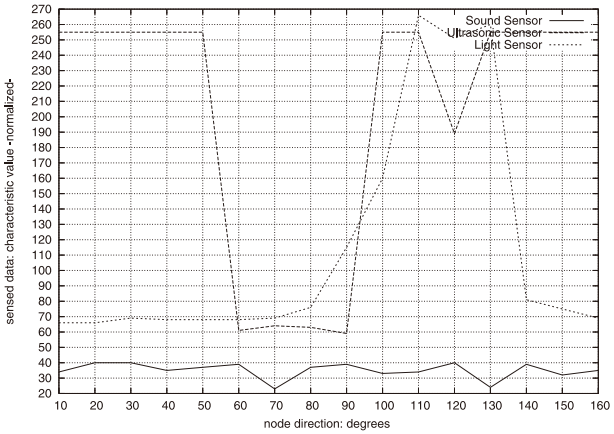


図 11　空間モデル評価実験：実験 2 の結果
Fig. 11　Space model experiment 2.

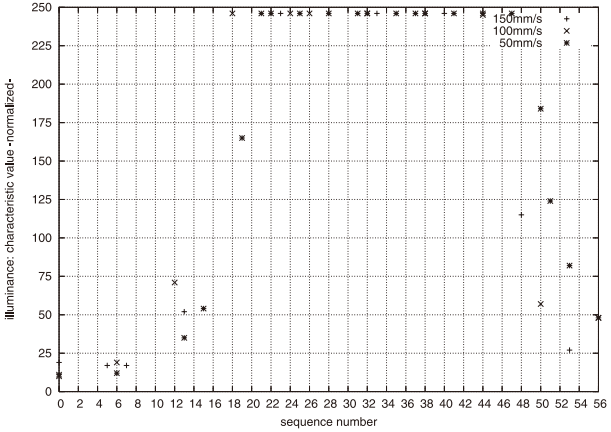


図 12　時間モデル評価実験：実験 3 の結果
Fig. 12　Time model experiment 3.

4.2　時間モデル評価実験

時間モデル評価実験では，静的なオブジェクトと動的なセンサを用いてセンシング時間の測定を行う実験 3 と，環境モニタリングアプリケーションで実装したロボティックセンサノードを使用し，センシング周波数と動的センサの関係を測定する実験 4 の 2 種類を行った．

実験 3 では，動かないオブジェクトを直線移動するセンサシステムによりセンシングするというシナリオで，uPart の照度センサ，iRobot Create [15] という移動可能なロボット，卓上ライトを用いて時空間モデルの実験を行った．本実験では，照度センサを直線移動するロボット上に設置し，卓上ライトからの照度を測るようにした．照度センサの設定と卓上ライトの明るさは固定し，ロボットの移動速度を秒速 50 mm，秒速 100 mm，秒速 150 mm と変化させた計測結果が **図 12** である．ロボットの移動距離は 2,000 mm であり，その中間点にロボットの軌跡から 450 mm 離れた場所にライトを設置した．図 12 の x 軸は移動距離 2,000 mm を 56 分割したそれぞれのポイントであり，0 がスタート地点で 56 がゴール地点を示す．また y 軸は照度値である．

ロボットの移動速度が秒速 50 mm のときはスタートからゴールまでに 40 秒かかっており，秒速 100 mm のときは 20 秒，秒速 150 mm のときは 15 秒となっている．センシング周波数は同じことから，移動速度が遅いほど細かくセンシングできていることが分かる．なお，移動速度が秒速 50 mm のときには理論的には 56 回センシングされるはずだが，センサの無線通信におけるパケットロス等の影響で，実際には 21 回しかセンシングされていない．特に移動速度が速い場合には，重要なタイミングでセンシングに失敗する恐れもあるので，モデル式を用いて移動速度に余裕を持たせて冗長にセンシングする等の工夫が必要になる．ここでは最もセンシング回数が多い秒速 50 mm の実験結果を用いて以下の考察を行う．本実験において，照度センサの値が 246 以上になっている場合を対象オブジェクト（ライト）を検知できていると仮定すると，式 (6) に $r = 63.6$（図 12 中の値が 246 になっている部分の距離と，スポットライトと軌跡間の距離から算出），$v = 50$，図 12 中の値が 246 になっている部分の通過時間 19 秒を代入すると，$\theta \approx 90$ になる．したがって本実験環境においては，照度センサは指向性を持っており，その検知角度が約 90 度になることが分かった．このセンサの指向性は，ハードウェアの仕様と照らし合わせると適当な値になっている．

実験 4 では，実験 1 で用いた uPart センサの読み取りサイクルを 576 ミリ秒，Phidgets モータの角速度は秒速約 3.4 度とし，センシング周波数と動的アクチュエータの関係である式 (16) について評価を行った．ただし，モータの制約により，モータは 20 度から 220 度の間を 10 度ずつ停止しながら段階的に動かし，各段階で 3 秒ずつ停止させた．このセンサのセンシング周波数と，1 回のセンシングあたりの角度（10 度）を式 (16) に代入すると，$\omega = \frac{1000}{576} \times 10$ となり理論上の最大角速度は秒速約 17.4 度になる．本実験でのモータの角速度は秒速約 3.4 度であり，各 10 度ごとに約 5 回センシングされる計算になる．

© 2008 Information Processing Society of Japan

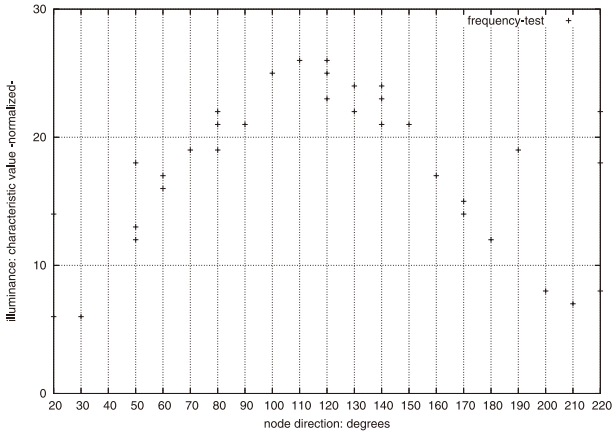


図 13　時間モデル評価実験：実験 4 の結果
Fig. 13　Time (frequency) model experiment 4.

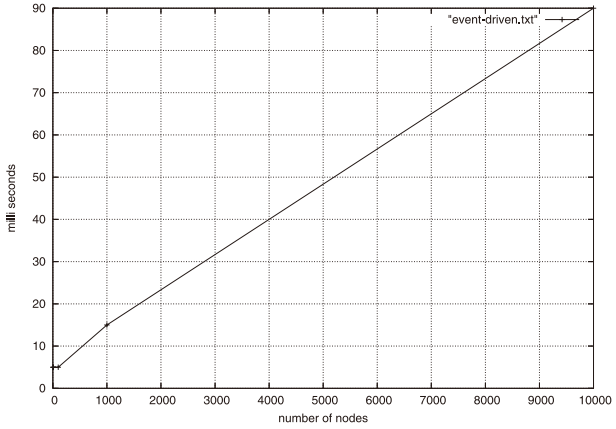


図 14　実時間モデル評価実験：実験 5 イベント配送時間
Fig. 14　Realtime model experiment 5: event distribution time.

結果を**図 13** に示す．各角度においてデータの取得数にばらつきが発生している．今回はワイヤレスセンサノードを使用したために，ネットワークの遅延，パケットロス等がセンシング数のばらつきに影響している．精度の高いデータを取得するには，あえて各角度でのセンシング数を増やしデータの平均をとる等の計算をし，また消費電力やネットワーク帯域に制限があり少ないセンシング数にしたい場合はセンサのセンシング周波数を下げるか，モータの回転速度を速めることによりセンシング数を削減できる．

4.3　実時間モデル評価実験

実時間モデル評価実験では，コンピュータに USB ケーブルで接続された Phidgets の RFID センサとタグを用いて，RFID タグがセンシング範囲内に滞在する時間とロボティックセンサノードの動作速度の関係を測定する実験 5 を行った．本実験では式 (18) より，RFID タグがセンシング範囲内に滞在している時間内に，タグの検知，タグに基づいたサービスの処理，アクチュエータの動作を終える必要がある．本実験で使用した RFID センサは，1 秒あたり 30 回のタグ認識をし，その検知範囲は，センサの大きさが 7 cm 四方で検知距離が 6 cm であるために，半径 9.5 cm となる．

まず式 (18) における α については，センサのセンシング周期が 1 秒あたり 30 回のタグ認識であることから，最大で 34 ミリ秒となる．また PT については，タグの有無の判定とその情報を関連するアクチュエータに通知する時間がかかる．タグの有無をイベントとして関連アクチュエータに通知する時間について，**図 14** に示した．今回の実験では関連するアクチュエータは，コンピュータ上の画面表示か Phidget のモータ 1 台としたため，イベント配送にかかる時間は 5 ミリ秒以下であった．また，β については，タグの存在に応じてコンピュータの画面に情報表示については 5 ミリ秒以下，モータのアクチュエーションについては角度に応じて変わるものの，最大移動角度である 220 度においては 700 ミリ秒となった．したがって，タグの検知からアクチュエーションまでの最大所要時間は 744 ミリ秒となり，タグが 745 ミリ秒以上センシング範囲内に滞在すれば実時間性を満たす．センシング範囲は直線距離にして 19 cm であるので，タグの移動速度は毎秒 25.5 cm 以下である必要があることが分かった．

4.4　ツールキット評価

Spinning Sensors ツールキットの評価について，本論文で問題としてあげていた即興的協調実現問題と検知範囲把握問題の解決方法を軸に評価と考察を行う．

即興的協調実現問題の解決方法として，ツールキットを用いたアプリケーション開発を実現した．本ツールキットでは，センサやアクチュエータの開発メーカや種別に依存しないでプログラミングができるようにハードウェアを抽象化し，またフュージョンという協調概念を導入し，センサとアクチュエータ間の通信を容易に変更できる機構にした．既述のセンサ実装クラスとアクチュエータ実装クラスを用いてアプリケーションを構築する場合は，200

© 2008 Information Processing Society of Japan

行弱で記述できた．本ツールキットを使用せずにすべてのプログラムをスクラッチから書く場合には約 400 行であったことから，プログラムコード量は約半分になった．センサやアクチュエータ等のハードウェアに特化した実装クラスと，アプリケーション用のフュージョンクラスが独立していることにより，様々なハードウェアによる協調を試したい場合や構築済みアプリケーションのセンサ部分だけを別種のセンサに変えたい場合等に簡単に変更ができる．また，ツールキットを用いたテストユーザによる使用感については，「ツールキット特有の抽象化やフュージョン等の概念や記法を学ぶ必要があるものの，プログラミング記法を習得してしまえば，その後の開発では従来より簡便なロボティックセンサノード構築が可能になった」，「ロボティックセンサノード構築に必要な基本的な協調設定や検知範囲計算等の機能が用意されており，その部分の実装が省略されるので便利である」等の好意的な意見をいただいた．

検知範囲把握問題の解決方法としては，ツールキットの設計段階においてセンサとアクチュエータの組合せによる時空間モデルの変化を 2.3 節に示した．これらのモデルにより，実機を実際に組み合わせることなく，ロボティックセンサノードのセンシング性能を導出できるようになった．また，3 章では，ツールキットの機能層の検知範囲解析処理を説明し，開発者がセンサやアクチュエータのパラメータとしてセンシング特性を実装クラスに記すことにより，フュージョンクラスにおいて実現されるセンシング範囲が算出される仕組みを構築した．本機能により，開発者は複雑なセンシングモデルの計算を行うことなく，センシング範囲を取得できる．4 章では実際のロボティックセンサノードを用いて評価実験を行い，各種時空間モデルにおける検知範囲の結果を示すとともに，モデルの妥当性を示した．汎用的に使用できるセンサやアクチュエータが豊富に存在する場合においては，センシングしたい範囲や頻度を指定しセンサやアクチュエータを決める方が開発者には便利な場合もある．しかし現在のように汎用的に組み合わせて利用可能なセンサやアクチュエータの種類が限られている場合においては，本論文における組合せによるセンシング範囲・時間の算出も有用になる．

5. 関 連 研 究

センサネットワーク環境用のアプリケーション構築キットとしては，文献 16) がある．本文献では，Crossbow 社の MOTE [17] をターゲットとして，無線センサネットワーク環境で特に重要なメモリリソースや電源リソースの低減のために，SNACK と呼ばれる新たなコンフィギュレーション言語を用意している．SNACK では，MOTE での開発言語である NesC の非効率さに着目をしており，本論文のような汎用性，時空間モデルの検討，ロボティックセンサノードへの適用等は現在のところは行われていない．

ロボット用ミドルウェアとしては，ロボット制御に関するソフトウェアのモジュール化を推進する RT Middleware [18] がある．RT Middleware は CORBA をベースにした分散コンポーネント技術であるが，使用するにあたって CORBA の動作環境を構築する手間がかかる．また，必ずしもセンサ特有の検知範囲やデータ処理に対応しておらず，センサに関する機能が充実していない．

センサネットワークシステムの検知範囲に関する関連研究を紹介する．文献 1) では，センサを 2 次元空間上を自由に移動可能なロボットに装着することにより，検知範囲の拡大を図っている．文献 19) では，センサを人間に寄生させる手法でセンサを動かしている．また，2 次元空間上を自由に移動可能なこれらの研究とは正反対に，文献 20) ではユビキタスコンピューティング実験環境に移動不能なセンサを配置して環境監視を行っている．我々の研究では，1 次元的な動きをするシンプルなロボティックアクチュエータを用いることによる検知範囲の拡大と，実験空間に静的に配置することによる低コストなアーキテクチャの双方の利点を生かす．

検知範囲の評価方法についても各種研究がなされている．文献 2)，3) では，ワイヤレスアドホックセンサネットワークシステムにおける露出量（Exposure）という概念を提唱した．本文献では，動く対象オブジェクトに対して，システムがそのオブジェクトを最も検知できない状況を最低露出量パスと定義し，その算出方法やシミュレーションを提示している．本論文と本文献の大きな違いは，本文献ではセンサノードが動かないという前提の下にシミュレーションやモデル化が行われているところにある．本論文では，センサが回転・移動することにより，検知範囲を拡大することができ，その検知範囲の算出やモデル化はより複雑になる．

6. ま と め

本論文では，既存のセンサシステムやセンサアクチュエータ統合システムにおける即興的協調実現問題と検知範囲把握問題について提示した．そしてそれらを解決するロボティックセンサノードモデルを時空間モデルとともに定義し，ロボティックセンサノード機構を実現するプログラミングツールキットである Spinning Sensors ツールキットを提案した．

Spinning Sensors ツールキットを使用することで，汎用のセンサとロボティックアクチュエータを用いてロボティックセンサノードを構築でき，静的なセンサノードを動的にし，セ

© 2008 Information Processing Society of Japan

ンシング範囲の拡大を実現できる．本論文ではまずロボティックセンサノードのセンシングに関して各種時空間モデルを定義し，ロボティックセンサノードの時間的，空間的センシング性能における有効性を示した．また，Spinning Sensors ツールキットを用いて環境モニタリングアプリケーションとラジコンロボットアプリケーションを実現し，ロボティックセンサノードにおける時空間モデルについて検証を行った．さらに，ロボティックセンサノードを用いた評価実験により，実機を用いてのセンサとアクチュエータの時空間モデルの評価を行い，空間モデル，時間モデル，事実時間モデルのすべてについて検証を行った．空間モデルでは指向性のあるセンサを回転させる意義について検証し，時間モデルでは移動するセンサによるオブジェクトセンシング時間の測定や，センサの周波数とセンサの回転の関係を示した．また実時間モデルではセンシング，計算，アクチュエーションといったセンサシステムの一連の処理について測定を行い，実時間性を満たすために必要な処理時間を実験とモデルより導き出した．

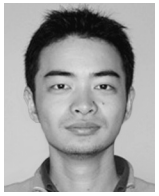
今後の課題としては，オブジェクトの移動方向や移動速度検知機構の開発，時空間モデルとツールキットの 2 次元や 3 次元への対応，複数ロボティックセンサノードへの対応や，ツールキットにおいてモデルのビジュアライゼーションを実現し，画面による時空間センシングモデルの確認機能等を実現したい．

参 考 文 献

1) LaMarca, A., Brunette, W., Koizumi, D., Lease, M., Sigurdsson, S., Sikorski, K., Fox, D. and Borriello, G.: Making Sensor Network Practical with Robotics, *Proc. 1st International Conference of Pervasive Computing* (Aug. 2002).
2) Meguerdichian, S., Koushanfar, F., Qu, G. and Potkonjak, M.: Exposure in wireless ad-hoc sensor networks, *International Conference on Mobile Computing and Networking* (*MOBICOM*) (2001).
3) Veltri, G., Huang, Q., Qu, G. and Potkonjak, M.: Minimal and maximal exposure path algorithms for wireless embedded sensor networks, *The 1st ACM Conference on Embedded Networked Sensor Systems* (*Sensys*) (2003).
4) Nakazawa, J., Tokuda, H., Edwards, W.K. and Ramachandran, U.: A bridging framework for universal interoperability in pervasive systems, *The 26th IEEE International Conference on Distributed Computing Systems* (2006).
5) Iwai, M. and Tokuda, H.: Evaluation of a robust middleware for enormous distributed task handling, *The 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications* (2005).
6) Kumar, R., Wolenetz, M., Agarwalla, B., Shin, J., Hutto, P., Paul, A. and Ramachandran, U.: Dfuse: A framework for distributed data fusion, *The 1st ACM Conference on Embedded Networked Sensor Systems* (*Sensys*) (2003).
7) Li, S., Son, S. and Stankovic, J.: Event detection services using data service middleware in distributed sensor networks, *The 2nd International Workshop on Information Processing in Sensor Networks* (*IPSN*) (2003).
8) Gosling, J., Joy, B. and Steele, G.: The java language specification (1999).
9) Kahn, J.M., Katz, R.H. and Pister, K.S.J.: Next century challenges: Mobile networking for “smart dust”, *International Conference on Mobile Computing and Networking* (*MOBICOM*) (1999).
10) Estrin, D., Govindan, R., Heidemann, J. and Kumar, S.: Next century challenges: Scalable coordination in sensor networks, *International Conference on Mobile Computing and Networking* (*MOBICOM*) (1999).
11) Beigl, M., Decker, C., Krohn, A., Riedel, T. and Zimmer, T.: uParts: Low Cost Sensor Networks at Scale, *Demonstration Proc. 7th International Conference of Ubiquitous Computing* (Sep. 2005).
12) LEGO Group: LEGO Mindstorms NXT (2006). http://mindstorms.lego.com/
13) Phidgets Inc.: Phidgets Analog Sensors and Servo Motors (2003). http://www.phidgets.com/
14) Weiser, M.: The computer for the twenty-century, *Scientific American*, Vol.265, No.3, pp. 94–104 (1991).
15) iRobot Corporation: iRobot Create (2006). http://www.irobot.com/create/explore/
16) Greenstein, B., Kohler, E. and Estrin, D.: A sensor network application construction kit (snack), *The 2nd ACM Conference on Embedded Networked Sensor Systems* (*Sensys*) (2004).
17) Hill, J. and Culler, D.: Mica: A wireless platform for deeply embedded networks, *IEEE Micro*, Vol.22, No.6, pp. 12–24 (2002).
18) Ando, N., Suehiro, T., Kitagaki, K., Kotoku, T. and Yoon, W.: Rt-component object model in rt-middleware – distributed component middleware for rt (robot technology), *IEEE International Symposium on Computational Intelligence in Robotics and Automation* (*CIRA*) (2005).
19) Laibowitz, M. and Paradiso, J.: Parasitic Mobility for Pervasive Sensor Networks, *Proc. 3rd International Conference of Pervasive Computing* (May 2005).
20) Intille, S., Larson, K., Tapia, E., Beaudin, J., Kaushik, P., Nawyn, J. and Rockinson, R.: Using a Live-In Laboratory for Ubiquitous Computing Research, *Proc. 4th International Conference of Pervasive Computing* (May 2006).

(平成 19 年 9 月 7 日受付)
(平成 20 年 2 月 5 日採録)

© 2008 Information Processing Society of Japan



青木　崇行（正会員）

2003 年慶應義塾大学大学院政策・メディア研究科修士．同年ソニー株式会社入社．画像信号処理の研究に従事．2006 年より慶應義塾大学大学院政策・メディア研究科博士課程在学中．主に，センサネットワーク，ロボティックセンサノード，ユビキタスシステム，画像処理等の研究に従事．

中澤　　仁（正会員）

2000 年慶應義塾大学大学院政策・メディア研究科修士．2003 年慶應義塾大学大学院政策・メディア研究科博士．2004 年から 2005 年にかけてジョージア工科大学訪問研究員．現在，慶應義塾大学大学院政策・メディア研究科講師．主に，分散オブジェクト指向システム，オブジェクト移送ミドルウェア，ホームネットワーク等の研究に従事．

徳田　英幸（正会員）

1977 年慶應義塾大学大学院工学研究科修士．1983 年ウォータールー大学 Ph.D.（Computer Science）．同年カーネギーメロン大学計算機科学科勤務．1990 年同学科研究准教授．現在，慶應義塾大学環境情報学部学部長．主に，分散リアルタイムシステム，マルチメディアシステム，超並列・超分散システム，ユビキタスシステム等の研究に従事．IEEE，ACM，日本ソフトウェア科学会各会員．

© 2008 Information Processing Society of Japan