2007－UBI－14 (9)
2007／5／23

neighborSense: 相対的な位置関係を用いた実世界の状況把握

柳沢 豊† 前川 卓也† 岡留 剛†

† 日本電信電話株式会社 NTT コミュニケーション科学基礎研究所
E-mail: †sasama@sys.i.kyoto-u.ac.jp, ††{takashi_hattori, yutaka}@cslab.kecl.ntt.co.jp,
houmi@idea.brl.ntt.co.jp

あらまし 筆者らは，コンテクストアウェアサービスのように，実世界で生じるイベントや状態の変化をトリガーとしてサービスを提供するアプリケーションのための，イベントの形式的な記述方法と，それに基づいた実世界の状況把握の方法に関する研究を進めている．本稿では特に「人が部屋に入った」「本を机の上に置いた」というような，物体間の相対的な位置関係が変化するイベントを記述する方法について述べる．具体的には，図形間の位相幾何学的な関係についての表現を，イベントの記述法に導入することで，記述の曖昧性や冗長性を削減する試みを行った．さらに本稿では，提案する方法に基づいて実世界のイベントを記述し把握するために必要な，物体間の「接触」を検知できる neighborSense と呼ぶセンサシステムについて述べる．

キーワード センシング，イベント記述，状況把握，位相幾何学的関係

neighborSense: Event Handling based on Relative Positioning

Yutaka YANAGISAWA†, Takuya MAEKAWA†, and
Takeshi OKADOME†

† NTT Communication Science Laboratories, NTT Corporation
E-mail: †sasama@sys.i.kyoto-u.ac.jp, ††{takashi_hattori, yutaka}@cslab.kecl.ntt.co.jp,
houmi@idea.brl.ntt.co.jp

Abstract Our aim of this work is the presenting an event description model for context-aware applications, which provides various services based on the events occurring in the real world. In this paper, especially we focused on a strict description model to represent 'relative' events, such that "a person enters in the room," "a book is put on the desk," and so on. To describe such events without redundancy, we introduce the topological diagrammatic relations into the event description model. Moreover, this paper describes our proposed *neighborSense* system to detect an object, which touches to another objects physically, for handling 'relative' events.

Key words sensing, event description model, context-aware, topological relations

1. はじめに

近年，実世界に埋め込まれた多数のセンサノードから集めたデータを使って，さまざまなサービスを提供しようとする研究が進んでいる．代表的なサービスとしては，コンテキストアウェアサービス [8] [7] [1]，ロケーションアウェアサービスがある [5] [4] [6]．これらのサービスでは，ユーザや実世界の物の状態や位置に応じて，ユーザに提供するサービスの内容を動的に変える．どのようなサービスを提供するかは，センサで取得したさまざまな実世界のデータを処理して決める．中でも，ユーザや物体の物理的な位置 (location) をあらわすデータは，サービスの内容や質を決定する上で重要なデータである．このような，物体の位置を取得するシステムは，測位システムあるいは位置検出システム (positioning system) などと呼ばれる [3]．

従来の測位システムでは，物体の位置データは通常 $\langle x, y, z \rangle$ のような座標値の組で表現された，絶対的な座標点データを直接取得．これは，絶対的な位置を取得し

ておけば，あらゆるサービスに使えるという考えに基づいている．これ自体は正しい思想である．しかし，人間が物体の位置を知ろうとしたり，物体の位置に関する条件を記述するときに，実際問題として絶対位置座標を使った表現が必要なことは少ない．いくつかの例を示す．

- 「私の本はどこにあるか？」という問いに対し，「座標 $\langle x, y, z \rangle$ に存在します」という絶対的な値で答えが得られても，人間には直感的にその位置を認識することは難しい．そのような絶対値による表現ではなく，他の物に対する相対的な位置表現，例えば「あなたの部屋の机の上にあります」というような表現のほうが，より直感的に人間が理解しやすい．
- 特定の場所にある物体が何かを知りたいとき，絶対的な位置表現を用いると，例えば「部屋の座標 $\langle x, y, z \rangle$ から高さ h 幅 w 奥行き d で囲まれる矩形領域にある物体は何ですか？」という形で問いを表現する必要がある．しかし，このような絶対的な表現での問いの表現よりも，例えば「本棚に入っている本は何がありますか?」のような物体間の相対的な位置関係を用いた表現のほうが，人間にとっては記述が容易である

このように，位置情報を扱う物体に関する問い合わせでは，ほとんどの場合は何か別の物体を基準として，その物体との相対的な位置関係を用いて表現するほうが，人間にとって理解しやすい．つまり，位置情報を扱うアプリケーションには，相対的な位置表現を用いた表記，および位置の認識ができる能力が求められる．そこで筆者らは，この “相対位置表現” を用いて形式的に物体の位置関係やイベントを記述する方法について議論する．

本稿では，図形間の位相的な関係を表す 8 つの単語 (disjoint, contains, inside, equal, meet, cover, covered by, overlap) に「support」を加えた 8+1 の関係を基本として，相対位置表現を形式化する方法を提案する．図形の位相的な関係とは，図形間の関係を推論するために提唱された関係であり．例えば「 図形 A の中に図形 B が入っており， 図形 B の中に図形 C が入っているならば，図形 A は図形 C を含む」というような推論に使われる．こうした推論は，地図データに対するデータマイニングや，空間推論の分野で利用されてきた．この図家関係は物理的に不変な現象に基づいているため，相対的な位置関係を一意に記述できる．この記述法を，物体の形状間の相対的な位置関係を記述する方法として用いる．

本稿ではさらに，提案する方法で記述されたイベントを，センサを使った検出するシステムである neighborSense について，その概要を述べる．このシステムでは，物体同士が接触したことを検知するセンサと，物体の形状の情報を組み合わせて用いることで，物体間の相対的な位置関係を推定することができる．

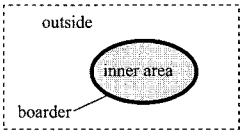


図 1 領　　域

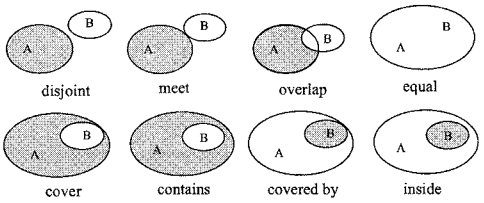


図 2 図形間の位相的な関係

以下，本稿ではまず 2 章で 9 つの図形間の関係を用いた，物体の相対位置表現について述べる．次に 3 章では実際にイベントを記述する方法について述べ，4 章では相対位置表現の基本となる，物体間の「接触」関係をセンサで検知する neighborSense システムについてその概要を述べる．

2. 相対位置表現

物体間の相対的な位置を表現する単語としては，上下左右，前後などの物体の並んでいる位置関係を表す単語や，中に入っている，囲まれているといった包含関係を示す単語，積まれている，吊るされているといった，物体の重さの支持／被支持関係を表す単語などがある．こうした，相対的な関係を用いて物体の位置や状態を表す試みは，これまでにもインタフェースの研究などを通して行われてきてはいる．しかし，各単語の指し示す意味がシステム毎に異なっており，これがユーザの混乱を招いたり，センシングを行う上での障害ともなっていた．この問題を解決するためには，相対的な位置表現の曖昧性を取り除き，どのシステムでも同じ単語が同じ物理的な状態を指し示すよう，定義を与える必要がある．

本章では，この問題の解法として，物体の相対的な位置関係を物体の形状と物体間の物理的な接触関係を用いて表現することにより，表現の曖昧性を取り除く方法について述べる．具体的な方法としては，図形間の関係を位相幾何学的な関係を用いて一意にかつ曖昧性なく表現する方法を導入する．まず 2.1 では，ベースとなる図形間の位相幾何学的な関係を表す方法について述べる．次に 2.2 で図形間の関係記述法を応用し，物体間の静的な位置関係を記述する方法（相対位置表現）について述べる．2.3 では物体の重量の伝播に着目した「支持」関係と呼ぶ相対的な関係を導入し，相対位置表現の表現力を

高める方法について述べる．そして 2.4 では実際に相対的位置表現を用いてイベントを記述する方法について述べる．

2.1 図形間の関係

実世界中にある，固体の物体は必ず物理的に何らかの形状をもっている．物体の形状は基本的に不変であり，また比較的容易に計測することができる．形状の変化が起こる場合でも，予めどのような形状に変化する可能性があるかを予測することは容易である．また，ある 2 つの物体が接しているかどうかについては，物理的に測定することで必ず一意に決定することが可能であり，曖昧性は生じない．こうしたことから，物体の形状と接触という情報をイベントを表す基本情報として用いれば，どのようなシステムにおいてもイベントを曖昧性なく一貫性を持って表現できると考えられる．

こうした形状（図形）の情報と，それらの図形間の相対的な関係を一意に記述しようとする試みは，図形推論と呼ばれる研究の中でいくつか行われてきた．その試みは，大きく分けるとユークリッド幾何学的なアプローチと，位相幾何学的なアプローチがある．前者のアプローチは，図形間の関係を，平行関係や相似関係，直交関係などといった，ユークリッド幾何学に基づく関係で記述する方法である．この方法は，図形の「絶対的な」位置や形状情報をベースに用いている．

一方で後者は，図形の関係を「交わっている」「包含している」「共有点をもたない」といった，位相的な関係を使って表す方法である．この方法は，図形の絶対的な形状には関係なく，図形と図形が相対的にどのような関係を持つかという点のみに注目して，関係を記述するものである．筆者らの目標は，物体間の相対的な関係に着目してイベントを記述するものであるため，後者のアプローチをもとに相対位置表現を行うアプローチをとる．位相的なアプローチの代表的なもののひとつに，Egenhöfer らの提唱する記述法がある．これは，各図形をまず図 1 に示すように「内側」「境界」「外側」の三つの空間をもつ閉じた領域であると定義し，他の図形のどの空間と共有点を持つかによって，関係を定義する方法である．，Egenhöfer らによれば，この定義に基づいて関係を表現すると，あらゆる 2 つの閉じた領域の関係は，必ず 8 つの関係 (disjoint, contains, inside, equal, meet, cover, covered by, and overlap) のいずれかに表現できることが証明可能である [2]．図 2 は，この 8 つの関係を図示したものである．ここではこの研究の結果を用い，これら 8 つの関係を用いて図形間の相対的な関係を記述する方法のみ定義する．

[定義 1] 領域 A と B が関係 R であるということを $R(A,B)$ と記述する．この記述 $R(A,B)$ のことを，topological diagramatic relation (TDR) と呼ぶ．

例えば，A と B が meets の関係をもつことは meet(A,B) と記述する．なお contains, cover, inside, covered by については，$R(A,B) = R(B,A)$ は成り立たない．これら以外の関係については，$R(A,B) = R(B,A)$ は成り立つ．一方で，図から容易に推測できるように inside(A,B) と contains(B,A) は等価な表記である．同様の等価性は cover と covered by についても言える．このとこから，本稿では TDR の関係 R としては，disjoint, meet, overlap, equal, contains, cover の 6 つのみを表記として用いるものとする．

ある領域 A, B について，$R_1(A,B) \wedge R_2(A,B)$ となるような R_1, R_2 の組みは存在しないことが証明されている．つまり，各関係は互いに背反である．また同時に，あらゆる 2 つの図形間の関係は，6 つの関係のいずれかで表現できることも証明されている．つまり，TDR を用いれば，図形間の関係は必ず一意に，かつ必ず表現できることを意味する(注1)

2.2 静的な位置関係の記述

TDR を用いることで，物体間の相対的な関係を図形の位相的な関係として記述することが可能であることは述べた．しかし，実際にこれを用いてイベントの記述を行うためには表現力が不足しており，論理演算子 (AND '$\wedge$', OR '$\vee$') や時間経過の表現を導入する必要がある．後者については，イベント記述の章で述べることとし，ここではまず静的な（時間経過のない）物体の関係記述の表現力を高めるために，論理演算子の導入を行う．なお，否定表現である NOT '$\neg$' は導入しない．これは，脚注 1 のとおり表現の一意性を損なうためである．なお，否定に近い表現として 'disjoint' 関係を使うことが可能である．

関係 'contains' と 'cover' は，実際のアプリケーションではほとんど同じ意味として扱ってよい場合が多い．実際に cover(A,B) は contains(A,B) に意味的に包含されているため，特に区別する必要がない限り，以下では contains(A,B) のみを用いて関係を記述するものとする．(注2)．区別が必要な場合は，contains$'$(A,B) という表記を用いるものとする．これは contains(A,B) $\wedge \neg$ cover(A,B) であるような関係を意味する．

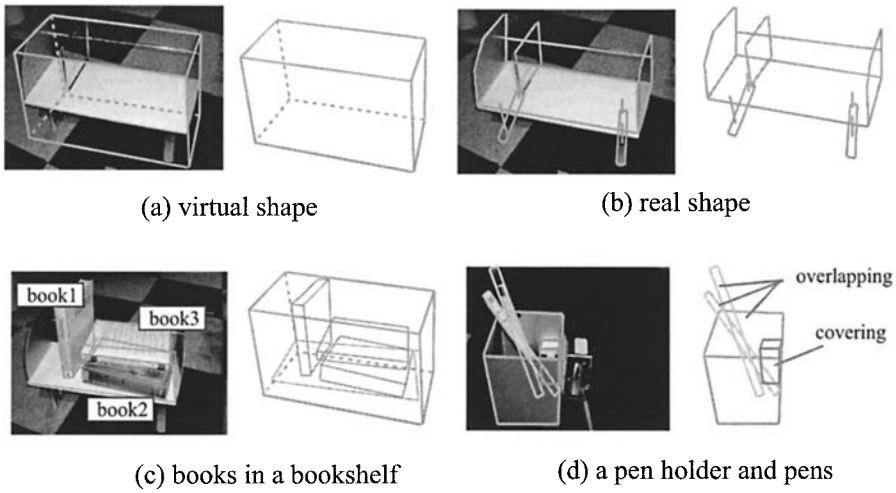
TDR と論理演算子を使って，実際にいくつかの静的な物体の相対関係を表記した例を示す．

（1） 本が箱の中にある: contains(box, book)

（2） 人が一階 (floor1) にいるが room1 にはいない: contains(floor1, person) $\wedge$ disjoint(room1,person)

(注1)：NOT ($\neg$) を用いる場合は，この限りではない点に注意．

(注2)：逆に，contains(A,B) は cover(A,B) に意味的に包含されてはない



(a) virtual shape　(b) real shape

(c) books in a bookshelf　(d) a pen holder and pens

図 3　物体間の相対的な関係

（3） 絵が壁に掛かっている: meet(wall, picture)

（4） ペンが机の上のペン立ての中にある: meet(table, penholder) ∧ (overlap(penholder,pen) ∨ contains(penholder,pen))

（5） book1, book2, book3 が本棚の中にある: contains(bookshelf,book1) ∧ contains(bookshelf,book2) ∧ contains(bookshelf,book3) ∧ meet(book1,book2) ∧ meet(book2,book3)

ここで，5 つめの例の記述には 'contains(bookshelf, book1)' という TDR を含んでいるが，このケースは 'meet(bookshelf, book1)' という表記でも記述できるようにも見える．なぜなら，本が本棚の底板のみに接している (meet) と解釈することもできるからである．こうした表記の混乱を防ぐためには，予め物体の「形状」をどう定義するかという点と，と物体をどのレベルまで分割して区別するかという点について，決定しておく必要がある．前者は，本棚をひとつの直方体であるとみなすか (すなわち図 3(a) のような仮想的な形状として扱うか)，図 3(b) のように本棚の板の部分のみを形状として厳密に扱うのか，という点である．物体の区別のレベルについては，本棚を「本棚」というレベルで扱うか，「本棚の底板」「本棚の縦板」というような部分にまで分割して区別するかということである．

いずれの方法をとった場合でも，形状や区別のレベルを固定して，すべての関係の記述において同じ形状，同じレベルを用いれば，一貫性を保つことはできる．例えば，本棚を厳密な形状で扱った場合は，'meet' を使って次のように書き換えることができる．

- meet(bookshelf, book1) ∧ meet(bookshelf, book2) ∧ meet(bookshelf, book3) ∧ meet(book1, book2) ∧ meet(book2, book3)

この記述はさらに，本棚の底板，縦板といった部分に区別して記述することもできる．いずれにしても，表現することは可能ではあるが，あらゆる状態記述において一貫して同じ形状，区別のレベルを用いることが必要である．

区別する物体のレベルを変更することで，物体間の表記の詳細度が高まることもある．例えば図 3(c) のようなケースは，本棚の各部位を区別すると，次のように表現できる．

- contains(bookshelf, book1) ∧ contains(bookshelf, book2) ∧ contains(bookshelf, book3) ∧ meet(book1, book2) ∧ meet(book2, book3) ∧ disjoint(bottom-board-of-bookshelf, book3)

これは，もし本棚の各部位を区別しない場合は，すべての本が単に本棚に含まれているという記述しかできないため，book3 が底板に触れていないという状態は記述できない．つまり，表現したい状態の詳細度を予め想定してから，物体の構成要素をどこまで区別するかについて考えておく必要がある．

これに関連して，関係 'overlap' は，しばしば関係 'contains' と意味的に等価であるケースがある．例えば図 3(d) のように「ペンがペン立てに入っている」という状態は overlap(penholder, pen) ∨ contains(penholder, pen) のように記述できるが，実質的には overlap(penholder, pen) という関係のことを指していて，contains(penholder, pen) という状態自体はほとんど生じない．すなわち「 overlap である状態を主として想定するが，con-

tains であるケースも含む」という状態である．このような状態は，棚やかごなどといった，完全には閉じていないが，他の物体を包含することのできる形状をもつ物体にはしばしば生じる状態であり，それらについて overlap ∨ contains という関係を厳密に記述することは煩わしいことが多い．

このため，ここでは 'ocontains' という表記を導入して，表現の簡略化を図る．すなわち ocontains(A, B) ↔ overlap(A,B) ∨ contains(A, B) を意味する．この表記を使うと，先の例は次のように書き換えられる．

- ocontains(penholder, pen)

なお，contains′ (A,B) に対応する ocontains′(A,B) という表記も導入することで，表記の一貫性を保てる．

2.3 「支持」関係

ここまでに，6 つの TDR (disjoin, meet, contains, equal, overlap, cover) をベースとした，物体間の相対位置表現について述べてきた．これらの関係だけでも，物体間の「図形的な特徴」に基づいて関係を表す上では必要十分ではある．これに加え，実際のアプリケーションにおいては「物体の重量を支える関係」を用いると，関係の表記がより簡潔で，なおかつ直感的に理解しやすいものになる場合がある．

例えば，テーブルの上にコップがあるという関係は，TDR を用いると次のように記述できる．

- meet(table, cup)

この記述は，厳密には「テーブルにコップが接している」という状態を意味しており，机の上面にコップが接しているかどうかは，この記述からでは判断できない．コップの上にテーブルが載っている状態であっても，そのような状態がありえるかはともかくとして，記述の示す意味としてはありえる状態である．より厳密に「コップがテーブルの上に載っている」ことを示す，最も簡単な方法としては（重量の）「支持」を意味する 'support' という関係を導入する方法がある．'support' は 'meet' もしくは 'cover' の特殊なケースと考えることができる．'support' を用いると，前述の関係は次のように書き換えられる．

- support(table, cup).

すなわちこの表現は「テーブルがコップを支えており，かつテーブルはコップと接している」ことを意味する．なお，明らかに support(A,B) ≠ support(B,A) である．筆者らは，この 'support' 関係もイベント記述に導入することで，より直感的な記述を可能にすることができると考えている．

なお支持関係は，図形的な関係ではないことから，厳密には形状や接触の情報だけでは検出することができない．しかし「物体は必ず何か他の物体に支えられている」という，地上に存在する物体に普遍的に適用可能な物理的な性質から，ある程度の推定をすることは可能である．この推定方法については他文献[11] にて述べる予定である．

3. イベント記述

ここまでは，TDR を使って，静止状態（時間の経過がない）での物体の状態を表す方法について述べた．本節ではさらに，TDR の組み合わせを使って，動的な物体間の位置関係の変化を伴うイベントの記述方法について述べる．以下，表記の簡単化のために，明らかである場合は $R_1(X_1, X_2)$ や $R_2(X_3, X_4)$ などと記述する代わりに，単に R_1 や R_2 と表記する．

[定義 2] $R_1 \wedge \neg R_2$ が時刻 $t_1 (t_1 < t)$ に真であり，なおかつ $R_1 \wedge R_2$ が時刻 $t_2 (t < t_2)$ に真であれば，イベント E を $E = R_1 \rightarrow^t R_2$ と表記する．イベントを $R_1 \rightarrow_1^t R_2 \wedge \neg R_1$ のように表記した場合は，R_2 は時刻 t 以降にのみ真となることを意味する．

やや定義が複雑なのは，イベントを記述する上でこのように定義したほうが実用上便利なことが多いためである．なお，$\neg R_2$ という表現も便宜的なものであり，否定を表す表現ではなく，R_2 以外の 5 つの関係のいずれかという意味を表す．例をいくつか示す．

- ある人が部屋 A から出て部屋 B に入った: contains(roomA, person) $\rightarrow^t$ contains(roomB,person)
- ある人が本を机の上に置いた: support(hand,book) $\rightarrow^{t_1}$ support(table,book) $\rightarrow^{t_2}$ disjoint(hand,book)
- ある人が CD ケースを箱から取り出した: contains(box, cdcase) $\rightarrow^{t_1}$ support(hand, cdcase) $\rightarrow^{t_2}$ disjoint(box,cdcase)
- ある人が本棚の中の bookA に bookB を並べて置いた：support(hand,bookA) $\rightarrow^t$ contains(shelf,bookA) ∧ meet(bookA,bookB)

t, t_1 といった表現は，ある時間的な境を意味する記号であって，特定の時間をさすものではない．ある時刻を境に，関係が変化することを意味する．2 つめのケースのように時系列に沿って順番に関係が真となる場合は，このように記述する．

前述の通り，$R_1 \rightarrow^t R_2$ というイベントについて，明示されない限り R_1 については R_2 が真となった後も，そのまま継続して真であるという点である．例えばケース 1 のような場合，テーブルの上に本が置かれた瞬間 t_1 は support(hand,book) は真であり続ける．その後，t_2 の後の条件に disjoint （すなわち ¬support) が記述されているため，ここで初めて support(hand,book) は偽となる．多くのイベントでは，t を境として直前の関係 R_1 が即座に偽となることは稀であり，このように前状態が継

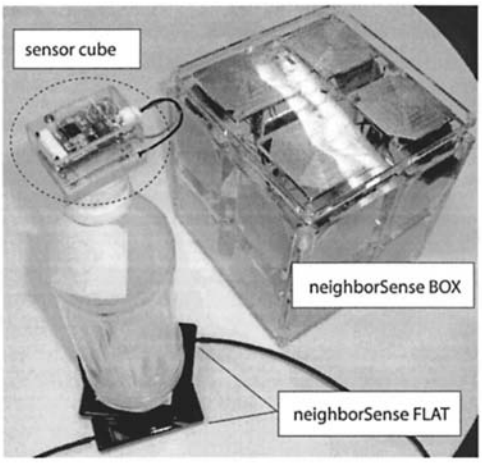


図 4　neighborSense の各センサ

続するものとして扱うほうが，イベントを記述しやすい．

4. neighborSense システム

前章まで述べたように，相対位置表現を構成する基本要素として，本稿では接触関係と支持関係をベースとして用いている．本章では，実際に物体間の接触関係を取得するシステムである neighborSense について述べる．neighborSense は，物理的な接触を検知する接触検知センサと，物体の動きや温度などを取得する Cube 型センサ，およびどの物体とどの物体が接触したかを判定する，物体管理サーバで構成される．以下では，それぞれの構成要素についてその概要を述べる．なお，接触検知センサの詳しい実装と性能の評価については，今後発表予定の文献 [11] にて詳細を記述する予定である．

4.1 接触検知センサ

接触検知センサは，センサ同士を一定距離以下に近づけると，近づいたことが検知できるタイプのセンサである．任意の物体の接近を検出できる，いわゆる近接センサとは異なり，あくまでセンサ同士が近づいたことが認識できるセンサである．具体的には，電磁誘導の原理を用いて磁場を発生させ，この磁場を検出することで他のセンサが接触したことを検知する．コイルに流す電流量を調整して発生する磁場の強さを変えることで，コイル間で低速度の遠隔通信を行うこともできる．

予め，各コイルごとに固有の ID を割り当てておき，一定の時間間隔で磁場を発生させて ID を送信する．他コイルでは，検知した磁場を電流に変換して ID を読み取ることで，どのコイルと接触したかを知ることができる．なお，ID の管理や接触の判断自体は，物体管理サーバで行い，センサでは磁場の検出と ID の送受信のみを行う．

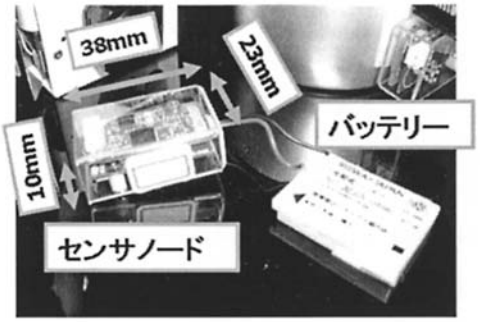


図 5　Cube 型センサ

筆者らは，接触検知センサとして BOX タイプと FLAT タイプの，2 つのタイプの実装を行っている．それぞれの実装したセンサの概観を図 4 に示す．図の右側が BOX タイプであり，左側の容器の底面に貼り付けられている黒いコースター状のものが FLAT タイプである．BOX タイプは，各面ごとに 4 つのコイルを持っており，それぞれ独立して接触を検知できる．さらに，6 面すべてにコイルを持っているため，BOX のどの面に他のセンサが接触したかを区別することもできる．BOX タイプは，その内側に後述する cube 型センサも内蔵することができ，BOX のどの面が下を向いているかといった，向きの情報や温度，加速度などの情報なども得ることができる．また，無線通信機能を持った組み込み PC ボード (gumstix) も内蔵しているため，外部 PC とデータを直接交換することもできる．

FLAT タイプは，BOX タイプのコイルのひとつを取り出して，個別の板状のケースに封入したものである．各センサにはコイルが 1 つだけ入っているため，他のセンサ 1 つとの接触を検知することのみ可能である．このタイプのセンサは，有線で物体管理サーバに直接接続して使う．

4.2 Cube 型センサ

Cube 型センサは，筆者らの研究グループでこれまで開発を行ってきた小型センサノードである [9]．このセンサノードは，Zigbee による無線通信機能をもち，1/30 秒毎にセンサからのデータを取得して，物体管理サーバにデータを送信することができる．搭載しているセンサは，三軸加速度センサ，二軸地磁気センサ，照度センサ，温度センサ，および赤外線人感センサである．携帯電話に用いる 3.3V のリチウムイオン電池を用いて，約 8 時間の連続使用が可能である．図 5 に，このセンサの外観を示す．

このセンサで得られるデータは，例えば物体が動く速度や物体表面の温度などであり，相対的位置表現で表されるイベントには直接は関係しない．しかし，イベント

に関する付帯的な情報を取得するための重要な役割を担う．例えば，人がある物体に触れたというイベントは，接触関係のみを取得すれば検知することができる．しかし，触れたあとに「動かした」か「そのまま保持して停止していた」かは，接触関係を取得しただけでは分からない．こうした動きを伴うイベントについては，加速度センサから得られたデータを用いることで，ある程度判別することができる．

現時点での neighborSense システムの実装では，Cube 型センサの情報を接触センサのデータを組み合わせて処理することは行っていないが，今後のプロジェクトの中でこのセンサのデータも加え，検出できるイベントの幅を広げる予定である．

4.3 物体管理サーバ

物体管理サーバは予め各物体の形状のデータを保持しており，接触検知センサおよび Cube 型センサからのデータを受け取り，イベントの発生状況を検知する機能を持つ．各センサから受け取ったデータをデータベース (PostgreSQL) にそれぞれ時刻のデータと共に格納しておき，ユーザがルールで指定したイベントの条件を，接触関係の変化の列（シーケンス）に変換し，そのシーケンスに該当するデータが存在する時区間を探す方法によって，該当するイベントが発生した時間（時区間）を得る．なお，イベントをシーケンスに変換する具体的な方法については，文献 [10] で述べている方法とほぼ同様であるため，本稿では詳細な説明は省略する．また現在の実装では，物体管理サーバは過去のセンサデータに対する検索機能しか有していないが，今後の改良によりリアルタイムなモニタリング機能も追加する予定である．

実装されているサーバは，4 台の計算機で構成されており，それぞれ 1 台で 8 個のセンサノードのデータを収集することができる，各サーバは Zigbee による通信機能をもち，Cube 型センサから直接データを受け取る．接触検知センサについては，FLAT タイプについては有線のシリアル通信にて直接データを取得する．BOX タイプについては，内蔵されている Linux 組み込み PC から無線 LAN と TCP/IP 通信を用いてデータを取得する．サーバプログラムは Windows XP で C++ を用いて実装し，データを格納するデータベースとしては PostgreSQL を用いている．

5. おわりに

本稿では，物体の位置に関する条件やイベントを，物体間の相対的な位置を表す単語群の組み合わせで表す，相対位置表現の提案を行った．また，この表現の基本となる「接触」関係を検知するシステムの概要について述べた．相対位置表現では，物体間の接触と物体の形状という，物理的に測定可能でかつ曖昧性の少ない情報を用いてイベントを記述できる点に利点がある．コンテクストアウェアシステムにおいて本手法を用いてルールを記述することで，曖昧性のない統一的な位置表現を用いて，サービスをユーザに提供できるようになると．

今後は，筆者らが開発を進めている実世界イベント検索エンジン (EventGo) [9]，リアルタイムイベントモニタ (EventCapture) [11]，およびモノ BLOG システムの三つのシステムで実際に neighborSense を用い，その性能評価を進める予定である．

文　献

[1] Harry Chen, Tim Finin, and Anupam Joshi. An Intelligent Broker for Context-Aware Systems. *Adjunct Proceedings of Ubicomp 2003*, pp. 183–184, October 2003.

[2] Max Egenhofer and Robert Franzosa. Point-set topological spatial relations. *International Journal of Geographical Information Systems 5(2)*, pp. 161–174, 1991.

[3] Jeffrey Hightower and Gaetano Borriella. Location systems for ubiquitous computing. *IEEE Computer*, Vol. 34, No. 8, pp. 57–66, 2001.

[4] Jeffrey Hightower, Sunny Consolvo, Anthony LaMarca, Ian E. Smith, and Jeff Hughes. Learning and recognizing the places we go. In *International Conference on Ubiquitous Computing (Ubicomp)*, pp. 159–176, 2005.

[5] Matthias Lampe, Martin Strassner, and Elgar Fleisch. A ubiquitous computing environment for aircraft maintenance. In *ACM symposium on Applied computing (SAC '04)*, pp. 1586–1592, New York, NY, USA, 2004. ACM Press.

[6] Jonathan Lester, Tanzeem Choudhury, and Gaetano Borriello. A practical approach to recognizing physical activities. In *International Conference on Pervasive Computing*, pp. 1–16, 2006.

[7] H. Si, Y. Kawahara, T. Igakura, T. Tonouchi, H. Morikawa, and T. Aoyama. A hybrid context-aware service platform based on stochastic and rule-description approaches. In *International Conference on Mobile and and Ubiquitous Systems: Networking and Services (Mobiquitous)*, 2006.

[8] Tsutomu Terada, Masahiko Tsukamoto, Keisuke Hayakawa, Tomoki Yoshihisa, Yasue Kishino, Atsushi Kashitani, and Shojiro Nishio. Ubiquitous chip: A rule-based i/o control device for ubiquitous computing. In *International Conference on Pervasive Computing*, pp. 238–253, 2004.

[9] 岡留剛, 前川卓也, 服部正嗣, 柳沢豊. センサネットワーク環境における実世界イベント検索システム. 情報処理学会論文誌, Vol. 48, No. 7, 2007, 採録決定.

[10] 前川卓也, 柳沢豊, 岡留剛. Tag and think: センサネットワークを前提としたモノ自身とその状態の推定. 情報処理学会 第 13 回ユビキタスコンピューティングシステム研究会 (UBI13), 講演番号 30, 2007.

[11] 柳沢豊, 前川卓也, 岡留剛. 物体の相対的な位置関係に基づいた実世界のイベント検知手法. マルチメディア，分散，協調とモバイル（DICOMO2007）シンポジウム, 2006, 発表予定.