DEWS2004 I-10-02

異種地図と異種センサの統合利用に伴う位置情報サービス開発ツールキットの実現方式

多賀　大泰†　　細川　宜秀††　　高橋　直久††

† 名古屋工業大学電気情報工学科　〒 466-8555 愛知県名古屋市昭和区御器所町
E-mail: †taga@moss.elcom.nitech.ac.jp, ††{hosokawa,naohisa}@elcom.nitech.ac.jp

あらまし　本稿は，異種地図と異種センサの統合利用を伴う位置情報サービスの開発環境の実現方式に関する研究についてまとめたものである．ここで，位置情報サービスとは，位置情報の利用を伴うサービスの総称である．本研究の異種地図と異種センサの統合利用を伴う位置情報サービスとは，シングルユーザによる地図とセンサの利用を対象とし、利用可能範囲の異なる位置検出センサ群を用いてユーザの位置を検出し，検出された位置が写像可能な地図上に分散配置された情報源群の利用を伴う位置情報サービスと定義する．そのサービスの特徴は，ユーザのあらゆる位置に対して情報提供を行える点にある．本方式のプロトタイプを実装し，その有効性を明らかにする．
キーワード　メディエータ, 空間 DB, モバイル DB, ユビキタスコンピューティング, 位置情報サービス（LBS）

An Implementation Method of a Toolkit for Developing Location-Based Services combining Heterogenous sensors and maps

Hiroyasu TAGA†, Yoshihide HOSOKAWA††, and Naohisa TAKAHASHI††

†
E-mail: †taga@moss.elcom.nitech.ac.jp, ††{hosokawa,naohisa}@elcom.nitech.ac.jp

Abstract　In this paper, we present an implementation method of a toolkit for developing LBS(Location-Based Services) combining heterogenous sensors and maps. The feature of this toolkit is to implement an automatic mechanism for selecting applicable maps and sensors according to user's activities and situation and so. By using our toolkit, it possible for LBS end-users to develop his LBS himself. We evaluated the feasibility of our toolkit by several experiment.
Key words　MEDIATOR, Space DB, mobile DB, ubiquitous computing, Location-Based Services(LBS)

1. はじめに

現在，GPS などの位置検出技術や RF-ID や IC タグなどの無線技術の発展・普及に伴って，カーナビゲーション・物品輸送・窃盗防止など一部の位置情報サービスが実用化され，社会における位置情報サービスの価値が高まっている状況にある．これらの位置情報サービスが，車の運転や商店の出入り口など特定の利用状況・利用目的における位置情報サービスであること，すなわち，それらの適用対象が利用者の社会活動のごく一部を対象としたサービスであることを考慮すると，利用者のあらゆる利用状況・利用目的に対して情報提供を行える位置情報サービスの実現は，社会における位置情報サービスの価値のさらなる向上に貢献するものと期待される．一方，利用目的と利用状況に依存した位置情報サービスの開発に関しては，異なる適用範囲を持つ地図とセンサを組み合わせることによって，位置情報サービス利用者の活動範囲を網羅することが必然である．具体的には，利用者の社会活動を網羅するために，市販されている広域地図に加え，利用者ごとに異なる活動範囲 (地下街・店内・オフィス) に対応した見取り図が必要となる．また，屋外を適用範囲とする GPS に加え，利用者ごとに異なる活動範囲に対応した位置検出技術の利用が必然となる．これより，利用状況・利用目的に応じた位置情報サービスの開発に次の 3 知識が要求される．

知識-1　複数の地図とセンサを統合利用するための操作に要する知識 (地図・センサの切り替え，利用可能範囲など)

知識-2　地図の直接操作に要する知識 (内部データ形式，縮尺，参照系など)

知識-3　センサの直接操作に要する知識 (内部データ形式，データ転送率，エラーなど)

さらに，使用する地図と使用するセンサに応じて，要求され

る 3 知識量は異なるので，利用目的と利用状況に応じた位置情報サービスの開発のための知識修得に要する時間が増大し，科発コストの増大につながる．開発コストの増大は，位置情報サービス利用の料金の増大につながり，社会における位置情報サービスの普及の妨げとなる．

本稿では，開発コストを削減して，利用目的・利用状況に応じた位置情報サービスを開発するためのツールキットの実現方式を提案する．その主要な特徴は，使用する地図とセンサを意識することなく，実行したい位置情報サービスの内容とその起動条件のみを記述することによって，位置情報サービスの開発を可能にする点にある．このツールキットの実現により，多くの利用目的・利用状況に応じた位置情報サービスの創出を可能にし，位置情報サービスの普及に貢献することが可能になる．

提案ツールキットを用いて現在主要な位置情報サービスであるナビゲーションシステムの開発を行い，提案ツールキットによる開発コストの削減効果を確認するための実験を行う．

2. 提案ツールキットの実現方式

提案方式の主要な特徴は，利用する位置センサと地図を直接操作するためのプログラムを記述することなく，複数の異種位置センサと異種地図を統合利用する位置情報サービスのプログラムを記述，実行可能にする点にある．

具体的には，提案方式は，位置情報サービスを次の 2 要素の組によって規定することによって，その特徴を実現する．

(要素-1) 利用者の位置に応じて利用する地図とセンサを決定するためのアルゴリズム

地図と位置センサは，位置情報サービスの適用範囲を決定する最も重要な要素なので，これらの決定アルゴリズムは，実現する位置情報サービスの適用範囲を決定することに対応する．

例えば，常に選択する地図を「東京都」，センサを「GPS」とした場合，東京都内で GPS が利用可能な屋外が，この位置情報サービスの適用範囲となる．ここで，ある屋内に入ったときに，その屋内における位置検出が可能なセンサを選択するようなアルゴリズムを組みこめば，その位置情報サービスの適用範囲をその屋内に拡大することが可能になる．このように，この要素は，位置情報サービスの適用範囲を規定する要素として位置付けられる．

(要素-2) 位置情報サービスの基本イベントに対するアクションの起動条件とアクション

位置情報サービスにおいて，利用者の位置に応じたプログラムの実行制御は本質的なものである．そのため，アクティブ・データベース [1] のような，ある条件を満たしたときに対応するプログラムを起動する枠組みは，位置情報サービスの実現に有効である．そこで，提案方式では，この枠組みに基づいて位置情報サービスの基本動作を規定する．提案方式とアクティブ・データベースとの違いは，提案方式が位置情報サービスにおいて本質的なイベント群を抽出している点にある．

本節では，まず，位置情報サービスの基本イベント群を抽出し，上記 (要素-2) の記述方式を規定する．次に，本ツールキットの提案方式を述べた上で，利用するセンサと地図の自動選択方式において，上記（要素-1）のセンサと地図を決定するためのアルゴリズムを述べる．

2.1 位置情報サービスにおける基本イベントの抽出と位置情報サービスの実行制御方式

ツールキットでは，位置センサと地図に関するイベントとして，次の 5 イベントを定義する．位置情報サービスのプログラムは 5 イベントに応じて起動するアクションの組み合わせとして記述する．

イベント 1　要求判定イベント：**Goal**

これは，位置情報サービスのプログラムの要求が達成されているかどうかの判定を行った時に発生するイベントを表す．

イベント 2　位置センサからの位置情報取得イベント：**getLocation**

これは，現在利用している位置センサから位置情報が取得された時に発生するイベントを表す．

イベント 3　チェックポイント実行イベント:**Checkpoint**

これは，位置情報サービスのプログラムにおいて，あらかじめプログラムが決めておいたチェックポイントの領域に入ったときに発生するイベントを表す．

イベント 4　位置センサ自動切り替えイベント：**shiftSensor**

これは，現在利用している位置センサが適用範囲外になる，もしくは動作可能に変わる場合において，ツールキットにより利用する位置センサが自動的に切り替えられた時に発生するイベントを表す．

イベント 5　地図自動切り替えイベント：**shiftMap**

これは，現在利用している地図が適用範囲外になる場合において，ツールキットにより利用する地図が自動的に切り替えられた時に発生するイベントを表す．

またツールキットが実行対象とする位置情報サービスのプログラムは，次のアクションによって記述する．

アクション 1 I_action

これは，変数の初期値代入やクラス宣言などの位置情報サービスのプログラム実行に必要となる，初期化アクションである．

アクション 2 G_action(f_{gc})

これは，位置情報サービスのプログラムの要求判定イベント Goal が発生した場合において，要求達成条件が真である時に起動されるアクションである．ここで，f_{gc} とは getLocation イベントが発生したときの，Goal の条件を設定している．

例えば，目的地 (destination) の半径 15 メートルの円領域 (circle) 内に入ったら，目的地に着いたことにするのならば，

f_{gc}**=Inside(UserCurrentCondition, circle(destination,15 meter))**

となる．'Inside' とは第一引数のユーザの現在地 UserCurrentCondition が，第二引数の circle(destination,15 meter) の領域内に入ったときに'true' を返すものである．

アクション 3 C_i_action(f^i_{cc})

これは，位置情報サービスのプログラムにおいて，チェックポイント実行イベント CheckPoint が発生したときに起動するアクションである．ここで，f^i_{cc} とは getLocation イベントが発生したときの，Checkpoint の条件を設定している．

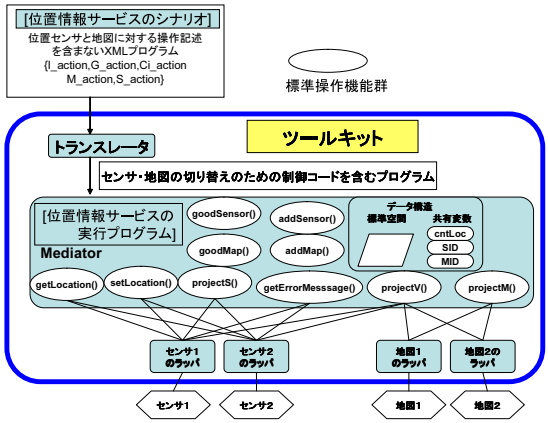

図 1　提案ツールキットの構成図

f_{cc}^{i} の書き方は G_action(f_{gc}) における，f_{gc} と同じだが，チェックポイントの場合は目的地とは異なり，一点とは限らない．よって，個々のチェックポイントがそれぞれ設定できるように，i 番目のチェックポイントを設定するという意味で f_{cc}^{i} としてある．

アクション４　S_action

これは，位置情報サービスのプログラムにおいて，位置センサ自動切り替えイベント shiftSensor が発生した時に起動するアクションである．

アクション５　M_action

これは，位置情報サービスのプログラムにおいて，地図自動切り替えイベント shiftMap が発生した時に起動するアクションである．

ツールキットでは位置情報サービスのプログラム LBSp(Location-Based Services Program) を５イベントに対応する，５アクションの組み合わせとして構成する．

LBSp={I_action(),S_action(),M_action(),
G_action(f_{gc}), C_1_action(f_{cc}^{1}),. . . ,C_n_action(f_{cc}^{n})}

ここで，I_action,S_action,M_action は，位置情報サービスのプログラミングの際に省略することが可能である．

またツールキットを用いた LBSp の実行手順を次に示す．

STEP1　現在利用している位置センサから，位置情報を取得する．このとき getLocation のイベントが発生する．

STEP2　位置情報サービスのプログラムの要求判定イベントに対し，要求達成条件が真であれば G_action を実行して，位置情報サービスのプログラムを終了させる．要求達成条件が偽であれば，STEP3 に進む

STEP3　Checkpoint,shiftMap,shiftSensor のそれぞれのイベントが発生したときに，そのイベントに対応するアクションを実行する．その shiftMap，shiftSensor を行う際に，goodMap，goodSensor により位置情報サービスに応じた最適な地図，最適なセンサが選ばれる．その後，STEP1 に戻り，要求達成条件が真になるまで手順を繰り返す．

```
String MID;
String SID;
I_action();
while(true){
    Coordinate cntLoc=getLocation(SID);
    If(f gc(projectM( MID , UserCondition ),projectS(SID,cntLoc))==true){
        G_action();
        break;
    }
    If(f cc(projectM( MID , UserCondition ),projectS(SID,cntLoc))==true){
        C_action();
        break;
    }
    String NewMID=goodMap(projectS(cntLoc));
    If(NewMID!=MID){ //M_actionの起動条件
        MID=NewMID;
        M_action();
    }
    String NewSID=goodSensor(SID,MID,projectS(cntLocs);
    If(NewSID!=SID){ //S_actionの起動条件
        SID=NewSID;
        S_action();
        setLocation(SID,cntLoc); //選択されたセンサの初期化
    }
}
```

図 2　アクションを運用する実行可能プログラムのテンプレート

2.2　提案方式の構成

位置情報サービスのプログラムを位置センサと地図に依存せずに設計，開発するために，ツールキットをメディエータ・ラッパ・アーキテクチュア（図 1）に基づき，次の４つのサブシステムによって構成する．

（サブシステム１）トランスレータ

トランスレータは位置センサと地図に依存せずに記述されたアクションを，図 2 に示すテンプレートに埋め込み，実行可能プログラムを生成する．

ツールキットにおけるトランスレータの実行手順は次のとおりである．

STEP 1　位置情報サービスのプログラムの記述内容から，適用するテンプレートを選択する．

ここでテンプレートとは，位置情報サービスのプログラムの記述内容より実行可能プログラムを生成するためのベースとなるプログラムコードで，位置情報サービスのプログラムにおける位置センサおよび地図の一連の操作を形式化したものである．図 2 は，テンプレートの例を示す．なお，アクションの中身は，開発者による任意の記述が可能である．そのため，異なる位置情報サービスにより，それぞれ自由なコーディングが可能である．

STEP 2　位置情報サービスのプログラムのアクションを，テンプレートの規定した位置に挿入する

（サブシステム２）位置情報サービスのプログラム実行部（mediater）

位置情報サービスのプログラム LBSp 実行部は，トランスレータによって生成された LBSp 実行可能プログラムを実行するものである．ここで LBSp 実行部は，位置センサと地図の標準操作機能群を実現し，その機能を介して位置情報サービスのプログラムが利用する位置センサと地図に関する操作の実行制御を行う．さらに，LBSp 実行部では標準空間を介して，地図と地図間，地図と位置センサ間，および位置センサと位置センサ間のデータの受け渡しを行う．

（サブシステム３）位置センサ・ラッパ

位置センサ・ラッパは，位置センサに対する標準操作機能を

位置センサ毎の固有操作にラップする．さらに位置センサと標準空間の間の座標の変換を行なうための機構を実現する．表 1 は，センサ (SID) から送られてきた現在地の値 (value) を格納するデータベースである．現在地の他には，現在地取得時間 (date)，エラー情報 (error) が格納される．

表 1　センサ値格納データベース (Sensor-ValueTable) の構造

SID	value	date	error
GPS	35.124,N,134.245,E	20041231235645	normal
cart	x:00125,y:00323	20040123164523	Initialization error

（サブシステム 4）地図ラッパ

地図ラッパは，地図と標準空間の間の座標の変換を行なうための機構を実現する．

ツールキットの全体構成は図 1 のようになる．

表 2，表 3 は，位置情報サービスを規定するための基本操作とデータ構造を表す．ツールキットを利用する開発者は，表 3 に示した SID，MID，cntLoc を用いることができる．そのため，センサや地図，現在地に応じたアクションが記述可能である．

表 3　ツールキットと位置情報サービスのプログラム間のデータ構造

データ名	説明
SID	位置情報サービスのプログラムが利用している位置センサの識別子
MID	位置情報サービスのプログラムが利用している地図の識別子
cntLoc	現在の位置センサの座標 (cntLoc[0〜2]={x,y,z})

2.3　getLocation による現在地の取得方式

getLocation は，位置情報サービスにおいて現在地をセンサ値格納データベース (表 1) から取得する基本機能である．図 3 に getLocation の SQL 文を示す．本提案システムでは，getLocation(SID) により，センサ ID(SID) が正常状態 (normal) であるときに現在地 (value) をデータベースから取り出す．ただし，選択されたセンサ ID の状態が正常でないときは，正常状態である別のセンサの中で，位置取得日時 (date) が最新であるセンサの現在地 (value) が取り出される．

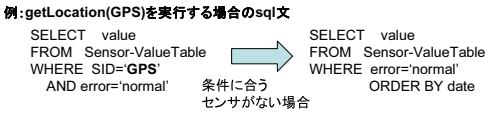

図 3　getLocation のデータベース検索の SQL 文

2.4　利用するセンサと地図の自動選択方式

goodMap と goodSensor は，位置情報サービスにおいて利用する地図と位置センサを自動選択するための提案ツールキットの基本機能である．さらに，goodMap と goodSensor における，地図とセンサの自動選択アルゴリズムを位置情報サービスに応じた変更を施すための方式について述べる．この変更方式は，本節冒頭で述べた位置情報サービスの (要素-1) を規定することに対応する．

表 4　地図領域データベース (Map-AreaTable) の構造

MID	area	InOut	jugdement
City	{ (x,y,z),(x,y,z) }	outdoor	8(detail)
Room	{ (x,y,z),(x,y,z) }	indoor	10(detail)

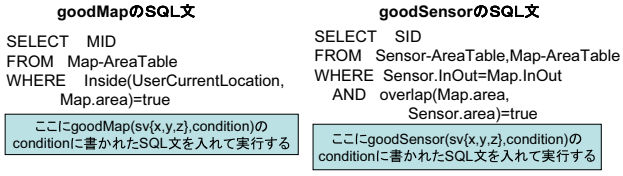

図 4　goodMap,goodSenosr のデータベース検索の SQL 文

2.4.1　goodMap の実現方式と地図の自動選択アルゴリズムの変更方式

ツールキットは，地図の自動切り替えを行うために，地図について次の 4 項目のデータを保持する．

（1） 地図の識別子 (MID)

（2） 利用可能範囲 (area)

（3） 建物内か建物外かを表すデータ (InOut)

（4） 判断基準 (judgement)：例-詳細度，地図内のランドマーク，更新日時

この 4 項目のデータは，表 4 の地図領域データベースに保持される．これらのデータは addMap により組み込むことが可能である．

goodMap の実行手順

STEP1　位置センサから得られた座標 sv{x,y,z} が，指定された地図（MID はその識別子）の利用可能範囲に存在するかどうかを判定する．その判定結果として，利用可能範囲内に sv{x,y,z} が存在している場合は，入力された MID を返す．すなわち地図の切り替えは行わない．また判定結果として，利用可能範囲内に sv{x,y,z}} が存在していない場合は，sv{x,y,z} を利用可能範囲内に含んでいる地図群を選択する

STEP2　Step1 で選択された地図群から，判断基準により最適な地図を選択し，その MID を返す．

具体的には，図 4 の SQL 文により，地図選択を行う．

UseCurrentLocation はユーザの現在地を表す．地図の利用範囲を示す area に UserCurrentLocation が含まれているとき，位置情報サービスの実行を継続するのに最もよい地図が選択される．この選択は，goodMap(svx,y,z,condition) の condition で与えられる条件に従って実行される．表 5 に condition の例を示す．このように，位置情報サービスのに応じた地図の選択制御を行うことが可能になる．

表 5　goodMap における condition の記述例

判断基準例	option の内容	condition の記述
詳細度の高い地図	詳細度	order by option
趣味に合う地図	ランドマークの種類	and option='music'
最近の地図	更新日時	and option>20040101

2.4.2　goodSensor の実現方式とセンサの自動選択アルゴリズムの変更方式

ツールキットは位置センサの自動切り替えを行うために，位

表 2　標準操作機能の一覧

標準操作機能	説明
getLocation(SID) → lsv{x,y,z}	現在利用している位置センサ (SID) から現在の座標 lsv{x,y,z} を取得する
getErrorMessage(SID) → error	現在利用している位置センサ (SID) のエラーメッセージ (error) を取得する
setLocation(SID)	切り替える位置センサ (SID) の初期値を設定する
projectS(SID,lsv{x,y,z}) → sv{x,y,z}	現在利用している位置センサ (SID) から取得された座標 lsv{x,y,z} を，標準空間の座標 sv{x,y,z} に変換する
projectM(MID,lmv{x,y,z}) → sv{x,y,z}	地図上の座標 lmv{x,y,z} を，標準空間の座標 sv{x,y,z} に変換する
projectV(lv{x,y,z}) → sv{x,y,z}	世界測地系の座標 lv{x,y,z} を，標準空間の座標 sv{x,y,z} に変換する
goodMap(sv{x,y,z},condition) → NewMID	与えられた標準空間上の座標 sv{x,y,z} を利用可能範囲に含む地図 (NewMID) を選択する
goodSensor(MID,SID,sv{x,y,z}, condition) → NewSID	地図 (MID)，位置センサ (SID)，標準空間上の現在の座標 sv{x,y,z} を利用可能範囲に含む位置センサ (NewSID) を選択する
addMap(MID,area,InOut,detail)	地図 (MID) を, 利用可能範囲 (area), 建物内外 (InOut), 詳細度 (detail) から, 地図領域 DB に組み込む
addSensor(SID,area,InOut,priority)	センサ (SID) を, 利用可能範囲 (area), 建物内外 (InOut), 優先度 (priority) から, センサ領域 DB に組み込む

表 6　センサ領域データベース (Sensor-AreaTable) の構造

SID	area	InOut	judgement
GPS	{ (x,y,z),(x,y,z) }	outdoor	5(priority)
tag	{ (x,y,z),(x,y,z) }	indoor	4(priority)

置センサについて次の 4 項目のデータを保持する．

（1） 位置センサの識別子 (SID)

（2） 利用可能範囲 (area)

（3） 建物内か建物外かを表すデータ (InOut)

（4） 判断基準 (judgement)：例-優先度，センサの種類，製造年月日

この 4 項目のデータは，表 6 のセンサ領域データベースに保持される．これらのデータは addSensor により組み込むことが可能である．なお，PC からのセンサデバイス操作のためのプログラムは，センサラッパ作成者が用意するものとする．

goodSensor の実行手順

STEP1　現在利用している位置センサ（SID はその識別子）の判定結果として，位置センサの動作が安定している，もしくは適用可能範囲内に座標 sv{x,y,z} が存在している場合は，入力された SID を返す．ここで，位置センサの切り替えは行わない．また利用している位置センサの判定結果として，位置センサの動作が不安定である，もしくは適用可能範囲内に sv{x,y,z} が含まれていない場合は，座標 sv{x,y,z} を適用可能範囲内に含む位置センサ群を選択する．

STEP2　Step1 の結果から，現在利用している地図の利用可能範囲に現在の座標があり，現在利用されている地図と重複している部分がある適用可能範囲を持つ位置センサ群を選択する．

STEP3　Step2 で選択されたセンサ群から，判断基準により最適な位置センサを選択し，その SID を返す．

具体的には，図 4 の SQL 文により，位置センサの選択を行う．

UseCurrentLocation はユーザの現在地を表す．センサの利用可能範囲を示す area と地図の利用可能範囲を示す area が重なっていて，かつセンサと地図の建物内外を示す InOut が一致するとき，位置情報サービスの実行を継続するのに最もよいセンサが選択される．この選択は，goodSensor(MID,SID,sv{x,y,z},condition) の condition で与えられる条件に従って実行される．表 7 に condition の例を示す．このように，位置情報サービスに応じたセンサの選択制御を行うことが可能になる．

表 7　goodSensor における condition の記述例

判断基準例	option の内容	condition の記述
優先度の高いセンサ	優先度	order by option
特定の種類のセンサ	センサの種類	and option='mobile'
新しいセンサ	製造年月日	and option>20040101

2.5　提案ツールキットへの地図と位置センサの登録方式

addMap と addSensor は，提案ツールキットに地図と位置センサを登録するための基本機能である．この機能によって，提案ツールキットを用いて実現される位置情報サービスから利用可能な地図と位置センサを追加することが可能になる．

2.5.1　addMap の実現方式

ツールキットにおける goodMap を用いるためには，地図領域データベースを構築する必要がある．このとき，位置情報サービスのプログラム作成者の持っている知識に対応するために，次の 2 つの addMap を実現する．

（1） **地図領域の絶対座標を知っている場合**

次の SQL 文を実行する．

INSERT INTO Map_Table
VALUES (MID,area,InOut,judgement)

MID は組み込む地図の識別子を示す．また，area は組み込む地図の利用可能範囲を示し，組み込む地図の北西端点と南東端点の世界測地系の座標 (絶対座標) をそれぞれ，lv_1, lv_2 として，

area={projectV($(lv_1\{x,y,z\}$**),projectV(**$lv_2\{x,y,z\}$**))}**

と記述する. ただし，{x,y,z} はそれぞれ，東経 (x) 北緯 (y) 高度 (z) を表し (以下同様)，projectV(lv{x,y,z}) により，世界測地系 lv{x,y,z} を標準座標 sv{x,y,z} に変換して，地図領域データベースの area 値となる．そして，InOut は地図利用範囲が建物の内外かを，indoor か outdoor で示し，judgement はセンサを選ぶときの判断基準を示す.

（2） **地図領域の相対座標を知っている場合**

この場合は，addMap により地図を組み込むときの SQL 文は，絶対座標のときと同じであるが，area の記述が異なる．

area は組み込む地図の利用可能範囲を示すが，組み込みたい地図が，より広域な地図 (wideMID) 上での対応する北西端点

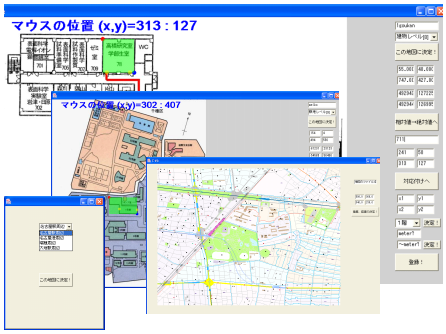

図 5　領域入力インターフェイスの実行画面

と南東端点の座標 (相対座標) をそれぞれ，lmv_1, lmv_2 として，

area={projectM(wideMID,$lmv_1\{x,y,z\}$**,**
projectM(wideMID,$lmv_2\{x,y,z\}$**)}**

と記述する．ここでは，projectM(wideMID,lmv{x,y,z}) により，地図 (wideMID) 上の座標 lmv{x,y,z} を標準座標 sv{x,y,z} に変換して，地図領域データベースの area 値となる．

このとき，相対座標の知識をもとに，直観的に地図領域データベースが構築できるように，マウス操作による領域入力インターフェイスを図 5 のように用意した．

2.5.2　addSensor の実現方式

ツールキットにおける goodSensor を用いるためには，センサ領域データベースを構築する必要がある．このとき，位置情報サービスのプログラム作成者の持っている知識に対応するために，次の 2 つの addSensor を実現する．

（1）　**センサ領域の絶対座標を知っている場合**

次の SQL 文を実行する．

INSERT INTO Sensor_Table
VALUES (SID,area,InOut,judgement)

SID は組み込むセンサの識別子を示す．また，area は地図の area と同様に，組み込むセンサの利用可能範囲を示し，その利用可能範囲の北西端点と南東端点の世界測地系の座標をそれぞれ，lv_1, lv_2 として，

area={projectV($(lv_1\{x,y,z\}$**),projectV(**$lv_2\{x,y,z\}$**))}**

と記述する．ここでは，projectV(lv{x,y,z}) により，世界測地系の座標 lv{x,y,z} を標準座標 sv{x,y,z} に変換して，センサ領域データベースの area 値となる．そして，InOut はセンサ利用範囲が建物の内外かを，indoor か outdoor で示し，judgement は地図を選ぶときの判断基準を示す．

（2）　**センサ領域の相対座標を知っている場合**

この場合は，addSensor によりセンサを組み込むときの SQL 文は，絶対座標のときと同じであるが，area の記述が異なる．

area は組み込むセンサの利用可能範囲を示すが，その利用可能範囲を，地図 (MID) 上での対応する領域範囲の北西端点と南東端点の座標 (相対座標) をそれぞれ，lmv_1, lmv_2 として，

area={projectM(MID,$lmv_1\{x,y,z\}$**,**
projectM(MID,$lmv_2\{x,y,z\}$**)}**

と記述する．ここでは，projectM(MID,lmv{x,y,z}) により，地図 (MID) 上の座標 lmv{x,y,z} を標準座標 sv{x,y,z} に変換して，センサ領域データベースの area 値となる．

このとき，領域入力インターフェイスにより，地図領域の入力と同様に，相対座標の知識をもとに，直観的にセンサ領域データベースが構築できる．

3.　実　　験

提案ツールキットを用いて現在主要な位置情報サービスであるナビゲーションシステムの開発を行い，提案ツールキットによる開発コストの削減効果を確認するための実験を行う．

この有効性の確認とは，位置情報サービスで使用されるセンサや地図の数，組み合わせが異なる場合でも，位置情報サービスのプログラムを作成する際に，センサや地図の知識がどれだけ必要なくなるのかを調べることである．

また，位置情報サービスのプログラム作成後も，地図，センサを新しく，依存性を考えないで組み込めることを確認する．

本論文では，位置情報サービスのプログラムの書き方として次に挙げる三段階において，それぞれ個々の知識別にプログラムコード量を比較することで，センサや地図を利用するために必要な知識・技能が削減できることを示す．

（1）　**実験プログラム-1**:ツールキットを用いない場合

実験プログラム-1 は，ツールキットで提案したものを用いない場合のプログラムコード量を数える．

（2）　**実験プログラム-2**:ツールキットの機能を用いた場合

実験プログラム-2 は，図 1 の [位置情報サービスの実行プログラム] にあたり，ツールキットの提案した機能を，位置情報サービスのプログラムに組み込む場合のプログラムコード量を数える．この場合は，新しく位置情報サービスを定義するなど，プログラマの好みに合った書き換えが，地図やセンサを意識しないで書く場合に有効である．

例えば，goodMap(sv{x,y,z},condition) の condition に，任意の位置情報サービスに適した SQL 文を書くことで，異なる位置情報サービスにも適応可能である．

（3）　**実験プログラム-3**:ツールキットのテンプレートを用いた場合

実験プログラム-3 は，図 1 の [位置情報サービスのシナリオ] にあたり，ツールキットで提案されたテンプレートを用いて，位置情報サービスのプログラミングを行った場合のプログラムコード量を数える．プログラミング段階-2 では細かな制御が可能だが，プログラム構造の組み立て方など，わずらわしくなる部分がある．一方，プログラミング段階-3 では，用意されたテンプレートに組み込むだけなので，そういったわずらわしさが軽減できる．

なお，この実験で用いたテンプレートに組み込むナビゲーションの記述例を図 6 に示す．ツールキットでは XML を用いて，テンプレートに組み込んでいる．この記述例では，センサ ID(SID) がタグセンサ (tag) であるとき，タグセンサをコンピュータにつなぐように指示を行う．また，センサ ID だけでなく，地図 ID(MID) や現在地 (cntLoc) を意識した記述も可能である．

3.1　実 験 方 法

地図，センサの数や組み合わせを変えたとき，またほかのメ

```
<?xml version="1.0" encoding="UTF-8"?>
<SN>
<l_action>
    MID="meiko";
    SID="GPS";
</l_action>
<S_action>
    if(SID=="tag"){
        System.out.println("please connect 'tag' to your computer");
    }
</S_action>
<G_action condition="inside(UserCurrentCondition,circle(300,100),15meter)">
    System.out.println("Congratulation!");
</G_action>
<M_action>
    Map.display(MID);
</M_action>
</SN>
```

図 6　テンプレートに組み込むナビゲーションの記述例

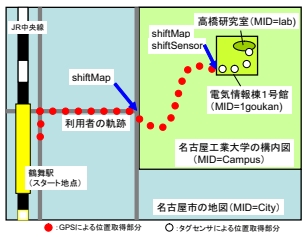

図 7　実験におけるナビゲーションの軌跡

ソッド（getLocation や shiftMap,shiftSensor）を用いたときにこのツールキットを使えばどれだけ手間が省けるのかを，前にあげた３つの実験プログラムで比較する．実験では，図 7 にあるように，JR 鶴舞駅から名古屋工業大学敷地内を通り，電気情報工学科棟の７階にある高橋研究室まで，ナビゲーションを行うものとする．

3.2　実 験 環 境

（1）　**用いた地図**

- **City**：名古屋市の地図
- **Campus**：名古屋工業大学の構内図
- **1goukan-1floor**：電気情報棟１号館１階の見取り図
- **1goukan-7floor**：電気情報棟１号館７階の見取り図
- **lab**：電気情報棟１号館７階にある高橋研究室の見取り図

実験では，組み込む地図の枚数を増やした．

（2）　**用いたセンサ**

- **手押し車**：研究用特注品（位置を取得する台車）
- **GPS センサ**：吉野電気株式会社製　GT-77
- **GPS センサ**：Garemin 製 eTrex Vista
- **気圧計**：同上（GPS の測定器に組み込まれている）
- **タグセンサ**：オムロン製 誘導式 RFID システム V720s

ここで GPS を二つ用いたのは，片方の GPS が壊れたとき，または一方の GPS の位置情報をもう一方の GPS により補正したいときに有効なためである．実験では，鶴舞駅から電気情報棟 1 号館の建物付近まではより誤差の少ない GPS を，建物内では手押し車とタグセンサを利用して位置を取得するものとする．

3.3　結果と考察

3.3.1　ツールキットにおける各標準操作機能等の動作確認

実際に，ツールキットを使った位置情報サービスのプログラムを実行させたところ，有効に動作することが確かめられた．

また，位置情報サービスのプログラム作成後からの addMap による，より詳細度の高い地図を組み込んだところ，goodMap によって，その地図が選ばれた．また，addSensor も同様に，より優先度の高いセンサを組み込んだところ，正常に goodSensor によって，適切にそのセンサが選ばれた．よって，新規の地図，新規のセンサを組み込む場合でも，新たに位置情報サービスのプログラムを書き直す必要がなく，依存性がないことが示せた．

表 9　位置情報サービスのプログラムに必要な知識の分類

分類	プログラムの分類	表 8 との対応	標準操作機能
メイン	メインプログラム	8，9	（なし）
地図	地図の状況認識プログラム	1，3	goodMap
	地図変換プログラム	6	projectM
	地図組み込みプログラム	10	addMap
センサ	センサの状況認識プログラム	2，4	goodSensor
	センサ変換プログラム	7	projectS
	センサ組み込みプログラム	11	addSensor
	センサ操作プログラム	5，12，13	getLocation

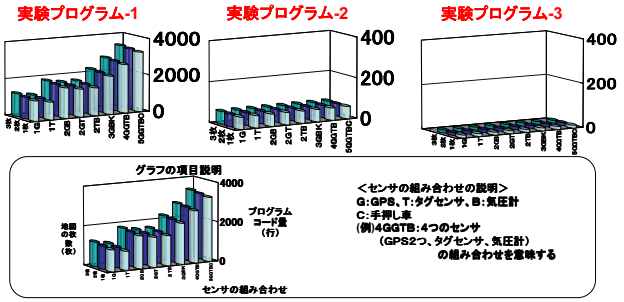

図 8　提案ツールキットによるプログラムの削減効果

表 10　地図が 1 枚 (City) のときのプログラムコード量 (行) の削減効果

実験 プログラム	1 G	1 T	2 GB	2 GT	2 TB	3G TB	4GG TB	5GG TBC
プログラム-1	1107	965	1673	1531	1531	2097	2663	3278
プログラム-2	61	61	61	61	61	61	61	61
プログラム-3	13	13	13	13	13	13	13	13

表 11　地図が 2 枚 (City,Campus) のときのプログラムコード量 (行) の削減効果

実験 プログラム	1 G	1 T	2 GB	2 GT	2 TB	3G TB	4GG TB	5GG TBC
プログラム-1	1169	1027	1735	1593	1593	2159	2725	3340
プログラム-2	61	61	61	61	61	61	61	61
プログラム-3	13	13	13	13	13	13	13	13

なお，地図が３枚以上のときも同様のため，本稿では割愛する．

3.3.2　プログラムコードの総量から見る結果と考察

図 8，表 10，表 11 より，総プログラム量の変化をみると，ツールキットを用いない実験プログラム-1 と比べ，ツールキットの機能を用いた場合の実験プログラム-2 では，96.4 %までプログラムコード量が削減できた．また，ツールキットのテンプレートを用いた場合の実験プログラムは，実験プログラム-1 に比べて，99.2 %まで削減できた．

この原因は，表 5 により確かめられる．すなわち，位置情報サービスを開発する上で必要なプログラムは，実験プログラム-1 では 13 種類であった．しかし，実験プログラム-2 では 4

表 8　各プログラムに要する提案ツールキットによる削減効果

個々の知識への分類	実験プログラム-1	実験プログラム-2	実験プログラム-3
1:地図の切替タイミングプログラム	O(m)	O(1)	0
2:センサの切替のタイミングプログラム	O(n)	O(1)	0
3:地図領域データベースへのアクセスプログラム	O(1)	0	0
4:センサ領域データベースへのアクセスプログラム	O(1)	0	0
5:センサ値格納データベースへのアクセスプログラム	O(1)	0	0
6:地図のデータ変換プログラム	O(m)	0	0
7:異種センサからのデータ変換プログラム	O(n)	0	0
8:切替地図の詳細情報	O(m)	O(1)〜O(m)	O(1)〜O(m)
9:切替センサの詳細情報	O(n)	0〜O(n)	0〜O(n)
10:地図領域データベースの構築（SQL）	O(m)	0	0
11:センサ領域データベースの構築（SQL）	O(n)	0	0
12:センサ値格納データベースの構築（SQL）	O(n)	0	0
13:PC からのデバイス操作のためのプログラム	O(n)	0	0

m:地図の枚数，n:センサの数を表す．また，切り替え地図の詳細情報が O(1)〜O(m) であるのは，
ゴール地点やチェックポイントを指定する場合に，地図 ID(MID) が必要になるためである．

表 12　実験プログラム-1 に対するプログラムコード量の削減率（%）

比較対象 (比較A):(比較B)	1G	1T	2GB	2GT	2TB	3GTB	4GGTB	5GGTBC	削減率の平均
実験プログラム-1:実験プログラム-2	95.0	94.4	96.6	96.3	96.3	97.3	97.8	98.2	96.4
実験プログラム-1:実験プログラム-3	98.9	98.8	99.3	99.2	99.2	99.4	99.5	99.6	99.2

削減率の導出法：削減率 (%) $= (1 - \frac{(\text{比較 } B \text{ のプログラムコード量})}{(\text{比較 } A \text{ のプログラムコード量})}) \times 100$　　（ただし地図 3 枚のとき）

種類，実験プログラム-3 では 2 種類と大きく削減されていることがわかる．よって，全く必要のなくなるプログラムがあることから，知識の削減効果が確かめられた．また，各実験プログラムにおいて，必要なプログラムのオーダから，どのプログラムに対しても削減されていることがわかる．以上により，全く必要のなくなる知識があるだけでなく，必要な知識に対してもオーダが小さくなるため，知識の削減効果があることが確認できた．

以上の要因から，位置情報サービスのプログラムを作成する際に，有効な位置情報のサービス開発ツールキットであることが確かめられた．

4. 関 連 研 究

Active Mobile DBMS[3] は，移動体端末が交信可能状態になったときに，アクションを起動するための機構を実現してくれている．その特徴は，交信可能イベント connect を定義し，そのイベントに対するアクションを実行するための機構を実現する点にある．一方，本ツールシステムは，異種地図・異種位置センサの統合利用を行うメディエタにおいて，実行中に発生する５つの本質的なイベントを定義し，その５つのイベントに対する５つのアクションとして記述・実行可能にするものである．

アクティブデータベースシステム [1] は，能動的に動作可能なデータベースシステムである．EAC ルールは，アクティブデータベースにおいて，能動的な動作記述を行うものである．このシステムは本ツールキットとは異なり，データベースシステム中に発生するイベント（更新など）に対してアクションを起動するものである．

文献 [2] は，異種センサ統合利用システムを実現している．複数のセンサ・ネットワーク群から独立に送られてくる位置情報を品質を落とさずに収集するための手法を実現している．本ツールしてステムは，現在の位置情報が，現在使用しているセンサの利用範囲外になったとき，ほかの利用可能なセンサに自動的に切り替えるための機構を実現することによって，ツールシステムによる位置情報サービスのプログラム実行中に継続して位置情報を獲得することを可能にする．

5. お わ り に

本稿では，異種地図・異種センサの統合利用を伴う位置情報サービスを開発するためのツールキットの実現方式を提案した．

実験により，異種地図・異種センサの統合利用を伴う位置情報サービスとして，屋内外を対象とした歩行者ナビゲーションサービスをとりあげ，提案方式のプロトタイプを実装し，そのナビゲーションサービス用の地図とセンサの優先度制御プロセスを本プロトタイプシステム上で開発することによって，提案方式の有効性を確かめた．

これによって，利用する地図とセンサを操作するための知識を必要とすることなく，また，利用する地図とセンサを操作するためプログラム記述を行うことなく，異種地図・異種センサの統合利用を伴う位置情報サービスの開発を行うことが可能になる．さらに，新しい地図とセンサを組みこめば，位置情報サービスのプログラムを変更することなく，その適用可能範囲を拡大することが可能になることが示された．

文　　献

[1] 石川博：アクティブデータベース，情報処理 Vol.35,No.2,pp.120-129(1994)

[2] 白石陽，安西祐一郎：位置情報に基づくセンサデータ統合のための逐次データ提供方式，情報処理学会データベースと Web 情報システムに関するシンポジウム予稿集,pp.153-160,(2002).

[3] Murase,T.Tsukamoto,MandNishino,S.:Active Mobile Database Systemfor Mobile ComputingEnvironment,IEICETransactionson InforMation and Systems,Vol.E81-D,No.5,pp427-433(1998).