

## 生活活動分析のためのGIS支援ツールの開発

荒井良雄\*・大木聖馬\*\*・武者忠彦\*\*

- I はじめに
- II 開発環境とデータ概要
- III GIS支援ツールの開発
- IV GIS支援ツールの適用
- V 結論
- 付録 Visual Basicのコード

### I はじめに

女性の社会進出や週休2日制の定着による余暇時間の増大によって、都市住民のライフスタイルが多様化し、現代都市の時空間構造も大きな変化を見せている。筆者らは、近年のこうした社会変化の実態を、時間地理学の枠組みに準拠した生活活動分析によって解明しようと、わが国のいくつかの都市で実際の生活活動調査を試み、分析作業を積み重ねてきた。その成果の大要は単行本の形ですでに上梓している<sup>1)</sup>。

こうした生活活動に関する調査分析作業は、調査技術的にも、分析手法的にも解決すべき事項が多く、これまでの研究蓄積が乏しいこともあって、新しいノウハウを独自に開発する必要に迫られることが多かった。一般に、研究活動の裏側にある技術的ノウハウは公にされることが少ないが、生活活動調査のように技術的困難の多い分野では、各個の研究者がそれぞれに開発・蓄積してきたノウハウを公開することによって、全体的な技術水準を高め、ひいては、さらに多くの研究者の参加を促して、研究の裾野を広げる努力が必要であると考え、かつて生活活動データのデータベース構築と分析システムに関する技術的報告を発表したのも<sup>2)</sup>、そうした認識からであった。

ところで、実際の生活活動記録をコーディングしてデータ化する際に、非常に問題になるのが活動場所の同定である。時間地理学の枠組みから当然のこととして、「どの

---

\* 東京大学人文地理学教室

\*\* 東京大学大学院生

活動がいつどこで行われたか」という情報は生活活動分析の基本である。しかし、「何時何分」という形で簡潔に表現できる活動時間に比べて、活動場所の空間的位置は、調査上もデータ表現上も著しく扱いにくい。上記の報告にも記したように、これまででは、調査票に記入された名称を基に、住宅地図や電話帳を用いて、活動場所を手作業で探し出して同定する作業を行ってきた。容易に想像されるように、この作業は大変に手間が掛かり、数千に及ぶ活動記録のすべてを処理するためには膨大な時間を要するために、生活活動分析の作業効率の上での最大のボトルネックとなっていた。また、同定した活動場所をデータとして表現するためのフォーマットも大きな問題であり、上記報告の段階では、数値・文字のみを扱うデータベースとしての制約から、「自宅からの距離」というきわめて限定された情報の形を取らざるを得なかった。

しかし、この問題に対して、1990年代に急速に発展・普及してきた地理情報システム（GIS）技術は解決のための有力な手段を提供しうると思われる。GISは地物をポイント、ライン、ポリゴンといった幾何学的要素として認識し、各種の地理的分析を加えることができる。生活活動分析における活動場所も、本質的にはポイントと考えることができるから、GIS上に構築した地域モデルの中に活動場所のポイントデータを導入すれば、これまで不可能だったさまざまな活動分析が可能になると考えられる。

一方、分析作業上のボトルネックであった活動場所の同定については、Address-matching等のGIS技術を応用することによって、作業効率を向上させることが考えられる。個々の建物名等のデータを含んだ、詳細な地域データが比較的容易に利用できるようになった今日、かつて、電話帳と住宅地図を頼りに目視で検索を行った作業をコンピュータ上で自動的に行う可能性が出てきたのである。

こうした事態の推移を受けて、筆者らは、GISを応用した生活活動分析支援システムの試作に着手した。本稿では、現時点で技術的に一応の目途のたった、Address-matchingによる活動場所同定作業の支援システムを中心に、GIS上での生活活動データベース構築の例を報告し、併せて、GISの空間分析機能を利用した活動分析の若干の試みを紹介する。

## II 開発環境とデータ概要

本研究で開発するGIS支援ツールは、最も手頃で手軽に利用することが可能なGIS

アプリケーションや汎用データベースアプリケーションを、段階的に利用して構築する。Address-matchingを行う際のリレーショナルデータベースとしては、Microsoft社のAccess2000を使用し、活動パスを描画し地図上における活動空間の分析には、ESRI社によって開発された代表的なGISソフトであるArcView3.0を使用した。また、ArcView3.0が利用するデータベースはdBase形式であり、この形式のファイルを容易に編集することができるMicrosoft社のExcel2000を補完的に用いている。

## 1 地図データの概要

本研究で利用する「地図データ」とは、川越市のゼンリン住宅地図（1995年に作成された川越市都市計画基本図を基に作成）を株式会社パスコに依頼してArcView上に移植したデータである。川越市を361のグリッド（格子）に分けて、各グリッドに属する地域の情報をレイヤーごとに分けて管理している。これはArcViewがレイヤー構造のGISであるため、レイヤー構造とは地理フィーチャー<sup>3)</sup>をそれぞれ一枚のレイヤー(layer)に一括管理し、それらを重ね合わせる(overlay)ことで全ての地理フィーチャーを一枚の地図に描画するという構造をとるものである。そのため、レイヤーごとのデータテーブルに地理フィーチャーはそれぞれ別個に管理されており、これら複数のデータテーブルを一括して管理するデータベースがArcViewなのである<sup>4)</sup>。(ArcViewではレイヤーのことを「テーマ」と呼ぶ。以下レイヤーのことをテーマと呼ぶことにする)

この地図データに含まれているテーマをまとめたのが第1表である。これらのテーマがすべてオーバーレイされることで、川越市の地理情報がすべて表示されることになる。一方、必要としている情報のみを表記したければ、該当するテーマだけをオーバーレイし、他のテーマはオーバーレイせずに非表示にすればよい。

これらテーマにはいずれもそれぞれの情報を格納した属性テーブルがあり、この属性テーブルがもとになって各情報が管理され、テーマとしてView<sup>5)</sup>画面上で描画されることになる。このテーブルはdBASE形式のデータベースファイルで、第1表のファイル形式のうち、「.dbf」が該当する<sup>6)</sup>。本研究では後述の活動データと地図データの整合性をはかるためにAddress-matchingする必要があることから、建物テーマ（ファイル名「TATEMONO」）のdBASE形式データベースファイルをAccessに移植する。

第1表 地図データに含まれるテーマの種類

No	テーマ名称	図形タイプ	ファイル名	ファイル形式
1	町（大字）界	ポリゴン	OAZA	.shp .shx .dbf
2	丁目（字）界	ポリゴン	CHOME	.shp .shx .dbf
3	街区界	ポリゴン	GAIKU	.shp .shx .dbf
4	地下街面, 施設界面	ポリゴン	CHIKA	.shp .shx .dbf
5	建物	ポリゴン	TATEMONO	.shp .shx .dbf
6	建物補助	ポリライン	TETEHOJO	.shp .shx .dbf
7	水域	ポリゴン	MIZU	.shp .shx .dbf
8	水域補助	ポリライン	MIZUHOJO	.shp .shx .dbf
9	道路	ポリライン	ROAD	.shp .shx .dbf
10	鉄道	ポリライン	TETUDOU	.shp .shx .dbf
11	歩道	ポリライン	HODOU	.shp .shx .dbf
12	等高線	ポリライン	CONTOUR	.shp .shx .dbf
13	地類界	ポリライン	CHIRUI	.shp .shx .dbf
14	施設	ポリライン	SSETU	.shp .shx .dbf
15	行政界	ポリライン	GYOUSEI	.shp .shx .dbf
16	記号	ポイント	SYMBOL	.shp .shx .dbf
17	建物名, 地番, 街区番号	ポイント	MOKUHYO	.shp .shx .dbf
18	基本注記	ポイント	CHUKI	.shp .shx .dbf

この建物テーマのデータベースファイルには川越市に存在する全建物に関する情報が収められており、レコード数は88752件である。本研究で必要となるフィールドとしては、まず「TATEMON\_ID」フィールドには各建物に固有のIDが割り当てられており、このテーブルの主キー<sup>7)</sup>となる。レコード数に対応して1から88752の数字が割り当てられている。次に「Acode」「Ccode」「Gcode」「Chiban」フィールドがあり、これらは各建物の住所に関する情報が収められている。「Acode」には大字のコード

が、「Ccode」には字丁目のコードが、「Gcode」には街区番号のコードが、「Chiban」には番地のコードがそれぞれ割り当てられている。たとえば、「旭町3-21-11」という住所の建物は各フィールドに対して以下のような数値をとる。

Acode :3 (大字旭町に該当するコード)

Ccode :3

Gcode :21

Chiban:11

よって活動場所の住所の完全な情報が入手できれば、これらのフィールドを用いることで容易にその場所を地図上で特定できる。後述の活動データはこのような完全な住所の情報は含んでいないが、活動の起点となる各世帯の住所は完全な住所情報が得られており、Address-matchingに際しこれらのフィールドを利用することができる。また次章で行う Address-matching の近隣検索において、これらフィールド項目を利用することになる。

最後の「HOUSENAME」というフィールドに、建物名称が格納されている。この建物名称と活動データの活動場所名を Address-matching することになる。この建物名称は、先述のゼンリン住宅地図の表記に準じているため客観性がきわめて強く、他のレコードと混同するようなことはほとんどない。

## 2 活動データの概要

ここで扱う活動データとは、生活活動に関するアンケート調査によって得られたデータである。このアンケート調査は1990年10月28日(日)午前0時から29日(月)午後12時までの48時間の間に行った活動の内容・時間・場所を日誌形式の調査票に逐一記録してもらった活動日誌法によって行った。調査対象世帯は原則として夫が勤労者である普通世帯を対象とし、自営業者の世帯や農家は除外しており<sup>8)</sup>、対象世帯の夫婦それぞれに回答を求めている。223世帯に回答を求め、そのうち204世帯403人の記録を回収した<sup>9)</sup>。この活動データによって大都市東京の近郊に住む都市住民の生活活動の分析が可能となる。

この活動データはすでにデータベース上にコード化されているが<sup>10)</sup>、大きく3つのテーブルに分けて格納され、世帯番号をキー項目としてリレーションが組まれている。

「基本台帳」テーブルには各世帯の基本的な属性情報（世帯主名、年齢、住所など）を格納している。また「活動（日）データ」テーブルと「活動（月）データ」テーブルには、それぞれ月曜日および日曜日の活動記録をコード化した活動レコードを格納している。これらテーブルのフィールド構成を第2表と第3表に示した。「基本台帳」テーブルでは世帯番号はレコードごとにユニークである（主キーとなる）が、「活動データ」テーブルでは同一の世帯番号を持つレコードが複数になるので、1対nのリレーションを組むことになる。

ところで、これらテーブルのフィールド構成は調査票からデータベース上にコード化する際の便宜を考慮して設定されており、本研究での支援ツール作成に際しては若干の不備がある。そのため最終的には本研究に適したフィールド構成を再構築する必要がある。ここでは、本研究で利用可能な従来のフィールドを選択し、その若干の説明を行う。また新たに設定するフィールドを模索するために、従来のフィールドの問題点を考えて、フィールド構成の再構築を行う

第2表 コーディング時の基本台帳テーブルのフィールド構成

No	名称	型	No	名称	型	No	名称	型
1	世帯番号	整数	23	家族4 免許	文字	45	妻就業形態	文字
2	氏名	文字	24	家族5	文字	46	夫勤務日	整数
3	住所	文字	25	家族5 年齢	整数	47	夫勤務開始時刻	整数
4	地区	文字	26	家族5 職業	文字	48	夫勤務終了時刻	整数
5	世帯類型	文字	27	家族5 免許	文字	49	夫通勤手段1	整数
6	ライフステージ	文字	28	家族6	文字	50	夫通勤手段2	整数
7	注記	文字	29	家族6 年齢	整数	51	夫通勤手段3	整数
8	夫氏名	文字	30	家族6 職業	文字	52	夫通勤手段4	整数
9	夫年齢	整数	31	家族6 免許	文字	53	夫通勤手段5	整数
10	夫職業	文字	32	家族7	文字	54	夫通勤時間	整数
11	夫免許	文字	33	家族7 年齢	整数	55	妻勤務日	整数
12	妻氏名	文字	34	家族7 職業	文字	56	妻勤務開始時刻	整数
13	妻年齢	整数	35	家族7 免許	文字	57	妻勤務終了時刻	整数
14	妻職業	文字	36	自家用車	整数	58	妻通勤手段1	整数
15	妻免許	文字	37	バイク	整数	59	妻通勤手段2	整数
16	家族3	文字	38	入居時期	整数	60	妻通勤手段3	整数
17	家族3 年齢	整数	39	住居形態	文字	61	妻通勤手段4	整数
18	家族3 職業	文字	40	所有形態	文字	62	妻通勤手段5	整数
19	家族3 免許	文字	41	夫勤務先	文字	63	妻通勤時間	整数
20	家族4	文字	42	夫勤務地	文字			
21	家族4 年齢	整数	43	妻勤務先	文字			
22	家族4 職業	文字	44	妻勤務地	文字			

第3表 コーディング時の活動データテーブルのフィールド構成

No	名称	型	No	名称	型	No	名称	型
1	世帯番号	整数	9	用件分類	文字	17	前同行分類	文字
2	個人番号	整数	10	同行者	文字	18	後移動時間	整数
3	総ストップ数	整数	11	同行分類	文字	19	後移動手段	整数
4	内ストップ数	整数	12	開始時刻	整数	20	後同行者	文字
5	サイクル番号	整数	13	終了時刻	整数	21	後同行分類	文字
6	ストップ番号	整数	14	前移動時間	整数	22	用件コード	整数
7	場所	文字	15	前移動手段	整数	23	自宅～	実数
8	用件	文字	16	前同行者	文字			

## a 世帯番号・個人番号

「世帯番号」フィールドは「基本台帳」テーブルの主キーであり、「活動データ」テーブルとのリレーションを組む際のキー項目である。「個人番号」フィールドには、夫：1，妻：2の数字が割り当てられる。この「世帯番号」と「個人番号」の2つのフィールドを組み合わせることによって、個人を特定することができる。

## b ストップ・サイクル

活動レコードの発生順序を整理しているのが「ストップ番号」と「サイクル番号」フィールドである。自宅から出て自宅へ帰るまでの一連の活動の連鎖(トリップチェーン)をサイクルと呼び、1日の最初からサイクルに連番をつけたのがサイクル番号、1つのサイクルの中で個々の活動に発生した順に連番をつけたのがストップ番号である。

このサイクル番号は各個人でカウントされている。たとえば、ある個人がn回の活動サイクルを行ったとすると、それぞれ1からnの数字が割り当てられ、また別の個人がm回の活動サイクルを行ったとすると、同様に1からmの数字が割り当てられる。つまり、サイクルだけで各レコードを見ていくと、個人間で数字が重複してしまい、あるサイクルがどの個人によって行われたものであるかを特定することができないという問題点がある。とりわけ本研究では、支援ツールによってトリップチェーンをGIS上で描画することを考えると、サイクルによって活動レコードを管理できるようにしておく必要がある<sup>11)</sup>。

## c 住所・場所

各世帯の自宅の住所は「基本台帳」テーブルの「住所」フィールドに収められてい

る。支援ツールでトリップチェーンを扱っていく際に、自宅は各トリップチェーンの起点と終点となるため、大変重要な情報である。また自宅を出て帰ってくるまでの間に行った活動の活動場所に関しての情報を収めたフィールドが「活動データ」テーブルの「場所」フィールドである。これは各サンプルが調査票に記入したものをそのまま各レコードに収めたものである。よって客観的情報に欠けるレコードが多く、**Address-matching**の際に問題となってくる。

#### d 移動手段・移動時間

活動レコードの活動場所への前後の移動の際にどのような移動手段を利用し、どれだけ時間がかかったかの情報が含まれている。移動の前後を分けて「前移動手段」「前移動時間」と「後移動手段」「後移動時間」フィールドが設けられている。

### 3 データテーブルの再構築

以上、地図データと活動データの概要をまとめた。本研究でこれらのデータを利用するにはフィールド構成にいくつかの問題点が含まれていた。そこでここではフィールド構成を再構築して、本研究に最適なデータテーブルを考える。

本研究の支援ツールを作成するには、活動データをArcView上に読み込めるようにする必要がある。そのためには活動データの活動場所情報が地図データのどの建物情報と一致しているかを一意に特定する必要がある。そこで行われる作業が**Address-matching**である。**Address-matching**をなるべく高速で処理させることで、ユーザーにストレスを与えない支援ツールを構築することができる。また、活動データのフィールドを全部用いようとしても、ArcViewにエクスポートする際に技術的な問題が発生する可能性がある。よって**Address-matching**の際には必要とするフィールドだけで構成された最小限のデータテーブルを利用することにし、必要に応じて従来のフィールド構成のデータテーブルを利用することにする。

#### a CODE 建物テーブル

このテーブルは地図データの各建物の完全な住所情報が収められている(第4表)。各世帯の自宅の住所を**Address-matching**する際にこのテーブルを利用する。

#### b HN 建物テーブル

このテーブルは地図データの各建物の名称を収めている(第5表)。活動場所の



Address-matchingを行う際にこのテーブルを利用するため、2つのフィールドだけで構成し、余分な情報は削除している。

c 大字テーブル

多くのテーブルで「ACODE」フィールドという、大字情報をコーディングしたものを利用しているが、各コードがどの大字に対応しているかを格納したテーブルがこの大字テーブルである(第6表)。活動データの活動場所のレコードには大字名を含んだものが多数見られる。そこでこのテーブルの大字名フィールドをAddress-matchingの際に参照し、活動場所の大字コードを特定する。

第4表 CODE 建物テーブルのフィールド構成

No	名称	型
1	TATEMON_ID	オートナンバー
2	ACODE	整数
3	CCODE	整数
4	GCODE	整数
5	CHIBAN	テキスト

第5表 HN 建物テーブルのフィールド構成

No	名称	型
1	HOUSENAME	テキスト
2	TATEMON_ID	オートナンバー

第6表 大字テーブルのフィールド構成

No	名称	型
1	大字名	テキスト
2	ACODE	整数

第7表 活動データテーブルのフィールド構成

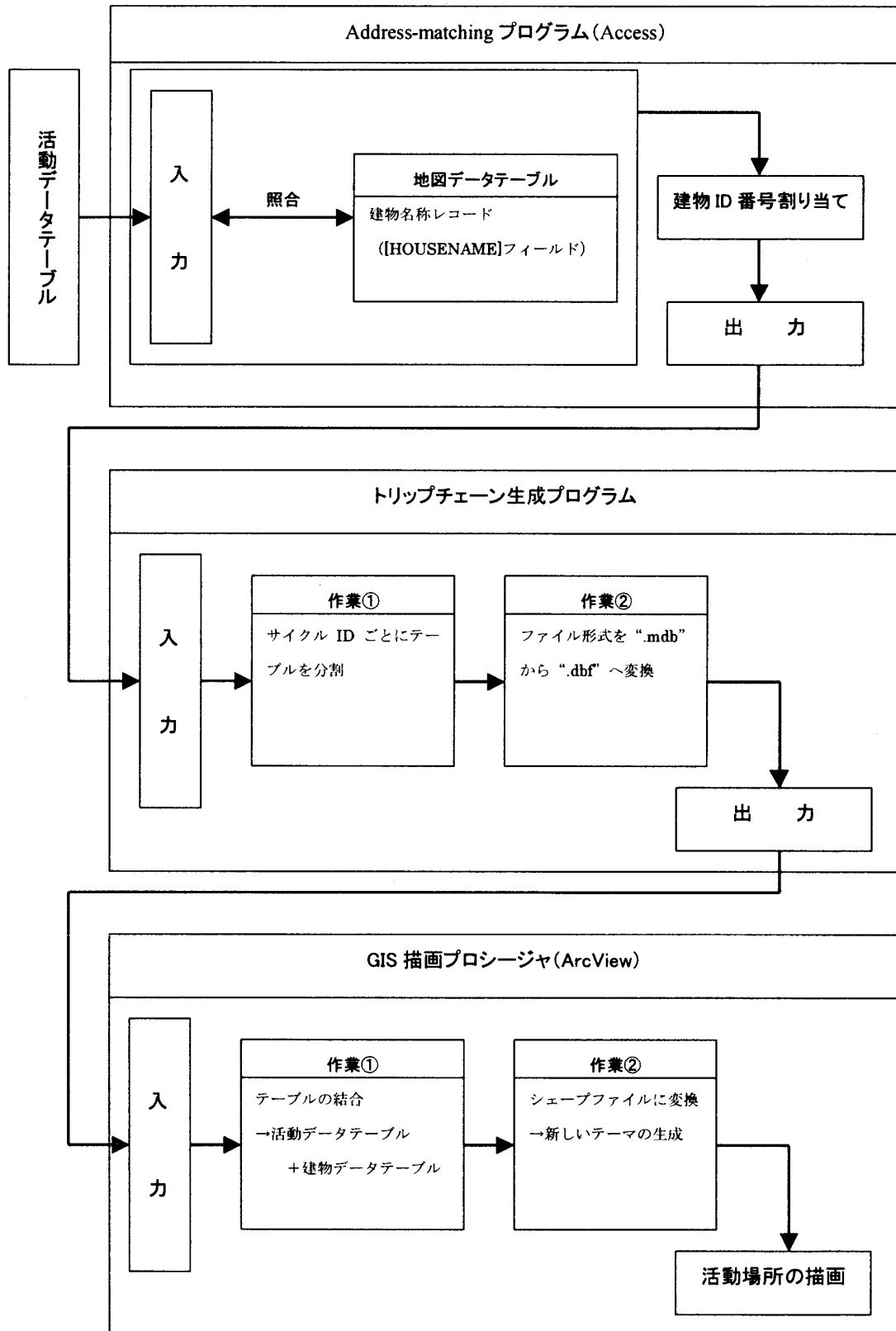
No	名称	型	No	名称	型
1	活動 ID	オートナンバー	7	注	テキスト
2	世帯番号	整数	8	TATEMON_ID	整数
3	個人番号	整数	9	前移動手段	整数
4	サイクル ID	整数	10	前移動時間	実数
5	ストップ番号	整数	11	後移動手段	整数
6	場所	テキスト	12	後移動時間	実数

#### d 活動データテーブル

アンケート調査によって得られた活動データは、さまざまな情報をそのテーブルに収めている。しかし、この情報を全て利用しようとする、データ処理速度が低下し、快適な支援ツールを作成することができない。ここでは活動データをArcView上に移植することに主眼を置き、移植に必要な最小限のフィールドで構成し直した(第7表)。こうすると、他の有益な情報が捨象され、生活活動空間自体の分析に不利益を被るような印象を受ける。しかし、捨象してしまうフィールドは、描画のために構築された活動データテーブルに、分析の必要に応じて付け加え、ArcView上に再移植するという手続きを踏めば、何ら問題はない。そのためにもまずは描画用の最小限のテーブルを作成する必要がある。ArcViewへの移植の方法は次章で述べる。

### III GIS支援ツールの開発

本研究で開発するGIS支援ツールは、生活活動空間をGISソフト上で描画するためのツールであり、その作業過程は大きく3つの段階に分けることができる(第1図)。最初の段階のAddress-matchingでは、活動場所のデータに地図上の位置情報を付加するために、活動データを入力して地図データと照合し、GIS上で読み込むことができる形式で活動データを返す。このAddress-matchingの精度がGIS上で描画される生活活動空間の精度を大きく左右する。次に、生活活動空間をトリップチェーンごとに分析するために、活動データテーブルをトリップチェーンごとに分割していく。このような作業の必要性は、ArcView上でテーブルの結合<sup>12)</sup>を行う際に、2つのテーブルの関係が1対1<sup>13)</sup>でないと、テーブルの結合がスムーズに行われずという機能上要請からくるものである。活動データは、全サンプルが曜日別に1つのテーブル納められており、駅などの活動場所は複数のサンプルで重複している。これを地図データ結合すると、データが欠落してしまう可能性がある。このような現象を防ぐために、トリップチェーンの分割が行われる。最後にArcView上で活動パスをトリップチェーンごとに描画することを行う。ArcViewはVBAなどの汎用プログラム言語との互換性がなく、一連の作業を自動化するといったことが困難である。よってここでもトリップチェーンごとに分割されたデータテーブル1つ1つに作業を加えていくことになる。以下で、



第1図 支援ツールの作業過程

それぞれの過程の詳細について述べていく。

## 1 Address-matching プログラム

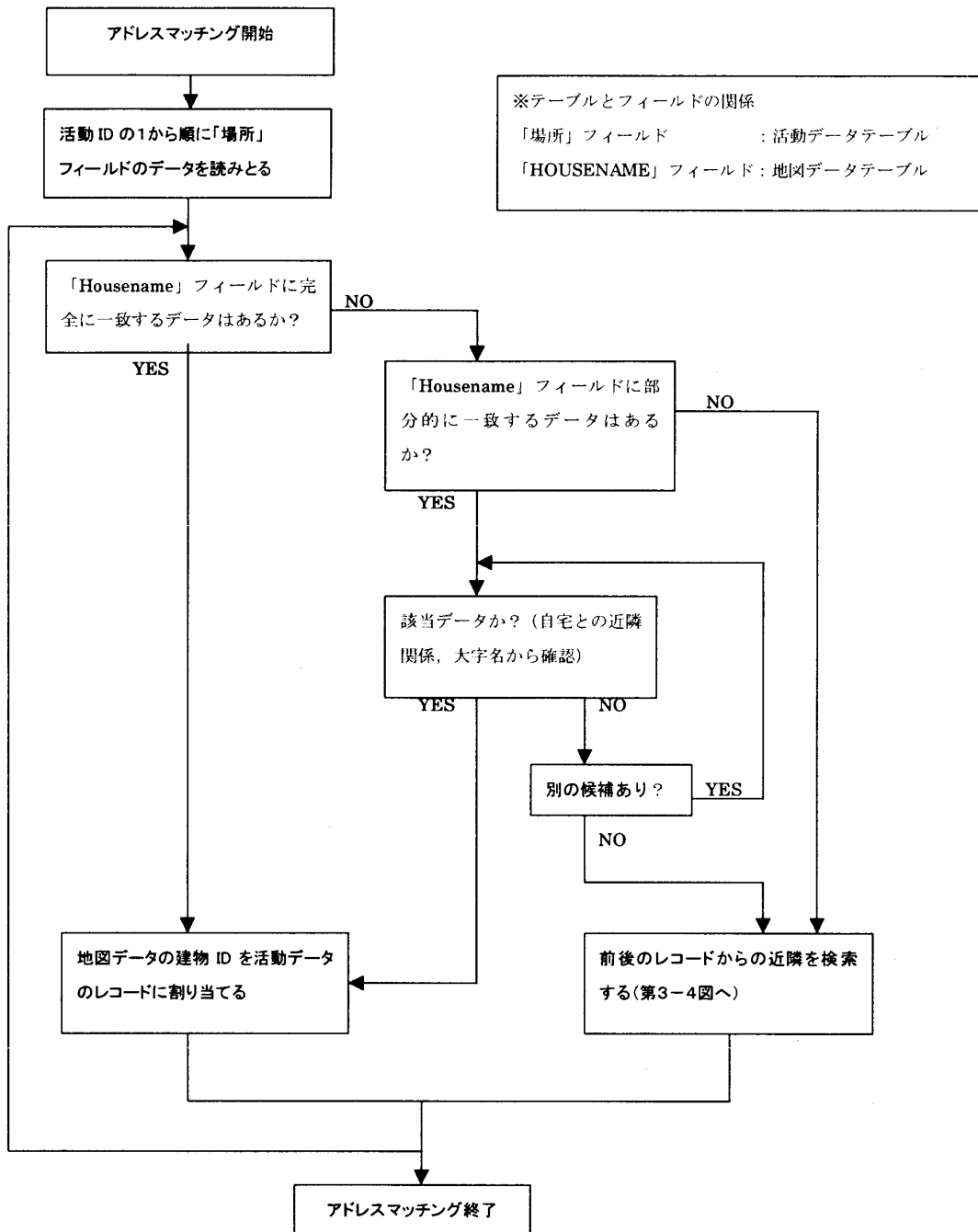
### a プログラムの性質と作業手順

Address-matchingはAccess上でVisual Basicを用いて専用フォームを作成し、ユーザーとの対話方式で各レコードに対して逐次行っていく。この対話方式をとる理由としては、Address-matchingの作業を単に自動化して一度に全てのレコードに対して行ってしまうような方式にすると、機械だけに全ての判断を委ねることになり、柔軟性に欠け、精度が落ちてしまうことによる。各レコードのAddress-matchingの結果をユーザーが各レコードごとにチェックし、その都度誤りを含んでいたら修正を施していく方が精度が高く、結果的に効率も良い。そこで、手作業では煩わしい部分（たとえば候補の検索やID番号<sup>14)</sup>の割り当てなど）を中心に自動化を行い、最終的なmatchingの判断はユーザー自らが行うことにする。その際にはユーザーの判断材料となる各種情報（活動場所や候補の建物の大字名、字丁目、番地など）も提示させるようにして、ユーザー支援型のプログラムを開発することに主眼を置く。

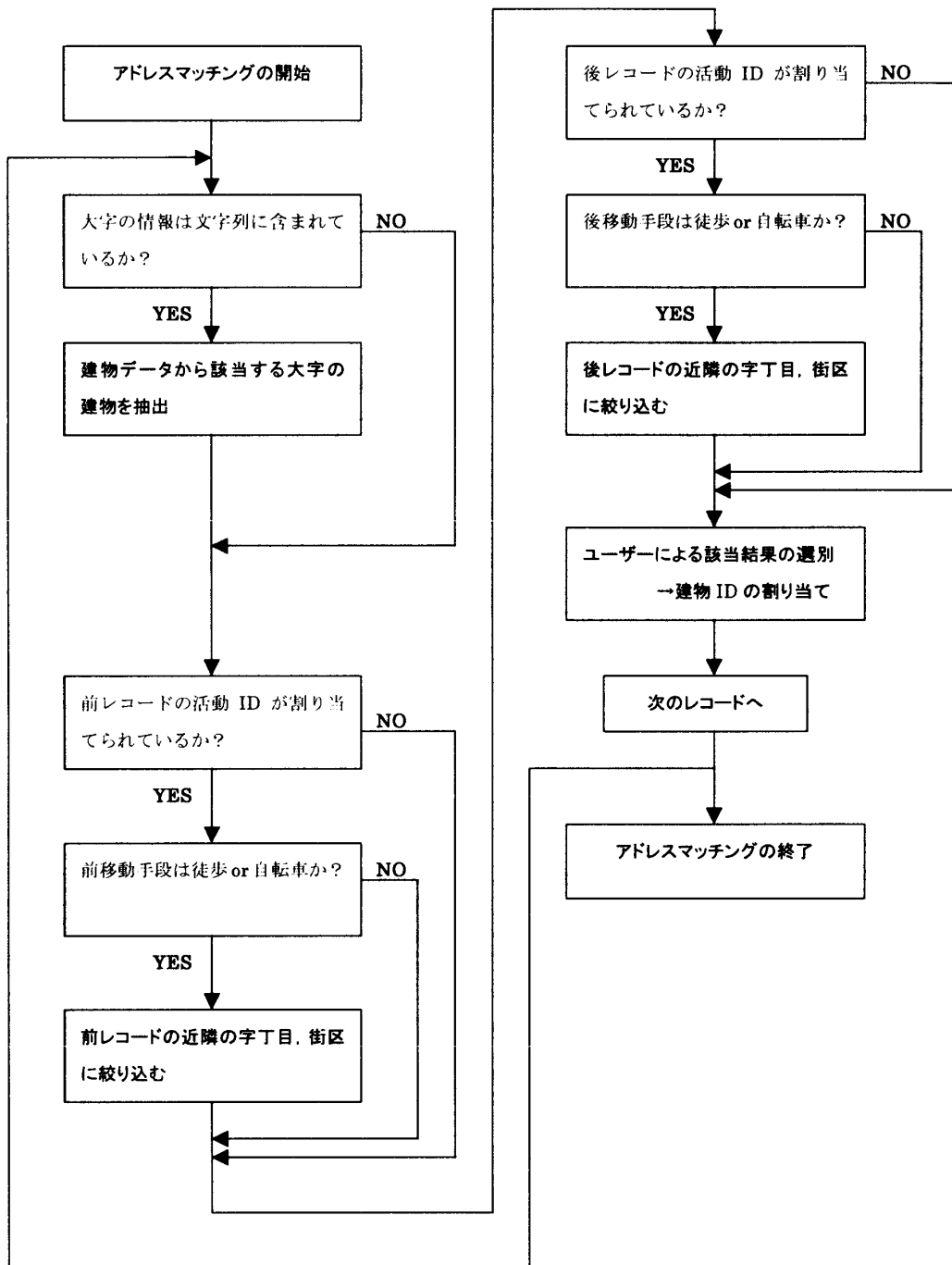
このAddress-matchingプログラムは、最初にサンプルの自宅のAddress-matchingから開始する。これは原則として活動サイクル（トリップチェーン）は自宅を起点として始まり、自宅を終点として終わるためである。活動データテーブルには自宅に関してのレコードは含まれていないため、基本台帳テーブルでAddress-matchingを行う必要がある。この基本台帳テーブルでは自宅に関するデータが完全な住所情報として与えられており、この住所情報を用いればAddress-matchingをほぼ100%の割合<sup>15)</sup>で達成することが可能だからである。基本台帳テーブルでの自宅のAddress-matching<sup>16)</sup>の次には、活動データテーブルの活動場所のAddress-matchingを行う。この活動場所のレコードについては客観性を欠くレコードが少なからず存在するため、柔軟性のあるAddress-matchingのプログラムを構築する必要がある。

### b アルゴリズム

Address-matchingプログラムのアルゴリズムを第2図と第3図に示した。活動場所のレコードが含む客観性の強弱によって、大きく2つのアルゴリズムにわけてプログラムは実行される。最初に第2図のアルゴリズムによって、客観性の強い活動場所レ



第2図 アドレスマッチングのアルゴリズム1



第3図 アドレスマッチングのアルゴリズム2

コードの **Address-matching** が行われる。このようなレコードは、地図表記に比較的近い表記がなされており、地図データに収められているきわめて客観性の強い建物名称との **Address-matching** がスムーズに行われやすい。ここで **Address-matching** することができない客観性の弱い活動場所レコードとは、記入者の主観的地理表現がその表記に多分に含まれたものであり、活動場所以外の情報も利用する第3図のアルゴリズム上で **Address-matching** の実現を目指していく。

第2図のアルゴリズムで最初に行われるのが完全一致検索<sup>17)</sup>である。一致候補が見つければ建物IDを割り当て次のレコードに移る。一致しない場合は部分一致検索<sup>18)</sup>を行う。部分一致検索では、どの部分を検索に用いるか、ユーザーが指示できるように設定しておく<sup>19)</sup>。一致すればIDを割り当て、一致しない場合はさらに条件をゆるめて検索を進めていく。そして最終的に一致しないデータは、次のアルゴリズムのプログラムに委ねる。このプログラムでの **Address-matching** は、かなり高い精度で達成される。

第3図のアルゴリズムのプログラムでは、活動前後の情報を利用して **Address-matching** の候補を絞り込んでいく。初めに活動場所レコードに大字の情報が含まれていないかを検索する。含まれている場合、その大字に活動場所が限定できるので、9万件近くの建物データからその大字に所属する建物のみを抽出する。次に、処理を施している **current** レコードの前後のレコードは建物IDを割り当てられているかどうか確認する。割り当てられていればそのレコードからの近隣関係を利用することができる。その際、移動手段が徒歩か自転車に限って、比較的近距离の移動と判断し(500m以内)<sup>20)</sup>、前後からの近隣関係から検索する。具体的には字丁目、街区で近いものを抽出し最終的なユーザーの判断材料を絞り込むことになる<sup>21)</sup>。したがって、活動場所の情報だけでは判断しかねるものを対象にこのアルゴリズムは実行されており、精度は若干低いものとなる。

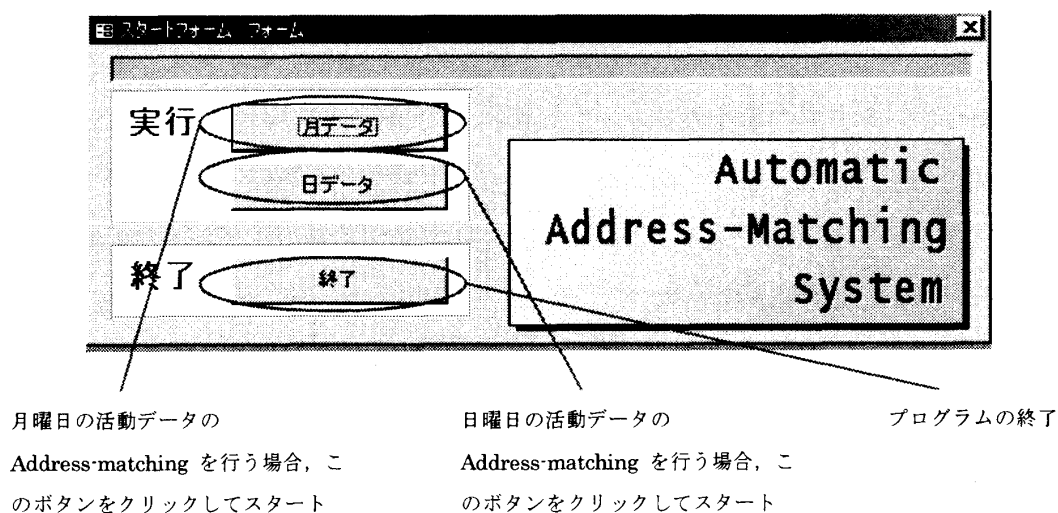
### C **Address-matching** 専用フォーム

アルゴリズムに従って、実際に **Address-matching** の専用フォーム<sup>22)</sup>を作成する。複雑な作業をバックグラウンドでコンピュータに行わせ、作業結果をフォームに反映させるようにする。

ここではプログラミング言語として、汎用的であり利用しているアプリケーションとの互換性が優れているVBA (Visual Basic for Applications)<sup>23)</sup> という言語を用いており、Address-matchingの作業を行うためにフォームに施してあるVBAの各種コードは付録に記載してある。以下、作成したフォームがどのように機能するかを説明していく。

#### スタートフォーム (第4図)

このフォームを最初に起動し、ここからAddress-matchingのフォームへと移る。活動データは日曜日と月曜日のものがあり、それぞれ別のテーブルに分けて格納されている。そこで日曜と月曜の作業は分けて行うことにし、それぞれの作業を選択できるように、2つのボタンを設けてある。「月データ」ボタンをクリックすれば、月曜日の活動データテーブル上でAddress-matchingを行い、「日データ」ボタンをクリックすれば日曜日のそれで行うことになる。生活活動調査のアンケートが行われた曜日・曜日数に応じて、このボタンは名称を変えたり、増やしたりすればよい。どちらのボタンをクリックしても、Address-matchingを行うフォームは同じものを扱う。



第4図 スタートフォーム



### Address-matching フォーム (第5図)

スタートフォームでAddress-matchingを行う活動の曜日を選ぶと、Address-matching フォームが開かれる。このフォーム上で具体的な作業を行っていく。

このフォームでは、「対象データ」という名で黒く囲まれた部分に活動レコードの詳細が表示される。活動データテーブルの全フィールドがこの黒枠内に表示されるが、必要に応じて基本台帳の情報を参照できるように、「基本台帳フォーム」へアクセスするボタンも配置してある。表示しているレコードのAddress-matchingは右下にある2つのボタンから行う。「建物名称」ボタンをクリックすると第2図のアルゴリズムに従って、「前後情報」ボタンをクリックすると第3図のアルゴリズムに従ってmatchingを実行する。最初にAddress-matchingの精度が高い前者のボタンから候補を検索し、見つからなかった場合に後者のボタンで候補を検索するという手順を踏む。候補が見つければ、下の「建物データ候補一覧」にその結果が表示される。最下部には見つかった候補数が表示されるようになっている。

この段階で、ユーザーはAddress-matchingの判断を行う必要がある。表示された候補の中に、まさしく活動レコードの活動場所と一致していると思われるものがあれば、それを活動場所として採用することになる。候補は複数見つかることが多いので、そのうちの最適なものを選ばなくてはならない。例えば、第5図では活動場所が「アトレ」となっている。「建物名称」ボタンをクリックすることで、3つの候補が検索された。このうち、どれが正しいものであるかをユーザーが判断しなくてはならない。この例では「アトレ」というスーパーでの買物活動であるため、マンションではない1番目の候補が実際の活動場所であるとユーザーは判断しなくてはならない。このユーザーによる判断が、Address-matchingの精度を上げることになり、ユーザーとプログラムとの対話がここで行われることになる。

候補が見つかったら、その候補を選択した状態（選択すると一番左側にマークがつく）で「取り込み」ボタンを押す。すると黒枠部分の「TATEMON\_ID」の欄に候補の「TATEMON\_ID」の値が代入される。

「建物名称」ボタンのクリックで候補が見つからない場合に、「場所」フィールドのレコードをユーザー自身が変更を加えて検索したい時もある。そういう場合は、「場

選択している候補の「TATEMON\_ID」の値を  
黒枠内の対象データの「TATEMON\_ID」に代  
入する

別なレコードへ移動する際にここにあるボタン  
をクリックする

「基本台帳フォー  
ム」を開く

「取り込み」  
ボタンを押  
して代入さ  
れた

候補を選  
択してい  
る状態

候補がいくつ見つかったかを示す。「2/3」とあ  
れば、3件の候補が見つかり、そのうちの上か  
ら2番目が選択されているということ。

この2つのボタンをクリックすることで実際に  
Address-matching を行い、その結果が直下の  
一覧に表示される。

### 第5図 Address-matching フォーム

所」フィールドをクリックしてユーザーが変更を加えることができるようにしてある。  
たとえば「ロジャース」というレコードを「ロチャース」と変えたい場合には、表示  
されているレコードをそのように直接変えてしまってもよい。変更後に「建物名称」ボ  
タンをクリックすれば、その変更に応じて検索を行う。この場合、活動データテー  
ブルのレコードは更新されないようにしているので、元の表記に戻したければ、そのま  
ま別なレコードに移動してしまえば、自動的に戻っている。

次のレコードの Address-matching に移るときは、右上の「次のデータ」ボタンを押

す。「指定IDのデータ」ボタンは活動ID<sup>24)</sup>を指定してAddress-matchingを行うときに用いる。真上にある「検索ID」の欄に指定したい活動IDの数字を入れてからクリックすると、指定した活動IDのレコードを表示する。Address-matchingを再開する時や特定の活動レコードを表示したいときに用いると便利である。「次の未割り当てデータ」ボタンは、現在表示しているレコード以降で、TATEMON\_IDをまだ割り当てていない最初のレコードへとジャンプするボタンである。

#### 基本台帳フォーム

このフォームはAddress-matchingフォームから開くことができる。Address-matchingフォームで表示中の活動レコードの世帯番号と同じ世帯番号のものが表示されるようになっている。活動場所が勤務先と判断し得る場合に、このフォームを開くことで、具体的な勤務先の名称や住所の情報を得ることができ、ユーザーのAddress-matchingの判断材料として利用することができる。

#### 汎用プログラム

ここで開発したAddress-matchingプログラムは、生活活動分析のための汎用ツールとして利用できるように開発した。本研究で利用した活動データテーブルは特定のフォーマットに基づいているが、このフォーマットは生活活動調査を行った場合、最低限必要とするもので構成されている<sup>25)</sup>。そのため、生活活動調査を行えば、同種の情報を必ず調査するはずであり、本研究の活動データテーブルと同様のフォーマットの活動データテーブルが作成可能なはずである。少なくとも、Address-matchingに必要な「活動場所」と前後の「移動時間」「移動手段」さえ入手できれば、Address-matchingプログラムは動き得る。

このプログラムを利用する際には、本稿の最後に記載したフォームのVBAコードを実際に記述してもらえばよい。前章で説明したものと同じ構造のデータテーブルを用意し、Accessのデータベースにこのフォームと一緒に保存すれば、プログラムを利用することができる。

## 2 トリップチェーン生成プログラム

Address-matchingプログラムによって活動データテーブルには地図データの建物を

参照する ID 番号が割り当てられた。そこで、この活動データテーブルをサイクル ID ごとに分割して、トリップチェーンで活動を管理していくマクロプログラムをExcel上で考案する。ここでExcelを利用する理由は、レコードの操作の自由度が高いためである。

まず、Address-matchingを終えてTATEMON\_IDを割り当てた活動データテーブルを、dBASE形式(拡張子“.dbf”)のテーブルとして「エクスポート」を実行する。これは、ArcViewで読み込めるテーブルがdBASE形式であることによる。ここで、活動データテーブルのフィールド数が多くなると、ArcViewでの操作で不具合が生じるため、描画に不必要なフィールドは削除してしまうとよい。また、トリップチェーンは自宅を起点・終点にしていることから、自宅の建物を参照するID番号も必要である。そこで次に、基本台帳テーブルも同様にエクスポートして、別のテーブルとして保存する。

このようにして用意された活動データテーブル、基本台帳テーブルを用いて、マクロを実行していく。マクロでは最初に、活動サイクルID番号順に昇順でソートする。これはデータ操作上、サイクルID番号が同じものを同一のトリップチェーンとして扱えるようにするためのものである。これに続き、データテーブルから切り出された各トリップチェーンを保存するための“trip\_chain.dbf”というworkbookが自動的に生成され、先頭列にはフィールド名が書き出されるようになっている。(第6図)そして、まず活動データテーブルがサイクルID番号単位のとまりでコピーされ、このworkbookにペーストされる。次に、この活動に対応する世帯番号のデータが基本台帳テーブルからコピーされworkbookの同じsheetにペーストされる。最後に、このsheetには

	A	B	C	D	E	F	G	H	I	J
1	世帯番号	活動ID	個人番号	サイクルID	ストップ番号	TATEMON	場所			
2		51	244	2	187	1	18815 アトレ			
3		51	246	2	187	3	26225 丸広百貨店			
4		51	245	2	187	2	26258 丸広百貨店前くらづくり			
5							78925			
6										
7										
8										
9										
10										

第6図 トリップチェーン

“c” + “サイクルID番号”という名称が自動的につけられる。このプロシージャが、データ終了まで順次繰り返される。

ここでもプログラミング言語として、汎用性のあるVBA (Visual Basic for Applications) を用いており、コードは付録に記載してある。

### 3 GIS 描画プロシージャ

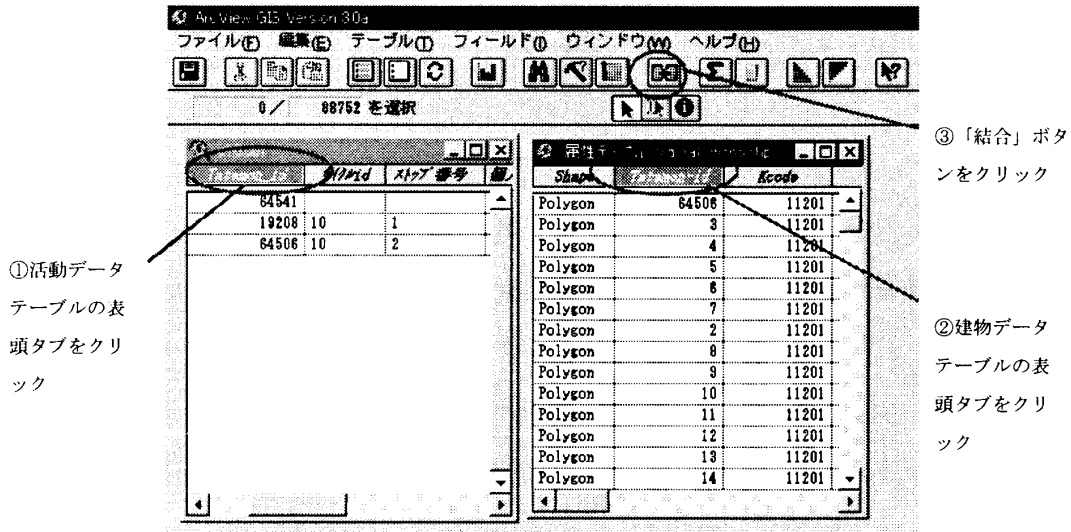
トリップチェーン生成プログラムによって出力されたトリップチェーンごとの活動データテーブルを入力データとして、実際に ArcView 上で生活活動パスを描画する。描画するには、トリップチェーンごとにファイルとして保存し、それを ArcView 上で建物データテーブルと結合してから描画する方法の他に、前節で作成したトリップチェーンテーブルと建物データテーブルを参照しながら、直接活動場所を選択する方法などがある。

#### a 活動場所の描画

前者の方法の場合、前節で生成されたトリップチェーンの dbf データテーブルを ArcView のテーブルに追加して、ArcView の建物データテーブルと取り込んだ活動データテーブルを結合させる。結合させる2つのテーブルを表示した状態にし、まず、活動データテーブルの [TATEMON\_ID] フィールドの表頭タブをクリックし、次に建物データテーブルの [TATEMON\_ID] フィールドの表頭タブをクリック<sup>26)</sup>、最後に2つのテーブルを結合する「結合」ボタンをクリックする(第7図)。こうすることで、建物データテーブルに活動データテーブルが加えられ、活動データが地図地図データと関連づけられたことになる。ところで、前節で触れたように、結合する前の活動データテーブルのフィールド数が多いと、テーブル結合を行った際に活動データテーブルのレコードのいくつかが欠損してしまう。これは ArcView 自体のアプリケーションの問題であるため、根本的な解決は不可能である。

次に結合したテーブルをもとに、新たなテーマを作成する。あらかじめ結合したテーブルにおいて、活動場所となるレコード<sup>27)</sup>を選択表示させておき、「シェープファイルに変換」コマンドを実行する。シェープファイル変換コマンドにより、結合したテーブルのうち選択表示されていたものだけが新たなテーマとなってシェープファイル<sup>28)</sup>

に保存され、建物のテーマや道路のテーマの上に活動データのテーマが重ねて表示される（第8図）。



第7図 テーブルの結合



第8図 活動場所の描画

b 活動パスの描画

活動パスの描画には、ArcViewの拡張機能であるNetwork Analystを利用する。Network Analystは道路ネットワーク上で選択された複数地点を結ぶ最短距離パスを表示できると同時に、各地点における活動起点からの累積距離を求めることができる。ここでNetwork Analystによって表示された最短距離パスのテーマをシェープファイルに変換することによって、活動パスの描画が完成する。

IV GIS 支援ツールの適用

前章でGIS支援ツールを開発し、その概要を説明してきた。ここでは最初に、このツールがどれだけの精度を持ったものであるかを検討した後、精緻な議論ではないものの、ツールの適用事例をいくつか示し、今後の研究展望とする。

1 支援ツールの精度

ここでいう「精度」とは、Address-matchingプログラムの精度である。というのは、ArcView上での描画は、Address-matchingによって割り当てられた建物IDを参照して行われるだけで、建物IDを参照しながらの描画自体をArcViewが誤ることはない。よって、再現された生活活動パスのリアリティも、参照する建物IDを割り当てるAddress-matchingの精度によって決まるのである。

前章で開発したAddress-matchingプログラムを実行した結果が第8表である。市外を除くと、日曜日で50.6%、月曜日で55.9%がこのプログラムによって建物IDが割り

第8表 Address-matchingの結果

	日	月
建物IDあり	655	705
市内	368	339
市外 <sup>29)</sup>	287	366
建物IDなし	359	267
活動場所不明	20	12
計	1034	984

当てられた。ここで2つのことを考えなくてはならない。判明しなかった残りのレコードと、判明はしたがその判明がどれだけ正確なものであるのかという判明の質である。まず判明しなかった残りのレコードを考えると、これらのレコードは、アンケート記入者にしかわからない記述になっているものと、固有名詞を用いた記述になっているが該当する建物が **Address-matching** で見つからなかったものに分けられる。前者の具体例を挙げると、「市内・親戚宅」「市内・〇〇町一丁目」「近所・八百屋」といった表記のものが挙げられる。これらは建物名称で検索しても、前後の活動の情報から検索しても特定することができないレコードである。プログラムの質を向上させても、データがこういった客観性に欠ける表記をしていては、**Address-matching** を達成することができない。

一方後者の具体例は「〇〇ストア」「レストラン××」「市内・△□宅」といった表記のものが挙げられる。固有名詞が使われていても建物名称で検索ができない。この原因は二つ考えられる。一つには、複数階で構成される建物、あるいは、いくつかの店舗が同居している建物では、そのうちの一つの名称に代表されてしまうため、存在していても建物名称で検索することができないのである。二つ目の理由としては、地図の作成年次が違うために、2つの年次間にその名称の建物がなくなってしまい、検索できない例である。これらは前後の活動場所から検索しても、決め手に欠け、やはり決定することができない。

次に判明の質についてである。どれだけ正確に **matching** したかは、アルゴリズムの観点からは前章ですでに説明した。問題となるのは、ユーザーの最終的な判断である。本研究では、なるべく建物名称での **matching** を優先させ、前後の活動場所からの検索は、どう考えてもこれしか考えられないというケースに限って採用しているが、**matching** の確率を優先させた場合には、判明の質が低下することに留意しなければならない。

## 2 生活活動空間距離

GIS 支援ツールによる生活活動空間分析の最大の利点は、活動場所をピンポイントで特定できることにある。それによって、これまではゾーン集計されていた個々の活動距離がメートル単位で測定される。計測にあたっては **ArcView** の拡張機能である **Net-**



work Analyst を使用し、地図データの存在する川越市域内での活動について、自宅から活動場所までの最短距離をそれぞれ求めるといった方法をとった。

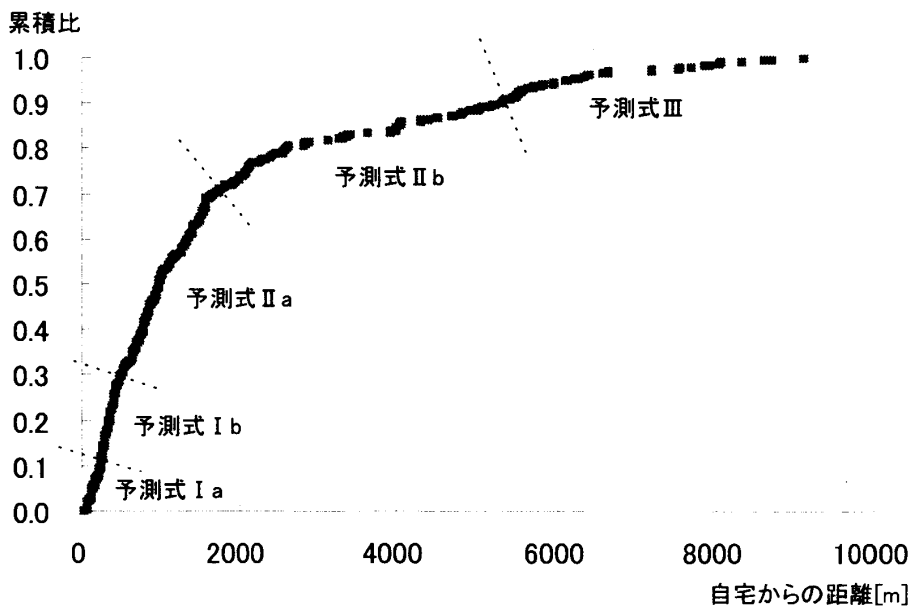
測定されたデータの分析手法には、本稿と同じデータを用いて自宅からの直線距離を500m単位で計測し、活動空間の数量化をおこなった荒井ほか(1996)における手法を採用している。第9図は、仕事以外の活動場所の空間分布を示したものである。ここで「仕事」の活動を区別したのは、「仕事」は固定性拘束性が強く他の行動と空間的性質が大きく異なることと、その多くが市域外での活動であることによる。図中の縦軸は活動数の累積比を示しており、その傾きが大きいほどその距離帯付近での活動数が多いことになる。このような累積比曲線を計量的に把握するために、以下のような指数関数の予測式によって曲線の傾きの度合を調べた。

$$C = 1 - A \exp(-\beta d)$$

ただし、 C : 累積比                       $\beta$  : 定数 (距離減衰パラメータ)

d : 自宅からの距離              A : 定数

荒井ほか(1996)では、上の予測式を用いて実績データにあてはめ、500m、6.0～7.0kmを境界に、全体を3区間(I・II・III)に分割した場合が最も適合度が高いという結果が得られている。この距離3帯構造仮説に基づいて、データの解釈を試みる。



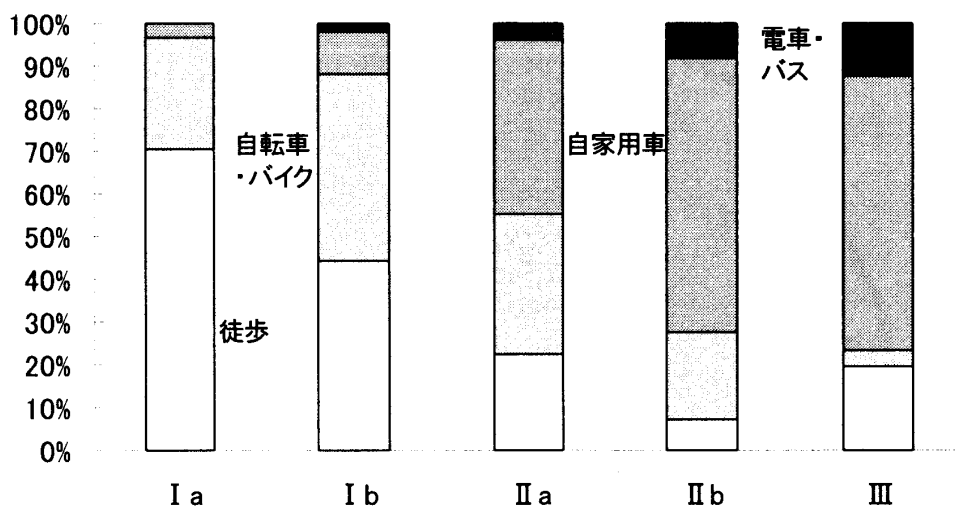
第9図 仕事以外の活動場所の分布

今回のデータについても、予測式の分割点を逐次変化させて予測の精度 ( $r^2$  値) を求め、全体としての精度が最もよくなるような分割をおこなった。この方法によって得られた結果が第9表である。

表中にあるように、今回の分析では累積比曲線のなかに屈曲点が4ヶ所みつき、全体は5区間に分割された。3帯構造仮説に対応させて、それぞれの距離帯を Ia・Ib・IIa・IIb・III とする。このような分割は、今回の分析が荒井ほか(1996)における分析をより精緻化したものであるという性格上、ある程度は予想された結果といえよう。それぞれの  $r^2$  値も1に近い値を示しており、適合性は十分である。またこの区分に従って、距離帯別の移動手段を集計したのが第10図である。

第9表 活動場所の累積比曲線の予測式への当てはめ結果

距離帯	I a	I b	II a	II b	III
区間	~245m	~476m	~1804m	~5319m	~9128m
距離減衰パラメータ	1.779	0.863	0.441	0.051	0.026
$r^2$ 値	0.99	0.995	0.994	0.98	0.944
累積比の上限	0.109	0.293	0.719	0.905	1



第10図 距離帯別移動手段<sup>30)</sup>

荒井ほかでは、自宅から500mがI帯に区分され、距離帯としての意味は「徒歩圏」として解釈された。今回の分析でこれに対応するのがIa帯とIb帯であり、I帯がほぼ半分に分割される結果となった。Ia帯は徒歩圏として解釈することができるが、Ib帯は徒歩だけでなく、自転車・バイクによる移動によっても特徴づけられるといえる。

その外側にあるII帯の上限は6.0～7.0kmであったが、IIa帯およびIIb帯がこれに対応する。IIa帯、IIb帯ともに、距離減衰パラメータはかなり低下し、距離の抵抗が小さい空間となっている。一方でIIa帯とIIb帯の間では距離パラメータで大きな差がみられ、移動手段においてもIIa帯で自転車・バイクが自家用車に匹敵する割合を占めるなど、距離帯区分として意味のある結果もみられた。

もっとも外側のIII帯では、距離パラメータの値も大きく低下する。また、5.5km付近に活動量が集中している距離帯がみられるのも特徴である。これは、川越市の都市構造と何らかの関係を示唆するものとも考えることもできる。

ここではGIS支援ツールによるメートル単位の生活活動分析が、従来の方法に比べてどれほど意味のある結果をもたらしたかについて、詳細な議論にまでは立ち入らないが、このツールを用いることによって距離帯の範囲が精緻に求められ、より細かい区分が可能になるということは明らかにされた。

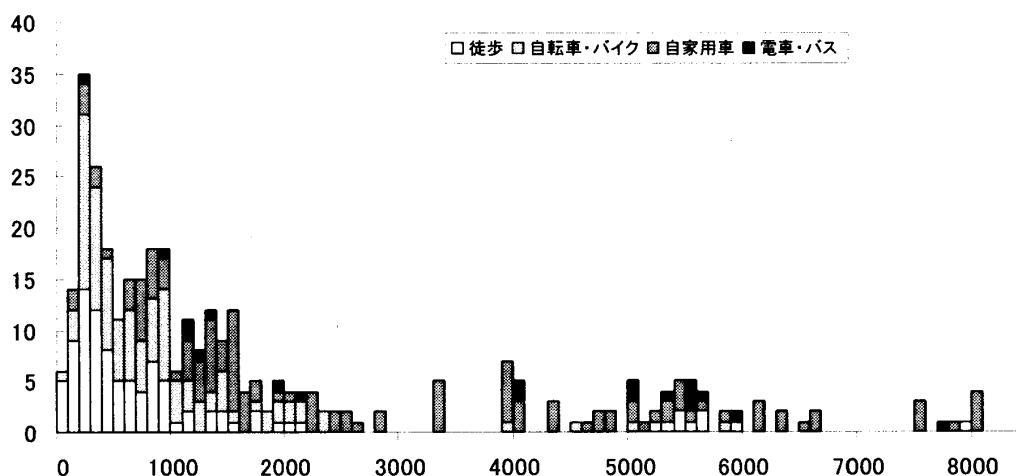
### 3 消費空間の空間的広がり

生活活動のなかでも買物行動に関する研究は、消費空間、購買圏といった概念によってその空間的広がりがより重要な研究対象となっている。これは、店舗を起点とした空間的広がりを求める商圈分析同様、消費者の買物行動の空間的広がりが、小売業にとってのマーケティング戦略において重要な要素となっているからである(佐藤1998)。これまで、商圈の分析としては、特定の店舗でのアンケート調査や、それに基づいて店舗からの距離を変数の一部とした回帰モデルの構築が、地理学の分野以外でも数多くなされてきているが、消費空間や購買圏については、その空間的特性はそれほど明らかにされていない。この原因としてはアンケート調査の難しさのほかに、その空間的広がりを計測する作業が煩雑になることがあげられる。商圈分析では起点としての店舗は一意に決まるが、消費空間の場合は、起点(この場合は自宅)はそれぞれサン

プルごとに異なるため、店舗までの距離を計測するのにも非常に時間がかかる。

GIS支援ツールによって買物行動の消費空間を分析する主な利点は2つ考えられる。ひとつは、Address-matchingによって上述の計測の作業量が大幅に軽減されることである。これによって、消費空間についての実測データを非集計的に扱うことが可能となる。もうひとつは、これまで直線距離や時間距離によって代替されてきた「距離」指標が、より実際の移動距離に近い形で求められることである。

第11図は、生活活動データのなかから買物が主な活動用件であるトリップを抽出し、100mごとの距離帯別度数とその移動手段を示したものである。もちろん店舗の種類によって商圈の大きさは異なるので、この図から直接的に何らかの結論を導くことはできないが、500m以内、5000～6000mにみられる活動量の増加は、コンビニエンスストアの商圈500m仮説やGMSの商圈5km仮説などと全く無関係ではないだろう。



第11図 買物行動の距離帯別移動手段

#### 4 生活活動空間と都市イメージ

GIS支援ツールを用いることによって、活動場所がピンポイントで特定されるとともに、活動パスも地図上に描画される。これによって「住んでいるまちの範囲」や「まちの中心地」といった、都市のイメージとして空間認知の領域で扱われていた事象を、実際の活動空間として実証的に捉えることができる。従来、位置情報を含んだ活動デー

タを扱う場合には、作業量の制約と、それらを地図上に正確に再現することの難しさからメッシュに置き換えてゾーン集計せざるを得なかったが、GISによってこれらの問題はクリアされ、大量のデータを実際の活動空間に極めて近い形で再現し、集計・分析が可能となった。

ケヴィン・リンチによれば、都市のイメージはPath, Edge, District, Node, Landmarkの5つの要素によって構成されるとされる<sup>31)</sup>。このうち3次元的要素であるLandmarkについてここで扱うことはできないが、それ以外の4つの要素については、都市のイメージと生活活動空間がどのように結びついているかを解明するための手がかりとなり得る。そこで、これらの要素に注目しながらGISの適用を検討する。

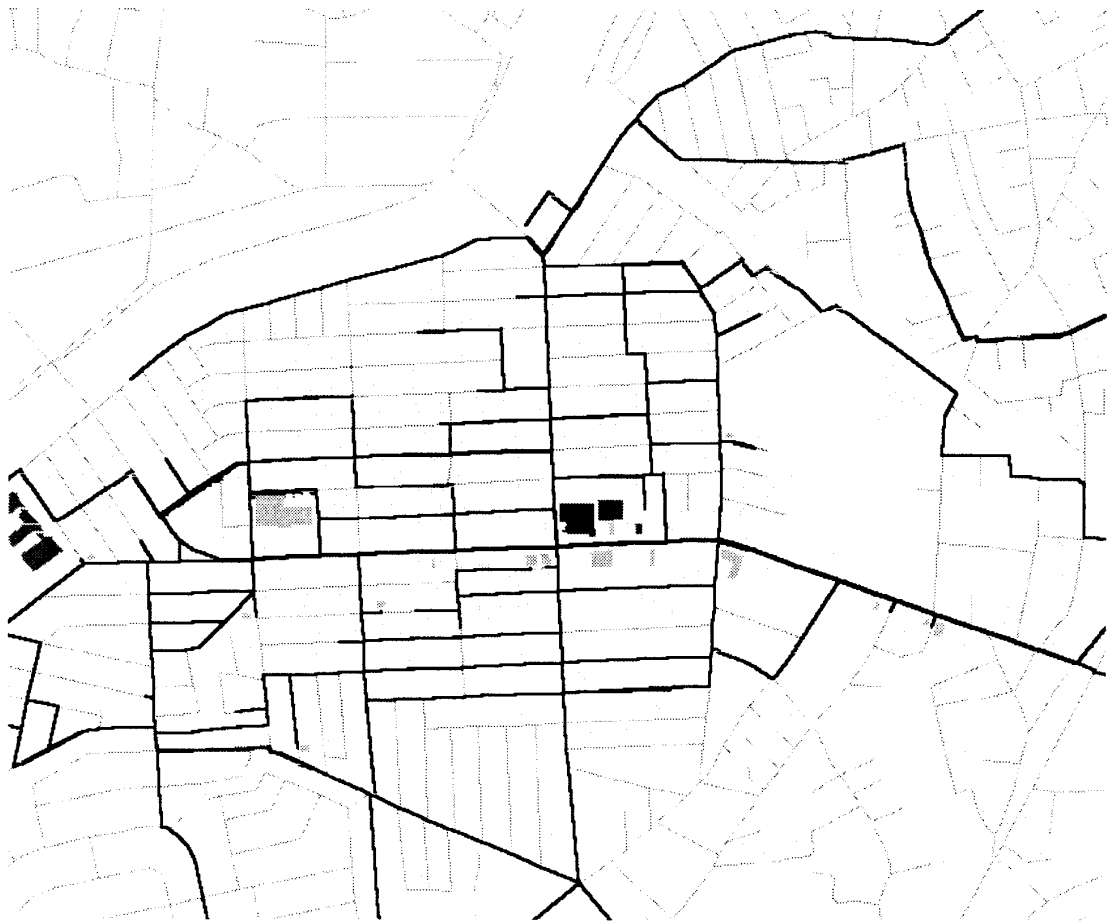
第12図は、川越市域内における仕事を除く全活動パスを描画したものである。ここからただちに都市イメージとの関連性を十分に議論することはできないが、Pathの通



第12図 川越市域内における仕事を除く活動パス

過数を段階的に表現するなどの加工をすることによって、これまでメッシュデータなどによって曖昧にしか捉えられなかった生活活動空間が、どのような範囲に、どのような密度で形成されているかが正確に再現される。それによって、市域レベルでは活動が空間的に集中している3つのDistrictが存在することや、本稿では試みていないものの、Pathの通過数をカウントすることで、「入間川」が空間行動のEdgeとなっている可能性を指摘することができる。

第13図は、よりミクロなスケールでの生活活動空間を示し、活動場所としての建物は活動の頻度によって階級分類されている。このレベルにおいては、河川（「小畔川」）が実際の行動のEdgeとなっている可能性や、活動の集中点としてのNodeになっている建物の分布状況がわかる。



第13図 川越市霞ヶ関地区における活動パスおよび活動場所

リンチのいう5つの都市イメージ構成要素は、人間の内的学習過程を明らかにするものであり、外的物理的形態の操作によってその効果を高めることが都市計画の命題であるといえる。しかしながらこれまで、その「効果」を実証する場合には、再び内的なイメージ構造に立ち戻らなければならなかった。リンチが目的とした都市の外見のわかりやすさ = **Imageability** の向上とは、イメージの向上ではなく、生活活動空間の変化を意味している。この点に関してこそ、今後GIS支援ツールが貢献し得る分野となるだろう。

## V 結論

### 1 生活活動調査とGIS

本研究では生活活動に関するアンケート調査の結果をデータとして利用したため、GIS支援ツールを開発・実行する段階で、データの性質自体にいくつかの不備が生じた。ここでは地理情報としての生活活動データを、とりわけGIS上に移植することを考えた場合に、どのようにして収集することが後の作業を効率的なものにし得るかを模索する。

データ自体の不備、欠陥は、そのデータを収集する段階で発生する。つき詰めて言えば、調査対象者が調査票に記入する段階で発生する。よってデータの不備、欠陥を防ぐためには、調査票を記入する段階で対策を講じておく必要がある。アンケートの記入に際しての注意事項で、詳細な指示を与えることが最も効果的な対策となるであろう。ただし、生活活動調査のように、アンケートの調査対象者への負担が大きい場合に、より詳細な指示を与えれば、さらに負担が増加し、アンケート回収率が低下するということも大いに考えられる。つまり、詳細な指示を与えることと調査対象者への負担を減らすことはトレードオフの関係にあると言える。そのため、要求・指示が過度なものにならないように配慮することは常に心がけておかななくてはならない。

具体的な指示・要求について考えていく。支援ツールを構築する際にデータの不備・欠陥が生じるのが、**Address-matching** プログラムを作成する段階であった。活動場所のレコードに主観的な表現が含まれたものがしばしば見受けられた。第II章でも触れ

たが、たとえば「〇〇町の友人宅」「××町の実家」「近くのスーパー」といった具合である。これだけの情報で、地図の上でその活動場所を特定するのはきわめて困難である。

よって、記載者本人にしかわからない表現を避けさせることが、最も効果的な対策といえる。理想的にはすべての活動場所の住所を書いてもらうことだが、これは現実的に不可能である。そこで、以下の要求・指示を提案する。

- ① 訪問先はフルネームで世帯主の名前を書く
- ② 活動場所は、知り得る限りの正式名称で書く
- ③ 市内での活動には先頭に「〇〇町」と町名をつける
- ④ 第三者でも活動場所がわかるように記入すること

①の指示により、活動が他の人の家への訪問である場合、はっきりとその活動場所が特定される。苗字だけの記入では候補が多数存在して絞りきれないので、フルネームで書いてもらうことで、大幅に候補を絞り込むことができる。②の指示によって、特にスーパーやコンビニなどでの買い物活動のケースで、どこで行われたのか正確に記入されるようになる。③の注意事項を加えると、調査対象者の負担がやや増えるが、同じ名称の候補が複数見つかった場合に見分けることが容易になり、Address-matchingプログラムの精度が向上する。④の指示は①から③をまとめたものであり、駄目押しの書くことで、調査対象者の具体的に書くことへの意識を定着させるものである。

## 2 結論と課題

本研究では生活活動分析のためのGIS支援ツールを開発することをその目的とした。とりわけこの支援ツールは、活動場所を空間的に非集計で分析できるようなプログラムが組み込めるように意図されたが、第3章で実際に開発した支援ツールはこの当初の目的を達成したと言える。従来では紙媒体の地図を用いて、逐次手作業で活動場所を探していく作業を行っていかねばならず、かなりの時間と労力を必要とした。しかしこの支援ツールを利用することによって、時間と労力を大幅に減らすことが可能である。とりわけ、Address-matchingにおけるその効果は大きい。この支援ツールを用いても、最終的なmatchingの判断は従来同様、研究者自身が行わなくてはならない



が、支援ツールに組み込まれた **Address-matching** のアルゴリズムはその判断を助ける際に必要な情報を、より簡便な方法で提供することができる。この支援ツールを用いることで、判断に要する時間と労力も削減することができるのである。

しかしながら、支援ツール自体にも課題はまだ残されている。この支援ツールは大きく3つの作業過程から構成されていたが、最後の描画プロシージャにおいては手作業が依然多くなっている。このプロシージャが「プログラム」と呼べる程度に、作業の多くが自動化されることが理想的である。これにはGISソフト **ArcView** の操作性の向上が望まれる。**ArcView** が **Microsoft** 社製ではないということもあるにせよ、**Access** や **Excel** などの併用する可能性の高いアプリケーションとの連携が、よりスムーズになることが望まれる。**Visual Basic** などの汎用的プログラムを利用することができるようになるだけでも、操作性は格段に向上する。こういったソフトウェア自体の向上を期待するには、より多くのユーザーが **ArcView** を利用し、ユーザーの要望を **ESRI** 社に反映させていくことも必要である。

また、支援ツール自体の精度を高めていく努力は、今後も必要である。コンピュータ技術は日々進歩しており、従来のコンピュータではできなかったような負担の重い作業がどんどんこなせるようになってきている。支援ツールもより操作性の優れたものを構築できるであろうし、精度も高めていくことができるであろう。

なかでも、支援ツールの精度に最も大きく関わってくる **Address-matching** プログラムのアルゴリズムは常に再検討していくことが必要である。アルゴリズム次第で同じ操作をしようとしても処理能力に大きな違いが生じてくる。たとえば、本研究の **Address-matching** プログラムでは、プログラミングの構成上、活動の用件によって候補地を絞り込むというプロセスを組み込めなかった。買い物は「スーパー」「店」などを含む語を絞り込むといったプロセスを踏めば、さらに支援ツールとしては精度の高いものになったであろう。ある特定の種類の用件と関わりの強い建物名称を予めリストアップしてテーブルにまとめておき、**Address-matching** の時にそのテーブルを参照するアルゴリズムを作成すれば克服できるであろう。

またこのプログラムにはより具体的な距離の概念を導入することができなかった。前後の活動場所からの移動距離がわかれば、活動場所の **Address-matching** がうまくいか

なくても、かなりの範囲で絞り込むことができる。そのために、地図データに距離を示すレコードが含まれている必要がある。たとえば、建物の絶対座標などが組み込まれていれば、座標を計算して一定の距離内の建物を検索することが可能になるであろう。これはデータの質を高めることで、ツールの精度を上げるという例である。いずれにしても、より簡便な方法を考案し続けるスタンスを忘れないことが必要である。

地理情報の分析手法は現在変わりつつある。今後はGISを利用した地理学の研究はますます増えていくと思われる。本研究はそのような過渡期に、GIS技術をいちはやく導入し、技術論を提示することで、移行がスムーズに行われることに貢献できれば幸いである。とりわけ、地理情報の収集法も含め、分析に際しての最適な地理情報データベースを構築する際の一助となることを期待したい。

本研究には、文部省科学研究費補助金（特定領域研究(B)(2)、研究代表者 高阪宏行、研究課題「人文地理学における空間データの構築、管理、分析手法の開発とその適用」、課題番号 10202205 および 基盤研究(C)(2)、研究代表者 荒井良雄、研究課題「日中都市における生活活動システムの時間地理学的比較研究」、課題番号 12680072）の一部を使用した。

## 注

- 1) 荒井・岡本・神谷・川口（1996）
- 2) 荒井（1995）
- 3) 建物や道路、鉄道路線などの地理情報のことで、それぞれポリゴン（多角形）、ライン（線）、ポイント（点）といった二次元情報を持つ。
- 4) ここでデータベースは「データテーブルを管理するアプリケーション本体」と定義し、データテーブルを「同一の構造を持つ複数のレコードを管理するファイル」と定義する。
- 5) ArcView内で地理情報が描画されるウィンドウのこと。
- 6) 他の「.shp」「.shx」「.sbx」「.sbn」はArcViewで管理する固有のファイル形式で、View画面での描画などの情報が格納されたファイルである。ちなみにArcViewは「.apr」という形式のファイルを包括的に利用する。これはプロジェクトと呼ばれ、ViewやTableなど、一切の操作をこのプロジェクトから行っている。
- 7) テーブル内の各レコードを識別するためのフィールドのことで、そのため主キーのフィー

ルド値は全て異なっていないてはならない。

- 8) このうち、母子家庭が5世帯、単身赴任のため平日には夫がいない世帯が2例ずつある。
- 9) 調査の詳細および結果の概要については岡本(1993)、荒井・岡本・神谷・川口(1996)。
- 10) 荒井(1995)にその詳細が述べられている。そこでは市販データベースソフト4th Dimensionを利用しているが、本研究に際して先述のMicrosoft Accessに移植した。
- 11) あるサイクルはどの個人によって行われ、いくつストップがあり、どのような場所で活動を行ったかなどが一意に特定できるようにする。
- 12) ここでのテーブルの結合とは、活動データテーブルとArcView上にある建物データテーブル(Address-matching時に利用した地図データテーブルとまったく同じものだが、AccessではなくArcView上で開かれているテーブルであるため、区別して「建物データテーブル」としている)の結合を意味する。
- 13) ArcViewは1対nのデータ構造にも対応してはいるが、操作が困難となり、また描画を行う上で柔軟性に欠けてしまう。本研究では汎用の支援ツールを作成するという目的からも、多少手間はかかるが、より操作が簡便であり、柔軟性に富む1対1のデータ構造を採用することにした。
- 14) 88752件の建物が地図データには収められている。その1つ1つを識別するために、各建物に1から88752の数字が割り当ててある。建物データの主キーとなる。
- 15) 本研究ではアンケート調査が1991年に、地図の作成が1995年に行われており、4年間のタイムラグが生じているため、同一の住所で世帯主名が異なるケースが見られた。ここでは、アンケート調査時のサンプルの自宅が復元できれば世帯主名が異なっても問題ではないので、住所番号さえ見つければ、そこをサンプルの自宅として採用した。また、まったくその住所の建物が見つからないケースでは、その住所に最も近い数字の住所をサンプルの住所として見立てることにした。
- 16) 自宅のAddress-matchingは複雑な構造をとらないため、ここでは省略する。
- 17) 読み込んだデータの文字全てが一致するかどうかを検索すること。ヒット率は非常に低い。ヒットすれば即座に一意に決まる利点がある。
- 18) 読み込んだデータの文字の一部が一致するかどうかを検索すること。ヒット率は高くなるが、即座には一意に決まらないので、さらに絞り込んで検索を進める必要がある。「\*文字列\*」のようにして検索する。「\*」の文字は任意の文字を意味するため、この文字で検索したい文字列をはさめば、部分的に含んでいるレコードが検索される。
- 19) ユーザーが指示することで柔軟性を補うようにする。たとえば「角栄ストア」という表記が活動データにあるが、実際の建物データでは「カクエーストアー」となっている。こういったケースでは、コンピューターにただ文字列を読み込ませて検索するだけでは満足いく結果が得られない。
- 20) 実際、徒歩、自転車による移動は自宅から500m以内で214件中141件が行われている。

1km 以内であれば 180 件と実に 9 割近くに及ぶ。

- 21) 活動場所の情報がまったく決め手にかかる場合はユーザーの独断に委ねるしかない。
- 22) そもそも Access ではテーブルでデータを全て管理しているが、さまざまな処理を行う上で、テーブルをそのまま利用するのは不便である。そこでフォームというデータ処理用の画面を作成し、そこでさまざまなデータ処理を行う。フォーム上で行った処理は参照しているテーブルに反映されることになる。
- 23) Address-matching を行う Access ではフォーム上のコマンドボタンなどのコードは全て VBA で書かれている。
- 24) 活動データテーブルのレコードを 1 から順番に数字を割り振っていったもの。
- 25) 具体的なフィールド構成については II 章を参照。
- 26) このクリックの順番を逆にすると、建物データテーブルが活動データテーブルに付加される形になり、ここでの描画に適さなくなる。
- 27) 結合前の活動データテーブルのレコード数と同じ数選択されることになる。
- 28) ArcView 固有のファイル形式で、テーマの描画やデータベースファイルなどを一元管理するファイル。シェープファイルを作成することで、テーマが作成される。
- 29) 川越市市外のデータは当然ながら川越市のゼンリン地図には載っていない、そのため建物データにも含まれていない。よってここでは全て“0”を割り当てて市外としている。
- 30) 活動の直前の移動手段を集計したため、Ⅲ帯で徒歩が含まれるのは、自宅からの主要な移動手段手段以外のものがふくまれているためである。
- 31) 詳細な定義は、リンチ、K. (1968) 参照。

## 文献

- 荒井良雄 1993. 生活活動空間の構造と変容に関する研究. 東京大学工学部博士論文.
- 荒井良雄 1995. 生活活動の時間地理学的分析のためのデータベースシステムの開発. 東京大学教養学部人文科学科紀要 101 (人文地理学): 15-40.
- 荒井良雄・岡本耕平・神谷浩夫・川口太郎 1996. 『都市の空間と時間』古今書院.
- 碓井照子 1995. GIS 研究の系譜と位相空間概念. 人文地理 47: 562-584.
- 岡本耕平 1985. 名古屋市における住民の個人特性と外出行動パターンとの関係—社会地域構造との関連を中心に—. 人文地理 37(6): 513-532.
- 岡本耕平 1993. 日本の都市住民の生活空間と生活時間: 資料. 東洋大学社会学部紀要 30(3): 50-120.
- 岡本耕平 1995. 大都市圏郊外住民の日常活動と都市のデイリー・リズム—埼玉県川越市および愛知県日進市の事例—. 地理学評論 68A: 1-26.
- 神谷浩夫・岡本耕平・荒井良雄・川口太郎 1990. 長野県下諏訪町における既婚女性の

- 就業に関する時間地理学的分析. 地理学評論 63A : 766-783.
- 川口太郎・神谷浩夫 1991. 都市における生活行動研究の視点. 人文地理 43 : 348-367.
- 佐藤俊雄 1998. 『マーケティング地理学』同文館.
- リンチ, K., 丹下健三訳 1968. 『都市のイメージ』岩波書店.
- リンチ, K., 三村翰弘訳 1984. 『居住環境の計画—すぐれた都市形態の理論—』彰国社.
- Baker, R.G.V. 1985. A Dynamic Model of Spatial Behaviour to A Planned Suburban Shopping Center. *Geographical Analysis* 17(4): 331-342.
- Hanson, S. 1980. Spatial Diversification and Multipurpose Travel: Implications for Choice Theory. *Geographical Analysis* 12(3): 245-257.
- Hanson, S., and Hanson, P. 1981. The Travel Activity Patterns of Urban Residents: Dimensions and Relationships to Socio-demographic Characteristics. *Economic Geography* 57: 332-347.
- Howard Slavin 1996. An integrated, dynamic approach to travel demand forecasting. *Transportation* 23: 313-350.
- Huff, J.O., and Hanson, S. 1986. Repetition and Viability in Urban Travel. *Geographical Analysis* 18(2): 97-114.
- Kitamura, R. (1988): An Evaluation of Activity-Based Travel Analysis. *Transportation* 15: 9-34.
- Kondo, K., and Kitamura, R. 1987. Time-space Constraints and the Formulation of Trip Chains. *Regional Science and Urban Economics* 17:49-65.
- Peter R. Stopher, David T. Hartgen & Yuanjun Li 1996). SMART: simulation model for activities, research and travel. *Transportation* 23: 293-312.
- Ryuichi Kitamura, Eric I. Pas, Clarisse V. Lula, T. Keith Lawton & Paul E. Benson 1996. The sequenced activity mobility simulator (SAMS): an integrated approach to modeling transportation, land use and air quality.

*Komaba Studies in Human Geography* 14 35-80 2000

## A Geographical Information System for human activity analysis: a technical report

Yoshio ARAI\*, Seima OHKI\*\* and Tadahiko MUSHA\*\*

Recently, the environment of the study of human geography has changed with the progress of computer technology. Especially, the progress of GIS (Geographical In-

formation System) is so remarkable that it provides us a new possibility of the study of human geography.

On the ground of this progress, this paper aims at the development of geographical information system for human activity analysis. The methodology and the technology of human activity analysis which has hardly been mentioned in the past studies of human geography are mainly reported here.

The past human activity analysis has studied mainly about the activity paths. There were few studies that analyzed the paths on the detail maps. On account of the technical constraints, they had tentatively analyzed human activity spaces by summing up the locations of both ends of the paths along rather large zones and drawing the paths on schematic diagrams.

Therefore, we develop the geographical information system which can draw human activity spaces on the detail maps to introduce new possibility to human activity analysis.

The developed system works through the three stage procedures. The first stage is Address-matching Program, which matches the activity data with the map data in terms of address on the database. The second stage is Trip-chain Producing Procedure, which divides the activity table by trip-chain. The third stage is ArcView Drawing Procedure, which draws human activity spaces on the real map.

The precision of human activity spaces which the whole system can draw depends on the acuteness of Address-matching Program. Address-matching Program functions according to the two algorithms: the one that uses the names of buildings, the other one that uses the information of the previous activity and next one. The later can suggest the set of candidate locations of path ends. This program could well achieve address-matching, except the data without the defects caused by the data coding. Next, the system draws human activity spaces on the detail maps.

## 付録 Visual Basic のコード

Ⅲ章で作成した Address-matching プログラムの各フォームおよびトリップチェーン生成プログラムの VBA コードは以下の通りである。

### スタートフォーム

```
Option Compare Database
```

```
Private Sub Form_Load()  
End Sub
```

```
Private Sub 月ボタン_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
    DoCmd.OpenForm "AddressMatching", , , ""  
    Form_AddressMatching.SetMonday
```

```
End Sub
```

```
Private Sub 終了_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
    DoCmd.Close
```

```
End Sub
```

```
Private Sub 日ボタン_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
    DoCmd.OpenForm "AddressMatching", , , ""  
    Form_AddressMatching.SetSunday
```

```
End Sub
```

### Address-matching フォーム

```
Option Explicit  
Option Compare Database
```

```
Dim db As Database      'Database (Current)  
Dim rs As Recordset    'RecordSet (活動データ)  
Dim rs2 As Recordset   'RecordSet (大字)
```

```
Dim Target As String   '対象テーブル
```

```
Dim Acode, Acode1, Acode2  
Dim Chiban1, Chiban2 As String  
Dim EnableAcode, EnableChiban1, EnableChiban2 As Boolean
```

```
Private Sub Form_Close()
```

```
    rs2.Close
```

```

rs.Close
db.Close

End Sub

Private Sub Form_Open(Cancel As Integer)

Set db = CurrentDb()
Set rs2 = db.OpenRecordset("大字", dbOpenDynaset, dbReadOnly)

ArcView用建物データのサブフォーム.Form.Filter = "TATEMON_ID = 0"

ArcView用建物データのサブフォーム.Form.FilterOn = True

Target = "活動データ (月)"

End Sub

Public Sub SetMonday()

Target = "活動データ (月)"
AddressMatchFirst

End Sub

Public Sub SetSunday()

Target = "活動データ (日)"
AddressMatchFirst

End Sub

Private Sub AddressMatchFirst()

Set rs = db.OpenRecordset(Target, dbOpenDynaset, dbDenyWrite)

If rs.EOF = True Or rs.BOF = True Then
DoCmd.Close
Exit Sub
End If
rs.MoveFirst
SetView

End Sub

Private Sub AddressMatchNext()

First:
rs.MoveNext

If rs.EOF = True Then
MsgBox ("最後までマッチング処理を行いました")
DoCmd.Close
Exit Sub
End If

If IsNull(rs!TATEMON_ID) = False Then
GoTo First
End If

End Sub

```



```
Private Sub SetView()
```

```
    詳細データの表示
```

```
    [場所] = rs!場所
```

```
    [活動ID] = rs!活動ID
```

```
    [TATEMON_ID] = rs!TATEMON_ID
```

```
    [サイクルID] = rs![サイクルID]
```

```
    [ストップ番号] = rs![ストップ番号]
```

```
    個人番号 = rs!個人番号
```

```
    [注] = rs!注
```

```
    [世帯番号] = rs!世帯番号
```

```
    [前移動手段] = rs!前移動手段
```

```
    [後移動手段] = rs!後移動手段
```

```
    [前移動時間] = rs!前移動時間
```

```
    [後移動時間] = rs!後移動時間
```

```
    [注] = rs!注
```

```
End Sub
```

```
Private Sub MatchExecute()
```

```
    Dim Name As String
```

```
    Dim FilterString As String
```

```
    Dim i, length, ptr As Integer
```

```
    If IsNull([場所]) = True Then Exit Sub
```

```
    '場所文字列が「・」で区切られている場合はその最後の文字を抽出
```

```
    Name = [場所]
```

```
    length = Len(Name)
```

```
    ptr = 0
```

```
    For i = 1 To length - 1
```

```
        If Mid$(Name, i, 1) = "・" Then ptr = i
```

```
    Next i
```

```
    If ptr > 0 Then Name = Right$(Name, length - ptr)
```

```
    FilterString = "HOUSENAME LIKE '*' + Name + '*'"
```

```
    ArcView用建物データのサブフォーム.Form.Filter = FilterString
```

```
    ArcView用建物データのサブフォーム.Form.FilterOn = True
```

```
End Sub
```

```
Private Sub TATEMON_ID_LostFocus()
```

```
    rs.Edit
```

```
    rs!TATEMON_ID = [TATEMON_ID]
```

```
    rs.Update
```

```
End Sub
```

```
Private Sub 決定ボタン_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
    MatchExecute
```

```
End Sub
```

```
Private Sub 検索_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
    If IsNull([検索ID]) = True Then Exit Sub
```

```
    rs.FindFirst ("活動ID = " + Str$([検索ID]))
```

```
    If rs.NoMatch = True Then
```

```

    MsgBox ("指定IDは見つかりません")
    rs.MoveFirst
End If

SetView

End Sub

Private Sub 後回し_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)

    rs.MoveNext

    If rs.EOF = True Then
        MsgBox ("最後までマッチング処理を行いました")
        DoCmd.Close
        Exit Sub
    End If

    SetView

End Sub

Private Sub 取り込み_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)

    On Error GoTo error_exit

    ' 選択された建物IDをGET!
    TATEMON_ID = ArcView用建物データのサブフォーム.Form!TATEMON_ID
    TATEMON_ID.SetFocus

    AddressMatchNext
    On Error GoTo 0
    Exit Sub
error_exit:
    MsgBox "レコードを選択してください"

End Sub

Private Sub 地域_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)

    Dim FilterString As String

    SpecialMatch1
    SpecialMatch2

    If EnableAcode = True Then

        'Acodeでの絞り込み
        FilterString = "ACODE =" + Str$(Acode)

        'さらに・・・
        If EnableChiban1 = True And Acode = Acode1 Then
            If EnableChiban2 = True And Acode = Acode2 Then
                'Chiban1,Chiban2が有効
                FilterString = FilterString + "AND CHIBAN IN( '" + Chiban1 + "', '" + Chiban2 + "')"
            Else
                'Chiban1が有効
                FilterString = FilterString + " AND CHIBAN = '" + Chiban1 + "'"
            End If
        Else
            If EnableChiban2 = True And Acode = Acode2 Then

```

```

        'Chiban2が有効
        FilterString = FilterString + " AND CHIBAN =" + Chiban2 + """"
    End If
End If
Else
    If EnableChiban1 = True Then
        If EnableChiban2 = True Then
            'Chiban1,Chiban2が有効
            FilterString = "( ACODE =" + Str$(Acode1) + ", AND CHIBAN =" + Chiban1 + ",) OR (,
            FilterString = FilterString + " ACODE =" + Str$(Acode2) + " AND CHIBAN =" + Chiban2 + """)"
        Else
            'Chiban1が有効
            FilterString = " ACODE =" + Str$(Acode1) + " AND CHIBAN =" + Chiban1 + """"
        End If
    Else
        If EnableChiban2 = True Then
            'Chiban2が有効
            FilterString = " ACODE =" + Str$(Acode2) + " AND CHIBAN =" + Chiban2 + """"
        End If
        MsgBox ("地域の絞込みに有効なデータはありません！")
        Exit Sub
    End If

End If
ArcView用建物データのサブフォーム.Form.Filter = FilterString

End Sub

Private Sub SpecialMatch1()

    'Step1 大字テーブルの各文字列と [場所] を比較し、ACODEを取得
    Dim s1, s2 As String
    Dim i, n1, n2 As Integer

    Dim t As String

    Acode = 0
    EnableAcode = False
    s2 = [場所]

    If rs2.EOF = True And rs2.BOF = True Then Exit Sub
    rs2.MoveFirst

    Do While rs2.EOF = False

        s1 = rs2!大字名
        'もしも“大字”で始まるデータなら“大字”をカット
        If Left$(s1, 2) = "大字" Then s1 = Right$(s1, Len(s1) - 2)
        n1 = Len(s1)
        n2 = Len(s2)
        For i = 1 To n2 - n1 + 1
            t = Mid$(s2, i, n1)
            If t = s1 Then
                Acode = rs2!Acode
                EnableAcode = True
                Exit Sub
            End If
        Next
        rs2.MoveNext
    
```

```

Loop
End Sub

Private Sub SpecialMatch2()

    Dim i As Integer
    Dim Result, FilterString As String

    If IsNull([ストップ 番号]) = True Then
        i = 0
    Else
        i = Int([ストップ 番号])
    End If

    '前レコードの探索
    EnableChiban1 = False
    If i >= 2 And [前移動手段] <= 2 Then
        FilterString = "サイクルID=" + Str$([サイクルID])
        FilterString = FilterString + " AND ストップ 番号 =" + Str$(i - 1)
        Result = DLookup("TATEMON_ID", Target, FilterString)

        If IsNull(Result) = False Then
            '発見！！
            FilterString = "TATEMON_ID=" + Str$(Result)
            Acode1 = DLookup("ACODE", "ArcView用建物データ", FilterString)
            Chiban1 = DLookup("Chiban", "ArcView用建物データ", FilterString)
            If IsNull(Acode1) = False And IsNull(Chiban1) = False Then EnableChiban1 = True
        End If
    End If

    '後レコードの探索
    EnableChiban2 = False
    If i >= 1 And [後移動手段] <= 2 Then
        FilterString = "サイクルID=" + Str$([サイクルID])
        FilterString = FilterString + " AND ストップ 番号 =" + Str$(i + 1)
        Result = DLookup("TATEMON_ID", Target, FilterString)

        If IsNull(Result) = False Then
            '発見！！
            FilterString = "TATEMON_ID=" + Result
            Acode2 = DLookup("ACODE", "ArcView用建物データ", FilterString)
            Chiban2 = DLookup("Chiban", "ArcView用建物データ", FilterString)
            If IsNull(Acode2) = False And IsNull(Chiban2) = False Then EnableChiban2 = True
        End If
    End If

End Sub

Private Sub 注_LostFocus()
    rs.Edit
    rs!注 = [注]
    rs.Update
End Sub

Private Sub 未割り当て_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    AddressMatchNext
    SetView

```

End Sub

Private Sub 基本台帳View\_Click()

```
Dim FilterString As String
If Len([世帯番号]) <= 0 Then Exit Sub
FilterString = "世帯番号=" + Str$([世帯番号])
DoCmd.OpenForm "基本台帳", , , FilterString
```

End Sub

## 基本台帳フォーム

Option Compare Database

Private Sub Form\_Current()

コンボ38 = 世帯番号

End Sub

Private Sub コンボ38\_AfterUpdate()

コントロールの値と一致するレコードを検索する

Dim rs As Object

Set rs = Me.Recordset.Clone

rs.FindFirst "[世帯番号]=" & Str(Me![コンボ38])

Me.Bookmark = rs.Bookmark

End Sub

## トリップチェーン生成プログラム

Sub TripChain()

Dim l As Integer

Dim k As Integer

Dim Fid As Object

Workbooks("活動データ(日).DBF").Activate

Cells.Select

Selection.Sort Key1:=Range("D2"), Order1:=xlAscending, Header:=xlGuess, \_  
OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom, SortMethod \_  
:=xlPinYin

Workbooks.Add

ActiveWorkbook.SaveAs Filename:="トリップチェーン.xls"

l = 2

Do

Workbooks("トリップチェーン.xls").Activate

Sheets.Add

Range(,"A1").Select

ActiveCell.FormulaR1C1 = "世帯番号"

Range(,"B1").Select

```

ActiveCell.FormulaR1C1 = "活動ID"
Range(,C1).Select
ActiveCell.FormulaR1C1 = "個人番号"
Range(,D1).Select
ActiveCell.FormulaR1C1 = "サイクルID"
Range(,E1).Select
ActiveCell.FormulaR1C1 = "ストップ 番号"
Range(,F1).Select
ActiveCell.FormulaR1C1 = "TATEMON_ID"
Range(,G1).Select
ActiveCell.FormulaR1C1 = "場所"

k = 1
Windows("活動データ (日).DBF").Activate

Do While Cells(l + k, 4) = Cells(l, 4)

    k = k + 1

Loop

Range(Cells(l, 1), Cells(l + k - 1, 7)).Copy _
Destination:=Workbooks("トリップチェーン.xls").ActiveSheet.Range("A2")

With Workbooks("基本台帳.xls").ActiveSheet.Range("A:A")
    Set Fid = .Find(Workbooks("トリップチェーン.xls"). _
ActiveSheet.Range("A2").Value)
End With

    Windows("基本台帳.xls").Activate
    Workbooks("トリップチェーン.xls").ActiveSheet.Cells(k + 2, 1) = _
Fid.Offset(0, 1).Range(,A1).Value
    Workbooks("トリップチェーン.xls").ActiveSheet.Cells(k + 2, 6) = _
Fid.Offset(0, 5).Range("A1").Value

    Workbooks("トリップチェーン.xls").ActiveSheet.Name = _
"c" + Str(Workbooks("トリップチェーン.xls").ActiveSheet.Range("D2"))

    l = l + k

Loop Until Windows("活動データ (日).DBF").ActiveSheet.Cells(l, 4).Value = ""

End Sub

```