

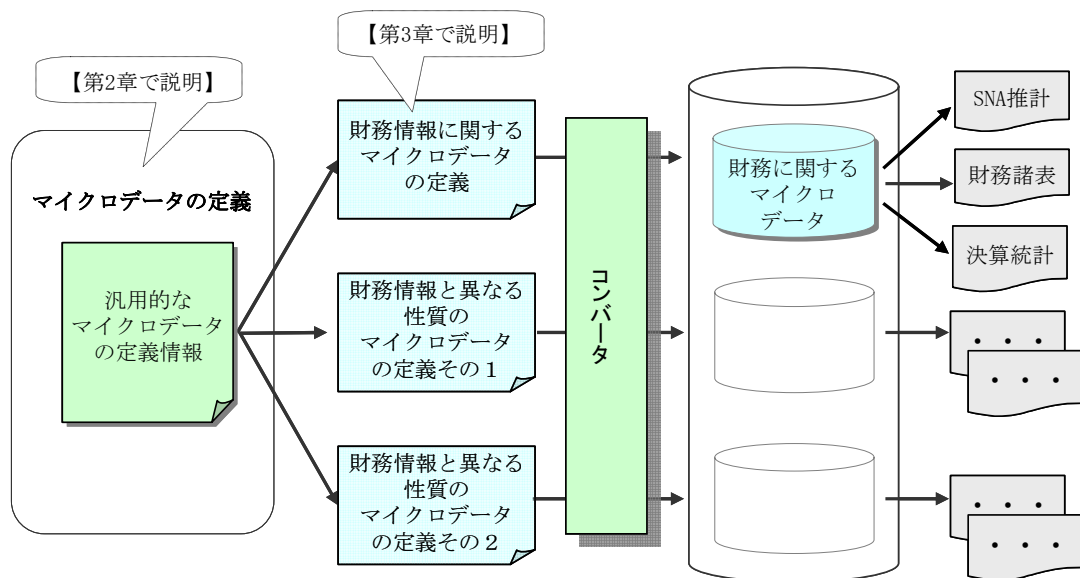
## 第2章 マイクロデータ

### 2.1 マイクロデータの仕様

第3章で述べる実証実験においては、地方公共団体の財務会計データを素材としてSNA推計、財務諸表、決算統計といった3つの異なった表章を行うことを実証しているが、本章で定義するマイクロデータの仕様はこの範囲に留まるものではなく、汎用的・普遍的な意味合いを持ち、本調査研究の最終的な目標である「マイクロデータから全ての統計諸表を軽快に表章する」というあらゆる表章において適用可能であることが求められる。

定義される内容はマイクロデータの生成、表章のプロセスで標準的に利用できるような特定のシステム開発ベンダーやソフトウェアに依存しない構造になっている必要があり、また、表章を行う際に使用される項目や内容は、常に変化が発生することが想定されるため、変化が発生してもマイクロデータの定義を変更することなく同一の定義で柔軟に対応する必要がある。

図 2-1：マイクロデータの仕様



汎用的・普遍的なマイクロデータの仕様とは何か、以下に利用者の視点に立って述べる。

## 2.2 属性データ管理

それぞれの表章用途にあったマイクロデータ的设计がされた際に、利用者やシステムが識別可能な状態で構造を定義する仕組みが必要となる。そのため、汎用的な構造となるよう以下の項目を利用してその構造を定義する。

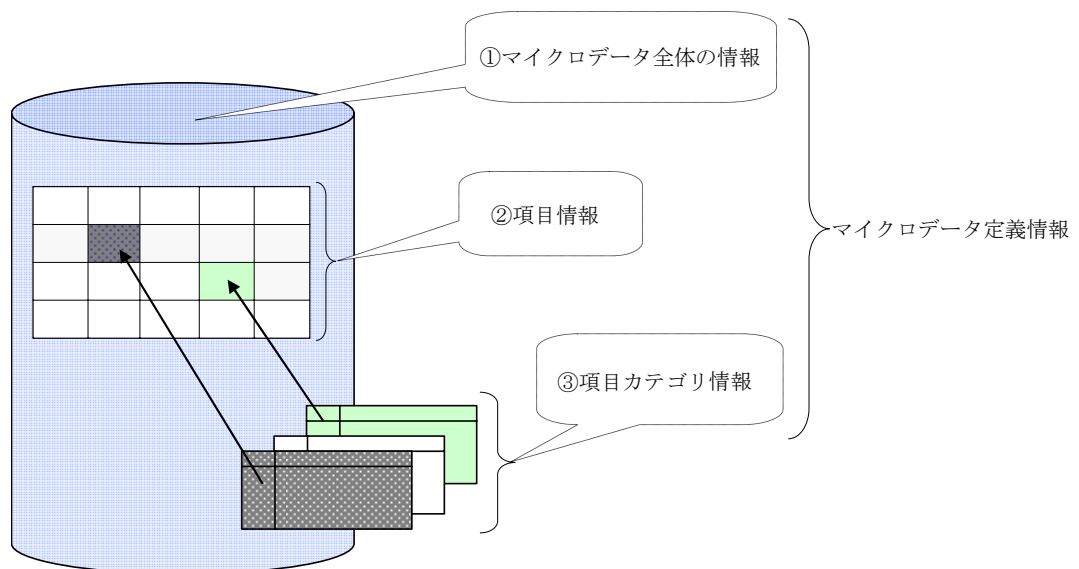
### 2.2.1 定義手法

マイクロデータの構造を定義する手法として、XML(eXtensible Markup Language)を採用する。XMLを採用した背景としては、OS やソフトウェア、言語、特定のシステム開発ベンダーに依存することがないためである。また、XML における周辺技術は豊富に存在し、システムが容易に定義内容を解析できることも採用した要因の一つである。

### 2.2.2 定義される内容

マイクロデータの構造として定義される内容を図 2-2 に示す。

図 2-2 : マイクロデータ構造定義の関連図



① マイクロデータ全体の情報

名称	タグ名	内容
マイクロデータ定義	fileDesc	マイクロデータの定義情報
マイクロデータ名称	namingrule	マイクロデータの名称、命名規則
マイクロデータ形式	format	マイクロデータの管理形式を記述する 例) CSV 形式、XML 形式、RDB、 オブジェクト

② マイクロデータ内で管理される項目情報

マイクロデータに管理される項目数分定義が行われる。

名称	タグ名	内容
項目情報	item	項目情報
項目名	name	項目の名称
項目型	type	データの種別、文字列、数値、日付、 コード等
項目長	length	最大バイト数
キー属性	index	キー項目であることを示す
省略属性	omit	省略可能かを示す。デフォルト値を設定
項目位置	location	項目位置
単位情報	unit	単位情報
カテゴリ情報	categoryid	カテゴリ情報へのリンク情報

③ マイクロデータ内で管理される項目カテゴリ情報

項目ごとに付随するカテゴリ情報を管理する。

カテゴリ情報とは、マイクロデータ内で管理される項目と表章用の項目間のマッピング情報など分類を管理する情報である。

名称	タグ名	内容
カテゴリ情報	category	カテゴリ情報
カテゴリ名称	name	カテゴリ名称
項目情報	item	項目情報
マッピング元情報	code	マッピング元の値
マッピング先情報	value	マッピング後の値

#### ④ スキーマ情報

以下に、XMLSchema を使用して定義を行った属性情報に関するスキーマ情報を示す。

```
<?xml version="1.0" encoding="shift_jis"?>
<xs:schemaxmlns="http://tempuri.org/fileDesc"
elementFormDefault="qualified"
targetNamespace="http://tempuri.org/fileDesc"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="fileDesc">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" ref="namingrule" />
        <xs:element ref="format" />
        <xs:element minOccurs="1" maxOccurs="unbounded"
ref="record" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="namingrule" type="xs:string" />
  <xs:element name="format" type="xs:string" />
  <xs:element name="record">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="unbounded" ref="item"
/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="item">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="name" />
        <xs:element ref="type" />
        <xs:element ref="length" />
        <xs:element minOccurs="0" maxOccurs="1" ref="index" />
        <xs:element minOccurs="0" maxOccurs="1" ref="omit" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

<xs:element minOccurs="0" maxOccurs="1" ref="location" />
<xs:element minOccurs="0" maxOccurs="1" ref="unit" />
<xs:element minOccurs="0" maxOccurs="1" ref="category" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="name" type="xs:string" />
<xs:element name="type" type="xs:string" />
<xs:element name="length" type="xs:string" />
<xs:element name="index" type="xs:string" />
<xs:element name="omit" type="xs:string" />
<xs:element name="location" type="xs:string" />
<xs:element name="unit" type="xs:string" />
<xs:element name="category" type="xs:string" />
</xs:schema>

```

#### ⑤ 同値交換方式によるマイクロデータの管理

従来のデータ管理方式では、同一のマイクロデータから目的の異なった複数の表章結果を得るためマイクロデータに含まれる一番細かな発生単位のデータ（以下、「レコード」と呼ぶ。）に対して、それぞれ、異なった数値情報をあらかじめ付加するという方式を採用していた。

このような構造の問題としては、同一のマイクロデータから表章すべき対象が多くなればなるほど、それぞれの数値を細分化してレコードに追加する必要があり、レコードが非常に大きなものになってしまうということがあげられる。

また、データの入力時にあらかじめ表章される全ての数値情報が把握できていなければ、データ入力することはできない。さらには、後で表章する内容が追加になった場合は全てのレコードに数値情報を付加するということになり、業務遂行的にもシステムの的にも現実的ではない。

例)

第3章でも詳細に触れるが、財務会計データを例に考えると取引自体は1つであるが、表章時には異なった数値で考える場面がある。

- (A) 財務会計的には、100万円で、作業を委託した。
- (B) 決算統計的の性質別に見ると、70万円の人件費と、30万円の物件費
- (C) SNA推計的には、80万円の消費と、20万円の資産

キー情報	A表章		B表章		C表章		D表章
伝票情報	委託	100万	人件費	70万	資産	56万	
					消費	14万	
			物品費	30万	資産	24万	
					消費	6万	

表章が増えれば、さらに、属性の付加、データの細分化が必要になる。

この問題を解決するため、既に発生しているデータに対して、属性情報の変換を行う個票単位のレコードを追加することで属性の変換を行う。ただし、単純に新たな属性情報を追加すると数値が重複されて計算されてしまう為、既に発生しているデータの相殺情報を追加する。つまり、同値の値を交換する方式により、あらかじめ定義されていない属性に関してもその後の変更が可能になる構造をとるように定義した。

図 2-3 に同値交換方式で管理されたマイクロデータのイメージを示す。

図 2-3 : 同値交換方式で管理されたマイクロデータ

【A表章】

キー情報	属性	金額
伝票情報	委託費	100万

} 表章される値

【B表章】

キー情報	属性	金額
伝票情報	委託費	100万
相殺情報	委託費	-100万
伝票情報	人件費	70万
伝票情報	物品費	30万

} 相殺され、0となる

} 表章される値

【C表章】

キー情報	属性	金額
伝票情報	委託費	100万
相殺情報	委託費	-100万
伝票情報	資産	80万
伝票情報	消費	20万

} 相殺され、0となる

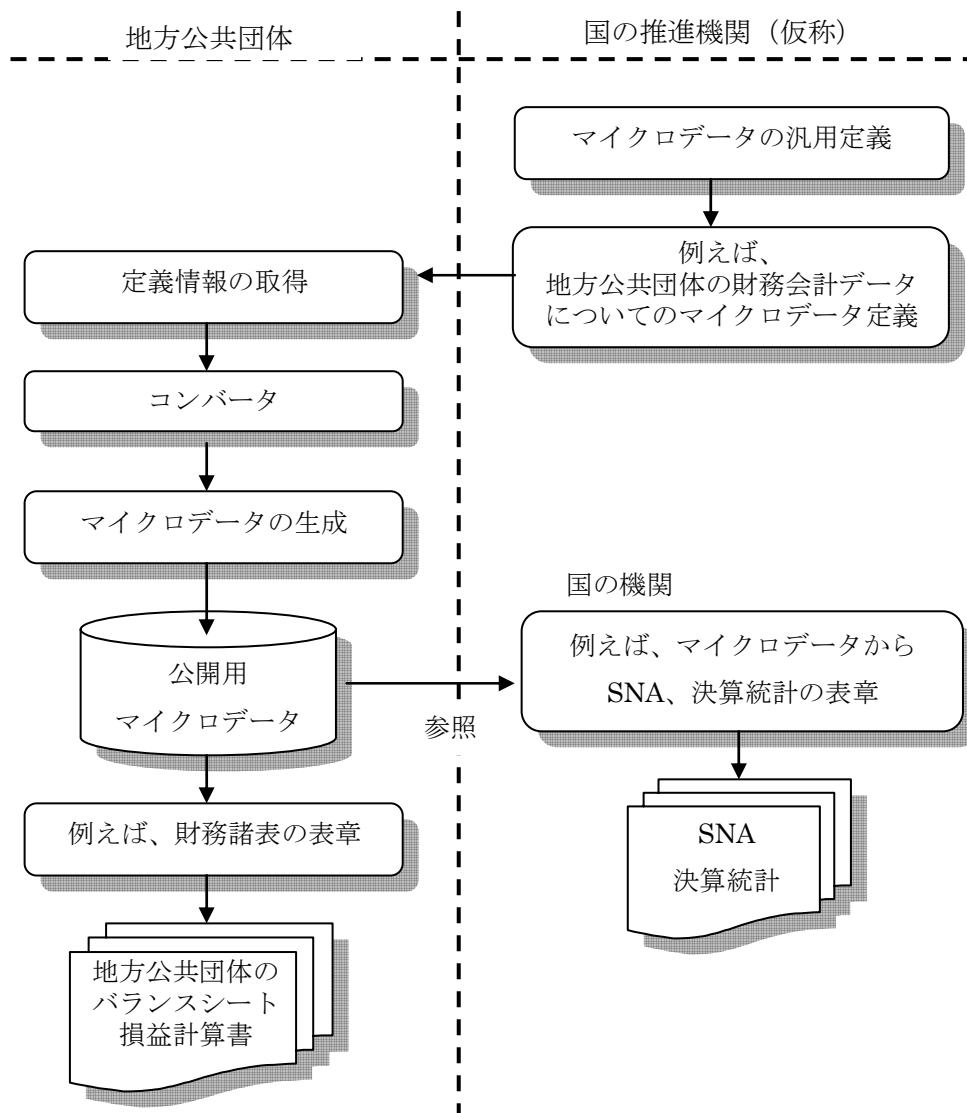
} 表章される値

※同値の相殺情報を発生させることで、属性情報を変更させる。  
※同値交換方式を使用した詳細な例は第3章で述べる。

### 2.3 マイクロデータの生成プロセス

地方公共団体を主とするデータの提供者（管理者）は、あらかじめ標準化されたマイクロデータの定義情報を取得し、定義にあわせてコンバータを作成もしくはシステムへの組み込み等その他の方法によりマイクロデータを生成し、公開用のマイクロデータとして登録する。

図 2-4 : マイクロデータの生成プロセス



### 第3章 プロトタイプによる実証実験の報告

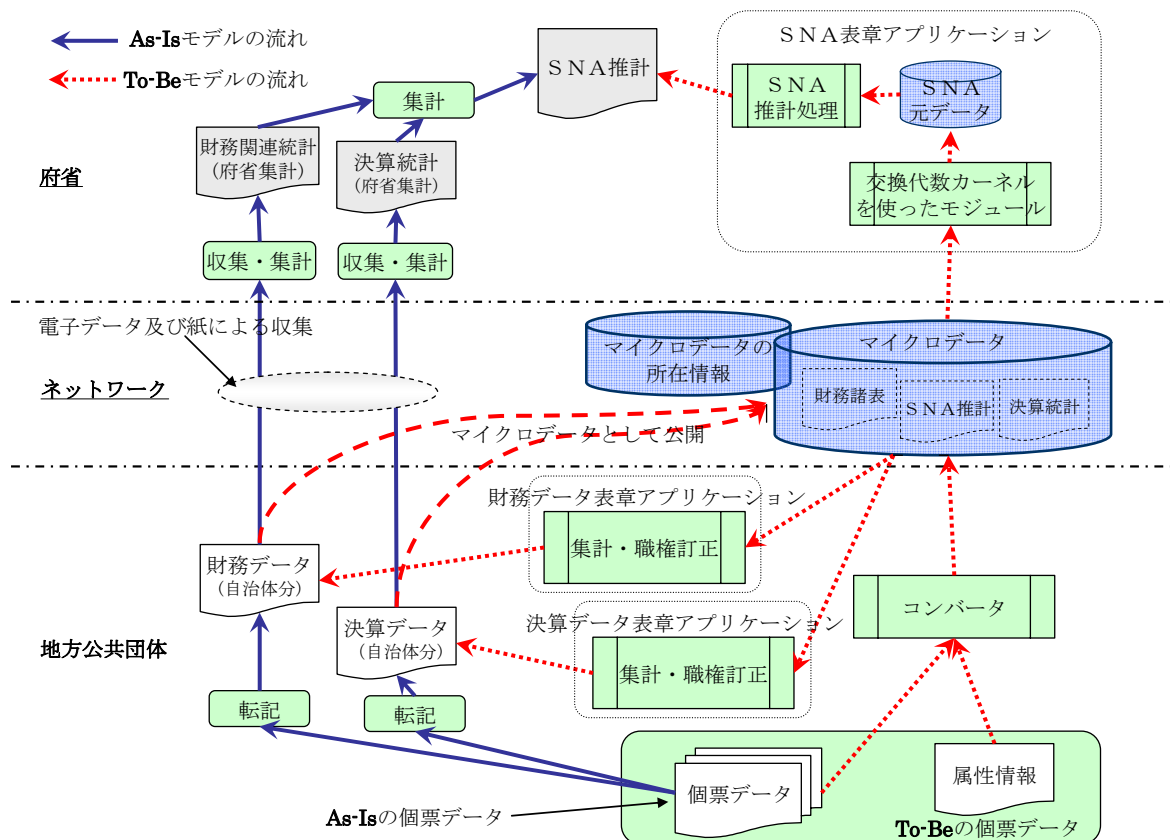
#### 3.1 実証実験の内容

##### 3.1.1 取り組んだテーマ

現在の表章の流れでは、図 3-1 の As-Is モデルのように国の機関から地方公共団体に向けて表章の単位に調査依頼がある。依頼を請けた地方公共団体ではその依頼ごとに報告書を作成することとなり、この作業が地方公共団体にかかなりの負担となっている。さらに、SNA 統計では決算統計など国の機関の統計調査の結果（一次統計）を元に表章する必要があり、タイムリーな集計は望めず、また、一次統計のデータにその正確性が左右されている。

本調査研究では、調査依頼を請けた地方公共団体に与える負担低減及びタイムリーな表章を実現することを推し進めるために、図 3-1 の To-Be モデルのように地方公共団体の個票データをマイクロデータとして公開し、このマイクロデータから各種の表章を行う方式の可能性について調査研究した。

図 3-1：表章の流れにおける現状とあるべき姿





### 3.1.2 実証実験の範囲

#### (1) 実証実験における検証の範囲

今回の実証実験では、研究テーマの中核である「マイクロデータから複数種の表章が可能である」ことを検証することを目的とした。その目的に則り、実在する地方公共団体の財務会計データからマイクロデータを生成し、そのマイクロデータを元にSNA推計、決算統計、財務諸表を表章するプロトタイプを作成し、下記の観点で検証を行なった。

- ① 地方公共団体のデータからマイクロデータを作成するプロセスの検証。
- ② マイクロデータから、SNA推計、決算統計、財務諸表が表章できることの検証。
- ③ 推計や統計の計算アルゴリズムや表章する科目の変換・変更に対して柔軟に対応できるアプリケーションアーキテクチャの検証。

#### (2) 実証実験における前提事項

今回設定した検証項目を絞り込むために以下の前提にて実証実験を実施した。

- ① マイクロデータの元となるデータ  
実在する地方公共団体から借用した実際の歳入データ、歳出データを使用する。
- ② マイクロデータの物理的な形式  
実証実験におけるマイクロデータは、CSVファイルとして実装する。
- ③ マイクロデータ作成に関する前提事項  
借用したデータ(CSV形式)から、個人情報などを削除し各表章に最低限必要な項目(列)のみを抽出して手作業によりマイクロデータ化を実施する。
- ④ 表章アプリケーション  
マイクロデータから各表章の整形前のデータを出力するアプリケーションと、表章への整形を行うアプリケーションを表章ごとに作成する。なお、整形前のデータを出力するアプリケーションは、推計や統計の計算アルゴリズムや表章する科目の変換・変更に対して柔軟に対応できることを目的として、交換代数カーネル<sup>4</sup>を用いることとする。
- ⑤ 表章対象の帳票  
表章する対象は下記の3帳票とし、表章可能な科目のみを表章する。  
イ) SNA推計において表章する帳票

<sup>4</sup> 東京工業大学大学院 総合理工学研究科の所有する交換代数カーネル。

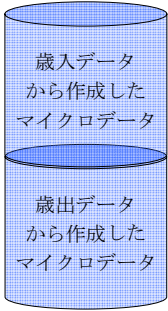
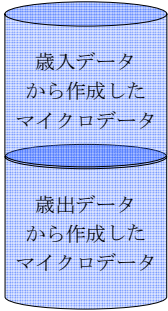
「国民経済年報 D付表 06 一般政府の部門別勘定」において、入手したデータで表章できる科目を対象とする。

- ロ) 決算統計において表章する帳票  
「決算統計年報 1-3-3 表平成 16 年度団体別歳入決算」
- ハ) 財務諸表において表章する帳票  
「地方公共団体のバランスシート」

⑥ 表章結果の検証

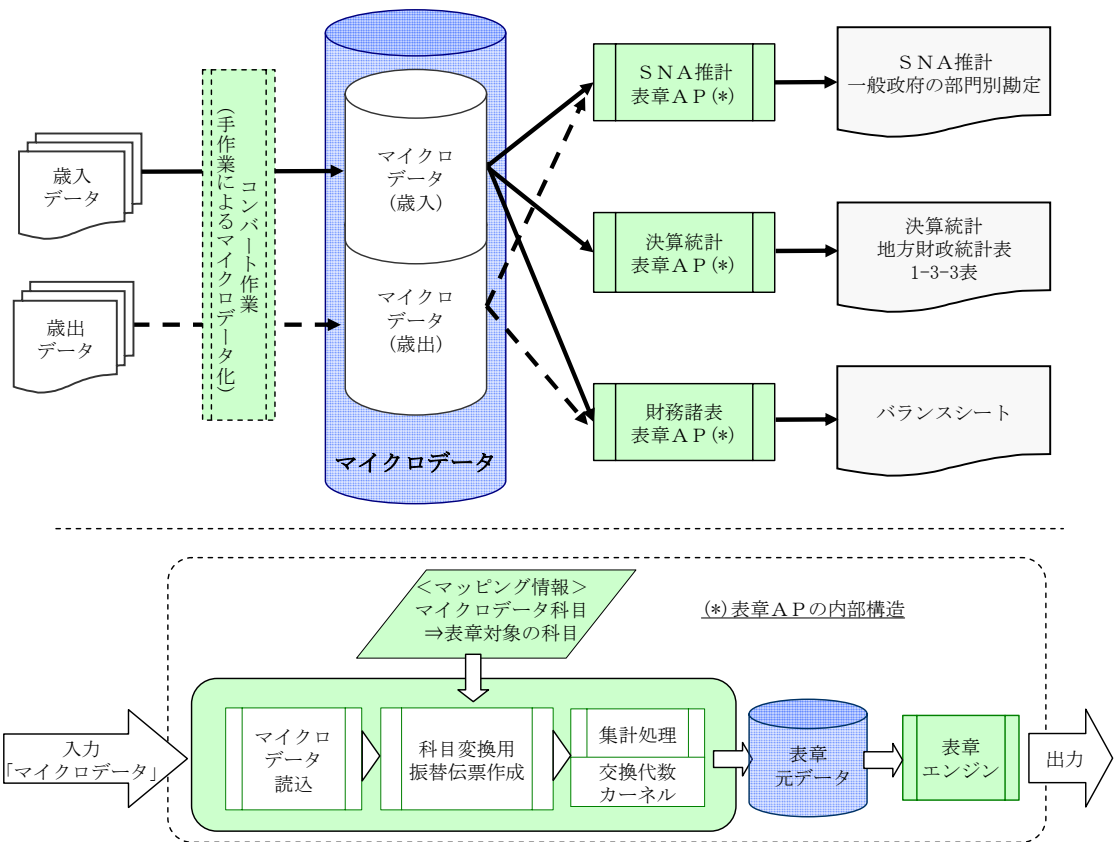
- イ) SNA 推計において表章する帳票  
借用したデータをもとに、机上にて計算した表章対象の科目ごとの集計値と突き合わせを行い検証する。
- ロ) 決算統計において表章する帳票  
借用した「平成 17 年度一般会計歳入歳出決算書」の内容と突き合わせを行い検証する。
- ハ) 財務諸表において表章する帳票  
公表されている地方公共団体の実際の「平成 17 年度バランスシート」の内容と突き合わせを行い検証する。

表 3-2 : データ・帳票関連表

入手データ	コンバート処理	マイクロデータ	表章AP		出力表章	突合データ候補有無
歳入データ	<ul style="list-style-type: none"> <li>・個人情報の削除</li> <li>・属性の追加</li> </ul>		<ul style="list-style-type: none"> <li>・表章元データ作成AP</li> </ul>	<ul style="list-style-type: none"> <li>・SNA用表章AP</li> </ul>	SNA推計 「一般政府の部門別勘定」 (*)	机上計算結果との突合
				<ul style="list-style-type: none"> <li>・決算統計用表章AP</li> </ul>	決算統計 「地方財政統計年報 1-3-3表平成16年度団体別歳入決算」	公表済み平成17年度一般会計歳入歳出決算書
<ul style="list-style-type: none"> <li>・財務諸表表章AP</li> </ul>	財務諸表 「バランスシート」 (*)	机上計算結果との突合				
<ul style="list-style-type: none"> <li>・SNA用表章AP</li> </ul>	SNA推計 「一般政府の部門別勘定」 (*)			公表済み平成17年度一般会計歳入歳出決算書		
歳出データ	<ul style="list-style-type: none"> <li>・個人情報の削除</li> <li>・属性の追加</li> </ul>		<ul style="list-style-type: none"> <li>・財務諸表表章AP</li> </ul>		財務諸表 「バランスシート」 (*)	

(\*) : 歳入・歳出を1つの帳票として表章する。

図 3-3 : 処理モデル図



上記の観点で検証するために、実証実験では地方公共団体の実際の環境を用いた実験ではなく、データのみを借用してスタンドアロン環境にて実験を実施した。

### 3.1.3 実証実験環境

今回の実証実験では、マイクロデータから SNA 推計、決算統計、財務諸表が出力できることに主眼をおいた検証であるため、スタンドアロン環境にて実証実験を実施した。

実証実験環境を以下に列挙する。

#### (1) ハードウェア環境

VersaPro	
CPU	Intel® Pentium® M processor
クロック	1.20 GHz

	メモリ	764MB
	ハードディスク	40GB
	その他	12.1 型 TFT カラー液晶(XGA)、100BASE-TX/10BASE-T など

## (2) ソフトウェア環境

実行・開発共通		
	OS	Windows XP Professional Service Pack 2
	Java Version	JDK 1.6.0 – b105
	開発環境	Eclipse 3.2.0
	交換代数カーネル	2007/02/27 に東工大出口研究室より入手した版
	表章整形エンジン	市販製品を使用
	表章アプリケーション	交換代数カーネルを使用した、表章用元データ出力アプリケーション (今回開発)

## 3.2 実証実験用のマイクロデータ

### 3.2.1 地方公共団体の財務会計データの構成情報

「決算統計表」「財務諸表」「SNA 統計」を表章するため、借用した地方公共団体の歳入・歳出データから実証実験用マイクロデータを生成した。

実証実験用マイクロデータの構成情報を下表に示す。

ID	項目名称	形式	長さ	開始位置 (カラム数)
1	UUID	数値	16 バイト	1
2	伝票 No (UUID)	数値	16 バイト	2
3	地方団体コード	数値	6 バイト	3
4	年度	数値	4 バイト	4
5	補正仕訳名	文字	256 バイト	5
6	貸借名	文字	256 バイト	6
7	会計名	文字	256 バイト	7
8	科目名	文字	256 バイト	8
9	金額	数値	15 バイト	9

## 【XML 例】

上記の定義情報を実際に XML 表記で定義した内容を下記に示す。

```
<?xml version="1.0" encoding="UTF-8"?>
<mdcp:fileDesc xmlns:mdcp="http://www.cao.go.jp/mdcp">
  <mdcp:namingrule></mdcp:namingrule>
  <mdcp:format>CSV 形式</mdcp:format>
  <mdcp:record>
    <mdcp:item>
      <mdcp:name>UUID</mdcp:name>
      <mdcp:type>数値</mdcp:type>
      <mdcp:length>16</mdcp:length>
      <mdcp:index>1</mdcp:index>
      <mdcp:omit></mdcp:omit>
      <mdcp:location>1</mdcp:location>
      <mdcp:unit></mdcp:unit>
      <mdcp:category></mdcp:category>
    </mdcp:item>
    <mdcp:item>
      <mdcp:name>地方団体コード</mdcp:name>
      <mdcp:type>数値</mdcp:type>
      <mdcp:length>6</mdcp:length>
      <mdcp:index>1</mdcp:index>
      <mdcp:omit> </mdcp:omit>
      <mdcp:location>2</mdcp:location>
      <mdcp:unit> </mdcp:unit>
      <mdcp:category></mdcp:category>
    </mdcp:item><mdcp:item>
      <mdcp:name>年度</mdcp:name>
      <mdcp:type>数値</mdcp:type>
      <mdcp:length>4</mdcp:length>
      <mdcp:index>1</mdcp:index>
      <mdcp:omit> </mdcp:omit>
      <mdcp:location>3</mdcp:location>
      <mdcp:unit> </mdcp:unit>
      <mdcp:category></mdcp:category>
    </mdcp:item><mdcp:item>
```

```

    <mdcp:name>補正仕分名</mdcp:name>
    <mdcp:type>文字</mdcp:type>
    <mdcp:length>256</mdcp:length>
    <mdcp:index>1</mdcp:index>
    <mdcp:omit> </mdcp:omit>
    <mdcp:location>4</mdcp:location>
    <mdcp:unit></mdcp:unit>
    <mdcp:category></mdcp:category>
</mdcp:item><mdcp:item>
    <mdcp:name>貸借名</mdcp:name>
    <mdcp:type>文字</mdcp:type>
    <mdcp:length>256</mdcp:length>
    <mdcp:index>1</mdcp:index>
    <mdcp:omit> </mdcp:omit>
    <mdcp:location>5</mdcp:location>
    <mdcp:unit> </mdcp:unit>
    <mdcp:category> </mdcp:category>
</mdcp:item><mdcp:item>
    <mdcp:name>会計名</mdcp:name>
    <mdcp:type>文字</mdcp:type>
    <mdcp:length>256</mdcp:length>
    <mdcp:index>1</mdcp:index>
    <mdcp:omit> </mdcp:omit>
    <mdcp:location>6</mdcp:location>
    <mdcp:unit></mdcp:unit>
    <mdcp:category></mdcp:category>
</mdcp:item><mdcp:item>
    <mdcp:name>科目名</mdcp:name>
    <mdcp:type>文字</mdcp:type>
    <mdcp:length>256</mdcp:length>
    <mdcp:index>1</mdcp:index>
    <mdcp:omit> </mdcp:omit>
    <mdcp:location>7</mdcp:location>
    <mdcp:unit></mdcp:unit>
    <mdcp:category></mdcp:category>
</mdcp:item><mdcp:item>
    <mdcp:name>金額</mdcp:name>

```

```

        <mdcp:type>数値</mdcp:type>
        <mdcp:length>15</mdcp:length>
        <mdcp:index> </mdcp:index>
        <mdcp:omit> </mdcp:omit>
        <mdcp:location>8</mdcp:location>
        <mdcp:unit></mdcp:unit>
        <mdcp:category></mdcp:category>
    </mdcp:item>
</mdcp:record>
</mdcp:fileDesc>

```

### 3.2.2 地方公共団体の財務会計データ項目の詳細情報

実証実験データの項目の詳細情報を以下に示す。

#### (1) UUID

項目名称	UUID
概要	データを位置に一意に特定するユニークなコード
データサンプル	0000000000000001
備考	補正仕訳等を行う際に、相手の特定やデータのバージョン管理等を行う事を想定。

#### 【UUID】とは

世界中でユニークな 128bit 幅の 2 進数値である。UUID を作成した時間や作成に使用したマシンに装着されているネットワーク・カードの MAC アドレスなどを数値の一部に組み入れたりしている。イーサネットの MAC アドレスは（登録制なので）世界中でユニークであることが保証されているため、データを一意に識別することが可能となる。

#### (2) 伝票 No (UUID)

項目名称	伝票 No (UUID)
概要	伝票単位を表すコード
データサンプル	0000000000000001
備考	伝票を特定するための代表の UUID を使用して定義する。

(3) 地方団体コード

項目名称	地方団体コード
概要	公共団体を特定するコード
データサンプル	193658
備考	

【地方団体コード】とは

地方団体コードは、日本の地方公共団体（都道府県・市町村・特別区・行政区・一部事務組合等）につけられた、コード番号のことである。JIS 地名コード、地方自治体コード、都道府県コード、市町村コードなどとも呼ばれる。

(4) 年度

項目名称	年度
概要	時系列を表す年度
データサンプル	2007
備考	時系列データを何処まで細分化するかは検討が必要。

(5) 補正仕訳名称

項目名称	補正仕訳
概要	複数の補正仕訳が行われる際に、どの補正仕訳を採用するかどうかの名称
データサンプル	SNA
備考	アウトプットの種別によって異なる名称が設定される。

【補正仕訳処理】とは

既に存在するデータの修正、変更を行う際に使用する。科目の振り直し処理や変更を行う事が出来る。

この際に、異なる用途での振替伝票が発生している場合は、どの伝票を有効にするかどうか補正仕訳名称を使用して判定する。

例)

(A) 財務会計的には、100万円で、作業を委託した。

(B) 決算統計的の性質別に見ると、70万円の人件費と、30万円の物件費

(C) SNA推計的には、80万円の消費と、20万円の資産

※A、B、Cのどの数値を参照すべきか補正仕訳名称を使用して判定



(6) 貸借名

項目名称	貸借名
概要	借方、貸方を識別する名称
データサンプル	借方
備考	借方、貸方

(7) 会計名

項目名称	会計名
概要	会計の種別を特定するための名称
データサンプル	一般会計
備考	一般会計、特別会計といった異なる会計を管理する

(8) 科目名

項目名称	科目名称
概要	取引のあった科目を表す名称
データサンプル	地方税
備考	歳入/歳出名称、款名称、項名称、目名称、節名称、細節名称

(9) 金額

項目名称	金額
概要	実際金額
データサンプル	10000
備考	

※基本的にはマイナスの数値は入らない。今回、地方公共団体より借用した歳入・歳出のデータは、現金主義の単式簿記の管理であったため、マイナスの数値が存在した。

3.2.3 レコード情報単位について

実証実験データは、付加価値の発生時点もしくは取引が実際に発生した時点で記録を計上することが原則となる発生主義の原則に基づくデータである。また、実証実験データは、付加価値または取引が発生した場合、相手方勘定科目も発生することを原

則とする。付加価値または取引の発生と、実証実験データのレコード発生の関係を以下に示す。

(1) 借方／貸方の勘定科目が 1 対 1 で発生した場合

借方		貸方	
現金	1,000	自転車駐車場使用料	1,000

実証実験データ

No	貸借	科目	金額	伝票 No. (代表の UUID)
1	借	現金	1,000	1
2	貸	自転車駐車場使用料	1,000	1

(2) 借方／貸方の勘定科目が 1 対 N で発生した場合

借方		貸方	
現金	2,000	行政財産使用料 (建築物)	1,000
		行政財産使用料 (土地)	1,000

実証実験データ

No	貸借	科目	金額	伝票 No. (代表の UUID)
3	借	現金	2,000	3
4	貸	行政財産使用料(建築物)	1,000	3
5	貸	行政財産使用料(土地)	1,000	3

(3) 借方／貸方の勘定科目が N 対 1 で発生した場合

借方		貸方	
現金	500	自転車駐車場使用料	1,000
売掛金	500		

実証実験データ

No	貸借	科目	金額	伝票 No. (代表の UUID)
6	借	現金	500	6
7	借	売掛金	500	6
8	貸	自転車駐車場使用料	1,000	6