

# SICORE, THE INSEE AUTOMATIC CODING SYSTEM

Pierrette Schuhl  
Insee, France

## ABSTRACT

When processing survey data, the coding stage is becoming increasingly automated, since the productivity gains made possible are substantial. This is why automatic coding techniques are gradually introduced in most statistical institutes. At INSEE, France, automatic coding exists for more than 15 years through the QUID software. This method had been used in many applications since 1983. However it had some drawbacks and needed to be improved. Since 1993 a new software and a new management structure for the automatic coding are being developed : SICORE. This article attempts to describe this SICORE system (software and management structure) and presents results of its first applications.

## KEYWORDS

Automatic Coding, Knowledge Base, Language, Pattern Recognition, QUID, SICORE

## INTRODUCTION

At INSEE, the french national institute of statistics, automatic coding has been used for more than 10 years through the QUID software created by Jacques Lorigny in 1982. However it had some drawbacks. For examples : the lack of generality in the use of additional variables and the lack of rapidity with which the coding tree is built.

Furthermore, the used of these techniques showed the necessity for a structure to prepare and validate the knowledge bases (reference file, synonyms, empty words, logical rules to take into account additional variables, ...). No tools and no methodology were available for the analysis and the update of these knowledge bases in QUID.

The SICORE project, launched in 1993 by Pascal Rivière<sup>1</sup>, aimed to improve and generalize QUID concepts, with the intention to establish a general system for automatic coding.

The four main objectives were :

- Construct **the knowledge base** for some crucial variables such as Occupation and Town.
- Create an **adequate management structure** : experts for each variable, working groups, expert users group, ...
- Write a **generalized software package** able to handle the coding itself, the pre-coding (preparation of the knowledge bases) and the post-coding (analysis of the non-coded records). It had to be user-friendly and available for any variable and any language.
- Provide a **documented methodology**, to guide statisticians who want to incorporate automated coding when processing their survey data, to help them build and modify the knowledge bases.

The SICORE project will soon be completed, and the SICORE software and structure are already used by statisticians. It was used and applied on some surveys, in production or in test, in many configurations and for many wording types such as : country, region, town, occupation, name and address of establishment, human activity, place of vacation, financial product. Some expert sessions are created, as well as a methodology to update the knowledge bases.

In this paper we will describe the SICORE system : its architecture and its main characteristics. We will also present the main results of its first applications.

## 1. THE KNOWLEDGE BASE

Automatic coding requires a certain amount of information on each variable to be coded. Six kinds of informations (six files) are required :

- **The reference file**, or learning file, defines the correspondence between texts and codes; it is a *codeset*;
- **The normalization rules** defines how the text will be normalized before automatic coding : maximum number of words, maximum length of each word, empty characters, empty words (or groups of words), synonyms (words or groups of words);
- **The logical rules**. To code some variables, we need to take into account additional variables. In this file,

---

<sup>1</sup> Pascal Rivière was in charge of the SICORE project between 1993 and 1995.

decision tables are defined by sets of rules that take into account the values of additional variables. The structure of each rule is very simple :

*If  $x_1 \in A_1$  and  $x_2 \in A_2$  and ... and  $x_n \in A_n$  Then Code= $C$*

( $x_i$  are additional variables,  $A_i$  are sets of values, and  $C$  is either a final code or another decision table name)

- **The transcoding rules** : the values of additional variables in the survey can differ from the values that appear in the logical rules. Consequently, survey values have to be translated into values that can be understood by the logical rules. These translations are called transcoding rules;
- **The record layout of the file to be coded** defines the positions of the text and the additional variables in the file to be coded;
- **The parameters of the learning algorithm** : SICORE needs some parameters about the reference file (position of the text, position of the code, number and length of words retained), how words are split to build the coding tree, and also about the order based on which the word pieces are taken into account to build the coding tree.

These six files have been brought together in a unique file named *knowledge base*.

Automatic coding cannot be performed without knowledge on the variable to be coded. One knowledge base (six files) is needed for each variable (and for each language). It has to be elaborated by an expert on the variable and has to be regularly updated. That is why we need an adequate management structure.

## 2. HOW DOES THE SICORE SYSTEM WORK ?

Here we will just briefly describe how SICORE runs.

SICORE operates in two stages : first, a learning phase (using the knowledge base) and secondly, the coding phase itself.

## 2.1 THE LEARNING PHASE

SICORE first reads the learning parameters, normalization rules, logical rules, transcoding rules, record layouts and, of course, the reference file. It "compiles" the all these knowledge files : every knowledge file is loaded into an internal structure, which will be used for automatic codings.

In particular, the reference file is loaded into a tree structure, called *coding tree*, which facilitates and speeds up text recognitions.

This coding tree is built by the algorithm developed by Jacques Lorigny in 1982 and recently improved by Pascal Rivière. Two steps are required to build the tree : the normalization and the building of the tree.

- The normalization step takes the reference file as input. It reads the text and removes empty words and empty characters, replaces words (or groups of words) by their synonyms, limit the number of words. Finally, it split each word into pieces of 1 character (monograms) or 2 characters (bigrams) or 3 characters (trigrams) or 4 characters (quadrigrams), depending the parameter given. (see example below)

Example :

### "Occupation" Reference File Example

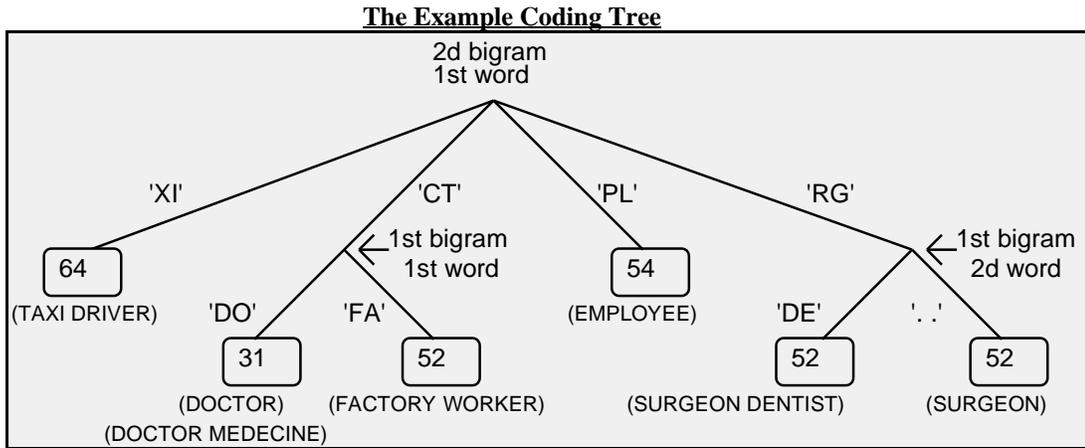
	TEXT	CODE
TAXI	DRIVER	64
DOCTOR		31
DOCTOR	OF MEDECIN	31
FACTORY	WORKER	52
EMPLOYEE		54
SURGEON		34
SURGEON	AND DENTIST	31

### Normalization and Truncation in Bigrams of Example Words

TAXI					DRIVER								
DOCTOR													
DOCTOR					MEDECINE								
FACTORY					WORKER								
EMPLOYEE													
SURGEON													
SURGEON					DENTIST								

- The second step takes, as input, the normalized reference file obtained in the first step. It computes the position of the word piece which gives the biggest amount of information (Shannon information) for the text recognition. Then, it builds all the branches which correspond to this position. For each branch, it computes again the second position given the biggest amount of information and builds the next branches. This process is repeated until each branch uniquely identify a code.

This computation gives the order based on which the word pieces are taken into account to build the coding tree. However, it is possible to force the program to build the coding tree beginning with a specific set of word pieces.



## 2.2 THE CODING PHASE

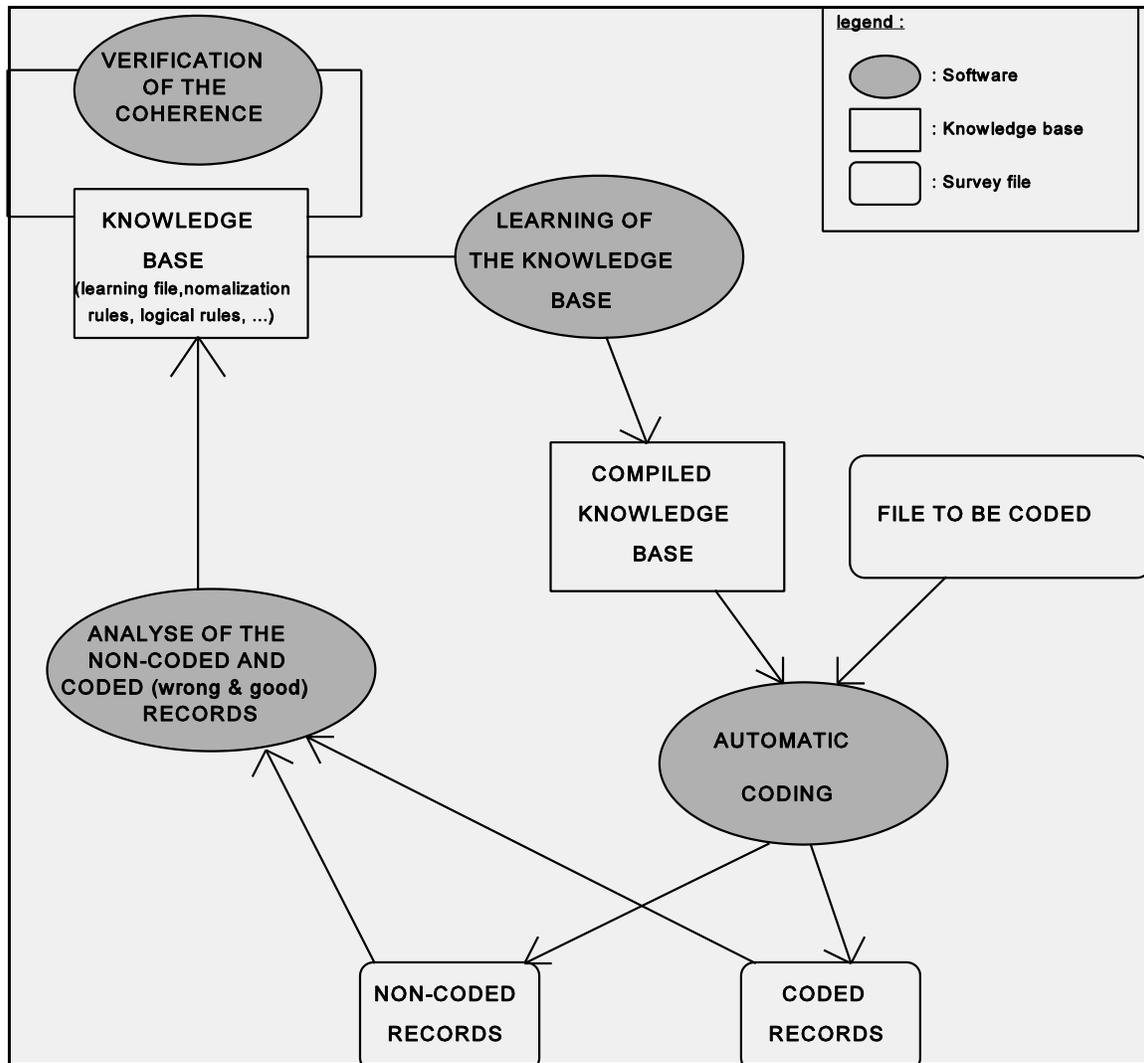
The coding phase is made of three steps : the normalization, the pattern recognition and the decision for partial success.

- Τηρ\_φιστ\_στεπ the normalization step takes the file to be coded as input and normalizes it like described previously for the reference file into the learning phase.
- Τηρ\_σεχονδ\_στεπ the pattern recognition algorithm, takes the normalized text as input and determines a code using the coding tree. There are three possible outcomes :
  - Failure : SICORE does not recognize the pattern of the text; it stops here;
  - Complete success : SICORE recognizes the pattern of the text and this text can be directly coded; it provides the code and stops here;
  - Partial success : SICORE recognizes the pattern of the text but, alone, this text is too ambiguous to be coded. SICORE then provides an intermediate code which is the name of a set of rules and the process continues with the third step.
- Φορ\_τηρ\_παρτιαλ\_συχχεσσ\_τηρ\_τηιρδ\_στεπ\_χονισισ\_ιν\_φινδινγ\_τηρ\_χοδε\_βψ\_αππλψινγ the logical rules to the additional variables available.

At the end, SICORE returns a diagnostic code to indicate if it recognized the text. If the text was recognized, the diagnostic code also determines if it was coded directly or with logical rules. Finally, SICORE evaluates how well the logical rules performed when they were used.

The SICORE system can be described as shown on the next graph.

## The SICORE system



### 3. AN ADEQUATE MANAGEMENT STRUCTURE

The SICORE system requires an adequate management structure for two main reasons : first, to insure that the knowledge base (six files) is updated regularly; secondly, to properly incorporate automatic coding in the processing of survey data or administrative data.

Concerning the first reason, we introduced for each important variable (Occupation and Town, at the beginning) the function of **variable-expert**. This expert has to elaborate the knowledge base and to regularly update it. This role is crucial for the quality and the efficiency of the coding.

But the variable-expert cannot decide all alone. The automatic coding users are entitled to know and "have their say on" how the expert decides to code. Then we had to create **working groups** including the variable-expert and some users representatives. These groups meet twice a year.

We also created an **expert users club**. The variable-experts meet together (as needed) in order to share their experiences.

The management of this structure (working groups, expert users club, variable-experts) is under the responsibility of an other expert, the **SICORE-expert**. For now, this person is a member of the statistical methodology division at INSEE.

The second reason, the incorporation of the automatic coding in a processing of survey data, also needs a specific structure, which is centered around the SICORE-expert too. The purpose of this structure is to ensure that all concerned parties join forces to attain the common goal. The three parties involved are : the survey (represented by a statistician and a computer engineer), SICORE (represented by the SICORE-expert and a computer engineer) and the variable to be coded (represented by the variable-expert).

Without this double structure it would be very difficult to insure coherence between many automatic codings of the same variable.

#### **4. THE DOCUMENTED METHODOLOGY**

One of the SICORE project main objectives was to provide a documented methodology. As of now, three documents have already been written :

- The users guide for the expert interface;
- A dictionary wich contain all the important words and concepts used by SICORE;
- The methodology guide, wich is the most important document. It describes how SICORE works, how to construct the knowledge bases and also how to verify the knowledge bases coherence.

We are also writting the programmer guide but it is not quite finished.

These documents are only available in french at the moment. They soon will be translated in english.

#### **5. SOME IMPORTANT RESULTS**

SICORE has already been tested with many variables : occupation (two and four digits), town, region, country, financial product, place of vacation, name and address of establishment, human activity. For some surveys, the automatic coding of the occupation variable (two digits) is in production.

To know if SICORE works well, three criterions have to be examined together :

- the *efficiency* : percentage of records that are automatically coded;
- the *accuracy* : percentage of coded records that are well coded;
- the *speed* : average time to code one record.

Efficiency can easily be computed but highly depends on the test file. Moreover, if the reference file is improved by adding the non-coded records of the test file, the efficiency increases in an artificial way; the user has to take care of that bias.

Accuracy is very difficult to obtain. To measure it, the whole file must be coded by an expert. Therefore, accuracy can only be approximated by taking a sample of the coded records and analysing it.

The third criterion is the easiest one, as it does not really depend on the file to be coded. Generally, a coding by SICORE does not exceed 0.1 millisecond (CPU time) per record on an IBM 3090. On a PC 486, it takes a little bit more (see table 2). To conclude, the coding algorithm is very fast and the speed criterion should not be taken into consideration to compare the results.

However, we have to mention the fact that coding must be preceded by the reading of the knowledge base, which includes learning the reference file. The time required for this step depends on the size of the reference file, and of the algorithm parameters. Excluding the variable "name & address of an establishment", one of the biggest reference files is the one of "town" (44000 lines); its learning takes less than 1 minute on a PC 486 DX2.

The reference file of "name & address of establishment" has more than four millions lines (4,600,000 lines). Learning that file requires a huge amount of memory : 1.8 gigabytes ! A test was recently made on a DEC  
α-σερϖερ\_\_ιτ\_τοοκ\_\_μινυτεσ\_το\_λεαρν\_τηε\_ωηολε\_φιλε\_ανδ\_τηε\_αϖεραγε\_χοδιγγ\_τιμε\_ωασ\_απ  
προξιματελψ\_\_\_\_μιλλισεχονδ\_περ\_ρεχορδ\_\_

Τηε\_μαιν\_χηαραχτεριστιχοσ\_οφ\_τηε\_διφφερεντ\_τεσσο\_ορ\_αππλιχατιονσ\_ωε\_μαδε\_αρε\_δεσχριβεδ\_ιν  
\_τηε\_ταβλεσ\_\_ανδ\_\_βελω\_\_

Τηε\_τιμε\_ρεθυιρεδ\_το\_ελαβορατε\_τηε\_κνωλεδγε\_βασε\_ισ\_ϖερψ\_διφφιχυλτ\_το\_εστιματε\_\_Ιτ\_δεπεν  
δο\_ον\_τηε\_ϖαριαβλε\_χομπλεξιτψ\_ανδ\_ον\_τηε\_αμουντ\_οφ\_κνωλεδγε\_ωε\_ηαϖε\_ατ\_τηε\_βεγιινγγ\_οφ  
\_τηε\_ωορκ\_\_Φορ\_εξαμπλεσ\_\_

- Ον\_ονε\_ηανδ\_\_φορ\_τηε\_εασιεστ\_ϖαριαβλε\_\_τηε\_φινανχιαλ\_προδυχτ\_ωε\_φυστ\_ηαϖε\_το\_φι  
νδ\_α\_φιλε\_χονταιινγγ\_τηε\_λιστ\_οφ\_φινανχιαλ\_προδυχτσ\_\_το\_ρεαρρανγγε\_ιτ\_αχχορδιγγ\_το\_τη  
ε\_ρεφερενχε\_φιλε\_φορματ\_\_το\_δεφινε\_τηε\_λεαρνινγγ\_παραμετεροσ\_ανδ\_σομε\_σψνονψμοσ\_\_Ιτ\_τ  
οοκοσ\_νο\_μορε\_τηαν\_τωο\_δαψσ\_\_Ιτ\_εσ\_ϖερψ\_εασψ\_το\_εστιματε\_τηε\_τιμε\_ρεθυιρεδ\_το\_ελαβορ  
ατε\_τηεσε\_κνωλεδγε\_βασε\_\_
- Ον\_τηε\_οτηερ\_ηανδ\_\_φορ\_τηε\_μοστ\_χομπλεξ\_ϖαριαβλε\_\_τηε\_οχχυπατιον\_\_ωε\_ρεχοϖερεδ\_αλ  
λ\_τηε\_ινφορματιον\_ελαβορατεδ\_σινχε\_τηε\_βεγιινγγ\_οφ\_τηε\_αυτοματιχ\_χοδιγγ\_οφ\_τηισ\_ϖαρι  
αβλε\_ατ\_ΙΝΣΕΕ\_ιν\_\_\_\_Τηερεφορε\_\_ιφ\_ωε\_θυαντιψ\_ονλψ\_ουρ\_ωορκ\_ωε\_υνδερεστιματε\_τ  
ηε\_τιμε\_\_ανδ\_ιφ\_ωε\_χουντ\_αλλ\_τηε\_ωορκ\_σινχε\_\_\_\_ψεαρσ\_\_ωε\_οϖερεστιματε\_ιτ\_\_Ατ\_  
α\_ρεσουλτ\_\_ιτ\_ισ\_ϖερψ\_διφφιχυλτ\_το\_εστιματε\_τηε\_ρεθυιρεδ\_τιμε\_\_Τηατ\_ισ\_ωηψ\_τηε\_σιγγ\_∇  
\_∇\_αππεαρ\_φορ\_τηισ\_ϖαριαβλε\_ιν\_τηε\_ταβλε\_\_

**Table 1 : Some figures about knowledge bases**

Variable	Size of the reference file	Number of synonyms	Number of additional variables	Number of logical rule tables	Time to learn the reference file <sup>2</sup>	Time to elaborate the knowledge base
Occupation, 4 digits (a)	10,500	1,006	14	2,679	12"	?
Occupation, 2 digits (a)	10,500	1,006	10 to 18	2,679	12"	?
Occupation, 2 digits (b)	115,000	540	0	0	10'07"	?
Place of vacation	42,000	38	0	0	1'34"	few days
Financial product	4,000	12	0	0	4"	2 days
Human activity (during a day)	6,000	700	7	50	10"	2 months
Name and address of establishment	4,600,000	305	0	0	30' on DEC	1 week
Town	44,000	38	1	1,836	57"	2 weeks
Region	470	38	0	0	> 1"	few days
Nationality/Country	728	37	0	0	> 1"	few days

**Table 2 : Automatic coding results**

Variable	Survey	Efficiency	Accuracy	Speed <sup>2</sup>
Occupation, 4 digits (a)	Labor Force Survey	80%	> 90%	6.6 ms
	1990 Census	66%	> 90%	5.0 ms
Occupation, 2 digits (a)	Survey on living conditions	76%	95%	5.0 ms
Occupation, 2 digits (b)	Administrative source <sup>3</sup>	82%	> 95%	11.9 ms
Place of vacation	Survey on living conditions	93%	99%	1.8 ms
Financial product	Survey on households investments	61%	Good	0.9 ms
Human activity (during a day)	Time used Survey	70%	90%	5.1 ms
Name and address of establishment	1990 Census	49%	> 90%	0.4 ms
Town	1990 Census	94 to 99%	99%	1.8 ms

<sup>2</sup> This time was obtained on a PC 486 except for the variable "Name and address of an establishment" which was tested on a DEC  $\alpha$ -server. It is given in milliseconds (ms).

<sup>(a)</sup> Occupation declared by individuals

<sup>(b)</sup> Occupation declared by enterprises

<sup>3</sup> "Déclaration annuelle de données sociales" (Annual declaration of social data by enterprises)

## 6. THE SOFTWARE PACKAGE

All the programs have been written in C language. The whole software package is available on PC (486 or more), with Windows or Windows-NT. Excluding the expert interface, the programs work on IBM/MVS mainframes and on Unix workstations.

The whole software package is *independent of the language* (french, english, ...) *and the variables used*.

The whole software package is made of three parts :

### 6.1 PART 1 : THE EXPERT INTERFACE

This expert interface is a user-friendly work-station to be used by the variable-expert to :

- elaborate and load knowledge files;
- save the knowledge files into a unique knowledge base;
- load and modify knowledge bases;
- test automatic coding using interactives methods (entering and modifying text and additional variables);
- test automatic coding on a file to be coded;
- update the knowledge base by exploring the coded files;
- explore the reference file : sorting and screening by code, words making up the text (as a function of position or not), partial codes;
- explore the decision tables : interdependencies between tables, possible codes for a given list (with the possibility of screening by means of an additional variable), possible paths leading from one table to another, partial coding using a list, etc;
- verify and ensure the quality of the knowledge bases obtained with this interface : ensuring the quality;

This interface is working on a PC (486 or more) with Windows or Windows-NT.

As of today, this interface is only in french but we hope to translate it soon.

### 6.2 PART 2 : THE APPLICATION PROGRAM INTERFACE (A.P.I.) PACKAGE

This A.P.I. package has been developed *to be called* into the processing of survey data. It was created to code automatically from the knowledge bases elaborated by an expert with the expert interface (see part 1).

### 6.3 PART 3 : THE OBJECT MODULES AND INCLUDE FILES PACKAGE

This programs package has been developed *to be incorporated and compiled* in the processing of survey data. Like part 2, it was created to code automatically from the knowledge bases elaborated by an expert with the expert interface (see part 1).

## 7. THE FUTURE

The SICORE project will be completed in may 1996. The knowledge bases for most important variables are already been created and updated regularly by a variable-expert. Many statisticians want to used SICORE in their survey data processing and we now have to help them to properly use SICORE.

The most important application will be the next France Population Census (in 1999) which will use SICORE to code automatically a lot of variables : Occupation (four digits), Town where the individual works, Name and address of establishment where the individual works (to obtain the activity of this establishment).

At INSEE, other applications intend to use SICORE :

- The Labor Force Survey, to code Occupation (four digits), Town where the individual works, and eventually a new variable : Diploma;
- The administrative source DADS (Annual declaration of social data by enterprises), to code Occupation (two digits at the moment, four digits in the future), Town where the individual works;
- All surveys on living condition, to code Occupation (two digits);
- Many regional surveys, to code Occupation (two digits);
- Some surveys on living condition, to code Financial product;

Moreover, a new userfriendly interface will be developed : the coder interface. It will help statisticians who work on PC Windows and want to separate the automatic coding from the rest of their survey data processing.

Also, other foreign institutes of statistics are interested in using SICORE to code their Census.

To conclude, the SICORE system is a great improvement from QUID. But we will not stop here. The research will continue to improve the quality of automatic coding and to generalize its use at INSEE.

## **BIBLIOGRAPHY**

LORIGNY, J., (1982) Questionnaire theory applied to wording recognition, *IEEE Congress at Les Arcs*, Ed. CNRS GR23, Paris VI

LORIGNY, J., (1988) QUID, une méthode générale de chiffrage automatique, *Survey methodology*, December 1988, vol.14, n 2, pp. 289-298

LYBERG, L., DEAN, P., (1992) Automated coding of survey responses: an international review, submitted to the Conference of European Statisticians, Work Session on Data Editing, Washington, March 1992

RIVIERE, P., (1994) The SICORE automatic coding system, Working Paper, Conference of European Statisticians, Cork, October 1994

RIVIERE, P., (1995) Outline of a theory of automated coding, Working Paper, Conference of European Statisticians, Athens, November 1995

RIVIERE, P., (1995) Applications of automated coding with SICORE, Room Paper, Conference of European Statisticians, Athens, November 1995

WENZOWSKI, M.J., (1988) ACTR - a generalized automatic coding system, *Survey methodology*, December 1988, vol.14, n 2, pp. 299-308.