Survey Methodology, December 1988
Vol. 14, No. 2, pp. 317–325
Statistics Canada

Postal Address Analysis

YVES DeGUIRE[1]

ABSTRACT

When we examine postal addresses as they might appear in an administrative file, we discover a complex syntax, a lack of standards, various ambiguities and many errors. Therefore, postal addresses represent a real challenge to any computer system using them. PAAS (Postal Address Analysis System) is currently under development at Statistics Canada and aims to replace an aging routine used throughout the Bureau to decode postal addresses. PAAS will provide a means by which computer applications will obtain the address components, the standardized version of these components and the corresponding Address Search Key (ASK).

KEY WORDS: Postal addresses; Administrative data; Parsing; Standardization; Search key.

1. INTRODUCTION

Postal address analysis can be defined as the process of identifying the basic components of an address which appears in free format, standardizing those components, and generating an identifier for that address. This process can be used, for example, in the pre-processing step of any record linkage application that uses an address field or in the generation of a key for database access. Statistics Canada, as part of its 1991 census research program, is conducting a study on the implementation of a national Address Register. Such a register contains basically, postal address information. This information must be analyzed carefully in order to produce a register and to assess its quality. The Address Register Research Team has recognized that fact and research into the area of automated postal address analysis was initiated.

This paper presents the results of this research on postal address analysis. The nature of an address and its related problems will be described. Also, some computer considerations will be discussed to explain why new software is needed for the Address Register and Statistics Canada. Finally, we will examine PAAS (Postal Address Analysis System); a system currently under development at Statistics Canada.

2. POSTAL ADDRESSES: STATEMENT OF THE PROBLEM

A postal address can be defined as a string of characters representing a location where an individual can pick up his mail. By location, we mean a physical place where the deliverer (like a postman) and the receiver agree in the matter of mail reception. It can be a dwelling, a postal box, a street or a rural route. To restrict our field of study, we are going to examine the addresses that are Canadian (French and English), that represent residential locations and that should result in correct mail delivery.

[1] Yves DeGuire, Research and General Systems, Statistics Canada, room 2405, Main Building, Tunney's Pasture, Ottawa, Ontario, K1A 0T6.

As one would expect, the flexibility in the address definition results in problems for any computerized application having to deal with postal addresses. Even a person is likely to encounter some problems with addresses with which he/she is not familiar. Three major problems are analyzed here.

2.1 The Syntax of a Canadian Postal Address is Complex

A postal address is composed of tokens (lexical items which can be considered as basic units in an address). A token can be either a delimiter, a term (or keyword), a word, a letter or a number. Figure 1 illustrates an example of token decomposition. Tokens can be combined to get address components which are larger address structures. In turn, a component can fall into three groups: designators, qualifiers and secondary words. Figure 1 gives also an example of a component decomposition. Valid addresses are composed of both a set of valid combinations of components and a set of valid combinations of tokens. However, it is more practical for implementation purposes, to define an address with token patterns (combinations of tokens). Token patterns can be generated from a formal postal address grammar (written in BNF for example) and used directly for constructing a postal address.

This syntax is fairly complex. First of all, the grammar is sizeable. We have analyzed a national sample of 30,000 addresses taken from six different administrative files. In these addresses, we found around 4,900 different token patterns. This is substantially higher than what is reported in Drew(1987) because we have analyzed addresses from many different files, not just one. Other interesting results concern the distribution of those patterns. Only 37 patterns are necessary to cover 50% of the addresses. So, there are a few common patterns, but most of the patterns are rather rare. Nevertheless, this analysis illustrates the complexity of postal address syntax by demonstrating that it is not restricted to just a few patterns. Secondly, as much as 600 different terms can be found in a good national sample of addresses. Thirdly, an address is usually in free format, *i.e.* the components (and the delimiters) can occur in any one of several positions.

2.2 Addresses Don't Follow Precise Standards

Addresses representing the same address location can be written in many ways as illustrated in Figure 2. The reason for this situation is the flexibility in postal address syntax and also human nature. In fact, people write addresses as they like and follow the "standards" in use in their immediate environment.

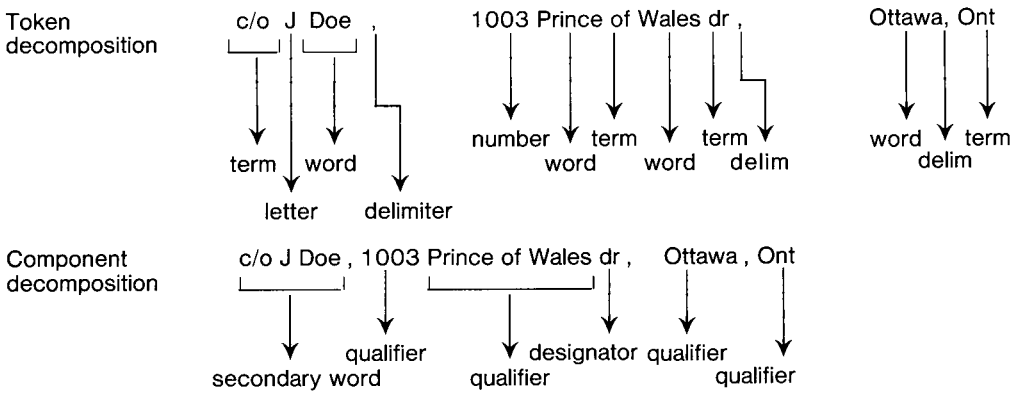

Figure 1. Two ways of decomposing a postal address.

2.3 Ambiguities Occur in Postal Addresses

A postal address can't be regarded only from a syntactic point of view. Its semantic (i.e. the meaning a postal address) must be examined as well. Sometimes, one address can potentially represent more than one location. We then face an ambiguity since we don't know how to interpret it. To do so, more knowledge is required in order to exclude the locations that don't exist and to identify the correct location. However, this knowledge doesn't always permit us to narrow down the location; we then face an unresolvable ambiguity. Figure 3 shows an example of an ambiguous address.

3. COMPUTER SYSTEMS CONSIDERATIONS

Now that we have a better understanding of postal addresses as well as their related problems, we will concentrate on the use of postal addresses in computer systems.

3.1 Computer Applications Requiring Address Information

Several types of application require address information. Some record linkage projects link individuals or dwellings (like in the construction of an Address Register) based on their postal addresses. Their linkage rules perform essentially on standardized address components. On the other hand, databases and computer files storing postal addresses are numerous. For example, postal addresses information for an Address Register must be stored in some fashion, either in a stand alone flat file or in some kind of integrated database. But what information is stored? Address components (standardized or not) could be. For follow-up or historical purposes, the original input address could be kept as well. However, retrieval from a large database (or a large flat file) requires an Address Search Key (ASK) to allow direct access (or direct matching) to a record identified by a postal address. Mailing labels processing is another area where postal addresses is a big concern. Address components, standardized or not, can form mailing labels.

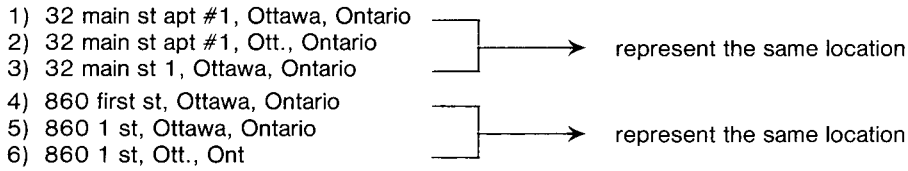

Figure 2. Examples of Addresses Which Represent the Same Location.

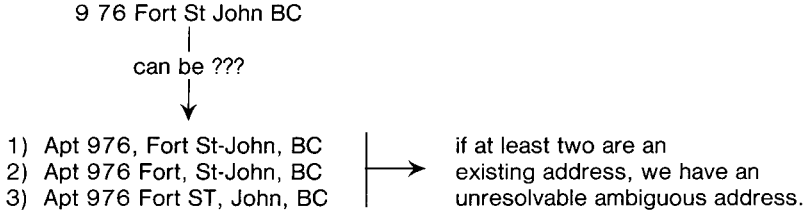

Figure 3. Example of an Ambiguity.

3.2 Three Basic Information Components

Therefore, three basic information components need to be derived from a free format postal address: the address components, the standardized components and the Address Search Key (ASK).

1. THE ADDRESS COMPONENTS

   They represent recognizable and useful portions of an address. The major address components are street number, street name, street direction, street designator, postal designator, postal qualifier, municipality name, province name, and postal code.

2. THE STANDARDIZED COMPONENTS

   They are the standardized version of the address components, where any style variations are removed.

3. THE ADDRESS SEARCH KEY (ASK)

   This is a compressed string, unique for a given address.

3.3 Postal Address Analysis System

A complete Postal Address Analysis System (a computer system that generates the three basic information components we need) represents an expert system in the field of postal addresses. Expert because you replace a specialist (like a postman) in address recognition. At Statistics Canada in the 1970's, two routines were developed to analyze postal addresses. ENCODA (component decompositon) and ASKGEN2 (standardization and ASK) were implemented for the Business Register Maintenance System. They served well until recently. With the advent of powerful computers, new software development techniques and the Address Register itself, they don't perform to today's standards.

- The encoding success rate is too low. A study using a national sample of addresses from many administrative files shows that ENCODA cannot properly decode an address, on average, 15% of the time. This is not acceptable since it could lead, in the case of the creation of a national Address Register, to over one million encoding failures.
- The user interface is poor. There is no comprehensive status produced at the completion of the analysis. As well, very few utilities are provided in order to ease programming burden.
- The functionality is incomplete. Standardized components and ASK are mixed up in the same data structure. Standardized components are truncated to allow data compression but ASK is very long because it is stored in fields of fixed length. Also, the software doesn't recognize address ambiguity.
- Maintenance of the software is a nightmare. New address patterns are difficult to incorporate into the routines because these are complex and tend to become more and more so with time. This is a sign of aging software.

To fulfill the requirements of an Address Register and of Statistics Canada in the area of address analysis, the development of a completely new system was initiated. The problem this time has been approached with expert system techniques, modular design and full scale implementation. This new system is called PAAS, for Postal Address Analysis System.

4. A POSTAL ADDRESS ANALYSIS SYSTEM: PAAS

PAAS is currently under development. Therefore, some results are preliminary, but in general very encouraging. We will review here the four basic functions of the system.

4.1 Address Parsing

The parsing function is the most important and complex function of PAAS. Here, PAAS accepts as input a free format address, scans it (breaks it into lexical items) and parses it (analyzes the syntax) to decode it into address components.

This parser generates the following items for every address processed (Figure 4 illustrates two examples of this output):

- A comprehensive Address Status code; such as V for valid, E for syntax error, *etc.*
- Identification of components in the input address.
- Components classification: every component is classified using a detailed code, so it is easy to understand the meaning of a component. This code is divided into three sub-codes:
  - TYPE code: indicates the group of components to which a component belongs. Example of TYPES are those for province (PR), municipality (MU), street (ST), *etc.*
  - CAT code: refines the group of components indicated by TYPE. Examples for the street TYPE (ST) are name (NA), number (NU), designator (DE), *etc.*
  - CLASS code: classifies a component by examining its characteristics. Examples are avenue (AV) or road (RD) classification of a street designator.
- Ambiguity detection: the PAAS parser flags any component that could change because of an ambiguity.

The PAAS parser was implemented using MPL. MPL is a meta-programming language. It allows us to generate programs or subroutines used for syntax analysis and automatic translation. The input to MPL is a set of specifications divided into the scanning (token recognition), the syntax rules and the semantics. The scanning represents the lexical analysis where the input is broken down into tokens. The syntax specification is similar to a BNF grammar specifications: the right-hand side symbols of a syntax rule are defined by the left-hand side symbols. Figure 5 gives examples of syntax rules. Finally, a semantic action can be associated with any rule and is used to handle some complex aspects of the syntax, as well as to perform other actions (such as updating a table of components). The MPL language is well suited to writing translation specifications and has been used at Statistics Canada to implement STATPAK (retrieval

	ADDRESS	COMPONENT	TYPE	CAT	CLASS	AMB_FLAG
(1)	32 Main st, Ottawa, Ont	32	ST	NU	**	
		Main	ST	NA	**	
	ADDRESS_STATUS=====> V	st	ST	DE	ST	
		Ottawa	MU	NA	**	
		Ont	PR	NA	35	
(2)	32 Main st Ottawa Ont	32	ST	NU	**	
		Main	ST	NA	**	
	ADDRESS_STATUS=====> A	st	ST	DE	ST	*
		Ottawa	MU	NA	**	*
		Ont	PR	NA	35	

Because the second example misses the commas to delimit the address, an ambiguity is flagged by PAAS.

Figure 4. Examples of the PAAS Parser Outputs.

and tabulation system for the census), NYSIIS (name encoding routine) and NAMEPARS (name parser). It saves development time (*e.g.* you don't need to write a detailed and custom program in a traditional programming language such as PL/1). The specifications in BNF are much easier to understand than is a program with a complex logic.

The PAAS parser involves a rather complicated syntax analysis and represent a fairly important MPL application. For example, a dictionary containing more than 600 terms assist in the scanning of addresses. As well, more than a hundred syntax rules implement the syntax analysis. In this syntax analysis, the initial tokens are transformed from a rule right- hand side to a rule left-hand side and become higher level address fragments (this is known as forward chaining) until the address is completely analyzed. During this process, the address components are identified and stored in a table by the semantic action of a rule. The invalid addresses are found whenever no rule is applicable. A sample set of rules to decode an address is illustrated in Figure 5. Finally, for some complex addresses, a special analysis is peformed through the use of the MPL semantic facility. This is required anytime an ambiguous term is encountered. In this case, PAAS analyses the surroundings of the ambiguous term.

In comparison with ENCODA, the PAAS parser is an improvement in the following are as:

- The quality of the parsing: the PAAS parser is able to decode more addresses successfully than ENCODA does. A series of parallel runs over identical national samples of addresses showed that PAAS is successful on more than 97% of addresses, while ENCODA properly handles only 85% of them.
- The indication of an address status: the status is more complete than ENCODA's which provides for only two possibilities: decoded address or blank address!
- The components: PAAS generates much more comprehensive component information than does ENCODA.
- The maintenance: the utilization of MPL helps in making the PAAS parser a lot easier to maintain than a huge algorithm such as is used by ENCODA.

4.2 Components Standardization

The standardization aims to remove any style variation in the address components defined in the parsing phase.

Unlike ASKGEN2, PAAS doesn't truncate any component and retains all the information in the components. This standardization is achieved basically in three different ways depending on the nature of the component:

1. CODABLE COMPONENTS

   Every component for which a limited number of values exist is standardized by replacing its value with the CLASS code of the component (this code uniquely identifies the standardized value of the component). Falling into this category are components such as the province name, street designator, *etc.*

2. NAME COMPONENT NOT NUMBERED

   To standardize a non-numbered name component, several rules must be applied to transform the original value into a standardized value. The rules vary from the removal of useless characters (*e.g.* quote, hyphen, *etc.*) to abbreviation replacement (*e.g.* Mtl becomes Montreal).

3. NAME COMPONENT NUMBERED

   A numbered name component is standardized by returning its name as a number. For example First becomes 1, Second 2, *etc.*

Ask

The Address Search Key should be unique and short.

Uniqueness is accomplished by concatenating in a pre-determined order the standardized components of an address (rather than a table as with ASKGEN2). We must note here that the ASK doesn't necessarily represent a unique identifier for dwellings. In rural areas for example, a postal address quite often represents many dwellings (*e.g.* RR #1 Ottawa Ontario).

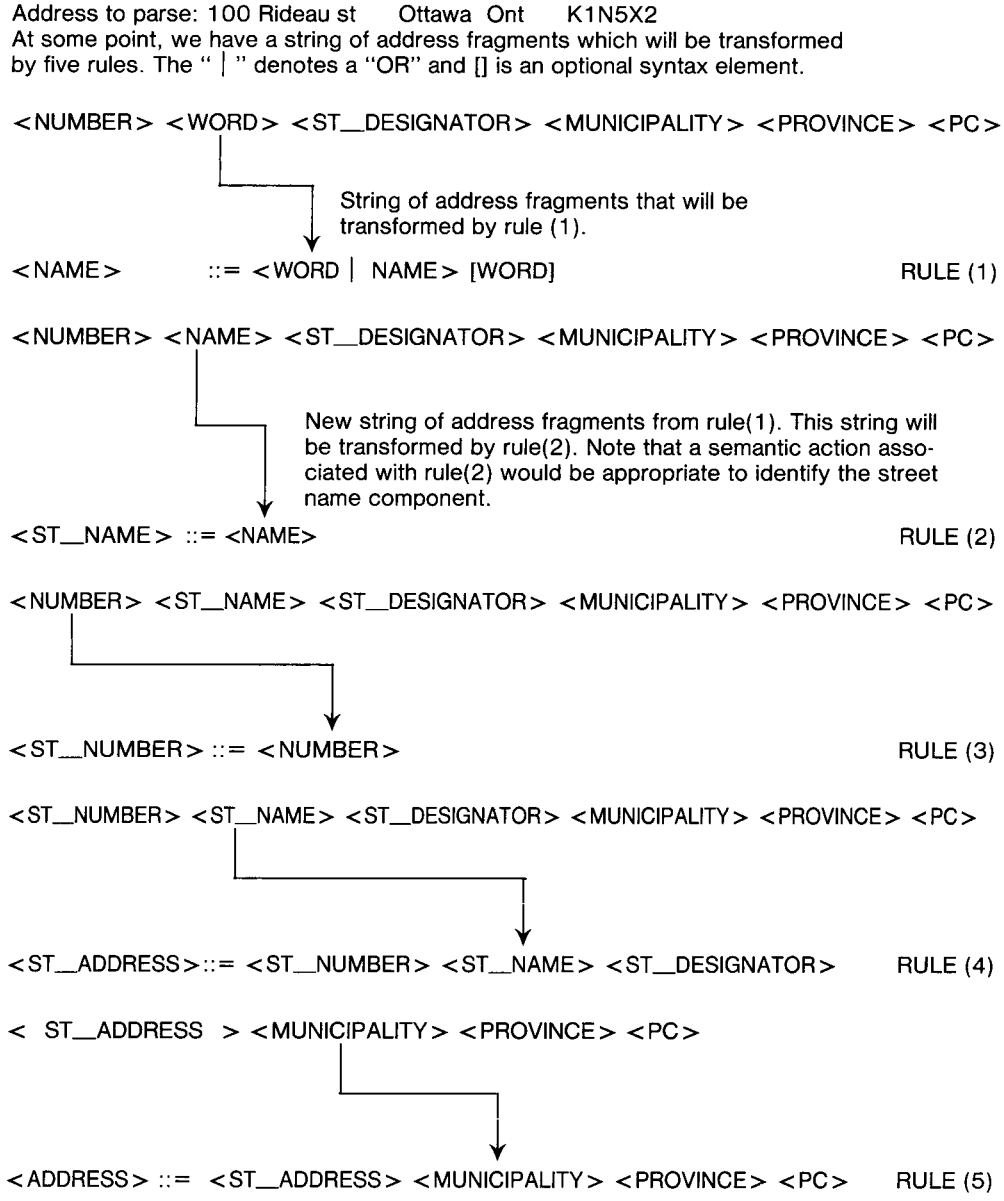

Figure 5. Rules for a Sample Address Syntax.

To shorten the key, different compression techniques can be used. However, compression takes time and we have to choose a technique that will be efficient. We are experimenting with two different techniques.

1. TRUNCATION
   here, the name components are truncated. This technique is not real compression and could affect the uniqueness of a key. However, it is simple and fast.
2. REAL COMPRESSION
   a compression technique that we are looking at consists basically of replacing common combinations of characters by a character code not in use for writing an address. Here, we will preserve the uniqueness but increase the complexity of generating and using a key. Therefore, a longer ASK calculation time is expected with this technique.

4.3 Ambiguity Resolution

Once an ambiguity is determined from the parsing, it must be resolved, either manually, or automatically by the PAAS system. PAAS uses a municipality name file (this file covers the whole country with around 6000 names and has as its source in the Postal Code Directory tape from Canada Post) in an attempt to resolve an ambiguity.

This methodology is limited to the problems related to municipality names. This is not so bad since these problems account for a good portion of the ambiguous situations, and are easy to detect and to resolve (they don't involve a large amount of data). Future work could examine the usefulness of detecting and resolving more situations.

Finally, no matter how good the software becomes, the unresolvable and the non-existant addresses will remain a problem and should be followed-up manually.

5. CONCLUSION

The results of postal address analysis as accomplished by PAAS are encouraging. It decodes a vast majority of addresses, outputs a very informative code for every component, standardizes and generates an ASK properly, and handles ambiguities. Also, PAAS integrates utilities and interfaces for users and maintainers.

Users have access to an interface which processes their addresses through the four basic functions as well as a facility that handles the addresses in error (on-line processing). A file processor program is also provided.

Also integrated into PAAS is a quality assurance tool for PAAS maintainers. PAAS will evolve in the future with the discoveries of new addresses and obsolete addresses. Making sure that the changes to the system are applied properly is tricky. This maintainance tool ensures that a change to the software doesn't jeopardize any valid addresses properly analyzed in previous versions of the system.

ACKNOWLEDGEMENTS

The author would like to thank J.P. Lozano and M. Vriends who worked on the implementation of PAAS, and B.E. Hill and M. Elsaesser for their help in writing this paper. Acknowledgement is also due to J. Armstrong who experimented with PAAS as well as providing comments on an earlier version of this paper.

REFERENCES

BARRETT, WILLIAM A., BATES, RODNEY M., GUSTAFSON, DAVID A., and COUCH, JOHN D. (1986). *Compiler Construction*. Science Research Associates Inc.

CANADA POST CORPORATION (1986). *Postal Codes Directory: Atlantic, Quebec, Ontario and Western regions.*

DEGUIRE, Y. (1987). Research into the parsing and standardization of free format addresses at Statistics Canada. Internal report, Statistics Canada.

DREW, J. DOUGLAS, ARMSTRONG, JOHN, VAN BAAREN, ALEX, and DEGUIRE, YVES, (1987). Methodology for construction of address registers using several administrative sources. International Symposium on Statistical Uses of Administrative Data, Ottawa.

HILL, TED (1986). *MPL A Translator Writing System.* System Documentation, 1-4. Statistics Canada.

LOZANO, J.P. (1987). Postal Address Analysis System Study. Internal Report, Statistics Canada.

STATISTICS CANADA (1986). Record Linkage Software User Guide. System documentation, Research and General Systems.

STATISTICS CANADA (1988). Postal address analysis system (PAAS): Project charter (draft). Internal report. Research and General Systems.