United States
Department of
Agriculture

National
Agricultural
Statistics
Service

Research Division

STB Research Report
Number STB-95-02

October 1995

Record Linkage I: Evaluation of Commercially Available Record Linkage Software for Use in NASS

Charles Day

RECORD LINKAGE I: EVALUATION OF COMMERCIALLY AVAILABLE RECORD LINKAGE SOFTWARE FOR USE IN NASS, by Charles Day, Technology Research Section, Survey Technology Branch, Research Division, National Agricultural Statistics Service, U.S. Department of Agriculture, Washington, DC 20250-2000, October 1995, Report No. STB-95-01.

ABSTRACT

Record linkage is an important technique in NASS for minimizing the presence of duplicate names on its list sampling frame of farm operators and agribusinesses. In the late 1970's, NASS developed an automated record linkage system which runs on an IBM mainframe for this purpose. With changes in technology, the need has arisen for portability between platforms, integration with client/server technology, and interactive operation. Also, NASS desires to reduce resource expenditures on record linkage while maintaining the quality of the process.

The growing availability of commercial record linkage solutions has made unnecessary the development of a new record linkage system or an expensive and difficult rewrite of the old system. This report evaluates six commercially available record linkage software packages for their suitability for NASS's purposes. The report starts with a brief discussion of record linkage in NASS, then discusses the statistical theory behind the most popular probabilistic record linkage solution, that of Fellegi and Sunter. Next, the report discusses the requirements for a NASS record linkage system. Detailed reviews of the six software packages follow. Except for the review of AUTOMATCH, which NASS has tested extensively, these reviews are based on information provided by the software manufacturers. The report concludes that, **for NASS's purposes**, AUTOMATCH is the best choice. The report ends with a glossary of record linkage terminology and a checklist for the evaluation of record linkage software packages.

KEY WORDS

Record linkage; List sampling frame; Duplication; Software

This report was prepared for limited distribution to the research community outside the U.S. Department of Agriculture. The views expressed herein are not necessarily those of NASS or USDA.

ACKNOWLEDGEMENTS

The author would like to thank the members of the List Sampling Frame Section for their assistance in developing the requirements and explaining NASS's list sampling frame maintenance procedures, Kara Broadbent of the Ohio Research Unit for her work in testing AUTOMATCH, and the representatives of the manufacturers of the software reviewed in this report for their assistance in understanding their products.

Table of Contents

SUMMARY

Record linkage is a technique for associating records representing the same unit from one or more files representing the same population by comparing identifiers. The National Agricultural Statistics Service (NASS) uses record linkage to minimize duplication of names of farm operators or operations during construction and maintenance of its list sampling frame (list frame) of farmers and agricultural operations. NASS's current record linkage solution, which NASS uses with the Real Time Mail Maintenance System (RTMMS), is a set of programs written in COBOL and FORTRAN running on an IBM mainframe computer at Lockheed Martin Corporation. The Enhanced List Maintenance Operations (ELMO) system is replacing the RTMMS. Like the RTMMS, ELMO will need a record linkage system. Over the years, NASS has become concerned with the high level of resources associated with using its current record linkage system. Also, NASS needs to integrate applications with client/server technology, and to have applications that are portable from platform to platform. The agency also anticipates a need to perform record linkage interactively. NASS anticipates that porting its current record linkage system to new platforms would be difficult and expensive. With the advent of affordable, commercially available record linkage software, these packages have become the best candidates for the core of any replacement system. For all of these reasons, NASS decided to explore the use of commercially available record linkage software. This report is an evaluation of several commercially available packages. A second report, entitled "*Record Linkage II: Experience Using AUTOMATCH for Record Linkage in NASS*," detailing NASS's experience with the AUTOMATCH record linkage software package, is also available. This second report covers implementation issues which are not covered in this report.

NASS's current record linkage solution was an early application of the Fellegi-Sunter record linkage theory. Based on a 1969 *JASA* paper by Ivan Fellegi and Alan Sunter, this theory has become the most widely accepted record linkage method. The best of the commercially available packages use it. The current linkage system uses the New York State Intelligence Information System (NYSIIS) phonetic code to block records into groups for comparison to improve efficiency. The commercially available record linkage systems also use blocking, but allow for more blocking factors and multiple passes. They also allow comparison of alphabetic strings using NYSIIS and sophisticated string comparison metrics. Another important task of a record linkage system is parsing free-formatted name and address fields into their component parts. The current system does this using a complicated set of data manipulation and reformatting programs. The commercially available systems offer varying degrees of ability to perform this task. Some of the systems have extremely powerful, flexible systems; others have no standardization capability.

At the beginning of this project, many people were involved in developing requirements for the use of new linkage software in NASS. The new record linkage system must be statistically defensible. It must be low in both initial cost and development costs and user friendly for end users in State Statistical Offices (SSOs) or Headquarters. The software must be well documented and supported by a reliable vendor. The software must run on the hardware platform and under

the operating system chosen for linking applications. In addition, it must be compatible with the Sybase database package, and must run in batch and interactive modes. The software needs to allow for flexibility in blocking and in the use of different linking variables. It must be able to make efficient use of the information in NASS list frame records. The software should allow the choice of appropriate comparison functions for the different types of matching data available on the list frame. If the system uses the Fellegi-Sunter method, it must provide utilities for estimating m- and u-probabilities, and for allowing the user to set weight cutoffs. (Terms of art, like m-probability, are defined in the glossary in Appendix A.) The software must also handle files at least as large as NASS's largest list frame file. Any software chosen for NASS to use needs to handle the parsing and standardization of free-formatted name and address information, such as is found in NASS's list sources. Finally, the software must produce any desired reports or extract files of matches and nonmatches.

Six software packages were chosen for evaluation. Due to budgetary and resource constraints, only AUTOMATCH was purchased for testing. Reviews of specifications and discussions with developers were considered sufficient for evaluating the remaining packages. AUTOMATCH is a generalized record linkage system developed by MatchWare Technologies. The package uses the Fellegi-Sunter record linkage theory and sophisticated string comparison functions in its matching process. It is available for all of the platforms considered by NASS. It also comes with a powerful, customizable package for parsing and standardizing names and addresses. The Generalized Record Linkage System (GRLS) is a product of Statistics Canada. It also is a generalized, Fellegi-Sunter based record linkage package. Statistics Canada developed this package to be used with ORACLE databases, and GRLS derives much of its power and flexibility from the capabilities of ORACLE. This reliance on ORACLE makes it inappropriate for use with a Sybase database. Smart PID, a product of Advanced Linkage Technologies of America, is a set of PASCAL and C language modules that performs record linkage tasks using an "enhanced" version of the Fellegi-Sunter methodology. These modules are primarily designed for use in hospital Management Information Systems (MIS) environments and do not offer the flexibility needed for NASS's purposes. SSA-Name3 and Extensions are similarly a set of software modules. SSA-Name3 is a package for constructing efficient database searching keys based on NYSIIS phonetic codes and frequencies of the codes' occurrence. The Extensions package provides tools to score records selected by the SSA-Name3 product based on a scheme of variable comparisons and weighting constructed by the user to select matches. SSA-Name is not a statistically based matching system, and is expensive. It would, in addition, be expensive to create a system from its components. Both Merge/Purge Plus from Group 1 Software and Merge/Purge 4.3 from Postalsoft are nonstatistical matching systems best suited to their intended use of deduplicating mailing lists in turn-key direct mail solutions. Figure 3 on page 16 summarizes the reviews of the software packages.

The report recommends that AUTOMATCH be chosen as the core component of the agency's next record linkage system. AUTOMATCH is the least expensive of the packages, yet it offers the most functionality. It will require the fewest resources to meet NASS's specifications, and it is based on the proven Fellegi-Sunter record linkage methodology. No other package has a

name and address standardization capability as powerful as that offered by the AUTOSTAN software. It is available on all of the platforms that NASS is considering for record linkage. Finally, it is available in a form that NASS can use to create an interactive record linkage capability.

This report is not a general comparison of the six record linkage software packages. The recommendation for AUTOMATCH is based solely on NASS's particular requirements. This report does not endorse one software package over any other for uses outside NASS.

INTRODUCTION

What is *record linkage*? Record linkage, or matching, does not have a single definition. In their seminal 1959 paper in *Science* [1], Newcombe, *et al.* quote H. L. Dunn and J. T. Marshall, saying, "The term *record linkage* has been used to indicate the bringing together of two or more separately recorded pieces of information concerning a particular individual or family. Defined in this broad manner, it includes almost any use of a file of records to determine what has subsequently happened to people about whom one has some prior information." Fellegi and Sunter, in their 1969 *Journal of the American Statistical Association (JASA)* article [2], state, "The necessity for comparing the records contained in a file L_A with those in a file L_B in an effort to determine which pairs of records relate to the same population unit is one which arises in many contexts, most of which can be categorized as either (a) the construction or maintenance of a master file for a population, or (b) merging two files in order to extend the amount of information available for population units represented in both files." In its 1980 *Report on Exact and Statistical Matching Techniques* [3], the Subcommittee on Matching Techniques of the Federal Committee on Statistical Methodology (FCSM) said that, "Although the terms 'match,' 'exact match,' and 'statistical match' have been used frequently in the literature, the Subcommittee knows no generally agreed upon definitions of these terms. For purposes of this report, the Subcommittee has defined a match as a linkage of records from two or more files containing units from the same population. It has defined an exact match as a match in which the linkage of data for the same unit (e. g., person) from the different files is sought; linkages for units that are not the same occur only as the result of error. Exact matching normally requires the use of *identifiers*, for example, name, address, social security number."

In this report we will borrow from all these definitions. As used here, "record linkage" means the association of records representing the same unit from one or more files representing the same population by comparing *identifiers* for construction and maintenance of a master file for a population. Note that this definition allows for a linkage between records in the same file; this special case will be called *unduplication*. This definition also fits the FCSM's definition of an exact *match*, meaning that the linkage of records that do not represent the same unit in the population is an error; similarly, the failure to link two records that do represent the same unit in the population is also an error. The first type of error will be called a *false link*, the second type a *false nonlink*. A glossary of record linkage terminology can be found in Appendix A.

Uses of Record Linkage in NASS

NASS uses record linkage primarily for the maintenance of the list sampling frame (referred to after this as simply the *list frame*). To make accurate estimates, it is important that no unit in the population be represented more than once on the sampling frame, that is, that no *duplication* be present on the frame. If undetected duplication is present, then the probabilities of selection for each unit in the population will be calculated incorrectly and estimates produced using these probabilities will be biased.

NASS constructs its list frame from source lists of agricultural operators and operations. These source lists have various origins;

among the most important are the Farm Service Agency (FSA) lists of operators who have signed up for agricultural price support programs, lists of members of agricultural producers' associations, lists from State Departments of Agriculture, and lists from veterinarians associations. Operators are often listed on more than one of these lists, a common source of duplication on the list frame. NASS also updates the list frame by the addition of individual records representing operators or operations and by the alteration of existing records. These updates can also result in the creation of two records on the frame representing the same unit in the population. NASS uses record linkage to detect these duplicate records.

Before NASS adds a new list source to its list frame, it uses record linkage to compare the source to the list frame and detect any duplication between the two lists. The operator or operation names not already on the frame are then added to the frame. To detect duplication created by the alteration of records already on the list frame, NASS uses record linkage to compare the list frame to itself (an example of unduplication). This is done annually with a simplified program which matches records on Social Security Number (SSN), Employer Identification Number (EIN), or telephone number before sampling from the frame. States request a complete unduplication using the current record linkage system every few years or so.

A Brief History of Record Linkage in NASS

NASS first began using lists of agricultural operators and operations for probability surveys in the early 1960's; at that time it used different lists for different surveys. In the early 1970's, NASS began to develop software for a single list frame for each state for use in all probability surveys. By the mid-1970's, an effort to develop computer software to support that work was underway within NASS. Among the systems developed during that time was the record linkage system [4]. Incorporating the work of a dozen people over a period of four years, the record linkage system was based on the record linkage theory of Ivan Fellegi and Alan Sunter (the *Fellegi-Sunter theory*, described in a later section of this report). Some 20 years later this is still the most widely accepted probabilistic method for linking records. The record linkage system consists of a set of COBOL and FORTRAN programs which runs on an IBM mainframe at Lockheed Martin Corporation under a time-sharing arrangement. NASS built the first state list frames in 1977 from source lists using this system. The system was then enhanced, and several states had list frames built in 1979. By 1982, all states had a list frame built using the record linkage system. In 1983, NASS put a new database, the Real Time Mail Maintenance System, in place to handle the list frame. The record linkage system is used now to update this database.

As conceived, the record linkage system required several passes through the data, with manual review steps between. The Reformat 1, Reformat 2, Reformat 3, and Data Manipulation stages put the names and addresses on the list in a standard format, and tested the state, place name, ZIP Code. This *standardization* process is critical to any record linkage method. It allows the computer to "find" all of the parts of a name or an address, so that similar parts may be compared between records. Without such a process, it is not possible for the software to compare the records in any meaningful fashion. After the user had successfully

standardized the records, an attempt was made to perform what was referred to as an identical match. This procedure did a straightforward comparison of the identifiers on each record and, if they were identical, linked the records. This was done to improve processing efficiency by eliminating identical duplicate records before the remainder of the processing. Next, the records that remained after the identical match were linked using the *Record Linkage Subsystem (RECLSS)*. RECLSS used the Fellegi-Sunter methodology. It employed the *NYSIIS* phonetic coding system to compare names and to do *blocking* (described in a later section of this report). The system then made an attempt to associate the groups of records that linked into larger, related groups. Finally, the results of the linkage were reviewed by State Statistical Office personnel.

This original procedure, with its many steps and manual review requirements proved too costly to implement, as too many labor resources were required. As it is now used, the system makes a single pass through the data. Records that have standardization errors are discarded. The remainder is run through identical match and RECLSS. If the linkage is being done to add a new list source to the list frame, any records that do not link are added to the list frame as inactives, that is, records not subject to sampling. Additional information is collected on these records and they are made active (subject to sampling), or a decision is made that the record should be deleted after this information has been reviewed. If the linkage is being done to unduplicate a single file, then the results are sent to the State Statistical Office as a (usually very long) computer printout for review. Streamlining the process in this fashion has led to a higher level of linkage errors than originally envisioned.

Purpose of This Report

In the early 1990's NASS decided that it was time to replace the Real Time Mail Maintenance System. The new system, dubbed the *Enhanced List Maintenance Operations system (ELMO)*, is to be maintained in a Sybase database, using client/server technology. At the time the project documented in this report was begun, the ELMO Team had not decided whether each State Statistical Office would have a UNIX server with its own list frame or if there would be a centrally located UNIX server on which all 50 states' frames would reside.

The replacement of the RTMMS precipitated discussion of the need for replacement of the current record linkage system. The need for portability between platforms, integration with client/server technology, and *interactive* linkage, combined with a desire for a system which could be used as designed with fewer personnel resources, led to the decision to explore new record linkage solutions. The projected cost and difficulty of either developing a new system or porting the existing system to new platforms, led to the exploration of commercially available record linkage software.

The purpose of the project documented in this report is to locate, evaluate, and develop software to replace NASS's current record linkage system. This report documents the first stage of that project, the location and preliminary evaluation of commercially available record linkage software for current and future NASS needs.

A second report, entitled "*Record Linkage II: Experience Using AUTOMATCH in NASS*," covers the evaluation of AUTOMATCH's suitability for NASS's record linkage tasks, and recommendations for additional research and development work prior to implementation.

BACKGROUND

Comparison Outcomes and Methods of Comparing Character Strings

One of the key concepts in the Fellegi-Sunter theory is the idea of a *comparison outcome*. Simply stated, a comparison outcome is the result of comparing the values for the same identifier on two different records. Figure 1 summarizes the Fellegi-Sunter theory. In Figure 1, $\gamma^{(\text{first name})}$ is an example of a comparison outcome. Fellegi and Sunter discuss vectors of such outcomes, with each element of the vector corresponding to a different *linking variable*. γ_n is an example of a vector of comparison outcomes, with elements corresponding to first name, surname, address, SSN, phone number, etc. But before we discuss these vectors, it is important to understand how richly Fellegi and Sunter define the idea of a single comparison outcome.

At first it might seem that only two outcomes are possible for each variable, that the values on the two records agree or they disagree, but, in reality, the set of possibilities is much more varied. Consider the identifier gender. Only two outcomes seem possible, that the two records agree on the value of gender or that they disagree. But what if gender is missing on one of the records? A missing value does not imply either agreement or disagreement, but is simply a third possibility.

Now, think about a variable like surname. Agreement between Smith on one record and Smith on a second record gives some confidence that the records represent the same individual in the population, but there may be many Smiths in a file of names, and such an agreement may occur readily by chance. On the other hand, agreement on the surname Fellegi gives much higher confidence that the records represent the same individual; it would be unlikely for such an unusual name to occur by chance on both records.

The Fellegi-Sunter method is a probabilistic one. It is concerned with estimating, in some rigorous way, the likelihood that two records represent the same individual. If one wishes to use the outcomes of his or her comparisons to estimate such a likelihood, he or she must distinguish the outcome "surname is Smith and agrees," from the outcome "surname is Fellegi and agrees." These have to be considered distinct outcomes, since they infer different probabilities that the records represent the same unit in the population. So, values of a comparison outcome, like $\gamma^{(\text{first name})}$, look like "name is John and it agrees," rather than simply "agrees" or "disagrees."

But the complexities do not end there. Suppose one compares alphabetic strings, like names of people or streets. Do Fellegi and Felegi really disagree, or is one of them a spelling error? A couple of solutions have been proposed to this problem. One is to convert the strings using a phonetic coding scheme (one that converts "sound-alike" strings to the same string) and then compare the results of the conversion. Two such phonetic coding schemes are called *Soundex* and NYSIIS [5]. Using these codes for matching, one converts each name to its

Figure 1. The Fellegi-Sunter Theory

$\gamma_n = (\gamma^{(\text{first name})}, \gamma^{(\text{surname})}, \gamma^{(\text{address})}, \gamma^{(\text{ssn})}, \gamma^{(\text{phone number})}, \ldots)$, where $\gamma^{(\text{first name})}$ is the result of comparing the value of first name on two records, one from List A and one from List B, $\gamma^{(\text{surname})}$ is the result of comparing the surname on those two records, and so forth. $\gamma^{(\text{first name})}$ is called a comparison outcome and γ_n is called an outcome vector.

$\Gamma = \{\gamma_1, \ldots, \gamma_N\}$ is the set of *all possible* vectors of outcomes of comparing two records. In practice, this set contains millions of elements. Γ is sometimes called the comparison space.

Each γ_n has two conditional probabilities associated with it:
m-probability = $P(\gamma_n \mid$ the records being compared represent the same population unit)
u-probability = $P(\gamma_n \mid$ the records don't represent the same population unit)

To achieve a false link rate of μ and a false nonlink rate of λ with a minimum number of record pairs for human review, first, order the γ's in descending order of m-probability/u-probability. Then sum u-probabilities, beginning with γ_1 until the sum of the u-probabilities equals μ (at γ_μ). Next, beginning with γ_N, sum the m-probabilities until they total λ (at γ_λ). This partitions Γ, and yields the following decision rule:

γ_1 (highest m/u)
γ_2
γ_3
.
.
.
___ γ_μ
.
.
.
___ γ_λ
.
.
.
γ_{N-3}
γ_{N-2}
γ_{N-1}
γ_N (lowest m/u)

} Links ($\sum$ u-prob $= \mu$) (Pairs generating these outcome vectors are linked.)

} Possible links (Pairs generating these outcome vectors are reviewed manually.)

} Nonlinks ($\sum$ m-prob $= \lambda$) (Pairs generating these outcome vectors are not linked.)

code, and compares the codes to see if they agree. In this kind of a scheme, agreement or disagreement on each of the different possible codes are all considered distinct outcomes. This is the approach used in the current record linkage system.

A newer approach addresses the question, "How sure are we that two strings are the same?" Considering Fellegi and Felegi, one might be nearly certain that they are the same. The NYSIIS code for both of these names is FALAGA. What about Faleggi, or Fulleggi? The NYSIIS code for these names is FALAGA as well, but they seem, in some sense, farther from Fellegi. A *string comparison function* assigns a "weight" (not to be confused with a match weight in the Fellegi-Sunter theory) to the comparison of two strings based on the lengths of the strings, and the number of insertions, deletions, transpositions, and replacements required to turn one string into the other. One can use this "weight" to adjust the probabilities that two records represent the same unit based on an agreement. In other words, these weights can be used to define partial agreements. If the two strings are identical, we use the full agreement probability, but if the two strings vary, we use something less than the full agreement probability, and the more the strings disagree, the more the probability is reduced. This type of character string comparison also has the advantage of being insensitive to the structure of the strings; for example, it is as useful for Oriental names, where most of the discriminating power is in the vowels, as it is for Western European names, where more of the discriminating power is in consonants.

To review, a comparison outcome is the result of comparing the value of an identifier on one record to the value of the same identifier on another record. Agreement or disagreement on different values represent distinct outcomes, and, for character strings, it is possible to differentiate degrees of agreement and disagreement into different outcomes. This leads to the conclusion that, for any one identifier, such as $\gamma^{(\text{first name})}$ in Figure 1, there may be many thousands of different possible outcomes.

The Fellegi-Sunter Record Linkage Theory

In 1969, Ivan Fellegi and Alan Sunter of Statistics Canada published a paper in the *Journal of the American Statistical Association (JASA)* entitled "A Theory for Record Linkage." This paper provided the theoretical underpinning for much of NASS's current record linkage system. The theory contained in it (closely related to one suggested by Benjamin Tepping in a 1968 *JASA* article entitled "A Model for Optimum Linkage of Records" [6]) has become the most widely accepted probabilistic linkage method. (Other, information-theoretic approaches to record linkage have been suggested [7, 8, 9], but have not gained popularity.) Several of the software packages evaluated in this report use the Fellegi-Sunter method.

Fellegi and Sunter began by assuming the existence of two files of computer records. They referred to the files as List A and List B. They assumed that both files contained records representing units from the same population. The authors developed a theoretical approach for using a computer to associate each record in List A with the record or records in List B which represent the same unit in the population. This approach has the property that a minimum number of *record pairs* have to be reviewed by human beings

to achieve given levels of false link and false nonlink.

The authors assumed that each record in File A would be compared to each record in File B using a set of identifiers (like name, address, social security number, telephone number). For each pair of records, there would be a comparison outcome for each identifier. In Figure 1, $\gamma^{(\text{first name})}$ is an example of a comparison outcome. Taken together, these outcomes would form a vector (an ordered n-tuple) of comparison outcomes. In Figure 1, γ_n is an example of such a vector. Next, they considered all of the possible outcome vectors; that is, all of the possible combinations of outcomes of comparing the identifiers on two records. The set Γ in Figure 1 is an example of such a set. For each vector, that is, each combination of outcomes (each of the γ_n's in Figure 1), they considered two conditional probabilities. The first, which they called the *m-probability*, is the probability that this outcome vector occurs given that the records being compared represent the same unit in the population (that is, the pair of records belongs to the *matched set*). The second, the *u-probability*, is the probability that this outcome vector occurs given that the records being compared don't represent the same units in the population (that is, the record pair belongs to the *unmatched set*).

Fellegi and Sunter then suggested a general form for a decision rule, which divides the set of outcome vectors into three subsets. The first of these is the linked subset, for which the computer makes the inference that any record pair associated with one of these outcome vectors belongs in the matched set (that is, based on the outcomes of the comparisons of the linking variables, the computer infers that the records represent the same unit in the population). This subset corresponds to the subset marked "Links" in the diagram at the bottom of Figure 1.

The second group is the unlinked subset, for which the computer makes the inference that any record pair associated with one of these outcome vectors belongs in the unmatched set (that is, based on the outcomes of the comparisons of the linking variables, the computer infers the records do not represent the same unit in the population). This subset corresponds to the subset marked "Nonlinks" in Figure 1.

For the third subset, which Fellegi and Sunter called the *possible links*, the computer makes no inference; record pairs with these outcome vectors are referred for human review. This subset corresponds to the subset marked "Possible links" in Figure 1.

The heart of the Fellegi-Sunter theory is a theorem that states that, for given levels of false link and false nonlink (designated μ and λ in Figure 1), the decision rule can be defined in terms of the ratios of the m- and u-probabilities so that the set of possible links is as small as possible, that is, the system automatically makes decisions about as many record pairs as possible. (This optimality criterion, a minimum size for the set of records to be reviewed by humans, makes sense when one considers that the goal of automating record linkage is to let the computer do as much of the work as it can.) Fellegi and Sunter began by forming the ratio of the m-probability to the u-probability for each vector of outcomes (each γ_n). They then ordered the set of outcome vectors on this ratio. To form the linked subset, they began at the top (highest m/u ratio--the

outcome vectors most likely to represent a true match) and, in descending order of m/u, summed the u-probabilities (the probabilities that, in fact, record pairs with these comparison vectors do not represent the same unit) until they reached the given false link rate. All of the vectors which went into this sum (the vectors from γ_1 to γ_μ) fall into the linked subset; record pairs that generate these vectors when compared are linked.

To form the unlinked subset, they began at the bottom (lowest m/u ratio--the outcome vectors least likely to represent a true match) and, in ascending order, summed the m-probabilities (the probabilities that, in fact, record pairs with these comparison vectors do represent the same unit) until they reached the given false nonlink rate. All of the vectors which went into this sum (the vectors from γ_λ to γ_N) fall into the unlinked subset; record pairs that generate these vectors when compared are not linked. Everything left over (everything in the middle, all of the vectors from $\gamma_{\mu+1}$ to $\gamma_{\lambda-1}$) represented the possible links. Pairs that generated these vectors require human intervention to determine their link status.

It may not be intuitively obvious to the reader that this decision rule results in the minimum size set of possible links for the given levels of error. However, the goal of this paper is to present only as much of the theory as necessary to discuss its application in software packages. The ambitious reader will find a rigorous proof in Fellegi and Sunter's *JASA* article.

At this point, an attentive reader might still ask, "All of this is fine, but, to get the m- and u-probabilities, don't you already have to know the answer--that is, to know what records are in the matched and unmatched sets, so that these probabilities can be calculated? ... And if many variables are used and they can each have thousands of possible outcomes, isn't the set of possible outcome vectors unmanageably large--so large that estimating the m- and u-probabilities from a reasonable sized data set is impossible?"

These are both reasonable criticisms; however, Fellegi and Sunter suggested a method for making the theory operational that addressed these concerns. Fellegi and Sunter began with the idea that, for each record pair, they could reasonably approximate the m/u ratio for that pair's outcome vector.

First, the m- and u-probabilities associated with each possible comparison outcome can be estimated. For example, m- and u-probabilities can be estimated for each possible value of $\gamma^{(\text{surname})}$; the value "name is Smith and agrees" might have an u-probability of 0.05, while the value "name is Rybczinski and agrees" might have a u-probability of 0.001. But how can these probabilities be estimated without knowing the results?

These estimates can be made as follows. Consider the idea of an m-probability, the probability that, given that the two records represent the same unit in the population, the given outcome occurs. The only way identifiers, like name, address, social security number, phone number, and so forth, would not agree on two records representing the same population unit is if they were in error. So, the m-probability of the outcome "the value of the variable is "x" and agrees" is just 1-q, where q is the joint probability of an error in the value "x" on either record. Therefore, the m-probability for an agreement on these records is 1-q; likewise, the

m-probability for the outcome "value is "x" and disagrees" is q. One can easily see that m-probability is really a measure of accuracy.

Ideally, independent estimates of q will be available, but this is usually not the case. Some of the software packages reviewed in this report have utilities that allow the estimation of m-probabilities using iterative methods, beginning with an approximation of q made from experience. These methods usually perform fairly well even when the initial guess for q is poor.

Now consider the u-probabilities. A u-probability is the probability of getting an outcome given that the record pair is in the unmatched set. If two records are in the unmatched set, the only way they agree on an identifier is by random chance. So, the u-probability for an agreement is just p, where p is the proportion of records in the file with that value for the identifier. The u-probability for a disagreement is just the complement of that for an agreement, 1-p. Since the u-probability varies with the distinctiveness of a value, it is a measure of discriminating power. Fellegi-Sunter systems usually have utilities for estimating u-probabilities by making frequency counts of the files to be linked.

We now have estimates of m- and u-probabilities for individual comparison outcomes. How do we get from this to an estimate for the whole vector? The answer is to make a simplifying assumption, one that is important to remember when applying the Fellegi-Sunter theory. The identifiers are assumed to be independent. Using this assumption, the m/u probabilities for the individual linking variables can be multiplied together (since the outcomes are independent events) to get an m/u ratio for the entire vector. In practice, these probabilities are converted to logarithms and the logs are added. The resulting log of the m/u ratio is called a *linkage weight*.

In practice, rather than ordering the m/u ratios and setting cutoffs, one assigns cutoff weights to form a decision rule, and assumes record pairs with weights above the linked cutoff are matches. Record pairs with weights below the unlinked cutoff are assumed to be unmatched. Weights between the cutoffs are mapped to the possible linked subset.

Blocking

Another important linkage concept is *blocking*. Theoretically, it would be ideal to compare all possible record pairs, but, unfortunately, this problem is computationally intractable. Blocking is a method for reducing the number of comparisons to be made by using a highly accurate variable with moderate discriminating power to divide the files being compared into smaller groups, or blocks. Records are then compared only within blocks that have the same value for the variable. For example, one may use the NYSIIS code of the surname to block two files. (In addition to its use in comparing character string linkage variables, NYSIIS coding is also used for blocking. Research at NASS in the 1970's confirmed the superiority of NYSIIS over Soundex for blocking purposes [10].) Records in file A with a particular NYSIIS code for their surnames would then be compared only to records in file B with that same surname NYSIIS code. This greatly reduces the number of comparisons to be made, with the result that the linkage problem is then com-

Figure 2.--Results of Standardizing Names in Records A and B

	Record A	*Record B*
Title:	Mr.	
First Name:	Robert	Robert
Surname:	Smith	Smith
Suffix:	Jr.	

putationally tractable. A simple arithmetic example will suffice to show why. If two files of 10,000 records each are to be linked, the number of possible comparison pairs is $10^4 \times 10^4$ or 100 million pairs. If each file can be blocked into 100 blocks of 100 records each, then the number of comparisons to be made is 100 x (100 x 100) or one million comparisons. If the latter problem took three hours to solve on a computer, then the former would take two weeks.

Standardization of Names and Addresses

In the record linkage context, *standardization* refers to the ability to *parse* a character string, like a name or an address, into its component parts, place these parts into identifiable fields, and then convert each of these parts to standard forms or abbreviations. This is necessarily done before comparing the records for linkage purposes. The importance of this transformation cannot be overestimated. Usually, several parts compose an individual's name or address. These parts may occur in different orders, be abbreviated differently, or be omitted. For example the names "Smith, Mr. Bob, Jr." and "Robert Smith" may appear in two records (call them A and B respectively). "Smith, Mr. Bob, Jr." is composed of a surname "Smith," a title, "Mr.," an abbreviation of a first name, "Bob," and a suffix, "Jr." "Robert Smith" is composed of an (unabbreviated) first name, "Robert," and a surname "Smith." Without the ability to identify the different parts of a name and to place standard abbreviations for those parts into fixed fields, one could only compare the two character strings to see if they were the same. Comparing R to S, o to m, b to i, etc. would cause the computer to believe that the two name fields disagreed. With the ability to standardize names, the two names would appear as shown in the box above. The computer could then compare each corresponding part. It would discover that it had two missing values and two perfect agreements, rather than a disagreement on a single long string.

RECORD LINKAGE REQUIREMENTS

To evaluate record linkage software, it was necessary first to develop a set of requirements based on NASS's specific needs. In late 1992 and early 1993 the Technology Research Section drafted a set of requirements based on their knowledge of record linkage, the current NASS record linkage system, and the requirements of the ELMO system. This document was shared with six state list frame statisticians [11], the List Frame Section, the ELMO Team, and three people who had been heavily involved in the development of NASS's current record linkage system, Bill Arends (Director, Statistical Standards Staff), Dick Coulter (Wyoming State Statistician), and Ben Klugh (Arkansas State Statistician). The requirements were then revised, based on their comments. Key topics from this requirements document are discussed below. A

more general framework for comparing record linkage software packages, not specific to NASS's needs, is contained in Appendix B.

General Requirements

This section covers general requirements, including *statistical defensibility*, cost, and vendor support.

Statistical Defensibility--When NASS makes an estimate it must be *statistically defensible*. Whatever definition one adopts for statistical defensibility, it must require, at a minimum, that NASS be able to describe what was done at each stage of the process of forming an estimate. Since using record linkage to eliminate duplication from the sampling frame is a part of ensuring unbiasedness, it is important for NASS to be able to describe in detail how each stage of record linkage works. This makes "black box" or proprietary linkage systems unsuitable for NASS's purposes. It also makes "ad hoc" systems, whose statistical properties are not understood, unsuitable.

Cost and Simplicity--A record linkage system that is too costly, complicated, or difficult to use is of little value, no matter its technical sophistication. Initial affordability, low operational cost, and ease of use are design requirements for any new linkage system.

Documentation and Training--Any new software should come with complete user documentation. In addition, training should be provided by the vendor, at least to a core group of users who can train others.

Maintenance and Support--Any commercially available software incorporated into the system needs to be well documented and supported by a reliable vendor. Updates to the software should be available at reasonable cost as they are developed.

Readiness for Use--Resources for additional development are scarce. Whatever commercial off-the-shelf (COTS) software is chosen, it should be as nearly ready to meet NASS's requirements as possible without much additional systems development.

Hardware and Software Environment

This section covers the impact of the hardware and software environment in which record linkage software will run and the modes in which the software will be used.

Computer Hardware--The choice of a hardware platform for record linkage is dependent on the choice of a platform for ELMO and on a decision by the ELMO Team regarding where record linkage is to run. Ideally the agency should choose record linkage software that is available on a variety of platforms to allow flexibility in implementation strategies. Any new record linkage software should run on a minicomputer, workstation, or desktop computer.

Operating System and Database Software--The ELMO Team has chosen Sybase database software running under a UNIX operating system. Ideally, any new record linkage software can use embedded Structured Query Language (SQL) code to directly access the database, interface with Sybase, run under UNIX or DOS/Windows. (Note: Sybase offers a product called "Open Client" which consists of a library of C functions which can be used by a C program to generate SQL for accessing the database.)

Real-Time Duplication Checking--A long-term goal of any new software is to be available to link single-record additions or modified existing records to the database to check for duplication in real time. The system must run fast enough to make such real-time duplication checking realistic. If software is purchased from an outside vendor, it must support interactive real time duplication checks.

Capability of Running in Batch Mode--The record linkage system should run in *batch mode*. This is necessary to accomplish linkages using large files with a reasonable expenditure of resources.

Record Linkage Methodology

This section covers methodological requirements for record linkage software.

Flexible Blocking--The software should allow users to choose blocking variables. Further, blocking should be done so that the additional linkage error due to the blocking can be estimated, to minimize block size without increasing false nonlinks to an unacceptable degree.

Flexibility in Use of Linking Variables--Some list sources may be richer than others in potential linking information. Beyond a core set of identifiers (which should be the same as the required set of identifiers for database integrity on the ELMO database) the system should have some flexibility as to what fields to attempt a match on. Also, the system should handle missing linking variables. Ideally, users will choose the treatment they want for missing variables.

Efficient Use of Available Information--Any software used by NASS should use all of the identifying variables available. Further, it should partition the variables to extract the maximum amount of discriminating power. (For example, a name should be partitioned into all of its parts, as discussed in the section on standardization above, or a phone number should be partitioned into the area code, exchange, and last four random digits.) Information about resolution of previous linkage runs on the same file(s) should be available. If a record pair is considered to be a possible link by the record linkage algorithm, but has previously been determined to be a link or nonlink in earlier resolution work, and no changes have been made to either record, then the system should act on this information to eliminate the pair automatically from the manual review group.

Choice of An Appropriate Comparison Function--A key element of the record linkage problem is recognizing when two values for equivalent variables that are being compared are the same. This may not be trivial, due to problems such as finding equivalent variables in files from two different sources, recognizing equivalent values (say, a proper name and a nickname), or recognizing that one variable contains the same value as another variable, but with some sort of error introduced (like a transposition or a dropped digit). A new record linkage system must have an efficient and flexible way of making comparisons.

Use of Prior Linkage Results in Developing Accurate m-Probabilities--Results of prior linkages of similar files could be used to help estimate more accurate m-probabilities for subsequent linkages. New linkage software should have a utility for estimating m-probabilities.

Estimation of Linkage Cutoff Weights in Fellegi-Sunter Systems--Identifying the cutoff points for the linked and nonlinked decision is one of the most important tasks in doing a linkage using the Fellegi-Sunter method. These cutoffs depend on knowledge about the accuracy of variables and the frequency of occurrence of values of those variables. Current technology does not allow precise estimation of these cutoffs; they are usually estimated by performing a pass of a linkage and examining the results. The pass can then be rerun with the cutoffs adjusted as needed. These cutoffs will vary from file to file, and the cutoff points should be reestimated for each new pair of files to be linked. (It may be possible, when several linkages are performed with files from the same sources to use a predetermined set of cutoffs without losing too much efficiency.) Any new record linkage software must accomplish the task of estimating these cutoff points in an efficient, user-friendly fashion.

File Size Constraints/Limitations--The software must be able to handle files of 250,000 (or preferably more) NASS list frame records.

Data Handling

This section covers issues regarding different types of data that will be used in record linkage and the requirements for handling those data.

Handling of Names--Many issues must be dealt with when handling names. NASS receives list sources with names in many formats. There is the usual split between sources with names in signature format (Charles Day) and surname on the left (Day, Charles). In addition, there are business names and names of partners. Some of these may appear in unusual formats. For example, for a few states NASS receives lists from FSA with the name in the format "Smith/Jones/Williams jt vent." These name forms are mixed with others in more conventional formats. Name standardization software for use in NASS must be powerful and customizable so that unusual name forms are handled properly. Not the least of these is the format (and possible variations in the format) of the name. In addition to format, there are the usual problems of nicknames, multiple names, initials only, titles (both present and omitted), suffixes (such as Jr.), "care of's," abbreviations, and misspellings. Standardization software must overcome all but the last of these challenges. Misspelling is usually handled either by phonetic coding or by a string comparison function of some kind.

Handling Different Kinds of Operating Arrangements--Different types of operating arrangements require different procedures. Linkage software needs to be able to treat business and partnership names differently than individual names, either by using multiple passes, or by providing for the use of different kinds of name comparisons in a single pass.

Addresses--Addresses present a similar problem to names. Again, the same address may be presented in different formats. It may contain abbreviations, building names, room numbers, names of intersecting streets, rural routes, highway contract routes, box numbers, apartment numbers, abbreviations, inconsistencies (e.g., Sawyer, VA when there is no place named Sawyer in Virginia) and misspellings. This will be complicated for the next few years by the planned conversion of rural route addresses to locatable

("911") addresses. The same considerations with respect to the power to discriminate between different sampling units with similar addresses while retaining the ability to associate records representing the same unit that have dissimilar addresses apply here as with names.

Verification of Place Names and ZIP Codes--The current record linkage system uses a dictionary of place names with a range of legal 5-digit ZIP Codes to verify place names and ZIP Codes. Some system for accomplishing this task needs to be part of the new record linkage system. This may be accomplished using an ELMO utility.

Assignment of Longitude and Latitude--The current record linkage subsystem assigns a longitude and latitude to each farm record based on the city in the address and then calculates distance between locations based on this information for linkage purposes. New linkage software should use such a measure of geographic "nearness" in the linking process.

Other Data--When other identifiers (such as SSN, EIN, telephone number, or control data) are available, a linkage system should use them. Linkage software should be flexible enough to use whatever combination of common identifiers is found on two files.

Post-linkage Functions

This section covers the functions needed after record linkage itself has been performed.

Resolution of Record Linkage Output--The current RECLSS generates printed output and requires batch input of corrections. Any new record linkage system or modification of the RECLSS should provide an online (ideally graphical) interface for resolution of "possible" links. The resolution system must have sufficient functionality to do all of the tasks that are currently handled with the resolution printouts, including recording comments, splitting the review between multiple reviewers, and keeping track of calls to farmers. An online resolution capability should be available even where the linkage itself is performed in batch mode. The option of generating paper output should be available.

Generation of Reports and Extract Files--Record linkage software should include utilities for the generation of reports and the creation of files of linked and unlinked records. Reports can be used to evaluate the results of changes in linkage specifications, including the application of different blocking schemes, the employment of different linking variables, the use of multiple passes, and the setting of different cutoff weights. The software also needs to be able to extract files of linked and unlinked records in a format that makes them easy to use to update the ELMO database. The software should provide a utility for doing this.

Apply Database Integrity Standards to New List Sources--Reformatting of new list sources should be separate from the linkage step. Updating the database with new records or updating existing records will be done using ELMO utilities that enforce the active data dictionary in order to ensure database integrity.

Reporting of Statistics on the Linking Process--Record linkage software must report relevant statistics at each stage of the linking process. Ideally, the software should also allow estimation of linkage error rates.

REVIEWS OF SOFTWARE PACKAGES

This section presents reviews of the six software packages. Figure 3 summarizes these reviews.

AUTOMATCH/AUTOSTAN (MatchWare Technologies)

General--AUTOMATCH is a generalized record linkage system, applicable to a variety of uses. AUTOSTAN, its companion program, is a highly customizable system for the standardization of name and address information. The methodology employed in AUTOMATCH is based on the popular Fellegi-Sunter theory. None of the techniques used in the system is considered proprietary, thus, the system meets the minimum standards for statistical defensibility. Costs for AUTOMATCH vary, from $3,000 for a single copy for a DOS-based personal computer to $9,995 for a license for a UNIX server. (All prices are as of August 1995.) Complete documentation is provided with AUTOMATCH, as is a one or two day training period for one user. One year of maintenance and support (including upgrades to any new versions issued during the year) is included in the price. Additional years of support are available at a reasonable fee ($600 a year for a DOS copy). One concern with this package is that it is the product of a small firm that is highly dependent on its owner. If the agency adopted AUTOMATCH as its record linkage solution, a software escrow, in which MatchWare agrees to allow NASS access to source code should MatchWare cease operations, would need to be arranged, to protect the agency against the company's ceasing operations. It is possible, although unlikely, that the agency might have to do without outside support for the software.

Hardware and Software Environment--A final decision on a hardware platform for ELMO has not been made yet; a dedicated minicomputer at the Lockheed Martin facility in Orlando, Florida, now appears the most likely alternative. AUTOMATCH is available to run on a number of hardware and software platforms, including: IBM-compatible desktops running MS-DOS, OS/2, and Windows NT; minicomputers and workstations running UNIX; and custom mainframe computer versions. The software is also available as Windows dynamic link libraries (DLLs) and UNIX callable libraries for construction of interactive systems. Callable libraries for a UNIX server cost $5,000 in addition to the batch price quoted earlier. ELMO will be implemented using Sybase. Sybase Corporation itself has successfully used AUTOMATCH in an interactive mode with Sybase. An ASCII file would need to be extracted to run in a batch mode.

Record Linkage Methodology--AUTOMATCH is based on the Fellegi-Sunter record linkage theory. The system is designed for linking in multiple passes, with the unlinked records from each pass proceeding to the next pass. Passes may vary on blocking variables or linking variables. Allowing multiple passes with the same linking variables, but different blocking variables, is valuable, since it reduces missed linkages due to errors in the blocking variables. (For example, if the probability of error in two blocking variables were independent, and were 1% for each variable in each file, then the expected percentage of missed linkages from blocking would be

Figure 3.--Comparison of Selected Features of Record Linkage Software Packages

Feature	AUTO-MATCH	GRLS	SSA-Name3	Smart PID	Merge/ Purge Plus (Group 1)	Merge/ Purge 4.3 (Postal-soft)
Cost ($US as of 8/95)	$9,995	$21,650	$33,000	Nego-tiable	$10,000	$20,000
Type of Package	Gener-alized	Gener-alized	Com-ponents	Com-ponents	Direct Mail	Direct Mail
Runs under UNIX	X	X	X	X	X	X
Compatible with Sybase	X	[1]	X	X	X	X
Interactive capability	X[2]	X	X	X	X	X
Uses probabilistic link-ing methodology	X	X		X		
Choice of methods for comparing variables	X	X[3]				X[4]
Standardizes names	X		X	X	X[5]	X[6]
Standardizes addresses	X				X[5]	X[7]
Verifies ZIP Codes					X[8]	X[7]
Interactive "possible link" review	X	X	[10]	[10]		
Generates reports	X	X[9]	[10]	[10]	X	X
Extracts files of linked data	X	X[9]	[10]	[10]	X	X

[1]GRLS requires the files to be linked to be in an ORACLE database. [2] AUTOMATCH is capable of interactive operation with the purchase of callable libraries at an additional cost of $5,000. [3]In addition to a rich library of standard comparison types, custom types can be programmed in C. [4]Standard routines offer no choice of comparison types, but custom types can be programmed in C. [5]Name and address standardization requires the List-Conversion package at an additional cost of $7,500. [6]Name standardization requires the True-Name package at an additional cost of $20,000. [7]Address standardization and ZIP Code verification require the Code-1 package at an additional cost of $20,000. [8]ZIP Code verification requires the Address Correction and Encoding (ACE) package at an additional cost of $42,000. [9]Report generation and data extraction are accomplished through the use of the ORACLE database package. [10]These functions would be accomplished by a system built around the SSA-Name3 or Smart PID modules.

reduced from 1.99% for one pass to 0.0396% in the two-pass case.)

AUTOMATCH can handle numeric identifiers and name and address variables. The user may use these variables in any combination. The user may specify the treatment of missing values. Since the user defines the variables as he wishes, numeric identifiers, like phone numbers and SSNs (which have nonrandom components, see Jabine [12]) can be partitioned and the parts treated as separate variables. With AUTOSTAN, name and address variables can be partitioned into their separate parts as well. Information about the resolution of previous passes can be applied to subsequent passes automatically, so there is no need for a user to resolve the same possibly linked or duplicate pair more than once during a particular linkage. However, no provision exists for carrying this information from one linkage to another. (For example, it would be quite useful in unduplicating a file that is checked annually to be able to apply the results of previous linkage determinations to pairs of unchanged records which appear year after year as possible links. AUTOMATCH has no such capability.)

AUTOMATCH offers many options for comparing different kinds of variables. For comparing name variables, AUTOMATCH offers both an information-theoretic string comparison function and support for the two most popular phonetic coding systems, Soundex and a modified version of NYSIIS. In addition to a straight character for character comparison, AUTOMATCH offers many specialized types of comparison. These include allowing an absolute number of characters (digits) to be different, allowing for percentage differences, comparing dates and times, and a function for comparing distances using latitude and longitude.

AUTOMATCH calculates u-probabilities from the frequency of occurrence of different values for each linking variable from the files being linked. AUTOMATCH also has a utility for calculating m-probabilities from completed passes. This utility could be used to estimate starting m-probabilities for similar linkages from previous years' results. With this utility, a user can iteratively estimate m-probabilities from current linkages. AUTOMATCH also allows the user to produce a histogram of linkage weights to guide estimation of weight cutoffs. Perhaps more useful is the report generation feature, which allows custom report specification and specification of reports including only pairs with weights in a particular range. Examination of these reports is useful in setting cutoff weights. In several test projects, including one for Texas, which has NASS's largest list frame, no difficulties were encountered handling files of any size.

Data Handling--AUTOSTAN, AUTOMATCH's companion software for name and address standardization, is a powerful, customizable system for standardizing these fields. AUTOSTAN is used by associating a data field, such as primary name, with a "process." Each process consists of three files. The first is a "dictionary" which specifies the format of the output record; that is, it specifies the variables into which the field being standardized will be parsed. The second file, called a "classification" file, consists of a list of words (like surnames or titles or street types), each of which is associated with a particular class or type. The third file is a "pattern" file. This file consists of patterns of the various classes of

variables in the classification file, along with some generic classes, and rules for parsing each pattern into one or more of the output fields specified in the dictionary file. AUTOSTAN comes with some standard processes; users can use these processes as they are, customize them, or create their own processes from scratch. Unfortunately, no process that conforms to U. S. Postal Service (USPS) addressing standards is included (although a user could create such a process). Because no such pattern has been certified by the postal service, there is some question about the availability of postal discounts using AUTOSTAN-standardized addresses.

AUTOSTAN is capable of handling the types of name and address problems encountered in NASS. In addition, it is possible to create separate fields for handling different types of operating arrangements. One function which AUTOSTAN does not perform is the verification of place names and ZIP Codes and addition of latitudes and longitudes to records. If NASS chooses to use AUTOMATCH, then the agency must write or purchase additional software to perform this function. The Group 1 and Postalsoft products reviewed later are capable of performing this function.

Post-Linkage Functions--AUTOMATCH is a "mixed bag" when it comes to performing post-linkage functions such as possible link resolution, report generation, and data file extraction. Possible link resolution is done with the clerical review program. This program allows review of possible links and duplicate pairs. Actions can be taken to make the possible link pair a link, a nonlink, or leave it for later review, when more information can be obtained. Additional actions that swap a "duplicate" record for the linked record from the same file and allow the user to move backward and forward within the file of possible links are available. This utility falls short of NASS's needs in some critical areas. First, only one user at a time can work on the file of possible links. Second, there is no provision for the creation of hypertext comments in the same way that a person working with a printout could make remarks on the printout. Third, there is no access to the data records on the database. There is no provision for accessing comment fields on the database at the time of resolution. There is also no provision for resolving some records for which a decision on their status is easy, and returning later, after more information has been gathered, to resolve others. Finally, the interface is character based and requires some skill in specifying the report format to get a useful format for possible link resolution. (The software for reviewing possible links uses the custom report format, but it displays these data fields as three 80-character lines. Care has to be taken to design a format that does not break fields. The report generation utility uses this format as well, which is inconvenient.)

As mentioned above, AUTOMATCH has a report generation function which allows the user to specify custom report formats. These reports are generated as ASCII text files, with printer control characters embedded. One feature not included is the ability to add a line of space between groups of linked records. Besides the report generation function, AUTOMATCH also has a file extraction function. It is possible to specify an extract of any combination of variables from either File A or File B for linked records and for unlinked records. These extracts are necessary for updating ELMO. An ELMO utility would be used for actually

adding records to the database, or for eliminating duplicate records. AUTOMATCH also can generate a number of different levels of statistics about the linking process. The user may specify as little or as much information about each pass as he or she wishes [13].

***Generalized Record Linkage System (GRLS)* (Statistics Canada)**

General--GRLS is also a generalized record linkage system, applicable to many uses. Statistics Canada developed the GRLS system using ORACLE and custom C language routines, creating a powerful and highly customizable record linkage system. This power comes at a price: GRLS requires ORACLE database software and the UNIX operating system package to run. While ELMO remains a Sybase database, this limitation all but prohibits the use of GRLS in NASS, as it would require the purchase of two enterprise database packages, and a port of the data from one system to another each time record linkage was run. Unlike AUTOMATCH, which has companion name and address standardization software, GRLS does not offer this capability. Like AUTOMATCH, GRLS is based on the statistically defensible Fellegi-Sunter record linkage methodology.

As of August, 1995, the cost for a site license for GRLS is Can$30,000 (about US$21,650 at current exchange rates). This price includes installation, technical support, and one week of training. This price represents a significant increase over earlier prices, and Statistics Canada suggests that most of this increase represents funding for product improvements and for better technical support. Statistics Canada is committed to GRLS as its official, in-house record linkage solution, so there is little need for concern that the product will be unsupported in the future.

Hardware and Software Environment--As mentioned earlier, users of GRLS are limited to UNIX workstation and minicomputer environments. In addition, ORACLE and a C compiler must be present, and the files to be linked must be in an ORACLE database. Interactive linkage is a natural mode of operation for GRLS. Linkages can, of course, also be performed in batch mode.

Record Linkage Methodology--GRLS is a Fellegi-Sunter system. It performs multiple pass linkages. Like AUTOMATCH, passes may vary on blocking variables or linking variables. GRLS can handle both name and address and numeric identifiers. It is possible to specify the weighting treatment of missing variables. The user can partition variables in any way needed. For example, whether a phone number is treated as a single number or whether area code, exchange, and random digits are treated as different variables depends upon how the user defines them in his or her ORACLE database. GRLS assumes that the name and address variables have been partitioned before loading them into the database. GRLS has no independent facility for parsing names and addresses. Possible matches are resolved at the end of the linkage process rather than between passes. Unlike AUTOMATCH, GRLS uses the capabilities of ORACLE to mark record pairs so that the system can detect and automatically resolve possible matches which have been resolved in a previous linkage or unduplication effort. This eliminates the need to resolve the same pairs time after time. Also, GRLS allows the user to write the linkage rules in a more

English-like format than AUTOMATCH, which uses a parameter file approach. Both GRLS and AUTOMATCH allow the user to include blocking variables as matching variables so that weights properly reflect agreement on common or rare values of the blocking variable.

Many different comparison options for different kinds of variables are available in GRLS. These include an improved version of the string comparison metric used in AUTOMATCH (by setting options, this function can duplicate AUTOMATCH's comparison function). In addition, custom comparison functions can be programmed by the user in C and linked to the GRLS system. This is a powerful capability, as it allows the user to build any special knowledge he or she may have about a particular variable into the comparison function used for it. Also, it allows the sophisticated user to do any kind of comparison; he or she is not limited to the comparison types contained in the package.

GRLS uses an Excel spreadsheet to estimate u-probabilities from the relative frequencies of the values of the fields. M-probabilities would have to be estimated iteratively from a sample of the linked pairs. Like AUTOMATCH, GRLS can produce a histogram of linkage weights for linked and unlinked pairs; however, Statistics Canada recommends using the iterative method for setting cutoff weights. Since GRLS has all of the report generation capabilities of ORACLE at its command, it is easy to create reports for purposes of evaluating cutoff weights. GRLS can handle files of any size that ORACLE can handle.

Data Handling--GRLS requires that data be contained in an ORACLE database. GRLS does not have a powerful name and address parsing capability built in. The assumption is that the variables are parsed into their component parts when they are entered into the database. Some standardization may be accomplished using the Automated Coding by Text Recognition (ACTR) facility. This system allows the user to define standard codes or abbreviations for descriptive phrases or words. The lack of powerful name and address parsing is a serious limitation for NASS; many of NASS's lists have free-formatted name and address fields which require parsing. Other than this limitation, GRLS can perform any sort of manipulation on the data that can be done with ORACLE.

Post-Linkage Functions--GRLS shines in its impressive set of post-linkage capabilities. Again, the power of the ORACLE database package gives GRLS its advantage. GRLS can make any kind of extract or create any kind of report that can be created using ORACLE, giving the users virtually unlimited options. On top of this, the software includes a Graphical User Interface (GUI) based utility for resolving possible matches. Multiple users can resolve records at one time, and comments from the database can be accessed at the time of possible link resolution [14].

SSA-Name3 and Extensions **(Search Software America (SSA))**

Unlike AUTOMATCH and GRLS, which are generalized systems, ready to perform linkages with a minimum of installation and setup time, SSA-Name3 and Extensions are sets of software components for building a custom record linkage system. This is a different paradigm, and assumes that the developers and users of a record linkage system have special needs, not met by the

generalized programs, which justify the expenditure of development resources. Companies with large databases, such as Federal Express, Visa, GEICO, and AT&T, use SSA-Name3 to generate keys for efficient database searches. The Extensions product includes routines for "scoring" agreements to perform matches. Costs for SSA-Name3 vary; as of August, 1995, a license for a small UNIX box with 45 concurrent users is $24,000 for SSA-Name3 and $9,000 for the Extensions product. The price for SSA-Name3 for a high end UNIX system is $46,000. The purchase price includes documentation and one year of unlimited telephone technical support. Search Software America is a subsidiary of SPL World Group Software, which has 550 employees worldwide.

Hardware and Software Environment--SSA will create a version of SSA-Name3 to run on any platform. A developer can create either a batch system or an interactive system from the software modules. Depending on the type of system constructed, the linkage could be performed directly against the database.

Record Linkage Methodology--SSA-Name3 does not use the Fellegi-Sunter record linkage methodology. In effect, SSA treats the record linkage problem like a database search problem. The software uses compressed, fixed-length 5-byte keys, based on an enhanced NYSIIS name coding system, to efficiently locate potential matches. After cleaning, reformatting, and standardizing the name, SSA-Name3 computes statistics for the frequency of occurrence of standardized words. Based on the frequency of these words, SSA-Name3 builds keys. Each key can contain information from up to four words in the name. It then isolates the most productive key for search purposes. It then establishes "search sets and ranges" based on this key. Multiple keys are used to allow for differing word sequences in the name. Using these keys, a system can be built that establishes candidate sets for linkage.

The Extensions package provides comparison tools that allow the members of the candidate sets to be "scored" on a scale of 0 to 100 to determine whether two records fall into the match, suspect match, or no match categories. The developer can decide on cutoffs for these categories or allow a user to set the cutoffs. The scores are calculated according to user-defined schemes for combining the results of comparisons between identifiers such as names, addresses, phone numbers, and other numeric identifiers. The use of the comparison tools does not provide a statistical basis for comparing the records. Variables may be partitioned as needed. The system is not inherently a multiple pass system. Any capability for the system to use previous linkages' information in resolving current possible links would have to be built into the system surrounding the SSA-Name3 and Extensions modules.

Data Handling--SSA-Name3 accepts up to 256 characters in a free-formatted name field as input. SSA parses this name into at most eight words made up from a limited character set. The character table can be customized. The name is then formatted by processing it using a rule base of up to 64,000 rules which can be applied to improve recognition of common variations by "[removing] noise words, concatenat[ing] or attach[ing] prefixes and suffixes, replac[ing] abbreviations and nicknames, translat[ing] logical equivalents, etc. [15]." The rules can also be customized.

SSA does not provide address standardization.

Post-Linkage Functions--SSA does not provide routines for doing post-linkage functions. These are the developer's responsibility. This allows maximum flexibility, but also means that development resources will be required to build the post-linkage functions into the system [16].

***Smart PID* (Advanced Linkage Technologies of America (ALTA))**

ALTA is owned and operated by Jerry Weber and Max Arellano. The company focuses on providing record linkage functions to be integrated into hospital management information system (MIS) applications. Like SSA-Name3, Smart PID is a collection of software modules for the construction of record linkage systems. These modules are written in Pascal and C, and, while originally designed for use in an IBM MVS environment, have been successfully recompiled to run on other machines, including UNIX machines. Visual Basic has been used successfully in a Windows environment to create systems around the Smart PID modules. The methodology employed in Smart PID is based on the Fellegi-Sunter record linkage theory. Costs for Smart PID are negotiable, and are based on the number of records in the system. The costs are usually spread over several years, but a single up-front payment can be negotiated (based on the present-value of the yearly costs). Documentation is provided with Smart PID, as are a couple of days of Max Arellano's time to help with installation and training. The purchase price includes telephone support for development and use of the software. The same concern that exists with MatchWare exists with this company. It is a very small firm, and a software escrow would need to be negotiated to protect NASS against the firm's ceasing operations.

Hardware and Software Environment--The Smart PID modules can be recompiled to run in many environments. The software has successfully been used with a Sybase database. One drawback of this software for NASS purposes is that ALTA designed it for use almost exclusively in an interactive mode. While batch mode operation may be possible, it is not how the software is designed to be used.

Record Linkage Methodology--Smart PID is based on the Fellegi-Sunter record linkage theory, along with some enhancements that improve performance in the medical record look-up setting. The system is designed for a minimum of user intervention. This can be valuable in a setting where the user is unlikely to have expertise either in record linkage or in the use of Smart PID. The system attempts a match using up to four blocking strategies. The first is blocking on SSN (ideally, this should yield unique records); the second is blocking on the Soundex code of the first name and the birthday; the third is blocking on NYSIIS code of the last name. (The third scheme is equivalent to the NASS RECLSS.) There is also the capability for a system developer to specify a fourth, custom blocking strategy.

As a part of the strategy of allowing the user limited options, Smart PID allows the use of the following linking variables: Name, Address, Telephone Number, SSN/EIN, and birth date. It is possible to add other identifiers. The user also chooses the matched, unmatched, and possibly matched cutoff weights. Smart PID can parse and standard-

ize a name field using look-up tables that the user can customize. Address "standardization" is rudimentary, consisting of removing a substring from the beginning of the address field. Smart PID differs from the other Fellegi-Sunter systems discussed in this report in the use of two additional codes (besides the Fellegi-Sunter weight) to decide if a record pair is a match. These are the validity code, used to decide whether special circumstances exist which might invalidate a high Fellegi-Sunter weight, and the confirmatory code, used to help decide whether a low-weight possible match should be considered a match. For example, ALTA has used the validity code in cases where an individual has used his or her spouse's SSN. A match on an accurate, unique identifier such as SSN, combined with an identical address, surname, and phone number would normally yield a high Fellegi-Sunter weight, resulting in a match; however, use of the first name to generate a validity code would catch this false match. ALTA uses the confirmatory code in cases of sparse information, where the Fellegi-Sunter weight might be too low to find a match because of a large number of missing variables. Address, SSN, and phone number are generally used in forming this code. Both near-identical records that are not matches and sparse records can cause problems in NASS's files. Any provision for using information from previous linkages to resolve a current linkage would have to be made using the system in which the Smart PID modules were embedded. Smart PID does not offer optional methods for comparing identifiers. Smart PID calculates u-probabilities using frequency counts and m-probabilities based on empirical estimates of the accuracy of the variables. Any limitations on file size would come from the embedding system, not from the Smart PID modules. (Two hundred fifty thousand records is not a large file for Smart PID.)

Data Handling--Smart PID can standardize and parse name fields based on a customizable look-up table. No real address standardization is available. A concern with the Smart PID system is the limited number of variables that it uses. Also of concern is its reliance on birth date, which is not available in NASS's list frame files, and on SSN, which is not available for some records.

Post-Linkage Functions--Possible links are displayed in order by their Fellegi-Sunter weights, along with their validity and confirmatory codes. The system that is to be developed around the Smart PID modules handles resolution of possible links and creation of extract files and reports. Users have successfully used nonprogrammable databases, like dBase, to download the results of Smart PID comparisons and produce reports [17].

***Merge/Purge Plus and Code-1* (Group 1 Software)**

The Merge/Purge Plus and Code-1 products from Group 1 Software represent yet another type of record linkage software. Rather than being components from which a record linkage system is built or generalized record linkage packages, these packages have been developed for the specific purpose of unduplicating mailing lists and standardizing addresses for direct mail marketing and subscription organizations. Group 1 claims an installed base of over 3,000 users. As of August, 1995, on the GSA schedule, Merge/Purge Plus costs $10,000 for a single UNIX box (any number of users); its companion product for standardization for linkage purposes, List-Conversion, costs

$7,500. Code-1, the postal name and address standardization software costs $21,000. These prices include: installation; training at one of Group 1's eight training centers (one is in Washington, D.C.); maintenance; and, for Code-1, a new copy of its database of deliverable U.S. addresses every three months for two years. The price of the software includes unlimited telephone hotline support.

Hardware and Software Environment--Merge/Purge Plus runs in many UNIX environments, including Sun Sparc, Hewlett Packard, NCR, AT&T, and IBM AIX. It is also available in a Windows NT version. The software can operate either as an on-line system or in a batch mode.

Record Linkage Methodology--Merge/Purge Plus is not a Fellegi-Sunter system. It uses a simple approach, comparing the linking variables specified by the user using a character-for-character approach. Loose, tight, or medium agreements can be specified by the user. The system then uses counts of agreements and disagreements and weights provided by the user to compute an overall weight. Priorities can also be set so that records can match on a single variable or combination of a few variables. The system makes a single, unblocked pass through the data, relying on sorting the actual data file (or, at least, a file of match variables and an identifier) to aid efficiency.

Merge/Purge Plus can use a combination of name, address, and numeric variables for matching. List-Conversion parses and standardizes names and addresses and produces a file similar to that produced by AUTOSTAN, with the original data appended to the standardized name and address data. There is no provision for using information from prior linkages in new linkages. There are no options for different methods of comparing variables.

Data Handling--List-Conversion parses and standardizes free-formatted name and address lines for matching purposes. The Code-1 product reformats, corrects, and enhances address information. It performs this function by matching the address against a database of all the deliverable addresses in the United States. This is a reformatted version of the USPS file. The system cannot reformat an address it cannot find in the database [18]. The software meets USPS certification standards, and, unlike the packages reviewed above, performs a verification of ZIP Code and place names. Merge/Purge Plus does not directly access database products.

Post-Linkage Functions--Merge/Purge Plus can extract many different types of files, including any files NASS would need. In addition, many different kinds of linkage reports are available. Since Merge/Purge Plus does not produce possible matches, no possible match resolution software is needed. It is possible to list record pairs with a combined weight (overall weight for all variables) above a certain level [19].

***Merge/Purge 4.3, Address Correction and Encoding (ACE), and True Name* (Postalsoft)**

Like Merge/Purge Plus, Merge/Purge 4.3 has been developed for the purpose of unduplicating mailing lists and standardizing addresses for direct mail marketing. Postalsoft has an installed base of about 500 copies of Merge/Purge 4.3. The company has approximately 160 employees. As of Au-

gust, 1995, Merge/Purge 4.3 costs approximately $20,000-$25,000 for installation on a UNIX system with lists about the size of NASS's list frame (for the whole United States). Its companion software for address standardization purposes, ACE, costs $42,000, and Postalsoft's name standardization software, True Name, costs $20,000. These prices include documentation and 90 days of technical support and updates. One year of extended support and updated versions costs 16.5% of the purchase price of the product. Postalsoft generally produces 2-3 updates per year. Updated databases of deliverable addresses for ACE cost $4,500-$6,000 per year.

Hardware and Software Environment--Merge/Purge 4.3 runs in DOS, UNIX, and IBM mainframe operating system environments. The software operates in a batch mode and exists as callable libraries for building interactive systems. The package can access Sybase directly.

Record Linkage Methodology--Merge/Purge 4.3 is not a Fellegi-Sunter system. The software compares as many as 35 variables on a pair of records. For each variable the outcome of the comparison is a score between 0 and 100. The user can set the cutoff level that will be considered an agreement. Similarly, for the overall agreement score (also on a scale of 0-100) the user can set the minimum score for a pair to be considered a duplicate. Tools are available to output the raw scores. There is no statistical approach taken to determining the likelihood of a match.

Merge/Purge 4.3 uses a combination of name, address, and numeric variables. ACE and True Name can parse address and name information for matching. There is no provision for using information from prior linkages in new linkages. There are no optional routines for different methods of comparing variables included with the software, but Merge/Purge 4.3 Custom allows the use of user-programmed (in C) custom comparison routines.

Data Handling--ACE and True Name parse free-formatted name and address lines. ACE also attempts to correct addresses and add ZIP Codes besides putting addresses in USPS format. ACE uses a database of deliverable address information to which it matches the incoming address and corrects and supplements the address if necessary. As is true for Code-1, this product cannot parse an address it does not have on its database. On the other hand, it can probably flag undeliverable or nonexistent addresses. The software is USPS CASS certified. The package has been used successfully with a Sybase database and in interactive mode.

Post-Linkage Functions--Merge/Purge 4.3 can extract many different kinds of files and produce many reports. The extraction and report generation features are customizable. Since Merge/Purge 4.3 does not produce possible matches, no resolution software is needed [20].

CONCLUSIONS

The packages reviewed in this report fall into three categories. The first, including AUTOMATCH, and GRLS, contains generalized record linkage packages. These packages are ready to perform linkages as they come from the manufacturer. They are both powerful packages that give a systems developer or user a flexible tool, allowing him or her

maximum flexibility in specifying how linkages will be performed and how the results will be displayed. As a generalized system, they require the least additional software development resources to meet NASS's needs. Both employ the Fellegi-Sunter record linkage theory.

The next category contains packages termed "components" in this report. This category contains SSA-Name3 and Smart PID. Both packages require significant additional development resources to build a system around the linking subroutines provided by the manufacturer. In addition, both are aimed at specific target markets different from government statistical agencies. Smart PID uses the Fellegi-Sunter method, along with some enhancements. There does not appear to be statistical justification (other than the justification for using NYSIIS in name searching) for the methods used in the "scoring" routines of SSA-Name3's Extensions package.

The final category is that of mailing list management software. Merge/Purge Plus and Merge/Purge 4.0 fall into this category. Both packages provide the advantage of needing little additional development to create a usable system, but neither has a statistical foundation for its matching methods.

RECOMMENDATIONS

NASS should adopt AUTOMATCH and AUTOSTAN as its new record linkage solution and should commit the resources to adding the functionality needed to meet all of NASS's requirements. AUTOMATCH is the least expensive of the packages, yet it offers the most functionality for NASS applications. It will require the fewest resources to meet NASS's specifications. It is based on the proven Fellegi-Sunter record linkage methodology. No other package has a combined name and address standardization capability for record linkage purposes as powerful as that offered by the AUTOSTAN software. It is available on all of the platforms that NASS is considering for record linkage. Finally, it is available in a form that NASS can use in later versions of ELMO to create an interactive record linkage capability.

Two alternatives exist to the adoption of AUTOMATCH. The first is to develop in-house, either from scratch or by conversion of mainframe legacy code, NASS's own completely proprietary record linkage system. This is an expensive and time-consuming alternative which would only be justified by a belief that such a system would significantly outperform AUTOMATCH. There is no evidence that an in-house system would do so. AUTOMATCH has compared favorably to the current mainframe record linkage system in empirical tests [21]. While some methodological improvements to the current system might be possible, the return from the large investment of resources would likely not be great. Indeed, most of the methodological improvements would involve incorporation of methods available in AUTOMATCH or GRLS.

The second alternative is to choose another package that is commercially available as the basis for a new system. The most likely alternative package is Statistics Canada's Generalized Record Linkage System. It is a complete generalized solution for linkage, which offers a great deal of power and flexibility because of its close tie to the ORACLE database package. Alas, it is this

very strength that is GRLS's undoing for NASS. NASS's ELMO is a Sybase database, and GRLS simply does not work with Sybase. In addition, if GRLS were chosen, a name and address standardization package would still be needed, since GRLS has no standardization capability.

Smart PID is another Fellegi-Sunter alternative, but it would require more development resources to build a working system to meet NASS's needs than AUTOMATCH. In addition, its linkage approach is limited with respect to choice of variables, and depends on variables NASS does not have on its list frame files. Finally, it is more expensive.

SSA-Name3 and Extensions is also expensive and does not offer the assurance of the Fellegi-Sunter methodology. Again, it would cost more to build into a working system that met NASS's needs than AUTOMATCH, with no assurance that the final system would be as effective.

Both Merge/Purge Plus from Group 1 and Merge/Purge 4.3 from Postalsoft are not suitable to NASS because they rely on ad hoc methodologies. They are also expensive. They are best suited to their intended purpose of serving direct mail clients who do not have the expertise in record linkage techniques to effectively use more sophisticated packages, and need a package tied into a mailing list management system.

Taken together, these considerations all point to AUTOMATCH as the most appropriate choice as the core component for NASS's next record linkage solution.

This recommendation is based on NASS's existing and planned applications and is not in any manner a general recommendation on record linkage software outside of NASS.

REFERENCES

[1] Newcombe, H. B., Kennedy, J. M., Axford, S. J., and James, A. P. "Automatic Linkage of Vital Records," *Science*, Vol. 130, pp. 954-959.

[2] Fellegi, Ivan P. and Sunter, Alan B., "A Theory for Record Linkage," *Journal of the American Statistical Association*, Vol. 64, pp. 1183-1210.

[3] Federal Committee on Statistical Methodology, *Statistical Policy Working Paper Number 5: Report on Exact and Statistical Matching Techniques*, U.S. Department of Commerce, 1980, p. 1.

[4] As used in this report to refer to current practice in NASS, the term record linkage system refers not only to the Record Linkage Subsystem (RECLSS) but also to the attendant reformatting, data manipulation, identical match, and resolution programs necessary to complete the task of linking two sets of records and producing a single unduplicated file. For a description of this system, see Coulter, Richard W., and Mergerson, James W., "An Adaptation of a Record Linkage Theory in Constructing a List Sampling Frame," *Computing Science and Statistics: Proceedings of the Tenth Annual Symposium on the Interface*, National

Technical Information Service Special Publication Number 503, NTIS, 1978, pp. 416-420.

[5] A good description of the NYSIIS and Soundex phonetic coding systems can be found in Appendix H of Newcombe, Howard B., *Handbook of Record Linkage*, pp. 181-187, Oxford University Press, 1988.

[6] Tepping, Benjamin J., "A Model for Optimum Linkage of Records," *Journal of the American Statistical Association*, Vol. 63, pp. 1321-1332.

[7] Yu, C. T., Lam, K., and Salton, G., "Term Weighting in Information Retrieval Using the Term Precision Model," *Journal of the Association for Computing Machinery*, Vol. 29, pp. 152-170.

[8] Cooper, W. S. and Maron, M. E., "Foundations of Probabilistic and Utility Theoretic Indexing," *JACM*, Vol. 25, pp. 67-80.

[9] Van Rijsbergen, C. J., Harper, D. J., and Porter, M. F., "The Selection of Good Search Terms," *Information Processing & Management*, Vol. 17, pp. 77-91.

[10] Arends, William, "Selection of a Surname Coding Procedure for the SRS Record Linkage System," Sample Survey Research Branch, Research Division, Statistical Reporting Service (NASS), USDA, 1977.

[11] The state list frame statisticians who participated were Joe Ross (WA), Jerry Ramirez (VA), Dale Turnipseed (TX), Charles Ruckman (KS), Aubrey Bordelon (FL), and Roger Ott (IL).

[12] Jabine, Thomas B., "Properties of the Social Security Number Relevant to Its Use in Record Linkage," *Record Linkage Techniques--1985*, pp. 213-225.

[13] This review of the AUTOMATCH and AUTOSTAN software packages is based on the author's two year's of extensive experience with the packages, along with the experience of Kara Broadbent in the Ohio Research Unit.

[14] The evaluation of the Generalized Record Linkage System is based on reviews of documentation and conversations with Ted Hill and Martha Fair of Statistics Canada. The product reviewed will not be available until December 1995.

[15] Personal communication from Geoff Holloway of Search Software America.

[16] The review of SSA-Name3 and Extensions is based on a review of documentation, letters exchanged with Geoff Holloway of SSA, and conversations with Suyen Lyn of SSA.

[17] The review of Smart PID is based on an early DOS demo, review of documentation, and conversations with Jerry Weber of ALTA.

[18] Empirical tests at the Census Bureau indicated that the Code-1 product performed less well than their in-house address parsing software and AUTOSTAN. Note, however, that the latter two systems do not attempt address correction or ZIP coding, an important part of Code-1's capabilities for direct mailers.

[19] The review of Merge/Purge Plus and Code-1 is based on review of documentation and conversations with Tyrone Singh of Group 1.

[20] The review of Merge/Purge 4.0 and ACE is based on review of documentation and conversations with Ed Sugg of Postalsoft.

[21] Day, Charles, "Record Linkage II: Experience with AUTOMATCH in NASS," Survey Technology Branch, National Agricultural Statistics Service, USDA, forthcoming.

APPENDIX A--GLOSSARY

The following is a short glossary of record linkage and related terminology used in this report. The first occurrence in the text of each word in the glossary is *italicized*.

Batch Mode--A mode of operation in which the computer operates without input from the user.

Blocking--The division of one or more files which are to be linked into groups, or blocks, which agree in value for a set of variables (the *blocking variables*) with the intent that only record pairs within blocks with the same values will be compared. Blocking is done to reduce the number of comparisons to be made to a computationally tractable number.

Comparison Outcome--The result of comparing the values of the same linking variable on each of the records in a record pair. For example, if the first name on each of the records in a pair is John, then "first name is John and agrees" would be the outcome of comparing the two records on first name.

Duplication--The presence in a single file of multiple records representing the same unit in the population.

False link--The association of two records which do not represent the same unit in the population. Sometimes loosely called a nonmatch.

False nonlink--The failure to associate two records which do, in fact, represent the same unit in the population.

Fellegi-Sunter Theory--A theoretical approach to the problem of probabilistic record linkage. Put forward in a 1969 *JASA* article by Ivan Fellegi and Alan Sunter, the theory showed how to obtain the smallest number of records to review manually (possible links) in order to achieve given levels of false link and false nonlink using the probabilities of obtaining various comparison outcomes given that the records being compared do or do not represent the same unit in the population.

Identifiers--Words or codes, such as names, addresses, phone numbers, or Social Security numbers which can be used to discriminate between units in a population.

Interactive Mode--A mode of operation in which the user actively participates, in real time, by entering data and responding to prompts from the system.

Link--A link is said to occur if two records are associated by the record linkage method being used. The inference is that these records represent the same unit in the population, although the inference may be false in the event of a linkage error. Except when speaking strictly in the terminology used in the Fellegi-Sunter theory, link is a synonym for match.

Linkage Weight--In the Fellegi-Sunter theory, the log on the base two of the ratio of the m- and u-probabilities. Also used to refer to the sum of these logs for all of the variables involved in a particular linkage.

Linking Variables--Identifiers which are used in the process of linking records.

List Frame--In NASS, a list of farms and agricultural operations from which samples are chosen for NASS's estimation programs.

m-probability--The probability of an outcome or vector of outcomes occurring given that the pair of records being compared belongs in the matched set.

Match--A comparison pair whose records represent the same unit in the population. Except when speaking strictly in the terminology used in the Fellegi-Sunter theory, link is a synonym for match.

Matched set--The set of pairs of records which represent the same unit in the population; that is, the set of records which should be matched.

Nonlink--A nonlink is said to occur if two records are not associated by the record linkage method being used. The inference is that these records do not represent the same unit in the population, although the inference may be false in the event of a linkage error. Except when speaking strictly in the terminology of the Fellegi-Sunter theory, nonlink is a synonym for nonmatch.

Nonmatch--A comparison pair whose records do not represent the same unit in the population. Except when speaking strictly in the terminology used in the Fellegi-Sunter theory, nonlink is a synonym for nonmatch.

NYSIIS Code--A phonetic coding system used for overcoming spelling errors in alphabetic identifiers. (NYSIIS stands for New York State Intelligence Information System.)

Parsing--The process of dividing a name or address into its component parts.

Possible Link--A possible link is said to occur if no positive decision on the link status of a record pair can be made by the record linkage method being used.

Record Linkage--The association of records representing the same unit from one or more files representing the same population by comparing identifiers.

Record Pair--Two records, one from each of two files being linked.

Soundex--A phonetic coding system used for overcoming spelling errors in alphabetic identifiers.

Standardization--The process of substituting standard abbreviations for words in a name or address and of putting the name or address in a standard format. Sometimes also used to include the process of parsing the name or address.

Statistical Defensibility--NASS policy states, "An estimate ... is statistically defensible when ... the estimate is the product of a well-documented estimation strategy that is based on reasonable and clearly articulated assumptions." It further states that, "... estimation strategy includes ... determining the frame or frames," and "Where possible, unnecessary assumptions should be avoided." In a record linkage context, this means that the methodology used should be clear and should be grounded in statistical theory rather than ad hoc assumptions.

String Comparison Function--An arithmetic function that produces an agreement weight for the comparison between two strings of alphabetic characters based on the differences and similarities between the strings.

u-probability--The probability of an outcome or vector of outcomes occurring given that the pair of records being compared belongs in the unmatched set.

Unmatched set--The set of pairs of records which do not represent the same unit in the population; that is, the set of records which should not be matched.

APPENDIX B--A CHECKLIST FOR EVALUATING RECORD LINKAGE SOFTWARE

General

1.1 Is the software a generalized system or specific to a given application?

1.2 Is the software a:

Complete system, ready to perform linkages "out of the box?"

Set of components, requiring that a system be built around them? If so, how complete are the components?

Part of a larger system for performing integrated mailing list functions?

1.3 What types of linkages does the software support?

Unduplication (one file linked to itself)?

Linking two files?

Simultaneously linking multiple files?

Linking one or more files to a reference file (multiple-pass systems only)?

1.4 Can the software be used on the following computers:

Mainframes?

Mini-computer?

Workstation?

IBM-compatible microcomputer?

Macintosh?

1.5 Can the software run under the following operating systems:

MS/PC DOS?

OS/2?

Windows 3.1/95?

Windows NT?

UNIX?

VMS?

Mac OS?

Novell NetWare?

Mainframe OS (e.g. IBM MVS)?

1.6 For PC based systems, what level of processor is required? How much memory? How much hard drive space?

1.7 Can the system perform linkages interactively (in real time)? Can it operate in batch mode?

1.8 How fast is the software on the user's hardware and files the size of the user's files? If the software is interactive, is its performance adequate?

1.9 If the software is to be used as part of a statistical estimation system, are the methods used in the software statistically defensible?

1.10 Is the cost of developing a system for the intended purposes using the software within the available budget?

1.11 Is the vendor reliable? Can the vendor provide adequate technical support? Will they continue to exist for the projected life of the software? If this is in question, is a software escrow available? Is the user prepared to support the software him/herself?

1.12 How well is the software documented? Can a new user reasonably be expected to sit down with the manual and begin using the software, or will training be necessary? Does the vendor provide training? At what cost?

1.13 What features does the vendor plan to add in the near future (e. g., in the next version)?

1.14 Is there a user group? Who else is using the software? What features would they like to see added? Have they developed any custom solutions (e. g., front ends, comparison functions) they would be willing to share?

Linkage Methodology

2.1 What record linkage method is the software based on?

Fellegi-Sunter?

Information-Theoretic methods?

2.2 How much control does the user have over the linkage process? Is the system a "black box," or can the user set parameters to control the linkage process?

2.3 Does the software require any parameter files? If so, is there a utility provided for generating these files? How effectively does it automate the process? Can the utility be customized?

2.4 Does the user specify the linking variables and types of comparisons?

2.5 What kinds of comparison functions are available for different types of variables? Do the methods give proportional weights (that is, allow degrees of agreement)?

Character-for-character?

Phonetic code comparison (Soundex or NYSIIS variant)?

Information theoretic string comparison function?

Specialized numeric comparisons?

Distance comparisons?

Time/Date comparisons?

Ad hoc methods (e.g., allowing one or more characters different between strings)?

2.6 Can the user specify critical variables that must agree for a link to take place?

2.7 How does the system handle missing values for linkage variables?

Computes a weight like any other value?

Uses a median between agreement and disagreement weights?

Uses a zero weight?

Allows user the option to specify treatment?

2.8 Does the system allow array-valued variables (e.g., multiple values for phone number)? How do array-valued comparisons work? What is the maximum number of values in an array?

2.9 What is the maximum number of linking variables?

2.10 How does the software block records? Do users set blocking variables? Can a pass be blocked on more than one variable?

2.11 Does the software support multiple linkage passes with different blocking and different linkage variables?

2.12 Does the software contain or support routines for estimating linkage errors?

2.13 Does the matching algorithm use techniques that take advantage of dependence between variables?

Fellegi-Sunter Systems

3.1 How does the system determine m- and u-probabilities? Can the user set m- and u-probabilities? Does the software provide utilities to set m- and u-probabilities.

3.2 How does the system determine weight cutoffs? Are they set by the user? Does the software provide any utilities for determining weight cutoffs?

3.3 Does the software allow linkage weights to be fixed by the user? What about weights for missing values?

Data Management

4.1 In what file formats can the software use data?

Flat file?

SAS Dataset?

Database? If yes, what kind of database?

Dbase?

Fox Pro?

Xbase?

Informix?

Sybase?

ORACLE?

Other database package?

4.2 What is the maximum file size (number of records) that the software can handle?

4.3 How does the software manage records? Does it use temporary data files or sorted files? Does it use pointers?

4.4 Can the user specify subsets of the data files to be linked?

4.5 Does the software provide for "test matches," of a few hundred records to test the specifications?

4.6 Does the software provide a utility for viewing and manipulating data records?

Post-linkage Functions

5.1 Does the software provide a utility for review of possible links? If so, what kind of functionality is provided for? What kind of interface does the utility use, character-based or GUI? Does the utility allow for review between passes, or only at the end of the process? Can more than one person work on the record review simultaneously? Can records be "put aside" for later review? Is there any provision for adding comments to the reviewed record pairs in the form of hypertext?

5.2 Does the software provide for results of earlier linkages (particularly reviews of possible links) to be applied to the current linkage process?

5.3 Does the software provide a utility for generating reports on the linked, unlinked, duplicate, and possible link records? Can the report format be customized? Is the report viewed in character mode, or is the report review done in a graphical environment? Can the report be printed? If so, what kind of printer is required?

5.4 Does the software provide a utility for extracting files of linked and unlinked records? Can the user specify the format of such extracts?

5.5 Does the software generate statistics for evaluating the linkage process? Can the user customize the statistics generated by the system?

Standardization

6.1 Does the software provide a means of standardizing (parsing out the pieces of) name and address fields?

6.2 Does the software allow for partitioning of variables to maximize the use of the information contained in these variables (for example, partitioning a phone number into area code, exchange, and the last four random digits)?

6.3 Can name and address standardization be customized? Can different processes be used on different files?

6.4 Does address standardization meet U.S. Postal Service standards?

6.5 Does standardization change the original data fields, or does it append standardized fields to the original data record?

6.6 How well do the standardization routines work on the types of names the user wishes to link?

6.7 How well do the standardization routines work on the addresses the user will encounter? (E.g., how well does it handle rural addresses? Foreign addresses?)