

Homeless Management Information System (HMIS) Integration Strategies and Solutions

September 10, 2003

University of Massachusetts, Boston
John W. McCormack Graduate School of Policy Studies
Center for Social Policy

Under Subcontract With
Aspen Systems Corporation
Rockville, MD 20850

For the
U.S. Department of Housing and Urban Development
Contract, C-OPC-21201, Task Order 4

Acknowledgments

This paper was prepared by the Center for Social Policy (CSP), John W. McCormack Graduate School of Policy Studies (formerly the McCormack Institute of Public Affairs) at the University of Massachusetts, Boston, through a subcontract with Aspen Systems Corporation (Aspen). Aspen's contract (C-OPC-21201, Task Order 4) is with the Department of Housing and Urban Development's (HUD's) Office of Community Planning and Development. Brian Sokol and Michelle Hayes of CSP authored this paper with community example contributions from Jennifer Charbonet, Jon Deigert, Deb Little, Jan Marcason, and Alex Matisco. Donna Haig Friedman, also of CSP, provided valuable review and commentary.

Cynthia Hernan, Aspen's project director provided guidance and editing, in conjunction with HUD's Government Technical Monitor, Michael Roanhouse, and Government Technical Representative, Marty Horwath.

Table of Contents

Section 1: Introduction	1
Background	
The Challenge of Multiple Systems	
Integration and Other Options	
Report Purpose and Audience	
Section 2: System Integration Models	4
Overview	
Real Time, Two-Way Integration	
Periodic, Two-Way Integration	
Periodic, One-Way Integration to HMIS	
Periodic, One-Way Analysis Integration	
Section 3: Integration of Client-Level Data	8
Identified and De-Identified Integration	
Client Consent Procedures	
Section 4: Integration of Service-Level Data	11
Section 5: Data Integration Steps	13
Step 1: Creation of Local Data Standard	
Step 2: Data Conversion	
Step 3: Data Merging	
Step 4: Data Use	
Step 5: Analysis	
Section 6: Community Examples	24
City of St. Louis, Missouri	
Massachusetts	
Jacksonville, Florida	
Kansas City Metropolitan Area, Kansas and Missouri	
Lancaster County, Pennsylvania	
Section 7: Summary and Lessons Learned	29

Section 1: Introduction

Background

The goal of this paper is to highlight Homeless Management Information System (HMIS) integration strategies and solutions that communities can use to address local data integration challenges to help them meet the U.S. Department of Housing and Urban Development's (HUD's) requirement to have an HMIS by 2004. Communities are challenged to integrate data from multiple systems for the purpose of generating a more complete picture of the extent of homelessness and the demographics and needs of persons served within their jurisdictions. For this paper *integration* is defined as the process of combining data from multiple existing sources. This definition does not include one-time migration of data from legacy systems although some of the concepts may be relevant.

HMISs are computerized data collection applications designed to capture over time client-level information on the characteristics and service needs of adults and children experiencing homelessness. An HMIS is designed to aggregate client-level data to generate an unduplicated count of clients served within a community's system of homeless services, often referred to as the Continuum of Care (CoC). HMISs can also cover a statewide or regional area, and include several Continua. For those included in an unduplicated count, the HMIS can provide data on client characteristics and service utilization.

A stand-alone database designed to capture information about clients served in one particular program or agency is generally not considered to be an HMIS. However, this paper explores ways in which the information in single-agency databases—as well as data from other sources—can be integrated with the data captured in an HMIS as part of a full HMIS implementation.

In July 2003 HUD released the *Homeless Management Information Systems (HMIS) Data and Technical Standards Notice*.¹ These standards detail precisely what data HUD wants collected from each Continuum, including specific questions and value options. They also mandate privacy protections and information security measures that should be instituted during HMIS implementation.

The HMIS standards or *national standards* do not necessarily envision data integration from multiple sources. However, they do not preclude it. The standards comprise a starting point for a community's determination of which data elements to collect from individual systems and what privacy measures to implement. To implement an integration strategy, communities must create local data standards that may require data not requested from the national standards and more precise instructions regarding the exact data format. (See Section 5, Creation of Local Data Standard.)

The Challenge of Multiple Systems

In response, communities have begun to implement HMIS nationwide. Ideally, HMIS planners generally intend for all service providers to use the same system for data collection, making reporting easier. As a benefit to clients, the HMIS would reduce duplicative intake processes,

¹ *Federal Register*, Volume 68, No. 140; July 22, 2003. The document can be found at: <http://www.hud.gov/offices/cpd/homeless/index.cfm>.

provide access to streamlined referrals, and coordinate case management. The HMIS would also address each provider's specific needs for linking clients with needed services, measuring client progress and outcomes, and managing agency operations.

Unfortunately, the reality in many communities is far from this ideal. Often, communities begin to implement an HMIS, only to learn that many local service providers already have a myriad of customized systems. Each of these pre-existing (*legacy*) systems is different and each has been specifically designed to meet the needs of the individual agency or small groups of agencies for which it has been developed. These agencies have multiple funders who may, in fact, require that the agency use a different system from the HMIS or they may require reports on data that are not captured in the HMIS. There may also be multiple systems that track data across the Continuum on a specific population that overlaps with the homeless, such as runaway youth or victims of domestic violence. In short, communities must integrate data from multiple tracking systems.

Integration and Other Options

Three possible approaches exist for confronting multiple systems: a unitary system, parallel data entry, and data integration.

Unitary system. One approach is to mandate that all service providers abandon their other systems and use the HMIS. In turn, the HMIS itself can be upgraded or customized to meet the needs of all community partners. This approach has the advantage of centralizing all of the data collection processes and requiring the community to support a single system. Although this approach may work for some communities, others may not have the leverage to mandate universal compliance. The drawback is that the HMIS may not be able to reproduce the functionality of the old systems, which may adversely affect daily operations. Moving to a new system requires training all users and, more significantly, changing the culture within each organization. The impact of forcing the agencies to change their systems may create a wave of resentment that jeopardizes the success of the implementation.

Parallel data entry. A second approach is to have agencies enter data into multiple systems. Organizations keep their current systems, but they must also enter data in the HMIS. This approach is simple, and there are no technical obstacles to overcome. It allows agencies to keep the functionality of their current system and eliminates the need to customize the HMIS. In effect, each system is used for what it does best. The problem with this approach is that it burdens the data entry staff and will distract them from other organizational functions. A possibility exists that already overworked case managers will enter only the minimum required in the HMIS, especially if it is used only for reporting purposes. Although organizations may not lose their current systems, this approach requires increased staff resources, including dual data entry and training on two different MIS systems.

Data integration. The third approach is integration, the focus of this paper. With an integration approach, users can enter information in one system and the data can be merged into other systems. In general, that integration is invisible to the user. The tasks are automated and the burden shifted to technical staff. Thus integration allows users to maintain their previous systems. They do not have to be trained on a new system or enter data in more than one system to contribute to the overall community data collection effort. Integration solves many of the problems of the other approaches. However, it requires higher level skills and resources. Integration also involves an array of design and implementation challenges for a community, which are discussed later in detail.

Report Purpose and Audience

Data integration is a technical process. However, a successful implementation requires the active participation of people who understand the meaning of the data as they are used in the field. This paper covers technical issues but is written for an audience of non-technical human services professionals. The purpose is to provide non-technical professionals with a working vocabulary and a basic understanding of the processes and issues involved in data integration so that they are better able to participate in integration design and implementation. Aspects of system integration that require input from non-technical resources are emphasized throughout the paper.

Section 2 discusses the need for a community to determine clearly the purpose and scope of the integration effort, in particular whether the integrated data will be available to end users or integrated solely for analysis. Clarity of purpose and scope will help communities determine which integration model to employ. Four models are described, representing a spectrum of choices with regard to the frequency with which data are updated, the directions that data flow, and whether integrated data will be stored in a functional HMIS or an aggregate database for analysis purposes only.

Although some communities may ponder integration of data about particular clients to generate unduplicated reporting, other communities may be faced with the need to integrate information about the programs and services available in the community. Communities should consider both types of data as possible candidates for integration. Sections 3 and 4 include discussions of issues particular to client-level and service-level integration, respectively.

Section 5 presents five detailed steps of the integration process, including creating local data standards and converting, merging, using, and analyzing data. Each of these steps involves critical design decisions that affect the overall results of the integration project, the usability of the data, and the effectiveness of the system overall.

Section 6 presents community examples from local jurisdictions experienced in designing integration approaches. These examples range from efforts to integrate data from a handful of agencies for analysis purposes to a large long-term project that envisions synchronized integration of many large systems in real time. Section 7 is a summary discussion that includes a list of important questions for a community to ask during the design and implementation of a system integration strategy.

This paper is intended as a guide to technical options for data integration. Although it contains comments on related policy and legal considerations by way of illustration, a full treatment of either exceeds the scope of this paper. No jurisdiction should make decisions about which options to choose purely on technical grounds. All should consult appropriate authorities on legal restrictions (federal, state, or local) that may impinge upon some options.

Section 2: System Integration Models

Overview

Communities must determine the answers to several issues before they can decide the model of system integration that best meets their needs. The primary task in assessing which model is the best fit is to determine the purpose of the integration effort—analytical or functional. In a purely analytical integration, data are merged primarily for the purpose of reporting and analysis. In a functional integration, the average users of one system will be able to access data entered in another system to improve the efficiency of service provision.

Functional integration. Understanding the number and types of multiple systems that exist is crucial for determining the purpose of the integration. For example, in a situation where many clients frequently move among multiple large agencies, each with its own system, clients and providers will benefit from the efficiency of a functional integration in which records are available at one agency after the client has been served at another. However, there is little need for functional integration if only one small agency maintains its own system with clients that rarely use other programs. In this case, a functional integration effort produces few benefits to the clients or agency staff. The small agency's data may only be needed for analysis.

Understanding the extent of the integration requires an assessment of current information systems. When determining the number and types of systems, communities should look for any systems that collect data about homeless persons as well as systems that maintain information about available programs and services. (See Sections 3 and 4 for information particular to client-level and service-level data.) The review may determine that it is only necessary to merge data from one proprietary system into the HMIS. Other communities may uncover a need to integrate data from multiple large-scale systems (for example, an HMIS with several single agency systems, a healthcare system, and an information and referral directory). Some communities may even need to integrate data from two or more full-scale HMISs.

The purpose of data integration may be distinguished from the purpose of the HMIS implementation generally. The *HMIS Data and Technical Standards Notice* states:

HUD does not expect every CoC to implement the widest range of functionality for every homeless shelter and service provider in the short-run. HUD encourages CoCs to focus initially on developing demographic information about homeless clients. However, it should be noted that client assessment and service outcome modules are valuable tools to track client needs and progress.²

HUD has prioritized capturing demographic data and acknowledges that, at least initially, not every service provider will benefit from all the functionality of the HMIS implementation. Thus, it may make sense for a community to implement a primary HMIS with the goal that agencies that use that system will gain the advantages of its full functionality. At the same time, the community may decide that the purpose of the integration is to enable providers who are not using the HMIS to comply with the highest priority data collection and analytical requirements of the Continuum.

The purpose of integration will determine how frequently integration needs to take place. If there is a need for functional integration, such as knowing about bed availability across multiple

² [Federal Register](#), Volume 68, No. 140; Section 1.6

agencies, then real time or frequent periodic merging (e.g., daily or hourly) is needed. This will facilitate as immediate an update as possible between data entered into multiple systems. Some functional needs, such as integrating two resource directories, may require less frequent periodic merging, assuming that program information does not change.

The explicit functional need will also determine whether the integration should be one-way or two-way. One-way integration involves transferring data from one system to another. This model provides functional benefits to the users of the system receiving the data but not to users of the other systems. Two-way integration entails an exchange of information between two or more systems that provides all users access to information entered in the other systems.

Analytical integration. If community stakeholders deem that the purpose of integration is for reporting and analysis, they do not need to have real time access to data. They may opt for integration on a periodic basis such as monthly or semi-annually. Merging of information systems for analysis purposes involves the export of data from two or more systems and import into a third-party database (a primary HMIS) that is used for statistical analysis and reporting. Merging for analysis purposes requires only a one-way model.

In addition to the frequency of integration, the purpose will also determine which data elements must be merged. Functional needs may require the exchange of data necessary for day-to-day operations, such as bed types available. Analytical needs would only require the integration of data that can be aggregated and viewed in reports.

A well-defined purpose and clear understanding of the scope of the project, including the number and types of data systems in use will facilitate the key design decisions regarding frequency of data merging (real-time or periodic), directionality (one-way or two-way), type of central repository (aggregate database or primary HMIS), and which data elements to integrate. All of these issues need to be discussed among community partners, systems developers, system users, and other interested parties prior to development of any technical specifications. Although the purpose and scope of integration is the foundation of the integration process, other factors, such as resource availability and privacy concerns, may limit the options available.

There are typically four system integration models. Communities may find that one model best meets their needs given the purpose and resources available. Although this is not a discussion of all possible models, the four most widely used models are discussed below. The models presented are:

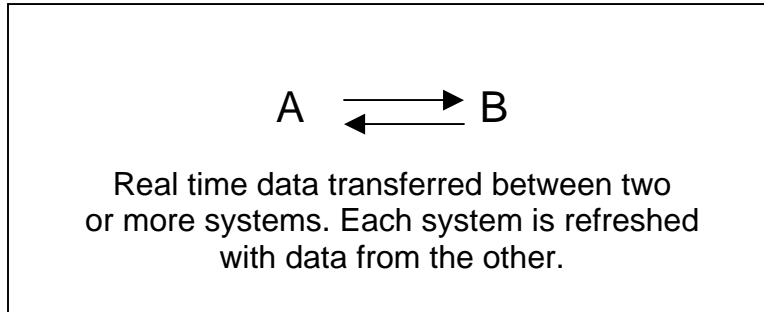
- Real time, two-way integration;
- Periodic, two-way integration;
- Periodic, one-way integration to HMIS; and
- Periodic, one-way analysis integration.

Real Time, Two-Way Integration

Real time, two-way integration involves the transfer of data between two or more systems in a synchronized fashion (see Figure 1). Each system is refreshed with data from the other on an ongoing basis whenever data are updated. The advantage of this model is that there is a real-time transfer of data among the systems. This model is appropriate, for example, when implementing a bed-reservation system allowing users of any system to view and reserve available beds in any shelter. In this case, it is desirable to have up-to-the-minute knowledge

that the bed is free and the ability to reserve it prior to sending a client across town. Although this is technically possible, many communities find that this option is cost prohibitive. Real-time integration requires significant development and programming to work effectively.

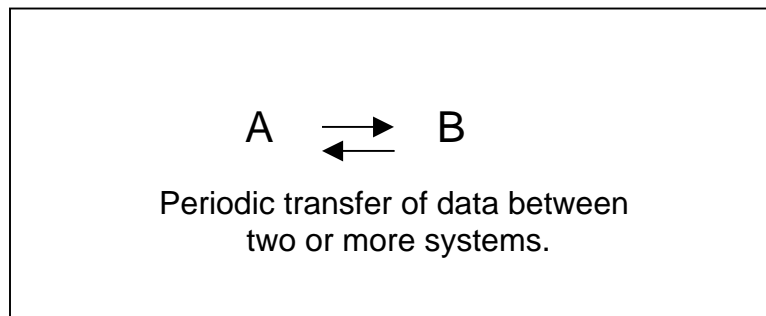
Figure 1: Real Time, Two-Way Integration



Periodic, Two-Way Integration

In periodic, two-way integration, data are transferred between two or more systems on a periodic basis (see Figure 2). For example, two different HMIS systems used in one community could exchange information on a nightly basis, transferring from one system to the other. If a client has a record of service in each system on a particular night, after integration, both systems could identify that a particular client was served at each agency. One type of integration for which this model may be ideal is a community choosing to integrate its HMIS with other resource directories. Over time, new community resources entered into any one of the systems will also be listed in the others. Although this option is less costly than a real time interface, it may still be cost prohibitive to many communities and does not reap the benefit of real time data transfer.

Figure 2: Periodic, Two-Way Integration

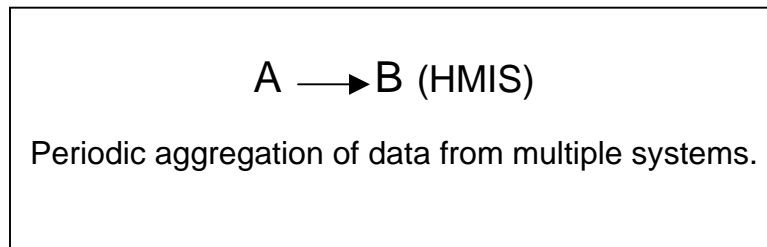


Periodic, One-Way Integration to HMIS

Periodic, one-way integration to an HMIS is a third system integration model (see Figure 3). This model is most commonly used when one or more programs use a system different from that of the majority of providers. For example, 20 agencies implement the same HMIS solution while one agency uses a stand-alone system. A strategy can be developed to incorporate the data from the stand-alone system into the HMIS at the database level. Again, this option does not provide real time access to data and data are not transferred among systems. The data only go one way—from the individual database to the central database. Most likely, this transfer will happen only on a monthly, quarterly, or annual basis. The advantages to this model are that data can be aggregated on an ongoing basis and users of the HMIS can have access to client-

level data from the stand-alone system. Because the data will be aggregated within the central HMIS, users can employ the reporting tools of the central HMIS to analyze the data.

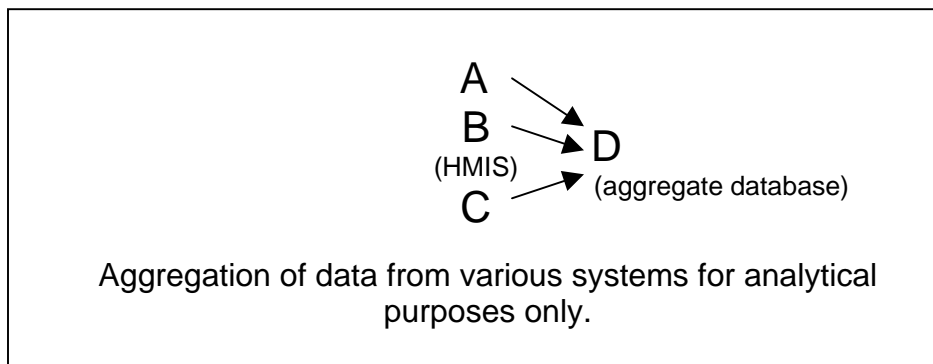
Figure 3: Periodic, One-Way Integration to HMIS



Periodic, One-Way Analysis Integration

The final model involves periodic, one-way analysis integration (see Figure 4). Many communities may have resources available only for this approach. All of the data from each of the contributing databases, including the primary HMIS, will be transferred into an aggregate database. Since the data will not be used as part of functional software, they can be much easier to analyze. Additionally, the aggregate database will not have access to the reporting functions of the HMIS software. This option is most likely the least expensive of the four and can assist communities in generating unduplicated counts and other reports for the clients they serve. However, in this model there is no real-time access to data and the data are time limited.

Figure 4: Periodic, One-Way Analysis Integration



The various integration models discussed above apply equally to client-level and service-level integration. However, the integration of these two types of data presents issues and challenges that are discussed in the following sections.

Section 3: Integration of Client-Level Data

The most common HMIS data integration need is to merge data about clients. Client-level data may include personal and demographic information, family relationships, residential history, assessments, income history, the services each client received, and the client's goals and outcomes. Valuable data about homeless clients may reside in many single-agency databases (as well as systems geared toward runaway youth, domestic violence systems, and county or state agency systems). Integrating client-level data involves special considerations in the areas of client consent and privacy as well as challenges such as determining when multiple records refer to the same client.

Identified and De-Identified Integration

Since the HMIS effort is geared toward creating an unduplicated count of clients served throughout a Continuum, it is essential to devise a means of recognizing when two records represent the same individual. Even in a single, centralized system—in which all data from multiple providers are entered directly into a common database—this task is not always straightforward. Many communities have implemented a closed model in which an agency cannot see information about clients entered by other agencies, thus duplicate records are created by design. Furthermore, when users do have access to a previously entered record for a specific individual, the user commonly creates a new record rather than search for the client and update the record previously entered.

The problem of identifying matching records for individuals is more challenging when multiple systems are involved, each of which might collect slightly differently data. The challenge is compounded by the concern for privacy, which may preclude use of the most common identifying information (e.g., Social Security Number [SSN]) from being sent to the central repository.

Several technically possible approaches are discussed below.

Fully identified clients. One approach is to send fully identified client information to the central repository. Having all of the client information allows the system to use standard identifiers such as SSN for record matching. Even this information can contain challenges, such as a record with a missing a SSN or an SSN that is missing a digit. Some community data sources may not collect SSNs at all. In these cases, the system needs to rely on other information, such as the client's name, which is not guaranteed to uniquely identify an individual. At the same time, inconsistent use of nicknames can create situations in which two different names can be used for the same individual. Conversely, two people may both be named John Burns or a single client may present at one agency as Jack and at another as John Burns.

A possible approach to this dilemma is to devise a formula in which some combinations of multiple fields indicate a likely match, even if other fields are not available. For example, if SSNs are missing, two clients with the same name, date of birth, and gender are considered the same individual. Formulas can become complex and can weigh fields differently for matches and disparities. That is, gender may be weighted heavily for disparities (if the genders do not match, the clients do not match), but lightly for matches (if the genders do match, it only minimally increases the likelihood that the records match). The task of client matching is particularly difficult for homeless clients who often lack such typical identifying information as address and phone numbers.

In some cases, using fully identified client information may be inconsistent with privacy and security concerns. To address the legal and ethical issues surrounding privacy, clients who contribute data to each source should consent to have their information shared with other systems before that data are integrated. To prevent unauthorized access, the information should be encrypted during transmission and in the central database. The security standards for data that are integrated should be at least as high as those applied to data entered directly into a primary HMIS.

Client code. The client code approach can be used when policies preclude the sharing of identifying information. Identifying information is removed from each client record and a code is generated using predetermined pieces of client information. For example, a code might be constructed out of isolated letters of a client's first and last names, the last four digits of an SSN, client gender, and part of the client's date of birth.³ Two records with identical codes are deemed to represent the same client. Before settling on a client code, statistical analyses should be conducted to determine the probability that multiple individuals will have the same code in a given geographic region. The derived probability would constitute part of the margin of error in any reports or studies.

The client code must balance two competing concerns: privacy and uniqueness. Including more elements of identifying information in the client code increases the likelihood that two distinct individuals will not have the same code. However, more elements also increase the likelihood that clients can be *re-identified* through the elements of the code. A person comparing the information in the code against other available databases such as voter registration lists or driver registrations may be able to determine the identity of the individual based on the client code. In addition, client codes are susceptible to data quality issues as some records may be missing some elements contained in the code. The system or the data analyst then has to decide what to do when two records match but not all elements of the code are present.

Use of de-identified data has been recognized by HIPAA (Health Insurance Portability and Accountability Act of 1996) as a way to allow the release of data without prior client written consent.⁴ De-identified data are client-level information that is stripped of all personal information that can plausibly be used to trace the record to a particular individual. HIPAA includes a list of data elements that are considered identifying information. This list includes names, zip codes, dates associated with an individual, telephone numbers, e-mail addresses, photos, and SSNs. Another important consideration when integrating de-identified data is that it may be necessary to remove data that are important for research.⁵ For example, if HIPAA standards were applied, most zip code information would have to be removed, which would seriously undermine the possibility of mapping prior residences of homeless persons without proper information sharing and utilization procedures in place.

Cryptographic solutions. An alternative solution to using the client code is to match records using fully identified client information but to encrypt that information. The term *encryption* refers to a method of scrambling information so that it is meaningless until it is unscrambled. *Asymmetric encryption* enables data to be encoded such that those who can encrypt do not

³ Note that the examples of elements comprising a client code given here are for illustrative purposes. Communities using such a code should seek clarification on what elements can be used to construct a code under governing laws, regulations, and policies.

⁴ HIPAA Privacy rule 45 CFR 160-164. See also www.hhs.gov/ocr/hipaa.

⁵ See D. Pettini, A.G. Breitenstein, L. Erickson, *Dataset De-identification: A Technical Overview* (January 2003), at <http://www.privasource.com/why/WP1.03.pdf>.

have the ability to unscramble. In this context this feature is important because it enables all agencies to encrypt the same data in the same way without any of those agencies being able to decrypt any of the other agencies' data. *One-way hash* functions are asymmetric functions that guarantee that the same text will always be scrambled in exactly the same way each time and that no two differing texts will result in the same scrambled output.

In the cryptographic approach, every contributing database must use software that enables encryption of the identifying information. Instead of matching client SSNs directly, the central database can match encrypted representations. By comparing the encrypted results, the database can tell whether the two original SSNs match, even if the two numbers are unknown.⁶

Using cryptography enables the system to match records based on all of a client's available identifying information without compromising the client's identity. This method does not have the same problems with maintaining both uniqueness and privacy that the client code method has. In many ways it is an ideal situation. However, the cryptographic approach requires more technical savvy than many communities may be able to afford or have expertise to employ. Of course, this solution is as vulnerable to lapses in data quality as the other models. In addition, this approach eliminates the ability of data analysts to eyeball the data for obviously invalid information such as SSNs consisting of all zeros.

Client Consent Procedures

Data sharing across systems or even among providers must not occur without proper consent and/or authorization. Typical HMIS implementations conform to the necessary requirements for data sharing, including obtaining written client consent when information is shared among providers. However, consent procedures may differ depending on the extent to which identifying information is shared. Communities must assess the specific federal substance abuse, HIV/AIDS, health, and other information sharing guidelines that are pertinent to the level of information sharing among provider organizations. Often these regulations include stipulations for the methods, mechanisms, vehicles, and timelines for client written consent protocols.

⁶ A similar approach is commonly used for password validation. To protect user privacy, databases often store only encrypted versions of user passwords. When users log in to their account, the database encrypts the value of the user types and compares the result to the encrypted value stored in the database. If the two encrypted values match, the user gains access.

Section 4: Integration of Service-Level Data

Communities are also challenged with the integration of service-level data with HMIS. Service-level data systems maintain information about agencies and programs, including services provided, location, hours of operation, and category of service (e.g., shelter providers, hospitals, and mental health centers). To effectively document service utilization patterns, movement of clients throughout the Continuum, and use of mainstream resources, communities must have the ability to record service needs and referrals as well as the services received. This requires a local HMIS to have an information and referral (I&R) component or to integrate with local I&R or 2-1-1 providers.

2-1-1 is a national effort to standardize information and referral services through a telephone directory. In July 2000 the Federal Communications Commission (FCC) assigned a three-digit dialing code—211—for access to information and referral information on health and human services. The FCC mandated that, by 2005, 2-1-1 services must achieve extensive utilization at the community level or the number may be reassigned for other purposes.⁷ Persons needing services can dial 211 from an area phone and be directly connected to their local provider, identify their need, and be given direct referrals to local area agencies that can assist in meeting their needs. 2-1-1 providers have found that in addition to persons calling for assistance, persons and organizations are using 2-1-1 as a mechanism to give donations. Thus 2-1-1 is not only a directory of human services, including health, mental health, substance abuse and homeless services, but has transformed into a community mechanism for giving and receiving.

2-1-1 has migrated from local resource directories administered by I&R providers to more expansive statewide collaboratives of centralized human service directories. I&Rs are organizations that administer a local listing of social services to improve access. Like 2-1-1, I&Rs offer a central calling number or office location providing assessment services to clients, linking them with appropriate referrals for services in their home community. Most I&R as well as 2-1-1 providers administer a list of social services by type for a geographic area.

With increased support and funding made available for 2-1-1 implementations across the country, communities are challenged with implementing both 2-1-1 and HMIS systems at the community level. The United Way of America and the Alliance of Information and Referral Systems (AIRS) are working in partnership to advance 2-1-1 nationwide.⁸ Although many communities (e.g., Dallas, Texas) are opting to simultaneously but separately deploy the 2-1-1 and HMIS, other communities are working to integrate their 2-1-1 data with the HMIS (e.g., Jacksonville, Florida). The HMIS documents client assessment information and service needs. 2-1-1 systems contain the resource directory information to expedite a client's access to resources needed to transition out of homelessness.

As communities move forward with these separate initiatives, some have found common ground in the exchange of information among the I&R, 2-1-1, and HMIS. Although some of the process is the same as integrating client-level data systems, there are some special considerations when integrating HMIS with service-level data systems. Many HMIS I&R directories as well as other community service-level directories (including 2-1-1) are organized around the AIRS taxonomy structure. The AIRS taxonomy provides a conceptual framework with standardized

⁷ Information obtained from www.211.org.

⁸ Information obtained from: <http://national.unitedway.org/211/>.

terminology and definitions for the human services field.⁹ This common structure expedites the data conversion and mapping process and facilitates an easier exchange of information among HMIS and 2-1-1 or other I&R systems.

Even with the standardized taxonomy in place, integration of service directories may entail challenges similar to those found in client-level integration. For example, just as clients can use nicknames, service programs can also be known under different names or may be spelled or abbreviated differently in separate databases. Names of fields and options may also be represented differently in distinct databases and thus need to be mapped and converted. A full discussion of data conversion as part of the overall integration process appears in the next section.

⁹ Information obtained from www.airs.org.

Section 5: Data Integration Steps

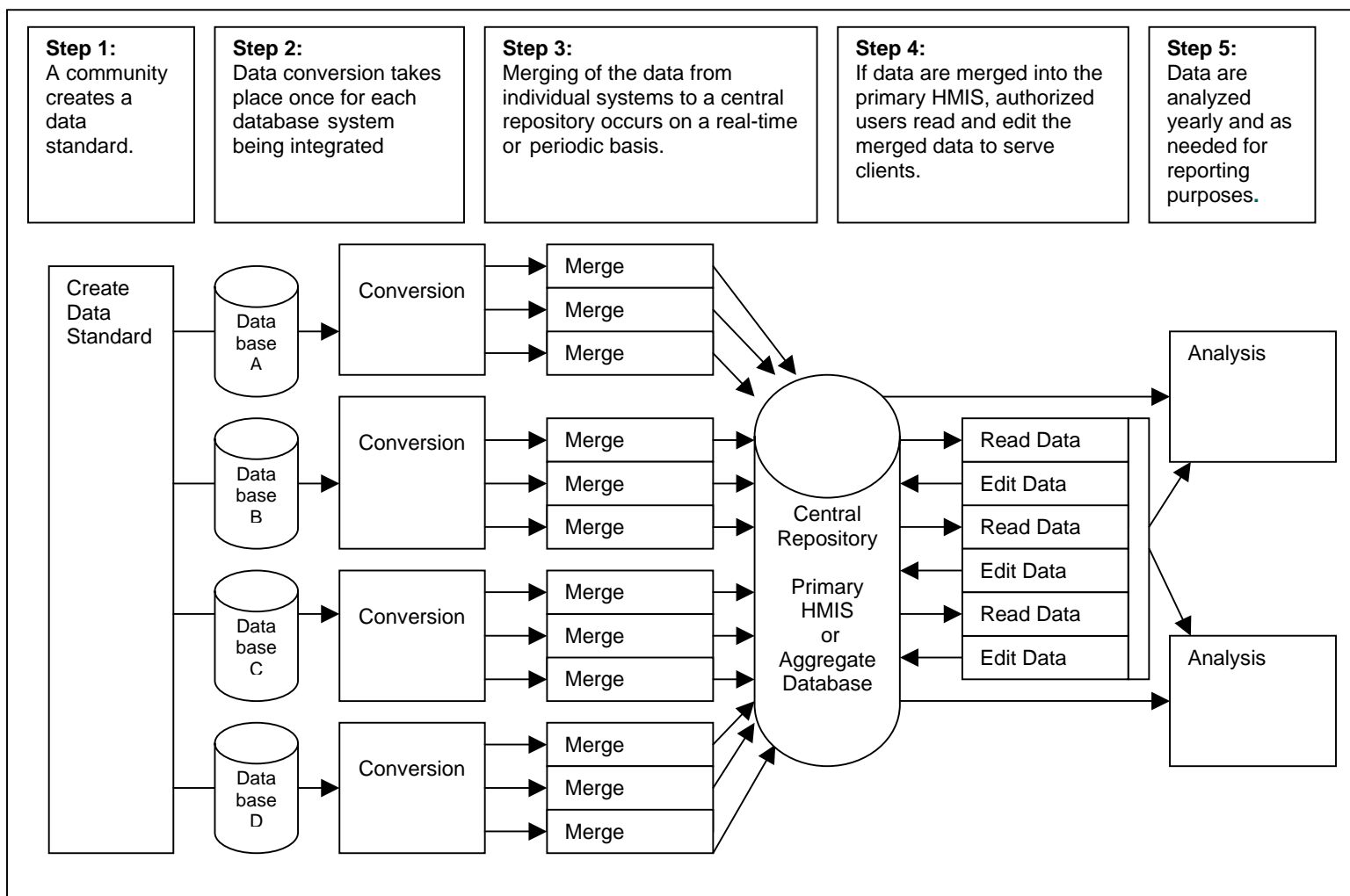
This section presents an overview of the system integration process. Once a community has defined the necessity, purpose, and scope of its data integration effort, knows the type of integration model it prefers to use, and has considered the specific implications of integrating client- and service-level data, it can take steps toward implementing its data integration strategy. Although many of the details of the data integration process will vary depending on the integration model chosen, the overall process can be broken down into five general steps:

- Step 1: Creation of a local data standard;
- Step 2: Data conversion;
- Step 3: Data merging;
- Step 4: Data use; and
- Step 5: Data analysis.

Figure 5 displays each of the steps in a one-way data integration process to an aggregate database or a primary HMIS. The overall steps for a two-way integration are much the same except that during the merging step, data will also flow back to the originating databases. The frequency and scope of each of these steps varies. A data standard is created for a whole community and rarely updated. In contrast, data merging in real-time could occur hundreds of times a day. Although some aspects of integration can be done simultaneously and automatically, each of these steps are necessary for a successful integration approach.

Although much of the discussion to follow is technical, many of the decisions during the technical process of integration need to involve both technical and programmatic experts. The examples used in this section primarily refer to the integration of client-level information, but the concepts are transferable to the integration of service directory information and other types of data as well.

Figure 5: One-Way Data Integration Process



Step 1: Creation of Local Data Standard

The primary challenge of data integration emerges from the fact that two databases may be storing the same type of information in different ways. For example, one database might store the first and last name separately, while others may store them in one field; one database may have a list of five race categories while another may have seven; or one database may record a client’s current age, while another records a client’s date of birth. Integration of such differently stored data requires reconciling.

A data standard is a document that details precisely what data can be integrated and in what format it should be stored. Each community involved in data integration should maintain a single data standard not subject to frequent change. Once the data standard is developed, experts in each individual system will be able to extract data from their particular system and transfer it to

the standard form. After the multiple-sources data are converted to a common format, they can be merged.¹⁰

A data standard serves many purposes. First, it determines the precise format in which the data should be sent. Two possible formats—comma separated files and XML—are discussed at length below. The data standard also describes *which* data can be integrated. Although many fields in each database intersect, each database will also collect data that the others do not collect. If a centralized HMIS is being used and the other databases are going to add their data to it, the data standard may consist of every data field available in the larger HMIS. Alternatively, the community might decide on a much smaller list of core data elements and include only those elements in the standard. While the need to develop the standard is equally important regardless of whether the model is one-way or two-way, these factors will affect the actual content of the standard. For example, a data standard for a functional integration may include information relevant to day-to-day operations that are not relevant for analytical purposes.

In addition to describing the possible data that can be included for each record, the data standard can also set rules about the minimal data elements. For example, the standard may not allow a client record without a date of birth or without an intake date. Or the standard may prohibit the inclusion of a service record that does not include information about the type of service. Finally, the data standard creates rules about each particular field. These rules may prescribe some or all of the following:

- **Minimum or maximum length.** Examples: First name must be between 2 and 30 characters; ZIP code must be exactly 5 characters.
- **Data type.** Examples: Numeric, date, characters only, Boolean (true/false).
- **Field format.** Examples: Dates must be MM/DD/YYYY, SSNs must not include dashes, and currency should not include commas or dollar signs.
- **Acceptable values.** Many fields will contain a list of acceptable values equivalent to those that appear in drop down boxes. The list of genders may include just male or female, or it may also include transgender. In addition to defining the universe of possibilities, the list determines what words or abbreviations to use for each value—for example, whether to use *California*, *Calif.*, or *CA*. Each value may also be linked to a number or some other code.

Example: Marital Status

0	Single
1	Married
2	Divorced
3	Widowed
4	Separated
5	Partnered/Living Together

- **Repetition.** Example: The data standard might mandate as few as one or as many as an unlimited number of choices to identify race.
- **Definition.** Many fields or field values may require precise definitions. Example: *Intake Date* may refer to a client’s appointment date, the day of a client’s initial eligibility

¹⁰ *The HMIS Data and Technical Standards Notice* is not intended to be used as a precise standard for system integration purposes. Thus it provides guidance on many but not all of the topics discussed here.

assessment, or the day the client began receiving services. The data standard should clarify the exact meaning.

Comma-separated files. There are several approaches to representing data, one of which should be chosen as the overall type for the data standard. The simplest method is to use a comma-separated value (CSV) file. In a CSV file, a comma separates each field. The data standard will indicate the order in which the fields should appear. Thus, the data standard might begin by calling for first name, last name, and then SSN. The data would appear as follows:

```
John,Doe,555-55-5555, ...  
Jane,Smith,666-66-6666, ...
```

Real-world data standards would contain many more fields.¹¹

This comma-separated approach is straightforward if these three conditions are met:

- The data requested in each client record are relatively static and do not change over time.
- Only one answer matches each field in the standard.
- The data do not need to refer to each other.

Data such as client name, SSN, birth information, veteran status, native language, and gender generally meet these conditions. Other demographic information such as race, marital status, and education level may also meet these criteria, although some problems may ensue because a client's marital status and highest education level may change over time.

The standard needs to be more complex for integrating data do not meet these criteria. Examples include:

- All of the services received by each client;
- Multiple income sources and amounts at distinct points in time;
- Residential, medical, or employment history; or
- Family composition.

Within the comma-separated format, two approaches capture these complexities. The first approach is to use separate files for each type of data. Thus, for example, the basic client information would be in one file and all services received by all clients would appear in another distinct file. The *services* file would include an identifier for each client, which would enable data analysts to link the information back to the client information. The services file format would then go on to include only information relevant to client services, such as service type and dates of service. There may be multiple service rows for each client in the service file, each row indicating a distinct service. Clients that received no services would simply not appear in the services file.

A second approach is to use only one file but include *Record-Type-Indicators*, numbers placed at the beginning of each row to indicate the type of data in each row. Thus the number *1* might

¹¹ Instead of commas, some standards call for semi-colons or vertical lines. A variation is to use a fixed-length standard, which specifies how many spaces each field should take up. Blank spaces are used to fill in the space between the actual length of each datum and the number of spaces. If the data are too long, then the value becomes truncated.

indicate individual information; 2 might indicate service information. In this case, the service record itself would not have to indicate which client received the service. Instead, this information can be determined by the order of the records. All the level 2 service records applicable to that client would follow each level 1 client record. Only after all of that client's information is exhausted, would the next level 1 (client) record be listed.

XML files. Extensible Markup Language (XML) is a more sophisticated type of data standard that naturally handles some of the issues that are difficult to support using comma-separated files. XML files can be easily transmitted over the web and interpreted by many standard database systems. XML represents data within *tags* that open and close. `<firstname>` is an example of an opening tag. `</firstname>` (note the forward slash) is an example of closing tag. The information in the tag tells you the type of data. Information between the opening and closing tag contains the data itself. Thus you might find the following in an XML file:
`<firstname>John</firstname>`.

XML allows you to group data together by putting tags inside of tags, as in the following example:

```
<client>
  <firstname>John</firstname>
  <lastname>Doe</lastname>
  <soc_sec_num>555-55-5555</soc_sec_num>
</client>
<client>
  <firstname>Jane</firstname>
  <lastname>Smith</lastname>
  <soc_sec_num>666-66-6666</soc_sec_num>
</client>
```

All data between the first opening client tag (`<client>`) and the first closing client tag (`</client>`) must relate to the same individual client. Only after closing a client tag can a new client tag open a new client record. Multiple pieces of the same type of information about the same client can be easily accommodated when necessary by simply adding two tags of the same type. Thus multiple services for a single client can be easily represented as follows:

```
<client>
  <firstname>John</firstname>
  <lastname>Doe</lastname>
  <soc_sec_num>555-55-5555</soc_sec_num>
  <service>meal</service>
  <service>bed</service>
</client>
```

In the above example, John Doe is shown to have received both a bed and a meal. Suppose multiple pieces of information need to be recorded about each service, such as the date of service as well as the service type.

In the following example, it is clear that John Doe received a bed and a meal on two consecutive dates. Notice how each distinct service is grouped within opening and closing

service tags and all four distinct services are contained within the client tags identifying which client received the service.

```
<client>
  <firstname>John</firstname>
  <lastname>Doe</lastname>
  <soc_sec_num>555-55-5555</soc_sec_num>
  <service>
    <service_type>meal</service_type>
    <service_date>7/7/2003</service_date>
  </service>
  <service>
    <service_type>bed</service_type>
    <service_date>7/7/2003</service_date>
  </service>
  <service>
    <service_type>meal</service_type>
    <service_date>7/8/2003</service_date>
  </service>
  <service>
    <service_type>bed</service_type>
    <service_date>7/8/2003</service_date>
  </service>
</client>
```

The XML standard, known as an XML Schema or a Document Type Definition (DTD), will indicate what tags are available and establish a hierarchy. For example, the standard could forbid the user from writing a <service> tag unless it is between <client> and </client> tags. The standard would also indicate whether a given tag could be used more than once inside another set of tags. For example, the standard may allow the <service> tag to be used multiple times for each client but allow the usage of the <soc_sec_num> tag only once.

Step 2: Data Conversion

Data conversion is the process of converting data from one format to another for integration purposes. This is often referred to as data mapping. The *data standard* mandates the way to represent data, whereas the *conversion* is the process of mapping the data in each individual database to the standard.

Mapping databases to the data standard needs to be done for all systems involved in integration. In one-way integration, it is necessary to map the data for the contributing databases, so that they can be exported in the proper format. The central repository (either the aggregate database or the primary HMIS) also needs to be mapped to import the standard data files into the tables used in the actual database. In two-way conversions, the mapping is necessary so that all systems can both export data into the standard format and import data coming from other systems into the standard format.

Field mapping can be seen as the process of determining where to look in an individual database for the data requested by the standard. If the data standard is looking for client *gender*, it is necessary to determine where in the database the client's gender is stored, even if

the database does not have a specific gender field. This case may be obvious: Sex in one database could be the same as *gender* in the standard. Other cases may be more difficult.

Value mapping determines the values in the database that mean the same thing. Values are often represented in the database as numbers. For example, the data standard may prescribe: Male = 1 and Female = 2, whereas a particular database may have: Female = 1 and Male = 2 or Male = M and Female = F. Value mapping deals with assigning the same values to fields regardless of how they are represented in a specific database.

Although there are many logical translations of values in mapping, response categories often must be consolidated. For example, a particular system may capture highest grade level completed (e.g., 10th grade) while the data standard may identify education categories as *some high school, high school graduate, GED*, etc. Although they cannot be mapped precisely, 10th grade can be accurately mapped to *some high school*. The mapping is accurate, but precision is lost in the translation. If, on the other hand, the database standard requested the exact grade level, and the particular database only had the categories, it would be impossible to get a factually accurate level in all cases. There is no way to tell whether for any given client *some high school* should be translated into 9th, 10th, or 11th grade, and yet it is also undesirable to leave the value blank. Consequently, decisions should be made about how to translate these cases. These decisions should be documented and shared with others in the community so that everyone is handling problems in the same ways. They should also be included in any report based on the data. Suppose the program decides to translate *some high school* as 10th grade. Without proper documentation, analysts would be left to wonder why so many more students are dropping out after the 10th grade than after the other grades. This example suggests that creators of the data standard should lean toward broader value options, so that the data from every system can be converted accurately, even if some precision is lost.

In many cases, correspondence is not immediately clear. Most people can match *sex* to *gender*, but many people do not know that *TANF* equals *AFDC* equals *Welfare*. The active involvement of people who work at the agency level is critical for a successful conversion process. The following are some case examples of field and value mapping possibilities. For each one, it is important to consider whether such a mapping is valid and who is the best person to know it.

Case 1: The standard may have a field that asks whether a client is homeless. A database for a homeless shelter may never actually ask whether the client is homeless. It is possible that the answer could be Yes for everyone. The answer is implied by the fact that a shelter provider entered the client into the system. Conversely, the question might be referring to a narrower definition of homelessness than used by the program. Similar situations may apply for a database used at veteran's shelter, a domestic violence facility, a home for runaway youth, or a provider catering only to either men or women.

Case 2: The standard may ask for the client's age category, such as under 18, or 18-24. A particular database may not have the field but may have the client's date of birth. The conversion process could use the date of birth to derive the age grouping. In this case, it is necessary to establish the date used for determining age. Possibilities include using one calendar date for everyone, using the clients' ages at intake, or using their ages at the earliest or latest date they received a service.

Case 3: The data standard may ask whether the client is employed. A particular database may not have that as a field but may collect employment income amount. It is possible to assume that a client who is receiving employment income is employed.

Case 4: The data standard may have a separate field for race and for ethnicity (Hispanic/Latino). A particular database may have only one field, where *Hispanic* is one of the values. The conversion program would need to include logic that maps the race field in the database to both the race and ethnicity field.

Case 5: This case is the reverse of Case 4. Suppose the data standard only has a race field that includes *Hispanic*, and the individual database has separate fields for race and ethnicity. The conversion program would need to map the two fields to a single field. Someone on the program or policy level should determine whether Hispanic clients that have another race listed should be listed under *Hispanic* or under the other race. The answer to this question should not be left for the technical staff.

Case 6: The standard may have fields for *Primary Disability* and *Secondary Disability*. A particular database may have checkboxes for *alcohol abuse*, *drug abuse*, *mental illness*, *physical health problems*, and others, any number of which may be checked and all of which may be choices allowed for the disabilities. The technical staff would be able to handle a case in which only one of the needs are checked by listing it as the client's primary disability. However, if more than one answer is given, non-technical staff should help to make the rules set for determining which disabilities should be considered primary and which secondary as well as what to do when more than two disabilities are listed.

In addition to mapping fields and values, the data conversion step may also include generating a client code based on particular data elements. If the client code method is being used, all systems must be able to create and/or generate a common client identifier. For more information see Section 3: Integration of Client-Level Data.

Step 3: Data Merging

Although data conversion can take place within the confines of each single database, eventually the data in the standard format must be merged. Data merging is the process by which data from two or more systems are combined.

As previously mentioned, the data merging itself can occur on a real-time or periodic basis. *Real-time* merging technically means that data are merged with another system whenever the original database is updated. *Periodic* merging means that data are merged at intervals and all the data within an interval have been updated. Periodic merging can be designed to occur automatically and as frequently as daily or even hourly, in which case, from the user's (as opposed to that of the software developer's) perspective there is very little difference between periodic and real-time. Therefore, frequent periodic merging is often referred to as *real time*. Periodic merging can also be designed to occur infrequently such as monthly or quarterly, which is reasonable if the purpose of the integration is for long-term analysis or if the data itself rarely changes, as might be the case with a resource directory.

In one-way integration, multiple databases will export data to one database. In two-way integration, each of the databases must be able to both send and receive data. In functional integrations—both one-way and two-way—where data are added directly to one or more live

systems, the merge process must also account for synchronization. Synchronization ensures that each of the databases receiving data has the most up-to-date information and that later changes are not overwritten by earlier ones made in a different database. Suppose a client went to one agency in March while pregnant and then to another agency in April after she bore the child. If data from the first database are merged into the second database in May, proper synchronization prevents the client's pregnancy status as of March to overwrite her status in April.

The typical process in data merging is as follows:

- **Export.** Data are extracted from a particular system in the format defined by the data standard.
- **Sender Validate.** Data are validated by the exporting agency for accuracy.
- **Transport.** Data are moved to the receiving location.
- **Receiver Validate.** Data are validated by the receiving agency.
- **Import.** Data are added to the receiver's database.

Automation can be applied to some or all of the steps in the merging process. For example, each site can be responsible for exporting data from their own application and one application can be developed that validates the data and identifies any problem areas. Another example would include the development of a single application that automatically exports valid data and sends the data to a central database on the Internet. When data are only migrated periodically and infrequently, the task might be assigned to the technical staff. For real-time or frequent data merging, automation requires little or no human intervention.

Export and Import. The technical import and export processes will vary depending on the format of the data standard (e.g., comma-separated files, XML, or another strategy) and on the type of data source. Most databases can easily import or export fields in a comma-separated format. Some databases also have the built-in ability to import and export XML data files, but others may require the acquisition of additional software to translate data into XML.

Not every record will be exported during every merge process. Rather, the export process includes certain conditions, such as a date range. Only records that were modified during that particular date range would be exported. Another possible condition is client consent. The export process would ensure that only records from clients who consented to share their data are sent.

In most cases, record matching occurs during the import phase. The system will determine whether a particular client is already present. If a client is already in the system, that client's record is updated. Otherwise, a new client record is created. The import process may also incorporate rules relating to synchronization. One possible rule might be that only records that are newer than the latest record about any given client should be imported. But synchronization is often much more complex. For example, in aggregating *income at entry* information, it may be most important to maintain the earliest assessment.¹²

Deletion of data is a challenge for the import and export processes. Suppose one of the programs in a resource directory ceases operations. It is difficult to devise an integration

¹² Where data are added to an aggregate database, it is possible to simply insert all records during import and postpone record matching and synchronization until the analysis step.

approach such that deleting the program from one database will cause it to be deleted from other databases. The most likely solution is for the databases to mark the program as inactive rather than deleting it entirely. This solution may not be adequate in the case of clients who request to have their records removed. Clients may have the right to have their information expunged from the database and that right may extend to other databases to which the information was sent.¹³

Validation. Validation consists of ensuring that the data in the files comply with the standard. Usually this is done through customized software. Standard tools exist to read XML documents and compare them to XML data standards (known as schemas or document type definitions) and determine whether the XML documents are valid. Other tools can validate comma-separated file formats.

A tool's response to invalid information varies depending on which tool is used and the nature of the inconsistency. One possible response is to reject the whole file, such that none of the data in the file is deemed valid. Second, a tool may accept or reject individual records. Thus if something is wrong with the data about one client, that client is declared invalid and removed from the data file. Third, customized tools might import a record even when particular fields within the record are invalid. For example, if the client's marital status is invalid, the tool might reject the marital status field but accept the rest of the client's record. Whether or not to reject a particular record based on an error in one field may depend on which field is invalid. An invalid SSN may prevent a client record from being matched and thus may have implications that differ from those of an invalid marital status.

Invalid data in an exported file may have a number of different underlying causes. The most obvious possibility is that the underlying data in the original system is invalid. However, it is also possible that mistakes were made during the process of data mapping. For example, in comma-separated formats, the programmer doing the mapping might have accidentally skipped one of the values. This is most likely to occur the first time data are converted from a particular system. Validation errors that are caught by the sender should be investigated and fixed before being transmitted.

Transport. Transport will generally consist of moving files over the Internet. The most common method of transporting files is File Transfer Protocol (FTP). With FTP, either the sender or the receiver maintains a folder that can be accessed by people with proper authorization. If the receiver is maintaining the folder, the sender will put the file in the receiver's folder. Otherwise, the receiver can get the file from the sender. This process can be manual or automated. An automated process that originates from the sender is often referred to as a *push* mechanism because the data are pushed out of one system into another. A process that originates from the receiver is called a *pull* mechanism. In more robust two-way integrations where many systems

¹³ Local policies and procedures as well as state and local law may differ on this. The *HMIS Data and Technical Standards Notice* does not necessarily reserve to clients the right to have their data expunged, but it does give clients the right to revoke consent to use and disclose protected personal information:

An HMIS user or developer must obtain the individual's consent prior to using or disclosing protected personal information. A Consent form must ... state that the individual has the right to revoke the consent in writing, except to the extent that the HMIS user or developer has taken action in reliance thereon.

Federal Register; Vol.68, No. 140, Section 4.3.

The implications of this statement for integration and data deletion require clarification.

