CONNECTIONS™

The Unique Records Portfolio

A guide to resolving duplicate records in health information systems

i *Public Health* INFORMATICS *Institute*

CONNECTIONS™

The Unique Records Portfolio

A guide to resolving duplicate records in health information systems

Public Health INFORMATICS *Institute*

The Unique Records Portfolio

Stephen Clyde, Susan Salkowitz
Principal Developers

Public Health Informatics Institute

David Ross, Ellen Wild, Alan Hinman, Terry Hastings
Editorial Reviewers

Lorrie Alvin
Project Manager

John Kiely
Communications Manager

Cathey Oldroyd
Visual Designer

About this document

Preparation of this publication was supported through a cooperative agreement between the Public Health Informatics Institute and the Genetic Services Branch of the Health Resources and Services Administration's Maternal and Child Health Bureau (HRSA/MCHB).

Original publication date: April 2006.
The Unique Records Portfolio is available online at www.phii.org

Suggested citation

Public Health Informatics Institute. (2006). *The Unique Records Portfolio*. Decatur, GA: Public Health Informatics Institute.

About Public Health Informatics Institute

The Public Health Informatics Institute is dedicated to advancing public health practitioners' ability to strategically apply and manage information systems.

The Institute assists federal, state, and local public health agencies and other public health stakeholders that are grappling with information systems challenges.

Our services provide clarity about the information systems problems to be solved and identify the solutions to those problems.

The Public Health Informatics Institute (parent organization of the *Connections* community of practice) is a component of The Task Force for Child Survival and Development.

For more information

Visit www.phii.org
Call toll-free (866) 815.9704
E-mail info@phii.org

Unique Records Workgroup Members

Delton Atkinson
DVS Re-Engineering Consultant
National Center for Health Statistics,
Division of Vital Statistics
Centers for Disease Control and Prevention

Mike Berry
Project Manager
HLN Consulting, LLC

Youseline Cherfilus
Public Health Epidemiologist
New York City Department of Health and Mental Hygiene

Stephen Clyde
Associate Professor
Utah State University

Roland Gamache
Director
State Health Data Center
Indiana State Department of Health

Denise Giles
Health Scientist
Division of Reproductive Health
Centers for Disease Control and Prevention

Nancy Hoffman
Deputy Director
Center for Health Information Management
and Evaluation (CHIME)
Missouri Department of Health and Senior Services

David Hollar
Assistant Professor
Department of Medical Genetics
University of Tennessee

Gail McMurchie
Project Manager
Office of Family Health
Immunization Program
Oregon Department of Human Services

Ted Palen
Director of Informatics
Clinical Research Unit
Colorado Permanente Medical Group

Susan Salkowitz
Principal, Health Information Systems Consultant
Salkowitz Associates, LLC

Mike Webb
IT Project Manager
Utah Department of Health

Warren Williams
Team Lead
National Immunization Program
Immunization Registry Support Branch
Centers for Disease Control and Prevention

Contributors

Noam Arzt
President
HLN Consulting, LLC

Rebecca Malouin
Epidemiologist
Division of Epidemiology Services
Michigan Department of Community Health

Kirstin Schwandt
Genomics Program Director
Maternal and Child Health Services
Indiana State Department of Health

Richard Urbano
Professor
Vanderbilt Kennedy Center and Department
of Pediatrics
Vanderbilt University

Nancy Wilber
Director
Center for Community Health
Community Health Services Information Initiatives
Bureau of Family and Community Health
Massachusetts Department of Public Health

Health Resources and Services Administration

Deborah Linzer
Senior Public Health Analyst
Genetic Services Branch
Maternal and Child Health Bureau

Michele Lloyd-Puryear
Chief
Genetic Services Branch
Maternal and Child Health Bureau

Marie Mann
Deputy Chief
Genetic Services Branch
Maternal and Child Health Bureau

Acknowledgements

The *Unique Records Portfolio* is the result of the vision, experience, and perseverance of many unique individuals collaborating through *Connections*, a community of practice facilitated by the Public Health Informatics Institute. *Connections* is funded by the Genetic Services Branch of the Health Resources and Services Administration's Maternal and Child Health Bureau (HRSA/MCHB). We extend our sincerest gratitude to all the people who contributed their time and effort in making sure that others may benefit from their experience and wisdom.

We recognize Stephen Clyde of Utah State University for his critical expertise in developing the *Conceptual Framework*, Susan Salkowitz of Salkowitz Associates, LLC, for her project management guidance, and Rebecca Malouin of the Michigan Department of Community Health for her valuable contributions to the *Metrics and Evaluation* chapter. We also thank our workgroup members and other contributors for their insights and commitment to this project.

We gratefully acknowledge the support and guidance of Deborah Linzer and our other colleagues at HRSA/MCHB.

The glue that holds together a community of practice is, of course, its many members throughout the country. *Connections* members voluntarily collaborate to share their successful practices and lessons learned on integrating child health information systems.

About *Connections*

Connections is a community of practice that assists public health agencies with planning and developing integrated information systems essential to improving the health of children.

The Public Health Informatics Institute launched *Connections* in fall 2004 to facilitate peer-to-peer knowledge sharing and problem solving among its 18 participating public health agencies throughout the country.

Connections is supported by the Genetic Services Branch of the Health Resources and Services Administration's Maternal and Child Health Bureau (HRSA/MCHB).

Connections member agencies

Colorado Department of Public Health and Environment

Georgia Division of Public Health

Indiana State Department of Health

Iowa Department of Public Health

Maine Center for Disease Control and Prevention

Massachusetts Department of Public Health

Michigan Department of Community Health

Minnesota Department of Health

Missouri Department of Health and Senior Services

New Jersey Department of Health and Senior Services

New York State Department of Health

New York City Department of Health and Mental Hygiene

Oklahoma State Department of Health

Oregon Department of Human Services

Rhode Island Department of Health

Tennessee, University of Tennessee

Utah Department of Health

Washington State Department of Health

“Small opportunities
are often the beginning
of great enterprises.”

— Demosthenes (384 BC - 322 BC)

The Unique Records Portfolio

Table of Contents

Preface

The nation's health system continues a steady, albeit slow, march toward using integrated information systems to support the delivery of clinical care, to ensure patient follow-up, and to inform public health programs and policy. Inaccurate, fragmented, and untimely information has led to an abundance of medical errors, missed opportunities for the provision of care, and a lack of coordination in health care services. The public health sector is often unable to appropriately aggregate data and rely on the quality of that data to conduct population-based surveillance and deliver services. In health care delivery, physicians often do not see a complete record for an individual, and thus make important health care decisions based on incomplete information. Data integration is increasingly viewed as a viable method of providing timelier, more appropriate services to individuals and the community, but obstacles remain.

At the core of the integration solution lies a problem: uniquely identifying individuals whose data are held in disparate data sources or information systems. Matching person-specific data from multiple sources to produce linked data sets or merged records is fundamental to the quality of data and the integrity of the information being provided by the integrated systems. Lack of quality data impacts the care a provider is able to give, creates liability and risks, and increases cost and inefficiencies (Connecting for Health, 2004).

Increasing demand to provide integrated data requires public health programs and health care provider organizations to adopt a strategic approach to the unique record identification problem. Linking data from disparate information systems forces an organization to be explicit about the intended uses of the linked data, to understand the risks associated with inaccurately matching data, and to establish a strategy that supports the goals of the record matching measured against its inherent complications and risks. The strategies developed to address duplicate records are often referred to as *deduplication* strategies.

While the day-to-day responsibility for an integrated health information system lies with the managers of these systems, senior-level executives of public health agencies and public health program managers must understand not only the challenges and issues related to matching, merging, and linking data across disparate information systems, but also what they must do to ensure the success of such an initiative. Integrating data or creating an integrated information system presents organizational challenges as well as technical challenges, and almost always requires change in procedures, policies, and the interactions of staff within these organizations.

The Portfolio focuses on understanding the problems associated with bringing together data from multiple, independent sources to form a consolidated view of an individual's record.

The concepts, examples, and tools in *The Unique Records Portfolio* are intended to help public health agencies and health care organizations address the deduplication challenge. For integrated information systems managers charged with developing a deduplication strategy, the *Portfolio* explains the major concepts they need for success, guiding them toward greater understanding of the problems and procedures associated with bringing together data from multiple, independent sources to form a consolidated view of an individual's record.

The *Portfolio* also articulates the challenges and solutions to developing deduplication strategies, and describes the implications that various approaches have on data use. Case examples of the deduplication strategies of several state integrated information systems provide real-world experience, and hands-on tools assist managers in thinking through the various aspects of the decisions they need to make.

The *Portfolio* also assists public health leaders—the senior executives in a health agency—in understanding the implications of integrated systems and deduplication strategies from resource and policy perspectives. And it assists public health program managers—those overseeing the programs that are participating in the integration—in understanding how systems integration and deduplication will impact individual programs.

Future improvements in the timeliness, quality, and appropriateness of care for children will result, in part, through improvements in the information used to support treatment, care, and follow-up processes, as well as surveillance. Integrating information is a need that is here to stay and is gaining momentum. Uniquely identifying the data for individuals lies at the heart of designing, developing, managing, and maintaining an integrated information system.

Connecting for Health. (2004). Achieving electronic connectivity in healthcare: A preliminary roadmap from the nation's public and private-sector healthcare leaders. Retrieved March 12, 2006, from **http://www.connectingforhealth.org/resources/cfh_aech_roadmap_072004.pdf**

Introduction

1 2 3 4 5 6 7 8

The Unique Records Portfolio (the *Portfolio*) is a product of *Connections*, a community of practice supported by the Genetic Services Branch of the Health Resources and Services Administration's Maternal and Child Health Bureau (HRSA/MCHB). *Connections* members represent 18 health departments that are involved in planning, developing, and implementing integrated child health information systems. The members are dedicated to using information to improve health outcomes, eliminate disparities in care, and advance quality of care for children, especially children with special health care needs. *Connections* members strive to develop information systems that produce the accurate and timely information required to create an integrated child health profile (i.e., a comprehensive record of a child's health information and health services).

The *Portfolio*'s concepts and tools are relevant to health care organizations as well as public health departments in addressing the challenge of uniquely identifying records in an integrated information system. As members of the *Connections* Unique Records workgroup, however, the authors bring their greatest depth of experience from the world of public health. As a result, the *Portfolio* is more specific to the needs of public health agencies, their leaders, program managers, and information systems managers.

The *Portfolio* addresses the challenge of duplicate records in integrated information systems by:

- Offering consistent terminology and definitions for concepts important to uniquely identifying individuals and their data contained within disparate information systems.
- Explaining the possible technical approaches (architectures) that can be used to create integrated information.
- Reviewing the benefits and limitations of each approach.
- Providing tools to help users think through the decisions they need to make (within the context of their roles) in determining how to design, implement, and use an integrated system.

The *Portfolio* also seeks to help public health leaders understand the importance of deduplication, its impact on data quality, and leadership roles in supporting deduplication strategies, policies, and procedures. The *Overview for Public Health Leaders* is written for this audience. For public health program managers who oversee programs participating in the integrated information system, the *Portfolio* articulates the implications of various architecture options for the quality of the integrated information and the limitations that each option places on data, data use, and data quality.

The *Conceptual Framework* sections II, V, and VI—along with the concepts in the Framework's margins—offer particular relevance for public health program managers.

The remaining chapters support health information systems managers by providing a clear and consistent approach to concepts and terminology they can use in presenting technical options to senior program management and public health leaders.

How the Portfolio is organized

The *Portfolio* explores the problem of unique person record identification within the integration of individual/disparate databases (the source databases) through a series of tabs or chapters. The *Portfolio* is organized as follows:

Chapter 1: Overview for Public Health Leaders

Provides an overview of the issues related to deduplication within an integrated information system. This chapter highlights the importance of the unique records identification problem and the impact of duplicate records on data quality and data use. It is written for public health leaders, health care executives, and other high-level decision makers.

Chapter 2: Conceptual Framework

Defines the technical infrastructure and the processes used to develop integrated information systems and facilitate the prevention, identification, and resolution of duplicate records. The Framework establishes the concepts and vocabulary for understanding and discussing data integration and deduplication. This chapter is illustrated with tables

and diagrams, contains references to more detailed material for additional reading, and is cross-referenced to other chapters in the *Portfolio*. Issues of importance to program managers and staff are highlighted to facilitate their use of this chapter.

Chapter 3: Case Examples

Presents a snapshot of the data integration approaches and deduplication practices of *Connections* member projects in Missouri, New York City, Rhode Island, and Utah. The cases provide a range of examples of integration system architecture and deduplication processes, and illustrate concepts presented in the Conceptual Framework, Metrics and Evaluation chapter, and Profile Questionnaire.

Chapter 4: Unique Records Profile Questionnaire

Provides a self-administered survey to help information systems managers categorize the nature of their integration system and provide a baseline for ongoing quality improvement. The *Profile Questionnaire* will also help managers make decisions about how best to organize their deduplication processes. It contains prompts about the definitions used in the *Conceptual Framework* so managers and staff may use uniform terminology to describe their processes, research and analyze their approaches, validate them, and find solutions to problem areas. Through its parent organization, the Public Health Informatics Institute, *Connections* will maintain a database online to allow participants to enter their survey data and compare their projects with other integration projects.

Chapter 5: Metrics and Evaluation

Discusses the need for metrics, as well as uses and benefits of metrics. This chapter describes the application of metrics, and presents specific metrics and their usefulness for different facets of quality improvement. Also included is a detailed tutorial on how to use, calculate, and interpret metrics for a matching process, and the steps to establish metrics for evaluation of an integration project.

Chapter 6: Self-Assessment Checklist

Contains two tools that provide step-by-step prompts on key considerations for developing and monitoring a deduplication strategy. The *Checklist for Planning Integration and Deduplication* guides information systems managers in evaluating their organization's readiness for integration and supporting deduplication activities. The *Checklist for Data Quality Assurance* provides a guide for performing continuous quality assurance on the integration process. The checklists can be used for project planning, staff training, and ongoing quality assurance, and can be expanded when adding new programs to the integration system.

Chapter 7: Glossary

Contains definitions for the terms used throughout the *Portfolio*. The definitions provide a common working vocabulary for deduplication issues and challenges.

How to use the Portfolio

Each chapter opens with a "how to" introduction and a table summarizing its contents. To use the *Portfolio* most effectively, the Unique Records workgroup recommends reading the *Conceptual Framework* first, followed by the *Profile Questionnaire*, *Metrics and Evaluation*, and *Case Examples*. Readers will then be prepared to administer the *Self-Assessment Checklists*. While reading the various chapters, they may also refer to the *Glossary*.

The chapters included in the *Portfolio* represent the issues and challenges that the *Connections* Unique Records workgroup thinks are most urgent in providing a foundation for understanding deduplication issues. The workgroup also identified potential future chapters, including one on operational issues and another on the responsibility of a central authority. If resources are available, these chapters will be added to the *Unique Records Portfolio* over the next two to three years.

Overview for Public Health Leaders

1

Overview for Public Health Leaders

1

Overview for Public Health Leaders

Public health agencies must prepare to meet the electronic health information challenge.

Information technology is transforming the landscape of health and health care. Electronic health records are changing the way in which health care providers assess, treat, and document patient care. Electronic health records also hold promise for individuals to gain access to their personal health information and participate more actively in their own health care. Public health agencies must also prepare to meet the electronic health information challenge if they are to deliver on their mandate to protect and improve the health of all people within their jurisdictions.

Behind the electronic health records movement is a national initiative to move from systems that are practice-centric to systems that are person-centric, with the goal of reducing medical errors and creating a more efficient, effective, and safer health care environment. Combining data about an individual's health to create an electronic health record yields more complete and timely information to improve quality of health care. Integrated public health information systems will improve the ability of public health practitioners to conduct accurate population-based surveillance, service delivery, and coordination of care.

Although most public health information systems are categorical, isolated silos that cannot exchange data, the importance of integration is gaining recognition. For example, many child health programs are already experiencing the benefits of integrating an individual's health information from multiple sources to produce, in effect, a child health profile. Physicians who have access to these systems are able to see,

at a glance, a child's newborn dried blood spot screening and newborn hearing screening results, as well as their immunization and lead screening status.

By accessing an integrated information system, public health practitioners are able to more effectively target and coordinate their outreach efforts, often covering several child health services in a single home visit. Children at risk for high lead levels, for example, can now receive more appropriate nutritional counseling from WIC staff who view the lead screening results. Nurses providing immunizations can see if a child needs a follow-up to a hearing screen. These success stories, along with the national electronic health records movement, have brought the importance of integration to the attention of public health leaders.

The integration of health information systems, however, still faces many challenges. Combining data demands that every public health and health care organization have the ability to correctly identify the personal data being merged into a new record or linked to form integrated information. Strategies for finding and eliminating duplicate records are essential to having an integrated information system that produces accurate, usable data. Without such strategies, the system, and therefore its data, cannot be considered reliable.

A *Connections* workgroup comprising representatives from state and local health departments that are engaged in integrating child health information systems developed *The Unique Records Portfolio*. The *Portfolio* helps guide public health practitioners in developing appropriate strategies, including policies and procedures to minimize, if not eliminate, duplicate records.

Failure to identify and resolve duplicate records compromises the quality, credibility, and usability of integrated information systems.

Duplicate Records: A Quality Assurance Issue

The problem of duplicate records for the same individual challenges all person-centric information systems. Duplicate records, which can exist in single systems, are even more problematic for systems that integrate an individual's information across programs and other systems. Failure to identify and resolve issues related to duplicate records compromises the quality, credibility, and usability of integrated information systems.

Early developers of immunization registries and child health integration systems grappled with huge backlogs of pending unmatched records. Costly projects had to be established to analyze the causes, apply software solutions, and mount a massive manual effort to resolve the duplicates. Many projects had to suspend or delay the deployment of their systems so that lack of data credibility would not jeopardize the entire integration project.

Deduplication is the set of processes that link, match, and merge data to integrate or create an integrated view of information for an individual. As a quality assurance measure, deduplication ranks as a top management issue and a challenge for integration projects, whether for private health care initiatives or public health.

In a survey of private health care initiatives working to develop interoperability among their health information systems and create an electronic health record for individuals, 80 percent responded that accurately linking patient data was "very difficult" or "moderately difficult." When respondents were grouped by advanced stage and early stage initiatives and organizations, 73 percent (advanced stage) and 90 percent (early stage) perceived accurately linking patient data to be very difficult or moderately difficult (Marchibroda & Covich Bordenick, 2005).

Resolving Duplicate Records

The good news is that in the last three years, national initiatives for improving interoperability between private health care partners have focused on technology issues of systems interoperability and the use of standards in improving quality of data. Available technologies to deploy integrated data systems, advances in telecommunications and networks, ubiquitous use of the Internet, and Web services contribute to a body of technical knowledge and practices specifically supporting integration projects and their deduplication processes. Deduplication involves not only software (e.g., matching algorithms), but also organizational (e.g., change management) and people challenges (e.g., staff training). Addressing these challenges is an information systems management responsibility that requires programmatic and technical input, deliberate choices, well-defined activities, and systematic processes.

Deduplication involves not only software but also organizational and people challenges.

The Unique Records Portfolio organizes this programmatic and technological knowledge base, with its practices and lessons learned, in formats that help senior management understand the importance and complexity of these issues. The *Portfolio* provides resources and tools to help information systems managers and their staffs engage with technical managers to discuss and apply effective solutions to their integration and deduplication problems. The *Portfolio* also addresses methods to document current practices and to measure the efficacy of deduplication strategy.

To produce high-quality data, an integrated information system must eliminate duplicate records and assign the correct data to each individual. The system requires strategies that include:

- A set of policies and procedures guiding the operation of the integration system.
- A technical architecture that supports the policies and procedures.
- An operational plan or set of activities that address the core data quality goals of the integration system.
- A method of evaluating the deduplication processes to determine how effectively and competently duplicate records are being reduced and resolved.

What Can Public Health Leaders Do?

Public health leaders must encourage their organizations to present a comprehensive deduplication strategy that helps validate their overall information system investment. They can also support the policies, procedures, and strategies by providing essential resources for deduplication project development and implementation. Public health leaders should also insist on receiving periodic evaluation reports on the results of deduplication efforts.

Senior leaders should learn about the impact of the improved information on public health programs.

Senior leaders guide the health information system integration project toward meaningful health goals. One of their key roles is to approve the health outcomes performance measures for the integrated information system. Senior leaders should request clear rationale for the overall deduplication strategy and technology architecture choices made to realize the strategy. *The Unique Records Portfolio* provides several tools with key questions that can be asked to help all levels of staff become comfortable with their specific technical choices. Senior leaders should also establish a routine forum or briefing in which they monitor the progress of the integration system project and learn about the impact of the improved information on the public health programs involved in the integration system.

Marchibroda, J. & Covich Bordenick, J. (2005). Emerging trends and issues in health information exchange. *Selected findings from eHealth Initiative Foundation's Second Annual Survey of State, Regional and Community-Based Health Information Exchange Initiatives and Organizations*.

CONNECTIONS™

Public Health Informatics Institute

750 Commerce Drive, Suite 400
Decatur, Georgia 30030

1.866.815.9704 | www.phii.org

Conceptual Framework for Unique Records Identification

2

Conceptual Framework for Unique Records Identification

2

Introduction

Before progress can be made toward preventing or resolving duplicate records, the concepts related to integration and resolution of duplicate records must be understood. The Conceptual Framework guides health information systems managers as they tackle the challenges of information system integration, deduplication, and quality assurance issues related to duplicate health records. Though the concepts in this chapter can be applied to information systems integration in general, the examples are derived from the experiences of *Connections* members in developing their own integrated child health information systems. The Conceptual Framework begins with fundamental concepts and challenges of integration and progresses to detailed strategies for resolving duplicate data.

This summary of the Conceptual Framework's main sections guides users through the document:

Section	Purpose and Content	Page
I. Data integration and Application Integration	Defines and distinguishes between data integration and application integration.	22
II. Challenges of integrating person-centric systems	Discusses four challenges closely related to resolving duplicate data, and establishes a context for concepts discussed in subsequent sections.	23
III. Deduplication: a key integration process	Describes deduplication in terms of generic roles that people or software play in the process of resolving duplicate data. It defines terms for describing and discussing deduplication.	26
IV. Common software architectures	Summarizes five prototypical system designs that represent a range of alternatives for how components in an integrated system can play different roles in the duplication process.	29
V. Comparison characteristics	Introduces characteristics that can be used to evaluate or compare system designs with respect to their impact on the duplication process.	48
VI. Matching	Describes matching techniques in terms of when, where, and how matching can take place and what data elements are often used.	51
VII. Linking	Describes three primary techniques for linking records in integrated systems.	59
VIII. Merging	Describes techniques for merging records in an integrated systems environment, prefaced by a discussion of the unique challenges that merging presents, and the preparations that lead to effective merging as part of the deduplication process.	63

Following are notes on how to use this chapter:

Although information systems managers will want to focus on the last three sections, which provide techniques, Sections 1-5 present definitions and concepts that are essential to the understanding of those sections.

Public health program managers should pay close attention to Section 2, the planning and data governance subsections of Section 5, and Section 6, and to concepts called out in the page margins.

Figures, tables, and sidebars are used throughout the Conceptual Framework. The figures, particularly in Common Software Architectures (Section 4), illustrate design concepts or system configurations discussed in the text. Although these are intended to be relatively self-explanatory, these are dependent on terms and symbols introduced in Section 3.

I. Data Integration and Application Integration

This section defines and distinguishes between data integration and application integration.

In general, building an integrated information system may involve *data integration, application integration,* or *both.*

- Data integration is the process of combining information from independent sources, such as vital records and newborn screening.

- Application integration is the process of providing end-user software to present a unified view of the data. More precisely, as Noam Arzt has noted, "Application integration for data presentation involves making integrated data available by presenting a unified view of data to a user through a *computer application* (computer application being broadly defined as anything from a personal computer to a Web browser to a smart card)." (Arzt & HLN Consulting LLC, 2005)

Although both types of integration must deal with duplicate or redundant data, the Conceptual Framework focuses on data integration and does not deal with issues related to user interfaces.

The underlying purpose of an integrated system is to improve the quality and availability of information. In concrete terms, this means giving public health and health care professionals immediate access to an individual's information (e.g., address, phone numbers, relatives, etc.) and medical and service information. The Markle Foundation states that such real-time access to integrated patient information could directly improve the ability to provide timely and appropriate medical services (Connecting for Health, 2004). This access must be done in a secure environment, in which confidential information remains private and individuals may decide whether their information can be shared.

II. Challenges of Integrating Person-Centric Systems

This section briefly discusses four challenges closely related to resolving duplicate data, and thus helps establish a context for concepts discussed in subsequent sections.

The integration of existing person-centric systems involves many challenges besides eliminating, or at least minimizing, duplicate or overlapping data. To provide a foundation for resolving duplicate data, program and technical staff involved in integrated systems should have, at minimum, a high-level comprehension of these related challenges:

1. Understanding the structure and semantics of data from the various data sources.
2. Establishing data sharing agreements between the sources and users of the data.
3. Presenting integrated data to users in a consistent and meaningful way.
4. Ensuring the security and confidentiality of shared data.

Addressing these challenges requires deliberate choices and activities: systematic processes. For discussion purposes, we will refer to such processes as integration processes. Some of these processes may eventually be automated, and some will remain entirely manual. Most, however, will involve human activities and software support guided by policies.

Understanding the data structures and semantics

The first challenge is an analysis and communication problem. Key stakeholders in an integration project and the information systems leaders must have a common understanding of:

Definitions

Integration Process

An *integration process* is any activity (manual or automated) that is needed to bring data together. Key integration processes:

- Handling duplicate or overlapping data
- Presenting integrated data to users in a consistent and meaningful way
- Ensuring the security and confidentiality of shared data
- Establishing data sharing agreements

Integration Infrastructure

An *integration infrastructure* is a collection of software components, outside of the individual data sources, that support integration processes. An integration infrastructure may include software matchers, mergers, and data converters, etc. (See Section 3 on Deduplication: A Key Integration Process, page26).

Integrated System

An *integrated system* is a collection of data sources and data users that are connected via an *integration infrastructure.*

Integration Project

An *integration project* involves building or adapting an *integration infrastructure* in support of specific integration processes to create an *integrated system.*

Participating Program

A public health or health care program that is part of an integrated system.

The Self-Assessment Checklist can provide direction to the planning of an analysis activity.

- The type of available data: personal information (e.g., names, addresses, phone numbers, birth dates, gender, relatives' names, etc.); medical information (e.g., primary care physician, birth anomalies, risk factors, newborn screening results, hearing screening status, immunization histories, etc.); and service information (e.g., early intervention services or service plans).
- The semantics (meaning) of the data and any subtle differences that may exist between data sources. For example, two data sources might both include information on primary care physician, but one might be the primary care physician at birth and one might be the primary care physician at the time of the last immunization.
- The quality of existing data, including its accuracy, reliability, completeness, and timeliness.
- Technical and operational details for accessing the data (e.g., means of connecting to the database, firewall issues, authentication and access control issues, etc.)

Conducting a thorough analysis of the data structures and semantics can be complex. Fortunately, an integration project can apply any number of widely accepted systems analysis processes to the problem. The Self-Assessment Checklist (page 111) can provide direction to the planning of an analysis activity.

Data Governance / Data Sharing and Use Agreements

A data-governance process must bring together data stewards and stakeholders.

The second challenge is not so much a problem of technical understanding as it is one of establishing agreement. Specifically, it involves making decisions on what data will be shared and the ways in which the data may be used. A data governance process must bring together data stewards and stakeholders, including key individuals from agencies responsible for the existing information systems and from agencies that will be using the integrated data, as well as technical people who are familiar with the data and data structure. These individuals need to collectively identify specific data sharing opportunities and their implications. The outcome of this process should be an agreement (formal or informal) on what kinds of information each data source will provide to the integrated system, how each type of data consumer can use that information, and the metrics that will be used to assess data quality.

Data Presentation

In general, application integration deals with how users see, navigate, and potentially modify integrated data. Since these data come from multiple sources, there may be subtle variations in their semantics or constraints that complicate the task of designing easy-to-understand and consistent screens or reports. Depending on how much data integration occurs within an integration infrastructure, application integration may involve only user interface design and application connectivity issues, or it may involve these issues and any or all of the challenges discussed in matching, linking, and merging data. Properly designed user interfaces and tools can increase the efficiency and effectiveness of the human review process for deduplication of records.

Data Security and Confidentiality

The fourth challenge is ensuring data security and confidentiality. Both are regulated (e.g., HIPAA (U.S. Department of Health and Human Services & Centers for Medicaid and Medicare Services, 2005) and FERPA (U.S. Department of Education, 2005) and required

functions to protect personal health information and they vary based on the sensitivity of the information and the type of user and intended use.

Processes to protect health information

1. Authentication

Authentication involves identifying, with a certain degree of confidence, those who are requesting access to the data. Authentication techniques range from simple user name/password schemes to sophisticated protocols that use multiple physical identifications (e.g., hand scans, thumb scans, magnetic keys, personal ID card, etc.) and remembered codes (e.g., user name and passwords). In general, these approaches can be categorized by what a user needs to know (e.g., user names, passwords, etc.), what a user must possess (e.g., magnetic keys, ID cards, etc.), and the use of physical properties (e.g., hand scans, thumb scans, etc.) (Denning & Denning, 1998). Typically, as the required level of confidence increases, the complexity of the system increases, and therefore, the cost and burden on the users.

So, in general, a system's designer will choose an authentication scheme that balances the level of confidence that it can guarantee with the sensitivity of the data that the users will be accessing. In integrated systems, participating information systems sometimes act as data users of other information systems. In such cases, they need to authenticate themselves, just like human users. However, they can employ different techniques (e.g., as public-private keys and request-challenge protocols) that provide a great deal of authentication confidence without a dramatic increase in cost.

2. Access Controls

Once a user is authenticated, access controls determine what that user can or cannot access (Tanenbaum & van Steen, 2002). Access controls can range from very coarse grain to fine grain. Coarse-grain access controls apply restrictions based on broad types of data and groups of users. For example, one such control may say that staff from the immunization program can access data from vital records, but no details about birth defects. Medium-grain access controls may allow more specific restrictions based on individual data fields or users. Fine-grain controls may allow restrictions or exceptions to be placed on individual records.

3. Transmission Privacy

Integrated systems typically require that data be transmitted between computers and servers, since the various data sources often are independent databases. Transmission can be across point-to-point communication lines, an internal network, or even a public network, such as the Internet. Regardless of the communication medium, data transmissions must remain private so they cannot be intercepted or discovered by anyone other than the intended destination. Some types of communication media are inherently easier to secure than others. For example, it is much easier to ensure transmissions across point-to-point communication lines than transmissions across the Internet. Setting up point-to-point communications, however, is considerably more costly, especially for geographically distributed integrated systems. The most common technique for ensuring transmission privacy is encryption, and dozens of public domain and commercial encryption mechanisms are readily available (Stallings, 1999). Other techniques are also available for securing transmissions. For example, one technique splits the data into arbitrary chunks (called packets) and sends each one along a different route to the destination

Data transmissions must remain private so they cannot be intercepted or discovered.

(Stallings, 1999). However, this technique and other alternatives are often considerably more complex and costly than encryption.

4. Auditing

Being able to assess the confidentiality and security of an integrated system is critical to its long-term success and may be required for information covered by HIPAA. Two basic kinds of audits can be considered:

- Security audits that periodically evaluate a system for weakness to potential security threats or breaches of confidentiality.
- Data-access audits that track what data have been accessed, when it was accessed, and who saw it.

5. Opt-out / Opt-in Capabilities

In integrated systems for public health and health care, an individual should have the ability to say, "Don't share my data with others." Such opt-out restrictions may cover all shared data for an individual, or certain parts of that data. For example, one individual might agree to share immunization data, but not birth defects information. The opt-out provision is handled differently in different jurisdictions or programs. Some systems include all information to be shared unless the individual formally indicates opt-out for certain information or features, such as reminders and recalls. Others require the individual to formally opt in, often at birth but also at different parts of the process. These provisions create challenges for data integration because of the additional constraints they place on accessibility of data, which in turn complicates the deduplication process.

The types of information that may be shielded from the view of the user by opt-out provisions, as well as by data sharing agreements, can reduce the effectiveness of front-end matching. (See Matching, page 51.) For example, if an individual's participation in a program includes an opt-out agreement, that address may not be viewed by a provider outside of the health department, then the provider's staff cannot use address as a discriminator to view possible, but not exact, matches. This can result in a lower hit rate, which may frustrate the user querying the information and lower the credibility and use of the system.

Data Challenges in Deduplication

Same-source Duplicate Records

Two or more records for the same person from the same source.

Multi-source Duplicate Records

Two or more records for the same person from different sources with similar types of data.

Overlapping Records

Two or more records for the same person from different sources with different types of data.

Similar Records for Different Individual (Overlaying Records)

Two or more records that appear to be for the same person, but are actually for different individuals.

III. Deduplication: A Key Integration Process

This section describes deduplication in terms of generic roles that people or software play in the process of resolving duplicate data and defines terms to form a common vocabulary for discussing deduplication.

Integrated systems vary in complexity, underlying software architecture, and the specific technologies they employ. To share person-centric data, however, all integrated systems must deal with multiple records that represent the same individual, regardless of whether those records come from different sources or the same source. In some cases, two or more records are for the same person and contain basically the same type of data. We refer to such records as either *same-source* or *multi-source duplicate records,* depending on where the original records reside. In other cases, the records are for the same person but contain different types of data. We refer to these as *overlapping records* (AHIMA MPI Task Force, 2004). Records that appear to be for the same individual,

but are not, are sometimes called *overlaying records* (AHIMA MPI Task Force, 2004).

Existing information systems do not yet use common unique record identifiers that would facilitate accurate identification of records from multiple sources as being for the same individual. Even if they did, the data are often fragmented, incomplete, and in some cases even inconsistent. The subtle details of these problems make data integration a serious challenge from technical and operational perspectives. In *The Unique Records Portfolio*, we refer to a set of processes that addresses these problems as ***deduplication***[1].

Two fundamental processes address deduplication: ***record matching*** and ***record coalescing***. Record matching is identification of previously unrelated records that represent the same individual. Record coalescing is the linking or merging of such records so users can readily access all the available data for a given individual, within the limits of confidentiality and inter-agency data-sharing policies.

Background

Almost as many strategies exist for deduplication as for integrated systems. Not surprisingly, a single "one size fits all" solution does not emerge as the ideal choice in all situations (Massachusetts Health Data Consortium Inc., 2004). Successful integration of data from independent information systems depends on understanding the nuances of each system's data, how those data come into being, and how they change over time. It also requires a basic understanding of matching, merging, and linking techniques and how those techniques can be married to best meet the demands of a given situation.

Developing a strategy for addressing data integration and deduplication is difficult and involves balancing goals against constraints and costs. Sorting out all the possible approaches against a set of requirements can feel overwhelming, even for experienced software engineers. A comprehensive strategy provides a framework for thinking about potential solutions in terms of basic components (data, applications, work processes, policies), their roles, and responsibilities.

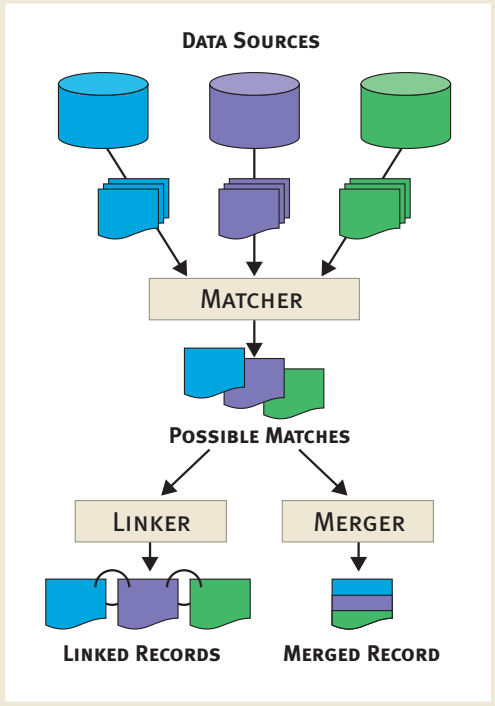


Figure 1: *Deduplication roles*

The thorough analysis required to support a deduplication strategy:

- shows how the basic components can form a variety of software architectures.
- discusses issues stemming from the types of data that may be involved in the integration.
- lists some characteristics of software architectures that are helpful in analyzing, evaluating, and comparing the relative advantages of deduplication processes.

Deduplication—Terms and Definitions

Deduplication is the term given to the set of processes that match, link and/or merge data.

A **deduplication strategy** includes at least four components:

1. A set of policies and procedures guiding the operation of the integration system.
2. A technical architecture that supports the policies and procedures.
3. An operational plan or set of activities that address the core data quality goals of the integration system.
4. A method of evaluating the deduplication processes to determine how effectively and competently duplicate records are being reduced and resolved.

[1] The term *Deduplication* is sometimes used to mean just the merging of records.

Basic Components of Deduplication

In general, deduplication involves components (software and people) that play one or more of the following roles:

	**Matcher**	A matcher is someone or something (i.e., a software program) that tries to determine whether two or more records are for the same individual. In an integrated system, the matcher role is often fulfilled by a software component that can identify high-confidence matches and suggest others for human review. New York City's integrated system, for example, uses an independent third-party software application called ChoiceMaker™. (See the New York City case example, page 81.)
	**Linker**	A record linker is a mechanism that logically connects records that are determined to be for the same individual. Linking does not require a software program. In fact, it may be a data-linking table or even just an ID field in an existing record.
	**Merger**	A merger is something or someone that combines multiple records for the same individual into a single record. In a manual deduplication process, the merger is a person who combines information about one individual from several files into one merged record. In an electronic system, the merger is typically an automated or semi-automated software program that modifies electronic records. Examples of interactive, semi-automated mergers are found in both Utah's and New York City's integrated systems. (See page 89, 81.)
	**Data Source**	A data source is someone or something that creates a new person record or provides new data to the system at large. Examples include participating information systems (e.g., birth records, immunization records, newborn dried blood spot results records, and data-entry software components, such as applications that record patient-registration data).
	**Data User**	In an integrated system, a data user is someone or something that accesses information from one or more data repositories (e.g., immunization and lead screening).

All deduplication strategies include data user, data source, matcher, and either *linker* or *merger* roles. Some use a combination of linking and merging, and therefore, include both. In most cases, the participating programs, such as newborn screening or immunization, play both data source and data user roles.

Some deduplication strategies include additional components that fulfill other special or supporting roles, such as:

	**ISD Repository**	An integrated system data (ISD) repository is a database within the integrated system, but not part of any particular data source. As the next section will show, an integrated system may have its own ISD repository to hold a registry or a master index of person records, demographic data for matching records, or fully integrated medical records. Depending on the architecture, the ISD repository may be called a central database, data vault, registry, master index, or a number of other terms.
	**Converter**	Converter software transforms data from one form to another. For a simple example, consider an integrated system that involves two data sources: One that stores dates in YYYYMMDD format and another that stores them in MM/DD/YYYY format. Before finding duplicates between these two sources or merging any data, a converter would need to transform the dates into one of the formats, or both formats to some other standard format. In typical deduplication processes, there could be many such conversions and many more that are considerably more complicated. Any software component that changes data from one form to another is playing a converter role.

As mentioned above, deduplication involves people and software. The software component may be part of the existing information systems or part of an integration infrastructure. In fully automated deduplication processes, software components must play all of the roles to conserve people hours. Yet human review always remains an option for confirming accuracy. In semi-automatic deduplication processes, some of the roles may be played by people, or people and supporting software components. For example, consider an integrated system involving the newborn screening, immunization, and vital statistics programs. Their information systems are the data sources and their staffs are the data users. Behind-the-scenes software components (i.e., something in the integration infrastructure) may play the role of a linker and an ISD. Finally, a person using an interactive software program might fulfill the matcher role. In this situation, the matcher software program could identify potential matchers and the person would confirm or reject them.

Deduplication involves people and software.

Individual components of an integrated system may play multiple roles. An existing information system can be both data source and a data user. For example, an immunization registry may provide data to others (i.e., act as a data source), and at the same time, use information from other sources such as vital statistics, lead screening, or newborn hearing screen (i.e., act as a data user). On the other hand, a vital records database may act only as a data source.

Also, roles may be shared or distributed across multiple components. For example, in a peer-to-peer architecture, each information system may play a matcher or merger role. In most integrated systems, multiple components play the converter role.

IV. Common Software Architectures

This section summarizes five prototypical system designs that represent a range of alternatives for how components in an integrated system can play different roles in the duplication process.

The assignment of roles to components of an integrated system and the way in which they relate to each other determines the system's design and construction. This overall structure is called the system's *architecture*. There are almost as many architectural possibilities as there are individual software systems, but most integrated systems fall into one of these generic architecture categories:

1. Central index
2. Peer-to-peer
3. Arm's-length information broker
4. Central database
5. Partitioned central database

This section describes each of these architectures and identifier-based techniques that can be combined with most architectures to enhance the deduplication process.

Related Terms

Record Locator Service (RLS), Central Repository MPI or Registry MPI (Massachusetts Health Data Consortium, Inc., 2004), Enterprise Master Patient Index (EMPI) (AHIMA MPI Task Force, 2004), Information Broker (Arzt & HLN Consulting LLC, 2005)

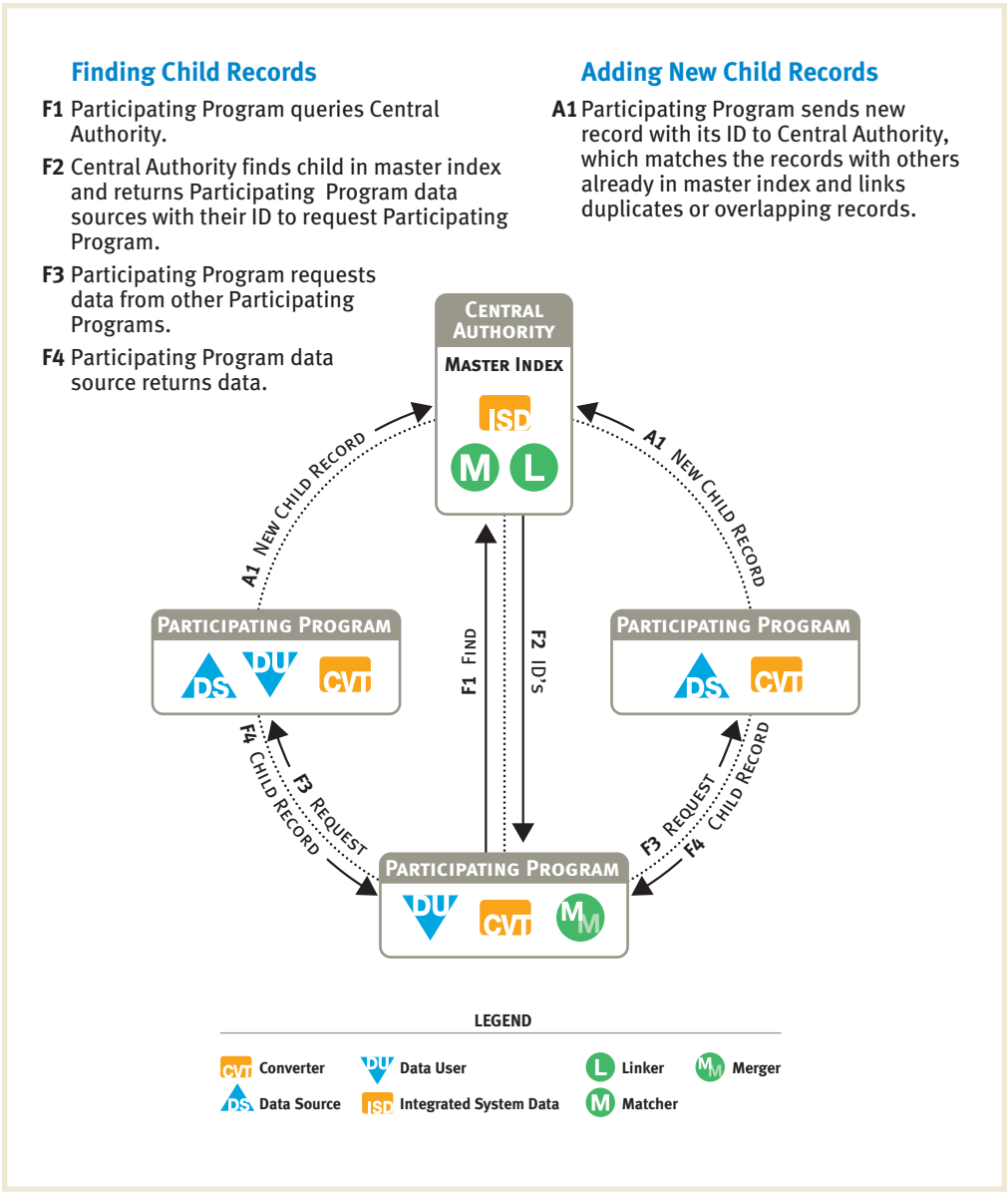


Figure 2: *Roles in a central index—locator architecture*

1. Central index

Overview

A central index contains information on the location of medical records for persons known to the system—a master person index (MPI). The central index, *also called a central repository MPI, registry MPI, enterprise MPI, or index of indices*, needs to hold enough demographic information that it can be used to perform effective matching, but does not contain any medical or program data.

In this architecture, the component that actually performs matching is often called a *record locator service (RLS)*, and can be either an integral part of the central index or a separate component. (Note: The word "service" used here is not related to the IT term "service-oriented architecture.") Once an RLS finds a person, it can follow one of three basic protocols for communication for retrieving data:

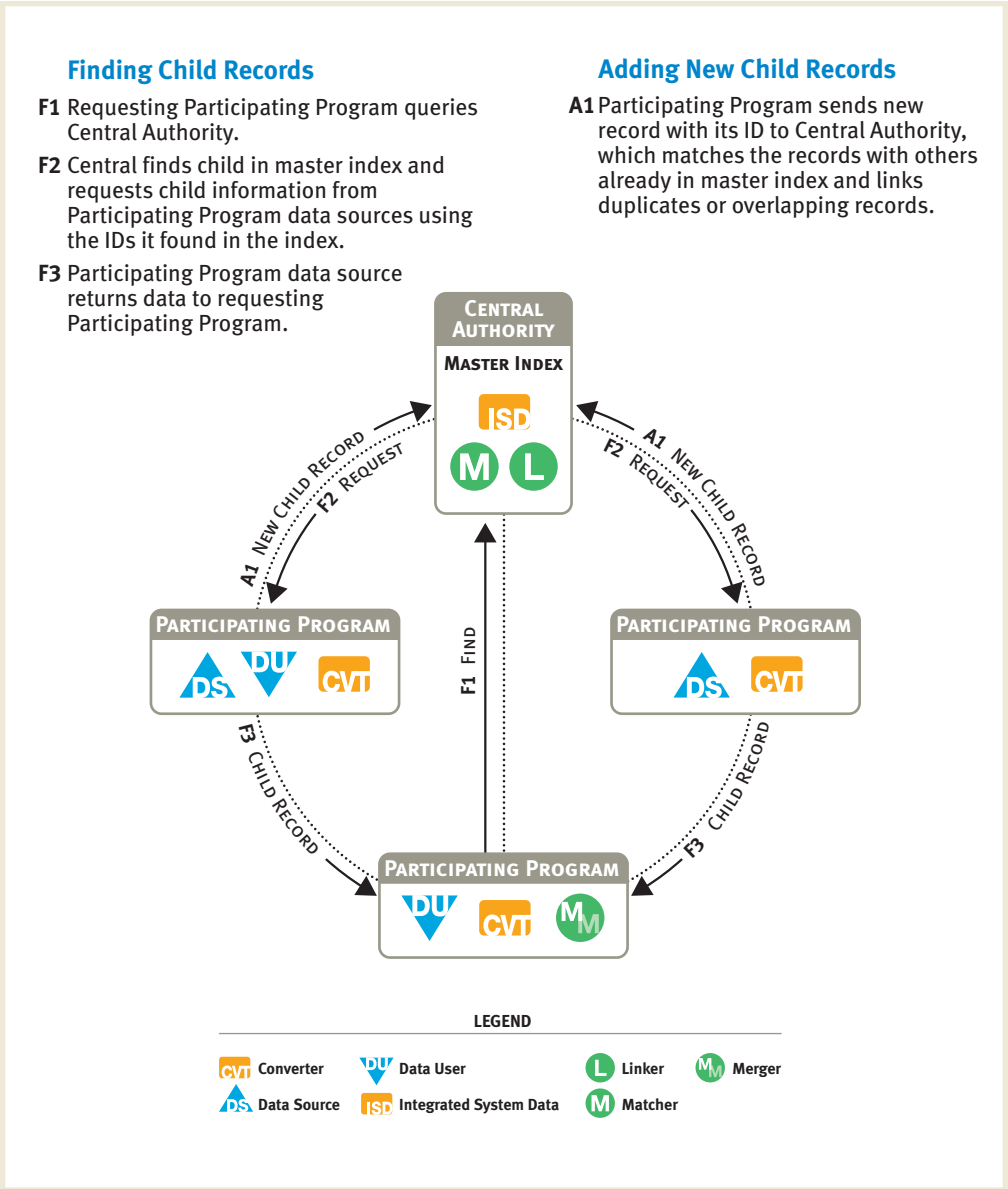


Figure 3: *Roles in a central index—facilitator architecture*

- *Locator:* The RLS tells the requesting program where to find the medical or program records (e.g., return the sources and person's IDs for those sources) and lets the requesting program access the individual's data directly from the sources. Figure 2 illustrates components, roles, and interactions for a typical central index architecture where the RLS is simply a locator.
- *Facilitator:* The RLS initiates communication between the requesting program and the data sources. Figure 3 shows the components, roles, and interactions for a central index architecture where the RLS acts as a facilitator.
- *Information broker:* The RLS acts as information broker by retrieving the available record for the requesting program. Figure 4 shows the components, roles, and interaction for central index architecture where the RLS acts as an information broker.

Central index architectures support linking directly via the master index.

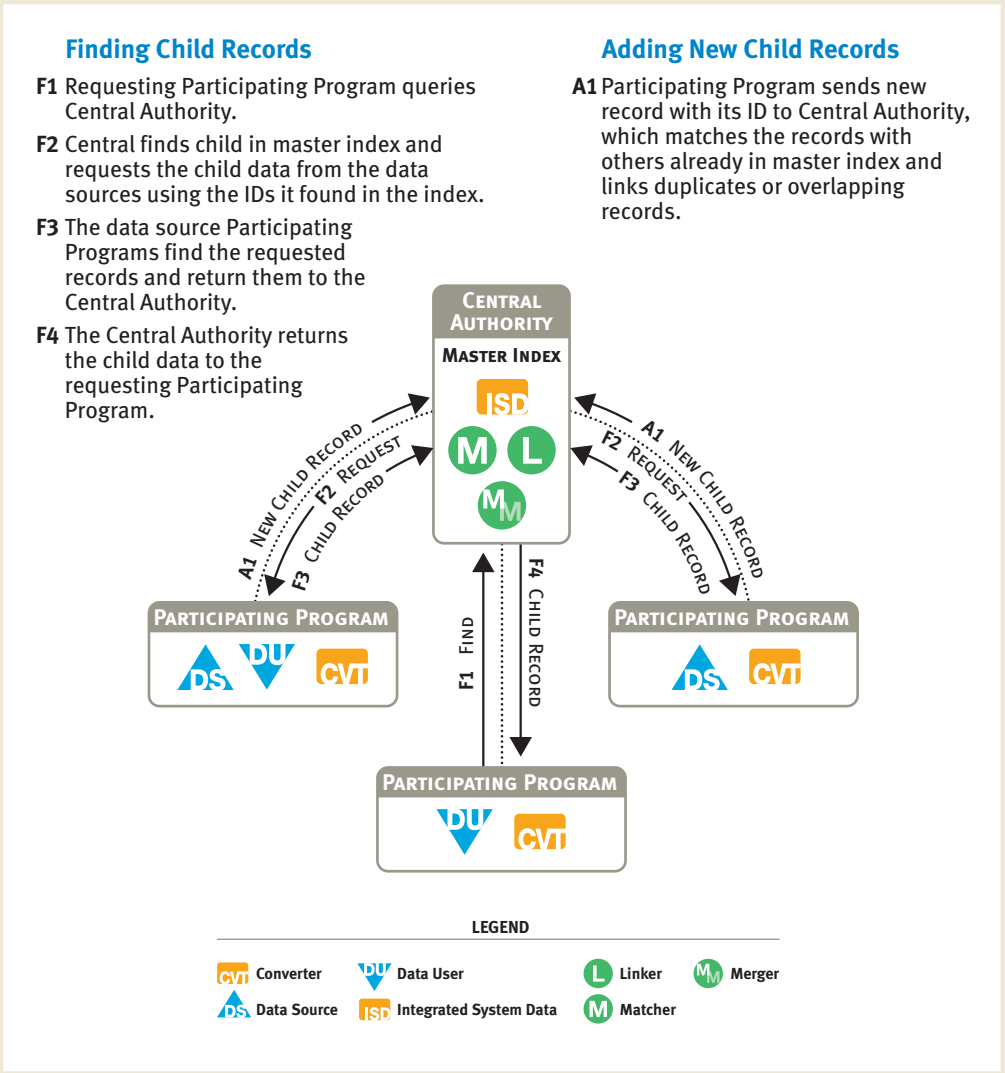


Figure 4: *Roles in a central index—information broker architecture*

Impact on Deduplication

1. Each of the participating programs operates independently of the central authority, so if the server goes down, the participating programs can still perform their primary functions. Working independently, however, they will not be able to find person information from other data sources or check whether a potentially new person already exists in another system.
2. Participating programs and the central authority rely on common data definitions (record fields and their intended use) to perform matching.
3. Participating programs must upload their person demographic and IDs periodically or immediately as person records are added, changed, or deleted. The frequency of the synchronization directly impacts the effectiveness of the RLS, and may affect its performance. Some problems associated with synchronization include:
   - Handling records for persons who have opted out.
   - If synchronization is done incrementally, which is typically the case for performance reasons, recognizing and removing records that a participating program has deleted can be problematic.

4. Central index architectures support linking directly via the master index.
5. Merging can be added to the system for data-cleaning purposes. Also, the architecture does not preclude each program's handling its own merging internally.
6. The three alternatives for record retrieval (Figures 2 - 4) vary in terms of performance, scalability, and security opportunities and challenges, but do not differ significantly with respect to deduplication.

Resources and Responsibilities

A central authority architecture requires significant hardware, software, and support resources.

Planning and Data Governance Issues

- The organizational context of the central index demands significant thought and planning. Specifically, the central index needs to be managed at a level within the organization that is consistent with the architecture. In other words, that managing group needs to have sufficient authority and detailed knowledge of the various participating programs to:
  - design a common data structure and require participating programs to use that data structure for uploading demographic information and IDs to the central index.
  - define policies and procedures for periodically uploading demographics and IDs to the central index.
  - establish and implement standards for the linking (and perhaps merging) of records.
- Data sharing agreements among the participating programs may impact the type of information that can be uploaded to the central index and used for matching. Policies and procedures for handling records (or not handling records) for those who opt out of data sharing require careful planning and design.

Role	Component(s)
Data source	Defines and distinguishes between data integration and application integration.
Data users	Discusses four challenges closely related to resolving duplicate data, and establishes a context for concepts discussed in subsequent sections.
Matcher	Describes deduplication in terms of generic roles that people or software play in the process of resolving duplicate data. It defines terms for describing and discussing deduplication.
Linker	Summarizes five prototypical system designs that represent a range of alternatives for how components in an integrated system can play different roles in the duplication process.
Merger	Introduces characteristics that can be used to evaluate or compare system designs with respect to their impact on the duplication process.
Global ID manager	Describes matching techniques in terms of when, where, and how matching can take place and what data elements are often used.
ISD repository	Describes three primary techniques for linking records in integrated systems.
Converter	Describes techniques for merging records in an integrated systems environment, prefaced by a discussion of the unique challenges that merging presents, and the preparations that lead to effective merging as part of the deduplication process.

Table 1: *Roles and components for central index architecture*

Risks and Dependencies

Participating program IDs are stored in the master index. This can make the master index a target for ID theft and abuse. It also creates a dependency between the RLS and the participating programs that can hinder the independent evolution of the systems.

Adaptations and Variations

- The central index can be partitioned and distributed for better performance.
- It can be replicated for better reliability.
- Even though the central index is basically a linking technique, merging can be added to the system and used to remove duplicate records from the same source.

Examples

New York City's Master Client Index is a central index approach, with an information-broker style of record retrieval. (See Case Examples, page 81.)

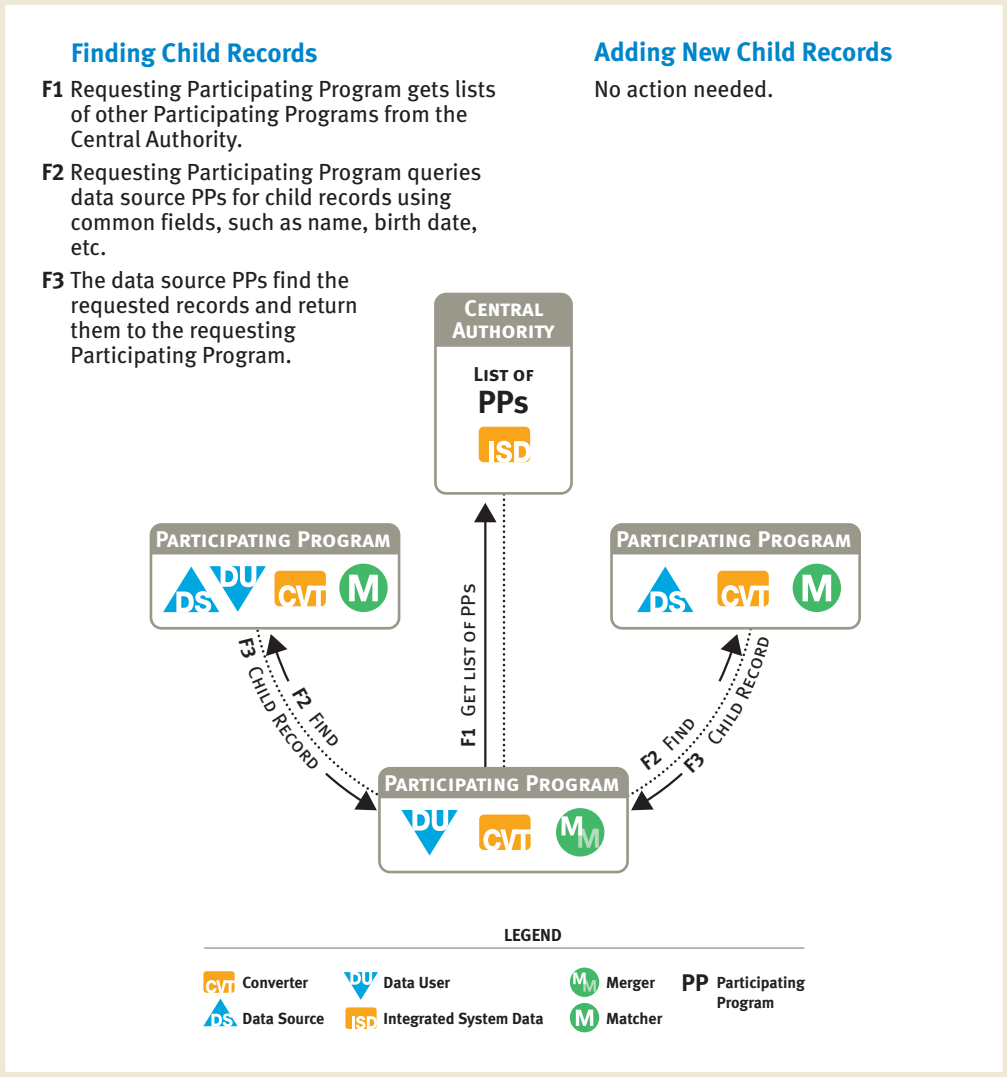


Figure 5: *Roles in a peer-to-peer architecture with a point-to-point communication protocol*

2. Peer-to-peer

Overview

In a peer-to-peer integration system, a data user (typically a participating program) locates information for a person by querying all data sources with some partial demographic information, such as a name and birth date. A peer-to-peer integration system (Figure 5) has three basic protocols:

1. *Targeted* or point-to-point. The data user sends a request directly to each data source. This requires that each data user know all the possible data sources.
2. *Broadcast*. The data user broadcasts the request to all data sources at the same time—like shouting out the request to all that can hear it. With this option, the data user doesn't have to know all the data sources.
3. *Facilitated*. The data user sends the request to an intermediate communication component that, in turn, sends it to all the data sources. With this option, the data user doesn't have to know all the data sources, but the intermediate communicator does.

A peer-to-peer architecture does not burden the central authority with the operation of substantial hardware and network resources.

Role	Component(s)
Data source	Participating programs or external data feeds.
Data users	Participating programs or external data warehouses.
Matcher	Each data source must have its own matcher.
Linker / Merger	Since data users receive records from multiple sources independent of each other, the data user must play the role of linking or merging those records.
Global ID manager	None
ISD repository	A directory of participating programs is necessary so data users can discover data sources.
Converter	Participating programs may include data converters for transforming information they exchange with other participating programs.

Table 2: *Roles and components for peer-to-peer architecture*

Resources and Responsibilities

- A peer-to-peer architecture does not burden the central authority with the operation of substantial hardware and network resources. The central authority only has to maintain a registry of participating programs.
- The central authority must take the lead in establishing standards for communication protocols and the structure and semantics (meaning). Compared to other architectures, each participating program must take more responsibility for security and confidentiality.

Planning and Data Governance Issues

With a peer-to-peer architecture, it is particularly important that data standards be established and that participating programs convert accordingly. Otherwise, each participating program must know and convert its data to match the structure and semantics of each of the other participating programs.

Impact on Deduplication

- Matching occurs with the participating programs, sometimes referred to as the *edges of the integrated system*, and may therefore suffer from inconsistencies. For example, consider an integrated system with two participating programs. Each one would have its own matcher. One matcher may be better at finding records with incomplete name information, while another is better at finding records with partial birth date information. As a result, a user may perceive the integrated system as being inconsistent since the integrated data for the target person can appear to change (i.e., contain different details, depending on the query used to find that person).
- Programs can participate as long as they adopt a standard communication protocol and provide the required record-matching capabilities.
- This architecture inherently uses a very loose form of record linking. Links are dynamically formed from the criteria specified in the record retrieval requests to two or more data sources. These links are not saved and are therefore hard to verify or track.
- This architecture does not support record merging across participating programs. Record merging can only take place within a single data source and is therefore independent of the data integration.

Risks and Dependencies

Changes to the structure or semantics to any data source may cause changes to all data users.

Examples

- Austin, Texas indigent care program for peer-to-peer evaluation for client eligibility.
- Regenstrief Institute for collecting information from various entities.

3. Arms-length information broker

The agents simplify potential data stewardship and program evolution problems.

Overview

Like central index architectures, an arms-length information broker (Figure 6) includes a centralized matcher supported by an ISD repository of demographic information. This, however, is as far as the similarities go. The ISD repository does not contain information about record sources and their specific IDs. Instead, it uses dedicated, program-specific agents to accomplish an arms-length linking of medical records. Each participating program or external data feed has a software agent that maps program-specific IDs to internal, non-public global IDs. The ISD repository uses those internal IDs to identify individuals in the system and, therefore, does not need to know the actual IDs from the participating programs. This reduces security vulnerabilities and dependencies among the participating programs.

When a participating program needs to retrieve data from other programs, it queries its specific agent, which transforms IDs to the internal format and passes the revised query on to the central server. In the process, data included in the query from the participating program's structure are converted to an internal structure used by the server. The

Finding Child Records

F1 Requesting Participating Program looks up child data via its agent, using its own ID for the child.

F2 The agent translates the Participating Program's ID for the child to an internal ID and sends the query to the Central Authority.

F3 Central Authority determines where it can find the requested data and asks for it via the appropriate agents.

F4 Each agent translates the internal ID to a Participating Program-specific ID and requests the data from the data source.

F5 Each data source retrieves the requested data and returns via its agent.

F6 Each agent translates the data from the Participating Program's format to a common format and sends it to the Central Authority.

F7 Central Authority combines the results and sends merged record back to the requesting agent.

F8 The requesting agent sends the merged record to the requesting agent.

Adding Child Records

A1 Participating Program sends new child data to agent.

A2 Agent translates it and sends to Central Authority.

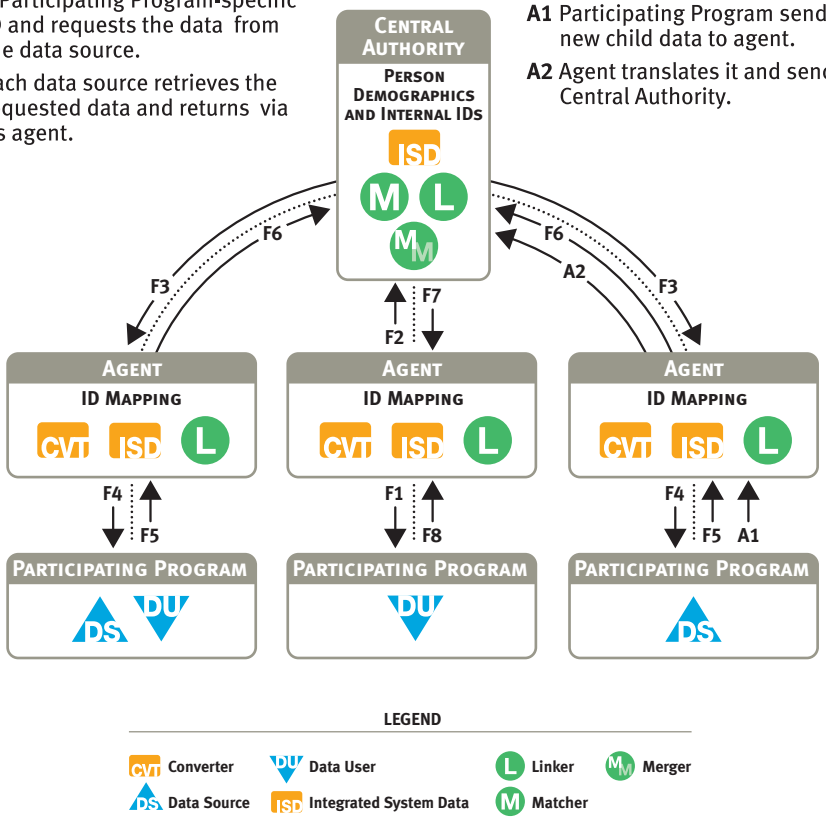


Figure 6: *Roles in an arms-length information architecture*

server determines where it needs to get the requested data and broadcasts the request for that data to all relevant participating programs. This step is similar to how queries work in a peer-to-peer approach. The server then waits for the data requested from the participating programs. After it receives the requested data, it merges them and returns the final integrated data to the requesting agent.

On the surface, an arms-length information broker may seem similar to an information broker type of central index, but there are some important differences from participating programs' point of view. Specifically, the arms-length information broker uses agents to play most of the roles that participating programs would otherwise play in an information broker. Although they add another layer to the architecture, the agents

simplify potential data-stewardship and program-evolution problems. For example, with an arms-length information broker, a participating program's IDs and data structures never go beyond its own agent so the server and other participating programs do not have to worry about their changing over time or the consequences of their becoming public. This reduces dependencies between components of the integrated system and thus allows participating programs to evolve independently. Also, many of the changes that a participating program would be required to make in becoming part of an integrated system can be relegated to the agent, thus simplifying what a participating program has to do to plug into the system. For example, it doesn't have to worry about converting data to a common data structure, communicating with the server, or handling global IDs.

Role	Component(s)
Data source	Participating programs or external data feeds.
Data users	Participating programs or external data warehouses, including private health care providers.
Matcher	The server includes a matcher.
Linker	Linking is handled by both the internal IDs and the ID mappings in the individual program agents.
Merger	The server has several options for handling merging: combine query results when participating programs return overlapping results; merge duplicates in the ISD when separate internal IDs are determined to be for the same individual; and merge records from the same source by simply changing the ID mappings in that source's agent.
Global ID manager	None
ISD repository	The server includes an ISD repository of demographic information used for matching and internal ID. Each agent also includes an ISD repository for ID mappings.
Converter	Each agent handles data conversion for its participating program or external data source.

Table 3: *Roles and components for arms-length information brokers*

Resources and Responsibilities

- A central authority needs to operate the server, and if necessary, host agents for those programs that can't host their own.
- A participating program must secure communications with its agent.
- A participating program may host its own agent, but if it does so, it must secure communications between the agent and the server.

Planning and Data Governance Issues

- For each participating program, decisions have to be made about what information it can share with others, what information it needs, and how it will communicate with its agent.
- In setting up ID mapping in the agent for a participating program, it is important to determine the following:
  - Does the participating program's information system use multiple records to represent a single person, or just one record?
  - How does the agent handle changes in IDs?

Impact on Deduplication

- Matching is centralized and therefore doesn't suffer from the inconsistencies that can plague the peer-to-peer architectures.
- This architecture uses record linking as its primary record coalescing technique, but unlike the central index architecture, the linking requires less coupling (fewer dependencies) between the participating programs and the server. The links are formed by combining the ID mappings in the individual agents and the core demographic data in the server. This two-level ID scheme isolates participating program IDs from others, thus helping eliminate unnecessary dependence between participating programs. It also provides convenient ways of managing links when IDs are changed or duplicates are merged.
- This also supports several types of merging. The server can use record merging to clean duplicates from its own ISD repository without immediately impacting the participating programs. Merges of duplicates from a single data source can be propagated back to that data source.

Architectures that do matching during information retrieval are more likely to appear slower.

Risks and Dependencies

- Establishing data-sharing agreements that are consistent with diverging confidentiality policies/rules.
- Uncoordinated changes to participating program databases that could invalidate shared data models.
- An ISD repository is potentially more vulnerable to security breaches since the IDs are held centrally. Delays can occur in assigning central IDs.

Lack of speed is not necessarily a problem inherent to the architecture and may be a result of implementation. Architectures that do matching during information retrieval, however, are more likely to appear slower than architectures that do matching during data loading, additions, or modification, as is the case with central index and arms-length information brokers.

Adaptations and Variations

An interesting variation may be to let the agents communicate with each other in a peer-to-peer fashion, instead of having all communications go through a central server. The advantage is that it would eliminate dependency on a central server. The disadvantage is that each agent would become considerably more complex, as it would have to implement a matcher.

Examples

Utah's CHARM system is an example of arms-length information broker. (See the Utah case example, page 89.)

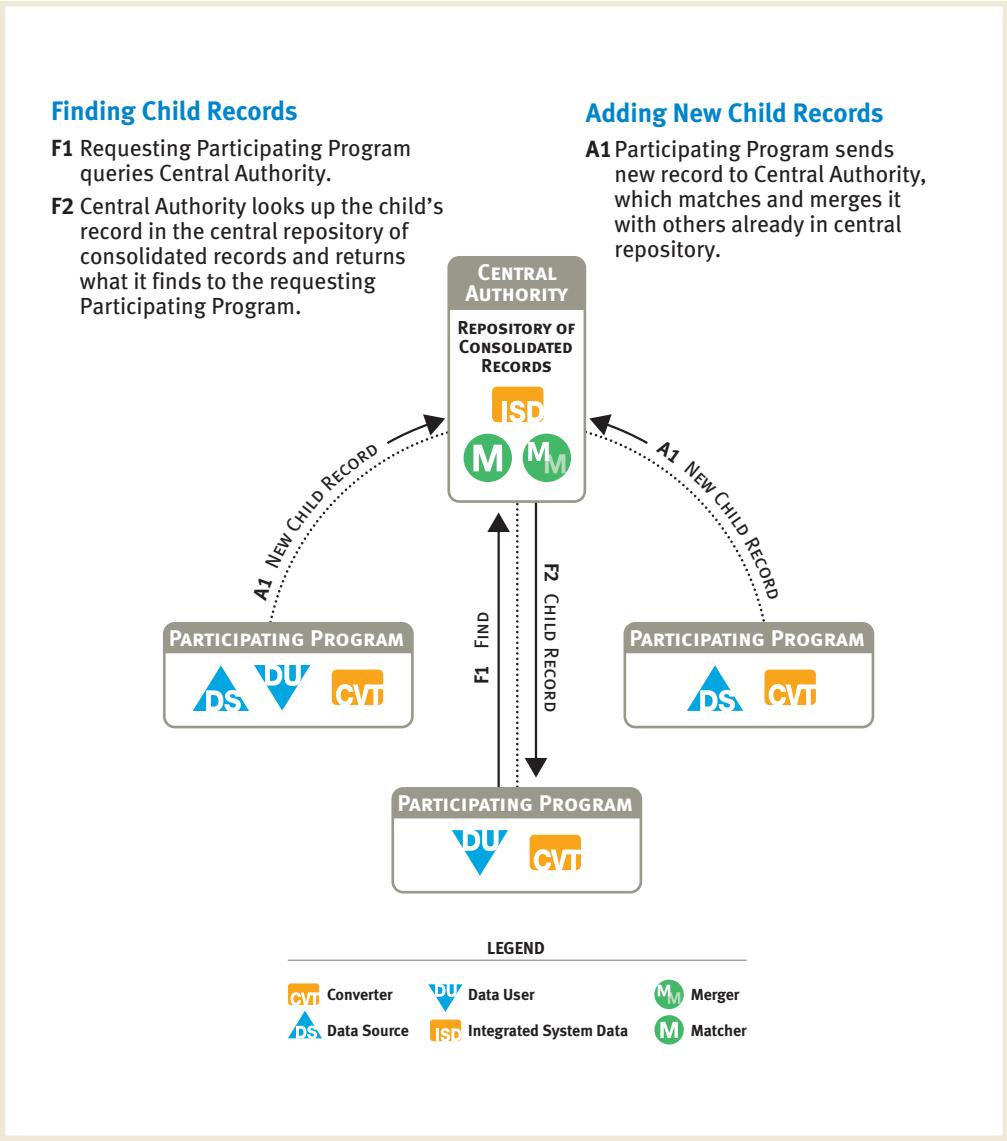


Figure 7: *Roles in a central database architecture*

4. Central database

Central Database
Related terms: Union Catalog (Arzt & HLN Consulting LLC, 2005)

Overview

In a central database architecture, a central authority assembles and operates a database of consolidated records, including medical, service, or program data (Figure 7). The participating programs may still keep their own databases, but those databases have to be periodically synchronized with the central database. This synchronization can be bi-directional, so changes made to records in the central database can be propagated back to corresponding records in the data sources. In others, the central database can support meaningful record update operations, in addition to the expected record retrievals.

Role	Component(s)
Data source	Participating programs or external data feeds, but with replication in the data vaults and central database.
Data users	Participating programs or external data warehouses.
Matcher	The central authority must host a matcher.
Linker	Although this architecture relies primarily on merging, the central authority may keep track of links back to the original sources. This can facilitate synchronization, particularly if it plans to propagate changes made to the central database back to the original data sources.
Merger	The central database handles all merging, which is the primary record coalescing technique.
Global ID manager	None
ISD repository	The central database is an ISD repository of merged records from all the data sources.
Converter	Each data source or data user is typically responsible for converting data to and from the standard structures used by the central database.

Table 4: *Roles and components for a central database architecture*

Resources and Responsibilities

The central authority must set up and maintain a central database. Because this database contains the shared information from all data sources, this may require a substantial amount of computer and network resources.

Planning and Data Governance Issues

- Participating programs are required to periodically supply their data to the central database. The complexity of this architecture is in this synchronization. It must deal with the following kinds of problems:
  - Conflicts
  - Deleted records
  - Records for those who opt out
- The central database relies on standards for data formats, message types, and communications techniques.
- The central database must be certain that changes made to the central database are propagated back to the source. Otherwise, participating programs may lose control of their data, and data inconsistencies could proliferate within the system.

Impact on Deduplication

- Matching is centralized, as with the central index and arms-length information broker architectures.
- The database operated by the central authority contains merged records of all people and their medical data. This database is sometimes called a "union catalog."
- Even though the central database contains merged records, it can contain links back to the original data sources.

Risks and Dependencies

Queries of consolidated records do not rely on real-time communication with the participating programs. The program systems can go down and the integrated system can still be available.

Examples

Missouri's MOHSAIC is a type of central database. (See the Missouri case example, page 76). Missouri client demographic information is captured once in an Oracle database and shared by all users. As additional components are added the data are included as part of the client's record. The system interfaces electronically with Medicaid, Vital Records, and the state public health laboratory. Users access the system through a secure network or through Web applications. There is no central index.

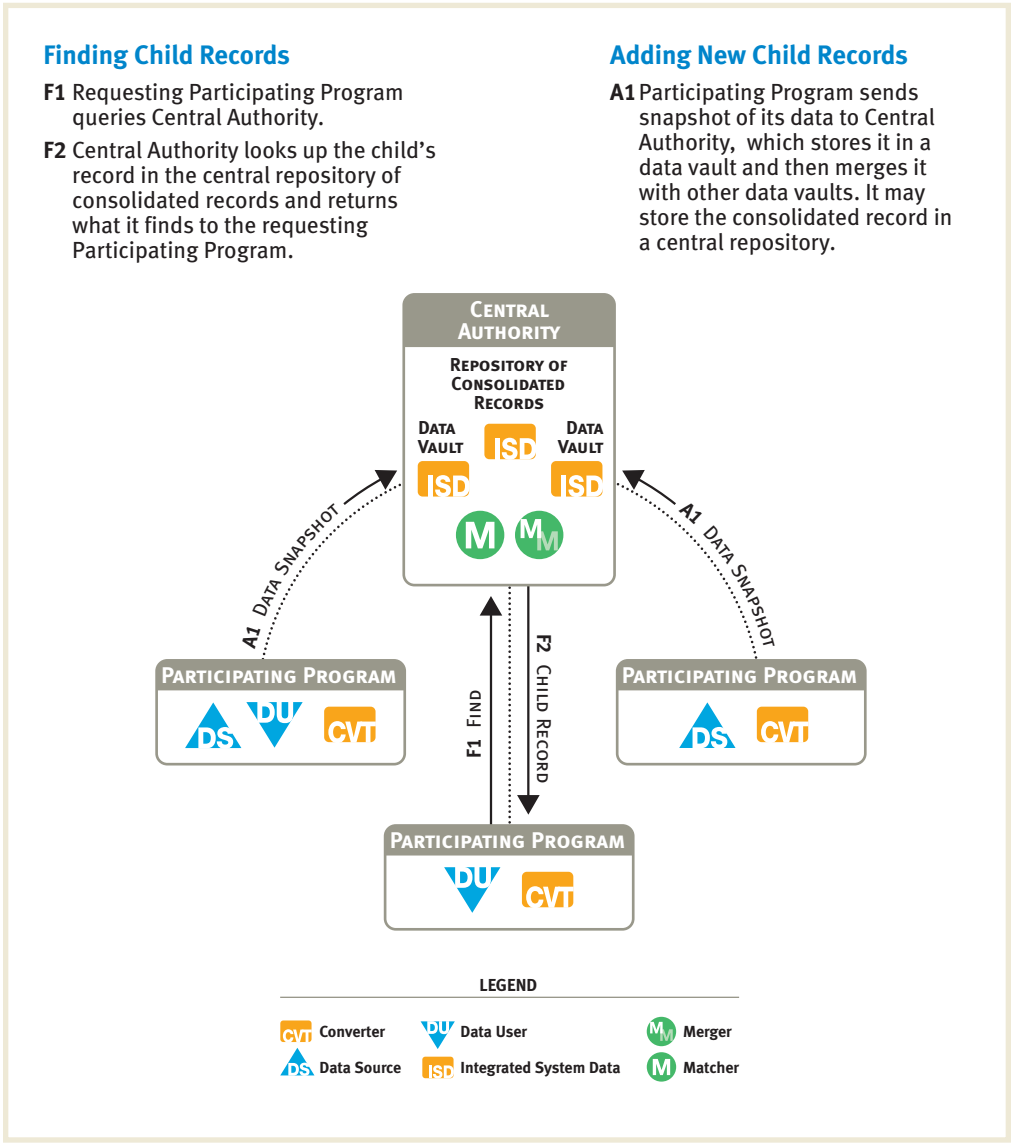


Figure 8: *Roles in a partitioned central database architecture*

5. Partitioned central database

Overview

Partitioned Central Database
Related terms: partitioned warehouse, data vaults (Arzt & HLN Consulting LLC, 2005)

Like a central database, a partitioned central database (Figure 8) is a database of consolidated records including medical, service, or program data. Instead of a single central database, however, the central authority maintains a collection of ISD repositories, called data vaults, typically one for each participating program. When a participating program synchronizes its data with the central authority, it simply synchronizes with data in its corresponding data vault. Since the synchronization doesn't involve any merging, it is a simpler operation than the synchronization in the central-database architecture. Merging can then occur independently by extracting records from the data vaults, combining them, and storing them in a central database. There are several options for when the merging occurs, including: on-the-fly when a record is requested; on a regular schedule (e.g., after business hours); and after a specific event such as a synchronization operation.

One other key difference between the central database architecture and the partitioned central database architecture is that the data in the data vaults (and the central database) are considered to be "read-only." Consequently, there is less need to keep links back to the original records in the data sources.

Role	Component(s)
Data source	Participating programs or external data feeds, but with replication in the central database.
Data users	Participating programs or external data warehouses.
Matcher	The central authority must host a matcher.
Linker	Not typically used with the architecture.
Merger	Data from the data vaults are merged either on-the-fly, periodically, or after some event. The merged records are stored in the central database.
Global ID manager	None
ISD repository	The data vaults and central database are ISD repositories.
Converter	Each data source or data user is typically responsible for converting data to and from the standard structures used by the central database.

Table 5: *Roles and components for a partitioned central database*

Resources and Responsibilities

The central authority must set up and maintain the data vaults and the central database. As with central database approaches, this architecture may require a substantial amount of computer and network resources.

Planning and Data Governance Issues

- As with the central database approaches, participating programs are required to periodically supply their data to the central database and deal with synchronization complexities.

- The Partitioned Central Database relies on standards for data formats, message types, and communications techniques.
- Synchronization can be simplified since each participating program can have its own data vault. Off-the-shelf database replication is often all that is needed.
- This approach may work well when there is limited need or no need to push data updates among the participating programs, and when there is a strong requirement for quick access to merged records, regardless of whether the original data sources are network accessible.
- This approach may also work well if the matching and record consolidating operations need to be outsourced.

Impact on Deduplication

- Matching is centralized, as with the central database, central index, and arms-length information broker architectures.
- This architecture relies primarily on merging. Records from the data vaults are combined to form a consolidated record for a person. For performance reasons, the central authority can store merged records in a central database.
- This architecture typically has little need for linking.
- Since matching and merging are centralized and operate on data that are likely to already be on the same machine (i.e., in the data vaults), these operations can be very efficient compared to other architectures with other factors considered equal.

Adaptations and Variations

- Instead of storing merged records in the central database, the central authority could create them on demand. In this case, the central database doesn't go away completely, but is greatly simplified and is like a master index that keeps track of how all the records link together.
- This approach can be married with other approaches, particularly the central index and the arms-length information broker.

Examples

- The Tennessee Volunteer e-Health Initiative Vault System is a partitioned central database. It uses a VUMC StarChart records system and includes comprehensive medical information, such as imaging. It follows the standard of the Consolidated Health Initiative (CHI) which includes HL7, SNOMED, and LOINC.
- The Indiana Health Information Exchange, organized by Regenstreif Institute, the State of Indiana, Marion County / Indianapolis, private health care organizations, and other participants, is a pioneering example of this architecture and also a leader in standards development (i.e., LOINC and HL7) (McDonald et al., 2005).

6. Identifier-based approach

Overview

This section describes an approach that can be used to simplify or improve the performance of person identification in other architectures but is otherwise external to the architecture itself. We include the identifier-based approach (Figure 9) in the summary of generic architectures because it is widely discussed as a design choice for identification, which, if used, would have a dramatic impact on the deduplication process.

As the name suggests, this approach uses identifiers, typically numeric or alphanumeric keys. An identifier uniquely identifies an individual known to the integrated system. If the identifier is external, it is carried or known by that person. For example, in

Identifier-Based Approach
Related terms: Unique Medical Identifier (Massachusetts Health Data Consortium, Inc., 2004)

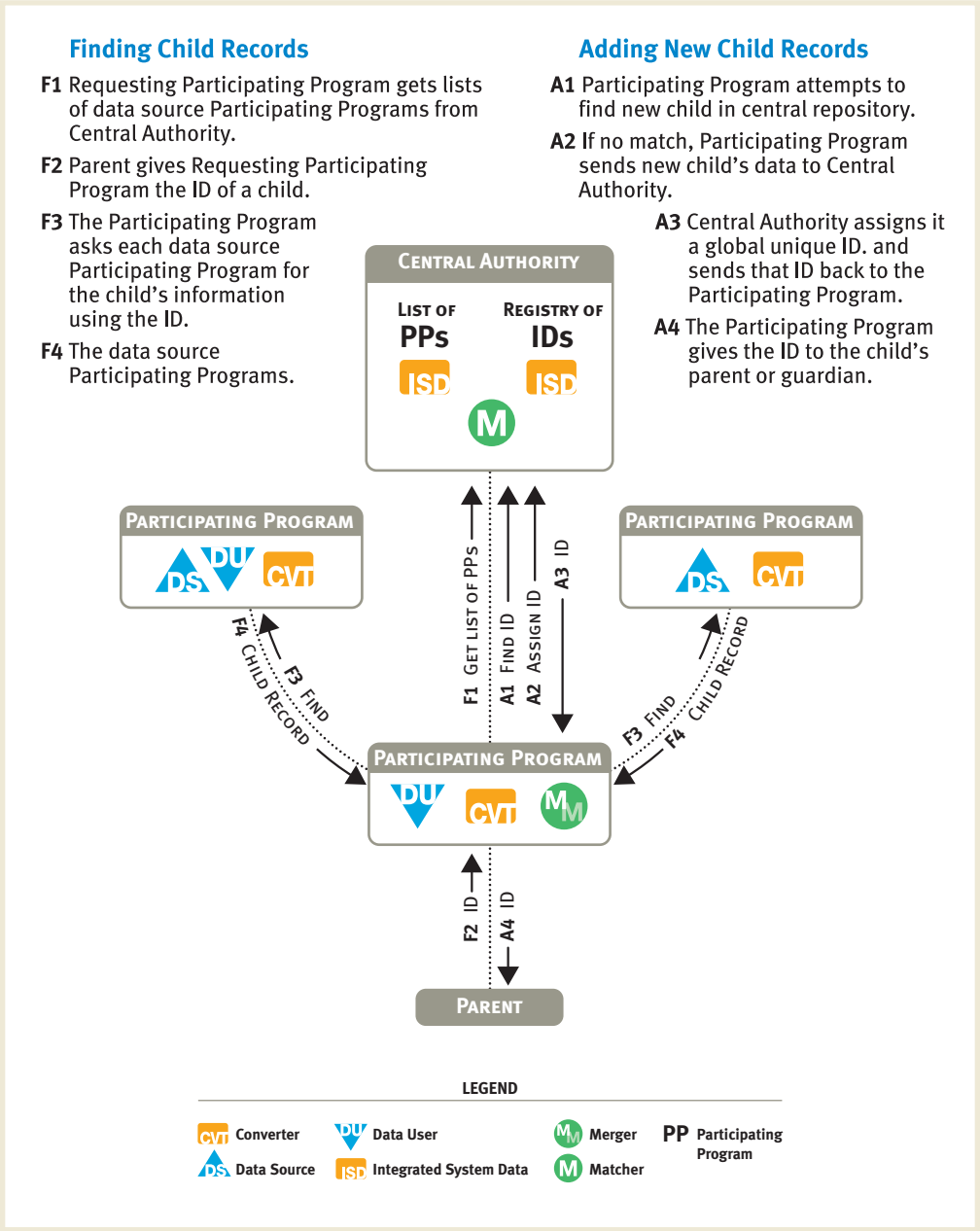


Figure 9: *Global unique ID added to the peer-to-peer architecture*

Medicare systems, recipients of service have Medicare numbers. The identifier may appear on a paper record or it may be hidden in a Smart Card or magnetic strip card. Either way, a patient needs to present it at the point of service, so a user of the integrated system can pull up that patient's integrated data. If a person doesn't have an ID, lost it, or forgot it, the user must be able to search a central registry using demographic information about the person. Assigning new IDs needs to be coordinated with a central authority to ensure that IDs are not duplicated and that new persons are properly registered.

Role	Component(s)
Data source	Participating programs and external data feeds.
Data users	Participating programs and external data warehouses.
Matcher	A component, coupled with a Global ID manager, that would be used in the process of assigning new IDs and in recovering lost or forgotten IDs.
Linker	A global identifier acts as a logical linking mechanism for accessing all of a person's medical data.
Merger	There's no inherent need for a merger, but the Global ID manager could include one to handle special cases in which, a person erroneously received two IDs.
Global ID manager	A central authority plays the role of global ID manager and would include a matcher that would be used in the process of assigning new IDs and in recovering lost or forgotten IDs.
ISD repository	The Global ID manager needs a repository that holds information about IDs and to whom they are assigned.
Converter	The participating program may include data converters as part of their communications with the other programs.

Table 6: *Impact of using external identifiers on deduplication*

Resources and Responsibilities

- The central authority must operate a global identifier manager.
- The central authority will probably have to provide a way to recover lost identifiers, which may take the form of a central index or central database.

Planning and Data Governance Issues

- Putting a system based on unique personal identifiers into practice has some significant challenges, including difficulties with incremental deployment (Connecting for Health, 2004; Connecting for Health Working Group on Accurately Linking Information for Health Care Quality and Safety, 2005; Massachusetts Health Data Consortium Inc., 2004).
- This approach can work when there is an organization that can manage and enforce the integrity of the identifiers. In other words, it can work for closed enterprise-wide systems, but it may not work at a national or even regional scale, where the integrity of the identifiers faces numerous threats.
- This approach overrides global opt-in or opt-out decisions as this decision is made when an individual presents (or withholds) identification at the point of service.

Impact on Deduplication

- Theoretically, this approach could greatly simplify matching because the global identifier is supposed to uniquely identify an individual. The uniqueness of the number

may be hard to guarantee, though, as the keepers of the identifier (the individuals) are external, and the identification might be stolen or fraudulently used by another person. A sophisticated matcher may still be needed to handle the recovery of lost identifiers.

- The identifier is a logical record link that coalesces a person's records from various data sources. So this approach uses record linking as its primary record coalescing technique.
- Although this approach does not preclude record merging, merging can be very difficult. It would have to deal with all the issues surrounding the changing and retiring of global identifiers.

Risks and Dependencies

Stolen identifiers can represent a significant security risk, if personal and confidential information is stored with the identifiers, as may be the case with Smart Cards. Furthermore, if the integrated system relies on the identifier as part of its authentication mechanism, stolen identifiers or identity fraud can present additional security risks, regardless of the system's architecture.

Adaptations and Variations

- Solutions for handling lost identifiers may use either central index or central database architectures.
- The central authority may act as a broker for queries among participating programs. Alternatively, participating programs may be allowed to query each other directly.

Examples

- Identifier-based approaches have been discussed at length in the context of whether a national identifier is needed to support interoperable health information exchange. There is currently no unique national identifier and this approach has not yet been used in an open integrated system without a strong central authority.
- Virginia is using this approach in a closed, enterprise-wide system.

Other approaches

As stated earlier, there are as many deduplication strategies as there are situations. The architectures discussed are by no means a complete list. These were selected for study and explanation because they represent some of the directions taken in child health information systems projects and can illustrate how different components can take on different roles. Some other approaches that go beyond the scope of this document include:

- **Portable-media architectures** that rely on medical records being stored on cards or other devices that individuals carry with them and directly control their use.
- **Hybrid architectures** that combine ideas from one or more of the architectures discussed.
- **Transitional approaches** that use a temporary architecture as a stepping-stone to some future architecture.

V. Comparison Characteristics

This section introduces various characteristics that can be used to evaluate or compare system architectures with respect to their impact on the duplication process.

Evaluating or comparing architectures for integrated information systems and, more specifically, how they impact deduplication can be difficult, even for experienced systems analysts and engineers / technical architects. The challenge is that there are many factors that can produce significant advantages or disadvantages. Nevertheless, the process of evaluating and comparing possible solutions is a critical part of any integration project. To simplify this process, this section lists and briefly discusses 10 of the most discriminating characteristics. It suggests ways of measuring or rating possible deduplication solutions with respect to each of these characteristics.

Location of resources

List the additional computer, network, and human resources that are needed for each architecture.

As illustrated in Section 7, Common Software Architectures, one of the most important differences among integrated system architectures is the distribution of key deduplication roles among the various software components. The distribution dictates the type and level of resources required at the participating programs or at a central authority. In some cases, as with central index, central database, and partitioned central database architectures, a central authority must operate a relatively robust server (or collection of servers) and provide human resources for support and for any interactive matching or merging between participating programs. A peer-to-peer architecture requires relatively few central resources and more local resources for each participating program, since they directly handle requests broadcast from other participating programs.

One method for evaluating architectures with respect to this characteristic is to simply list the additional computer, network, and human resources that each architecture requires for each participating program and at some central authority.

Degree and type of coordination required for data sharing and deduplication

This characteristic deals with necessary coordination among stakeholders to effect data sharing and typically involves formulating data sharing agreements, defining data models or data mapping specifications, and maintaining these items as participating systems evolve. Although such coordination depends more on organizational structures than on software, some architectures lend themselves to different types of coordination better than others. To evaluate candidate architectures with respect to coordination, consider the following issues for each one:

- *Central versus distributed control.* Does the architecture require a strong central authority for setting data or communication standards?
- *Big bang versus incremental implementation.* Does the architecture require the integrated system to be deployed in one big step or can it be rolled out in incremental steps?
- *Strong versus weak information boundaries.* Are there well-defined boundaries between the software components that minimize dependencies among the compo-

nents? Architectures with strong information boundaries isolate a participating program so the structure of its data can change independently of others, and so it can retain stewardship over its own data. Architectures with weak information boundaries may require the participating programs to adopt a common data model and may blur the data stewardship (i.e., who is responsible for various pieces of data). Arms-length information broker architectures, for example, facilitate strong information boundaries.

Degree and type of coordination required among underlying data models

Consider the degree and type of coordination among underlying data models explicitly and independently of the general architecture.

This characteristic is related to the previous ones but focuses on the coordination of any underlying data models. Some architectures, such as central index and central database, may require a *common data model*. Others, such as the peer-to-peer, may require each participating program to know something about the structure of every other program's data. In other words, they use *coordinated data models* as opposed to a *common data model*.

Unfortunately, degree and type of coordination is not entirely determined by the type of architecture. Each type of architecture offers sufficient flexibility, with significant variations in actual implementations. So, in evaluating specific solutions, it is important to consider the degree and type of coordination among underlying data models explicitly and independently of the general architecture. Looking at whether the coordination allows for independent versus lock-step evolution of the data models may shed some additional light on the degree and type of required coordination.

Degree and type of coordination required for communication

This characteristic focuses on communications among the components and the coordination that these communications require. As with the coordination required for the underlying data models, coordination for communications may depend on something common to and agreed upon by all stakeholders, or may require individual participants to know something about all the others.

Scalability

Scalability deals with an architecture's ability to grow with respect to the number of participating programs and the amount of information that it can handle. A simple way to characterize and compare scalability is to list ways an architecture either enables or deters growth. The following questions can help in this regard:

- If another participating program were to be added to the integrated system, how would this impact the load on the central authority and its resources?
- If another participating program were to be added to the integrated system, how would this impact the load on other participating programs and their resources?
- If a participating program significantly increased (e.g., >10 percent) the number of records it handled, how would this impact the load on the central authority and its resources?
- If a participating program significantly increased (e.g., >10 percent) the number of records it handled, how would this impact the load on other participating programs and their resources?

Availability

Systems vary in their level of availability to users during failures and system attacks. A high-level availability system allows the system to remain accessible, accurate, and secure in spite of failures. Like scalability, availability may be difficult to measure in concrete terms, but a general assessment could be made by using these key questions:

- If a participating program's information system becomes unavailable, can others still continue to work?
- If there is a central system and it goes down, can participating programs continue to work?
- If the network fails, to what degree can the participating programs continue to work?

Support for systemwide analysis

A simple way to characterize support for systemwide analysis is to document how the integrated system facilitates or deters quick access to bulk data.

This characteristic deals with whether the integrated system facilitates systemwide analysis of person data. To do this, the architecture must support efficient access to large amounts of person data without necessarily any identifying information. Solutions based on the central database and partitioned central database approaches typically lend themselves to this type of computation. Solutions based on a peer-to-peer architecture may have to provide extra features, such as data marts, to support systemwide analysis. A simple way to characterize support for systemwide analysis is to document how the integrated system facilitates or deters quick access to bulk data.

Timeliness

Timeliness deals with the currency of the shared data. Some architectures, such as the peer-to-peer and arms-length information broker, retrieve information directly from participating programs as needed and therefore provide access to the most current information. Others, like the central database, retrieve information from copies of the data. These copies can be out-of-date to some degree. To characterize the timeliness of the data for a particular solution, simply document the likelihood of the data being out-of-date for various time intervals (i.e., probability of the data being one hour out-of-date, one day out-of-date, one week out-of-date, etc.).

Adaptability

Adaptability is the ability to conform to different or changing circumstances. For deduplication, this includes the ability to use different matching, linking, and merging techniques. To analyze the adaptability of a possible solution, consider and document the following questions:

- Does it require a specific matching algorithm? (See Matching, page 51.)
- Does it allow linking or merging, or both? (See Linking, page 59, and Merging, page 63.)
- If it allows merging, does it require any specific merging techniques?

System performance

The system speed in terms of the time it takes to execute common operations can be another key discriminator among candidate architectures. From a deduplication perspective, users must have reliable, integrated data in a timely manner. Although the overall

complexity of an architecture can have an impact on perceived speed (and specifically, the average query's turn-around time), the selection of which components play the matching and merging roles has a bigger impact. For example, if matching is done by participating programs when they receive queries for information (as is the case in peer-to-peer architectures), then the turn-around time is slowed by one or more matching operations. Since matching is typically a time-intensive operation, it can cause peer-to-peer architecture to seem slower compared to other architectures, even though the peer-to-peer may be simpler to implement from an organizational perspective.

When evaluating the performance of candidate architectures, other factors besides average query turn-around time should be considered, such as:

- *Throughput*—How many operations (i.e., queries, additions, etc.) can the system handle in a fixed period of time?
- *Capacity*—How much information can the system handle before its performance drops below acceptable thresholds?
- *Response time*—How quickly does the system let the user know that it has received a request for data and is processing the request?

VI. Matching

This section explains matching techniques in terms of when, where, and how it can take place and what data is often used.

Matching (Figure 10) is the process of finding existing records that might be for the same person (Salkowitz & Clyde, 2003). The key issues for matching are:

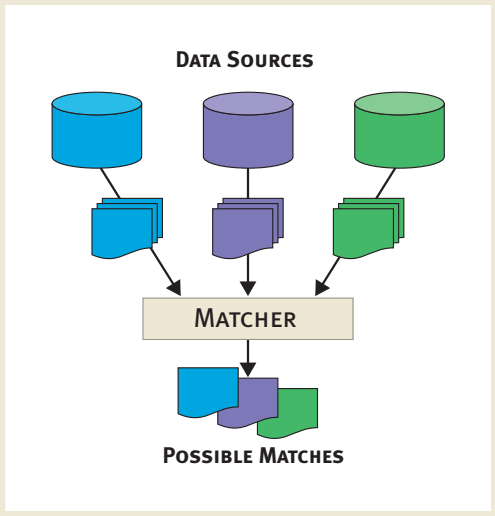


Figure 10: *Matcher roles*

- **When** does matching take place?
- **Where** in the integration system does it occur, and therefore, who is responsible for it?
- **How** is it done?
- **What** data fields are used for comparison?

When:
Front-end versus back-end matching

In general, matching can occur either as a user enters data into a system or afterward when the record is already in the database. The first approach, called *front-end matching*, can be interactive and takes advantage of additional knowledge that the user may have about an individual, especially if the individual is present. The second approach, called *back-end matching*, can work on many records at a time and may not require human interaction (Salkowitz & Clyde, 2003).

Front-end matching allows users to look for potential matches prior to adding a new record.

Back-end matching allows users to enter new individuals without initially worrying about duplication.

Systems that support front-end matching allow users to look for potential matches prior to adding a new record into the system or during the process of adding a new record. If a match is found, it is used instead of adding the new record. The aim of front-end matching is to minimize the number of duplicate records that actually get into the database. It can also take advantage of a user's first-hand knowledge about the person (Salkowitz & Clyde, 2003), but is also dependent on the quality of the user's training and persistence in effectively querying the database.

Back-end matching allows users to enter new individuals into the system without concern about duplication. The matching looks for duplicate and overlapping records behind the scenes at a later time. Because back-end matching does not rely on user interaction at the time of data entry, it is a good choice for batch data loading or external data feeds. The results of back-end matching, however, may not be as good as front-end matching because they cannot take advantage of first-hand knowledge about the individual and may end up in the human review queue at the end of the process.

When?	Pros	Cons
Front end	• Reduces initial generation of duplicate records. • Allows interaction with patient to clarify identity. • Keeps duplicate data from getting into the system.	• Resolution of duplicates may be out of locus of control and require other policy or procedures among participating programs. • Dependent on skill and training of user. • May be inefficient for batch data loading or other kinds of automated data feeds.
Back end	• Good for quickly loading lots of records into the system. • Can employ more sophisticated algorithms and make use of data that may not be displayed to the user as a result of opt-out or data-sharing restrictions.	• Cannot take advantage of first-hand knowledge of patient. • Reconciliation of uncertain matches can be time consuming and costly. • Duplicate or overlapping records may never get resolved, lessening the overall quality of the system's data.

Table 7: *Front-end versus back-end matching*

Where and who: Location

As described in the Deduplication background section (page 27), the overall architecture of the integration system determines what components play the matcher role. For some architectures (e.g., central index, arms-length information broker, central database, and partitioned database), a central component acts as the matcher. In these cases, an effective central authority should host the matching operation and have the ability to establish or help establish appropriate requirements for:

- How matching will be done (i.e., the algorithm, and what information participating programs can or should send the matcher). (See *How: Matching Algorithms*, page 52, and *What: Data Fields Used in Matching*, page 58.)
- What should be considered a definite match, uncertain match, and a definite non-match, depending on the algorithm.
- The procedures for linking or merging same-source duplicates, multi-source duplicates, and overlapping records. (See Linking, page 59, and Merging, page 63.)

For other architectures (e.g., peer-to-peer), participating programs act as the matchers. Because the participating programs are typically the data source or data users

of an integrated system, or both, they are sometimes referred to as being "on-the-edges" of the integration infrastructures, and matching is done by the participating program as "on-the-edge" matching. With on-the-edge matching, each participating program must host a matcher and deal with all the issues discussed above. With central matching, a central authority can help direct the deduplication process so there is better consistency across the integrated system. Table 8 lists pros and cons of the two choices of central matching and on-the-edge matching.

When?	Pros	Cons
Front end	• Consistency • Simpler data and communication standards	• Possible performance bottleneck • Risk of single point of failure
Back end	• No need for the central authority to operate a server, except to provide a directory of participating programs	• Multiple matcher implementations • Possible inconsistencies in matching quality and results

Table 8: *Central versus on-the-edge matching*

For either choice, system design techniques can be used to maximize the pros and mitigate the cons. For example, the potential performance bottleneck and single point of failure for the central matching approach can be softened by using a redundant server that operates concurrently with the main server. If the main server goes down, this back-up server keeps the system running seamlessly.

How: Matching algorithms

Hundreds of matching products are available commercially, and perhaps just as many, if not more, custom or in-house programs perform some kind of matching. The way each of these actually does matching is referred to as its *matching algorithm*.

Matching algorithms range from relatively simple to extremely complex, and they differ in significant ways. Figure 11 on page 54 illustrates this range as a linear spectrum with simple algorithms on the left and complex algorithms on the right. Some of the ways in which they differ are shown below the line. Although this one-dimensional spectrum is a tremendous simplification of the entire domain of matching algorithms, it provides a convenient overview of the relative complexity of algorithms and how they differ.

Single-step versus multi-step algorithms

Algorithms on the left side of the spectrum tend to treat the matching problem as if it were just a database record-retrieval problem and try to accomplish the match in one step. They rely on a fixed data structure with specific fields (e.g., name, address, phone, etc.) and allow a user (e.g., someone or something requesting matches) to search only by these fields. This approach allows the matcher to use standard database queries to do the actual matching. Such single-step query-based algorithms can be very fast and easy to implement but are limited in terms of matching strategy.

In contrast, those on the right side of the spectrum usually divide the matching process into at least two steps: one for retrieving candidate records and another for clustering potential matches. Although the first step involves retrieving records from a data-

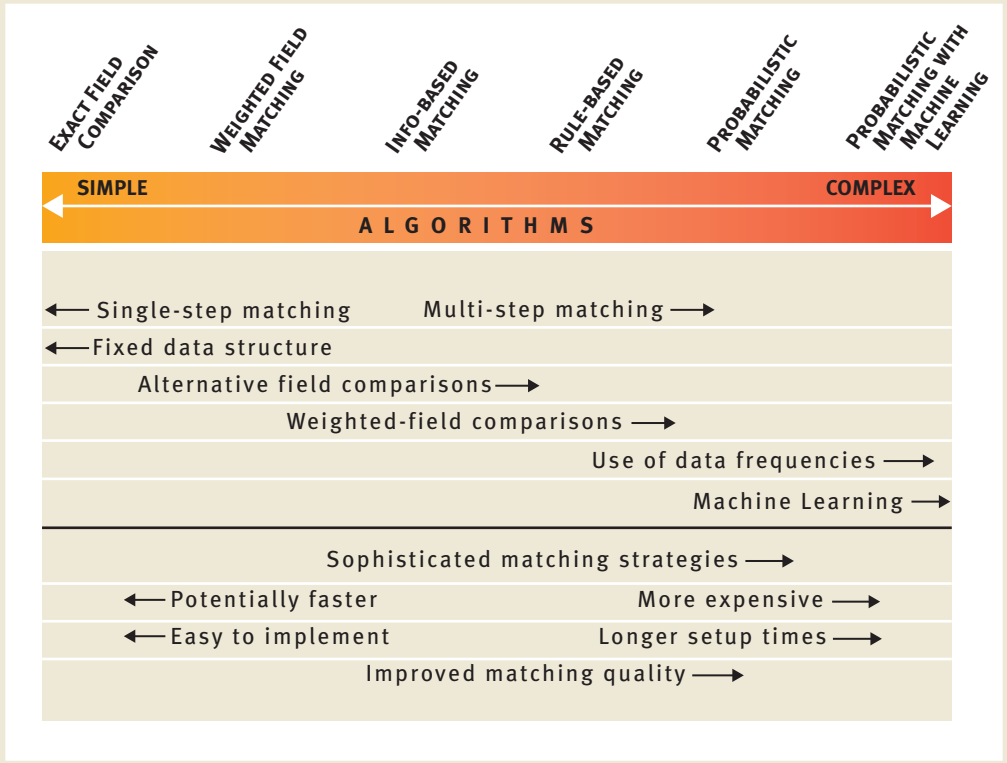


Figure 11: *Relative complexity of matching algorithms*

base, it differs from single-step query-based matching in intent. The purpose of this step is to select a set of candidate matches, not to find exact matches, so the query can be less exact. Therefore, the candidate set is more likely to contain multiple records for a given individual that are similar but not exactly the same, if such records exist. The second step then works through the candidate set and tries to cluster matching records. Multi-step matching algorithms can provide greater flexibility with their matching strategies but are more complicated to implement.

Field comparisons

Simpler algorithms use *exact-field comparison,* which allows users to provide possible values for some or all of the data fields, and the matcher searches for records that have those values. This approach works well with the single-step algorithm, because exact-field comparison lends itself to building efficient database queries. More advanced algorithms provide alternate comparison methods, such as:

- *Relative comparisons*
  In addition to the basic "equal" comparisons, relative field comparisons include "less than," "greater than," and "not equals" comparisons.
- *Range comparisons*
  This category includes comparisons for determining whether numerical or date values are within specified ranges. For example, a matcher may consider two person records to be the same if their names are the same and the birth dates are within three days of each other.

- *Containment comparisons*
  This category includes comparison functions that can determine whether a field value is either fully or partially contained within another.
- *Partial string*
  This category includes string comparison functions that limit the comparison to a specific number of characters (e.g., the first five letters of the last name).
- *Wildcards*
  Wildcards allow a matcher to compare values using string patterns. For example, a star "*" typically represents a wildcard that matches any sequence of characters. Using this wildcard, the string pattern, "joh*son", would match "johnson" and "johanson".
- *Fuzzy-date comparisons*
  Fuzzy-date comparisons allow the matcher greater flexibility in comparing data by allowing certain parts of dates to be left unspecified. "Jan-2006" is an example of a fuzzy date because the day is not specified and would match any other date in January 2006.
- *Edit-distance comparisons*
  This type of comparison gives the matcher information about how similar two values are. Specifically, it returns a number that indicates how many character insertions or deletions would be necessary to make the strings the same. For example, "johnson" and "johanson" would have an edit distance of 1, since only an "a" would have to be inserted into the first word after the "h" to make the second word. The words "danny" and "dani" would have an edit distance of 3.
- *Soundex Matching*
  Soundex comparisons match words (typically names) with different spellings but similar sequences of character sounds. They do this by first removing non-essential characters (i.e., all non-initial vowels, H's, Y's, and W's) from the words in the string and then, based on a set of rules, encode the remaining characters as a sequence of digits. These numerical sequences represent standardized sounds for key letters; they do not represent the pronunciation of the words in the strings. Two strings are then compared by these corresponding Soundex encodings. Robert Russell first proposed the original Soundex idea in 1918, long before the electronic information system. Since then, researchers have proposed many variations of the idea (Salkowitz & Clyde, 2003). Today, many database systems provide direct support for information retrieval based on Soundex comparisons.
- *Orthographic comparisons*
  Unlike Soundex comparisons, these functions attempt to compare words (typically names) based on their pronunciation (Salkowitz & Clyde, 2003).
  For example, with orthographic comparison, "danny" would make "dani" match.
- *Geo-distance comparisons*
  Geo-distance comparisons, which are only applicable to addresses, can give matchers an indication of how close (in miles, for example) two addresses are. They use standard address cleaning and geo-encoding software to convert addresses to global locations and then compute the distances between those locations.

Each type of field comparison provides some additional flexibility, and thus enables more sophisticated matching strategies. Simple matchers may support only a few different types of comparison; advanced matchers may include all of the above, and even more.

Weighted-field comparisons

Just to the right of the simplest algorithms in Figure 11 are those capable of assigning more importance to specific fields by giving them more weight in the overall computation of matching result. For example, the exact matching of an ID field could carry more weight than the exact matching of a name.

Typically, a weighted-field matching algorithm computes a match confidence by summing the results of individual field comparisons multiplied by a field weight. The bigger the weight, the more the results of that field comparison impact the final match confidence. If the confidence factor exceeds a pre-determined threshold, then the record is considered a match. Some algorithms have two thresholds that represent cut-off levels for high-confidence matches that need no further confirmation and probable matches that require human review before declaring them matches.

Information-based matching

Info-based matching can provide for richer matching strategies.

Beyond weighted-field comparisons, some matching algorithms try to extract and compare information, not just data. For example, a matcher may try to discriminate between twins even when data records don't explicitly contain fields that indicate whether individuals are twins. So, the matcher tries to infer whether two people are twins by looking at birth dates, birth locations, and mother information. Inferred information used in matching is sometimes referred to as *matching clues* (Borthwick, Buechi, & Goldberg).

Because information-based matching extracts and compares the meaning of data instead of just raw values, it can provide for richer matching strategies. Info-based matching is also one way of handling variations in data structures.

Rule-based matching

Further to the right in Figure 11 are rule-based matching algorithms, which allow for multiple, flexible matching strategies in the form of one or more rules. Rule-based matching algorithms are similar to weighted-field matching algorithms in that they can involve multiple field comparisons using a variety of advanced comparison functions. However, they don't compute confidence by summing the results of individual field comparisons. Instead, they apply a set of decision rules (i.e., "IF‹condition›THEN ‹action›" statements). The conditions consist of field comparisons, and the actions consist of "match" or "no-match" conclusions. If a rule's condition is true, then its action is taken.

Below is a basic example rule set. In the conditions, the "‹r›.‹field›" notation represents a field value, where r is either r1 (a subject record) or r2 (a candidate matching record), and field is a name of a field in the record structure (Salkowitz & Clyde, 2003).

1) IF r1.social_security_number = r2.social_security_number THEN match.

2) IF SoundexCompare(r1.last_name, r2.last_name) AND
SoundexCompare(r1.first_name, r2.first_name) AND
EditDistance(r1.birth_place, r2.place)‹2 AND
r1.birth_date = r2.birth_date AND
r1.multiplicity = r2.multiplicity AND
r1.birth_order = r2.birth_order THEN match.

Rule-based matching may be faster than simple information-based matching because it tries the highest-confidence strategies based on the most discriminating rules and broadens the search only when needed. Information-based matching must trundle through a series of comparison matches.

Rule-based matching over weighted-field matching can be faster than simple info-based matching.

For example, a rule-based algorithm would test rule set 1 first. In doing so, it would compare just the social security numbers of the two records. If they are exactly the same, it would declare the records as matching and it would not continue with the other field comparisons. This could result in dramatically speeding up the matching time over weighted-field matching (Salkowitz & Clyde, 2003).

Utah's CHARM project uses a rule-based matcher with five different rules for matching children and adult records. (See Utah case example, page 89.)

Use of data frequencies

Even further to the right in the spectrum are algorithms that use data frequencies to refine the matching process. For example, matching a last name of "Fitzgerald" would carry more weight than matching a last name of "Smith," assuming that "Smith" is a considerably more common last name than "Fitzgerald." Algorithms that take advantage of data frequencies are usually categorized as *probabilistic* and are more common in products based on machine-learning techniques. Weighted-field matching and rule-based algorithms, however, can also take advantage of data frequencies by integrating them into the weights and rule conditions.

Machine learning

On the far right of the Figure 11 spectrum are the algorithms that use machine-learning techniques to initially configure themselves and improve over time. The problem with weighted-field and rule-based matching is that someone has to figure out which fields are most useful in determining matches, how best to compare those fields, and how the result of these comparisons determine (or don't determine) matches. A machine-learning algorithm attempts to solve this problem by allowing the software to customize itself. It does this through a training process in which pairs of records are fed into the system along with their true match/no-match status. For each training pair, the system attempts to compute its own match/no-match result based on its current settings. If it gets the right answer, it reinforces the current settings. If it gets the wrong answer, it tries to figure out what would have helped produce the right answer and alters its settings a little in that direction. By running lots of training data through the system, it can eventually tune its own configuration to correctly compute all answers. At this point, the system should be able to accurately match other pairs of records not in the training data. The challenge with machine-learning algorithms is in creating a training set that represents all the problematic variations in the real data and will enable the algorithm to converge on a stable configuration (Salkowitz & Clyde, 2003).

The more complex the algorithm is, the better the quality of the results and the more expensive.

As part of New York City's integration project for immunization and lead screening data, a vendor developed deduplication software, called Choicemaker™, that used probabilistic matching with machine learning. Youseline Cherfilus, a representative from that project, said: "We can improve the accuracy of matching by 96 percent or better. We are

using the Master Child Index (MCI) to facilitate matching and be extensible to all New York City Department of Health databases."

New York City's MCI integrates data from its lead and immunization programs and provides a centralized deduplication service. MCI stores information from different programs for matching, and it provides core services (i.e., business rules governing the MCI). For more information, see the New York City case example, page 81.

Characteristics of matching algorithms

The final characteristics of matching algorithms shown in Figure 11 are speed, cost, implementation difficulty, and setup time. The spectrum represents some fundamental trade-offs. Typically, simpler algorithms are faster and easier to implement. Complex algorithms yield better quality results, but are more expensive and harder to set up.

Improved matching quality

Finally, one of the most important characteristics of a matching algorithm is the quality in terms of correctly finding duplicate and overlapping records, without incorrectly matching records. As mentioned above, the more sophisticated algorithms tend to produce better results but at a cost.

What: Data fields used in matching

Theoretically, matching algorithms can match records based on any piece of available data. In practice, however, off-the-shelf products and home-built matchers often make assumptions about what information is available and what will be the most discriminating. Rule-based and machine-learning algorithms typically offer more flexibility with respect to the pieces of data that can be used in matching. And the more flexibility there is, the better the chances of success.

Table 9 lists commonly available pieces of information used for matching person records, as well as others that may be discriminating but less common. Any given integration system will probably not have access to all these pieces of information or need them to achieve good results. They are listed here to stimulate ideas. If one piece of information is not available or cannot be shared in a particular integration system, other items can be considered.

In deciding what information to use in matching, the key issues are the quality and reliability of the information and whether it can be used for this purpose without violating confidentiality or security policies. Specific data sets in various states and jurisdictions may provide other opportunities (i.e., for infants, hospital of birth and birth weight in both birth and hospital discharge data; medical record number in hospital discharge data when linking mothers longitudinally—for the same hospital). Matchers must be creative in recognizing potential matching variables.

Deduplication processes need also to consider which data to use as matching variables and which to use to validate the match. One of the advantages of rule-based and machine-learning algorithms is that, if configured efficiently, they may use as little information as possible to determine matches (or non-matches) and only access additional information when needed to resolve ambiguous results.

NAME INFORMATION	Surname (i.e., last name) Given name (i.e., first name) Middle name or initial	Name prefix or suffix Aliases or previous names
PERSONAL INFORMATION	Gender Birth date Birthplace	Multiple birth Birth order Deceased?
PARENT INFORMATION—MOTHER	Surname (i.e., last name) Given name (i.e., first name) Birth (maiden) surname Middle name or initial	Name prefix or suffix Aliases or previous name Date of birth
PARENT INFORMATION—FATHER	Surname (i.e., last name) Given name (i.e., first name) Middle name or initial	Name prefix or suffix Aliases or previous name Date of birth
CONTACT INFORMATION FOR EACH	Phone numbers (current and previous) City and state (current and previous)	Street address (current and previous)
IDENTIFIERS FOR EACH	Social security number (where permitted) Statewide file number Medicaid number Medicare number	Driver license number Facility and facility-specific identifiers, (e.g., medical record numbers, chart numbers, case numbers, etc.)
MEDICAL DATA	Event dates (no details)	Primary care physician

Table 9: *Commonly available pieces of information for matching*

VII. Linking

This section describes three primary techniques for linking records in integrated systems.

Linking (Figure 12) is the process of creating logical connections between independent but matching records. Three fundamental approaches for creating links are:

- Inter-records references
- Linking table
- Integrated-systemwide unique identifiers

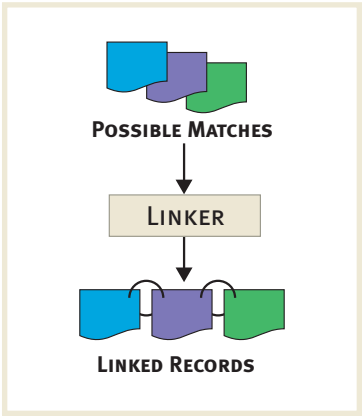
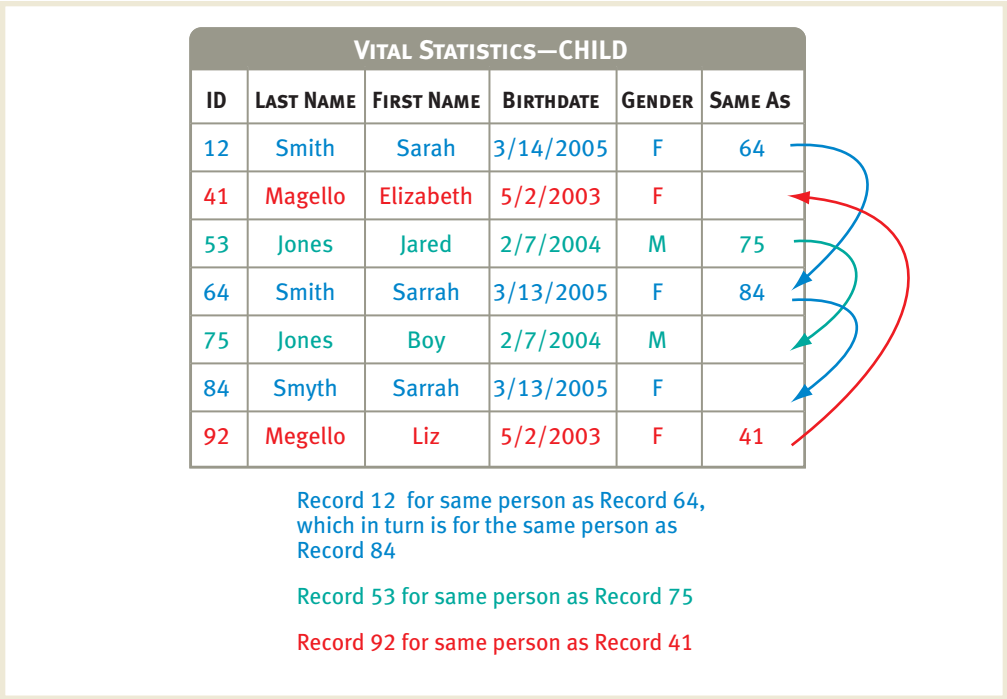


Figure 12: *Linker roles*

Inter-records references

Inter-records references (Figure 13) link matching records using record IDs or keys. Within a single database, this can be relatively simple and effective. Given two matching records, the ID of the second is stored in the first record, typically in an internal linking field. Similarly, given three matching records, the first record references the second, and the second references the third. Adding another record to a set of matching records only requires adding a link from the last record of the existing chain to the newly found matching record.



VITAL STATISTICS—CHILD

ID	LAST NAME	FIRST NAME	BIRTHDATE	GENDER	SAME AS
12	Smith	Sarah	3/14/2005	F	64
41	Magello	Elizabeth	5/2/2003	F	
53	Jones	Jared	2/7/2004	M	75
64	Smith	Sarrah	3/13/2005	F	84
75	Jones	Boy	2/7/2004	M	
84	Smyth	Sarrah	3/13/2005	F	
92	Megello	Liz	5/2/2003	F	41

Figure 13: *Inter-record references in a simple, single child database*

Within a single database, retrieving all available information for a person involves retrieving all the records in the chain, a query type called transitive closure. Some database managers support this type of query directly, but most support it through stored procedures.

Unfortunately, inter-records references are significantly more complex and problematic in integrated systems (Figure 14) and, therefore, are a not good choice. One problem is that the inter-records references must include something that identifies the participating program with the matching record. A second and more serious problem is that each participating program ends up storing IDs from one or more other participating programs. This creates unnecessary dependencies between the programs. For example, if one system changes an ID, it would have to make sure all of the other systems also change the ID in the inter-record references. Finally, retrieving specific information for a person may be very inefficient because it requires accessing each record in the chain one at a time until the desired information is found.

Note: Bi-directional linking techniques can speed up access times but can be more difficult to manage.

Linking table

The linking table approach avoids the dependencies and inefficiencies that plague inter-record references by removing the references from original records and storing them in a central linking table (Figure 15). A central authority would have to host this central linking table and provide services for adding new sets of matching records and retrieving matching records for a given person. An internal ID can be assigned to each new set of matching records and used as a means of clustering all the references to records in the set.

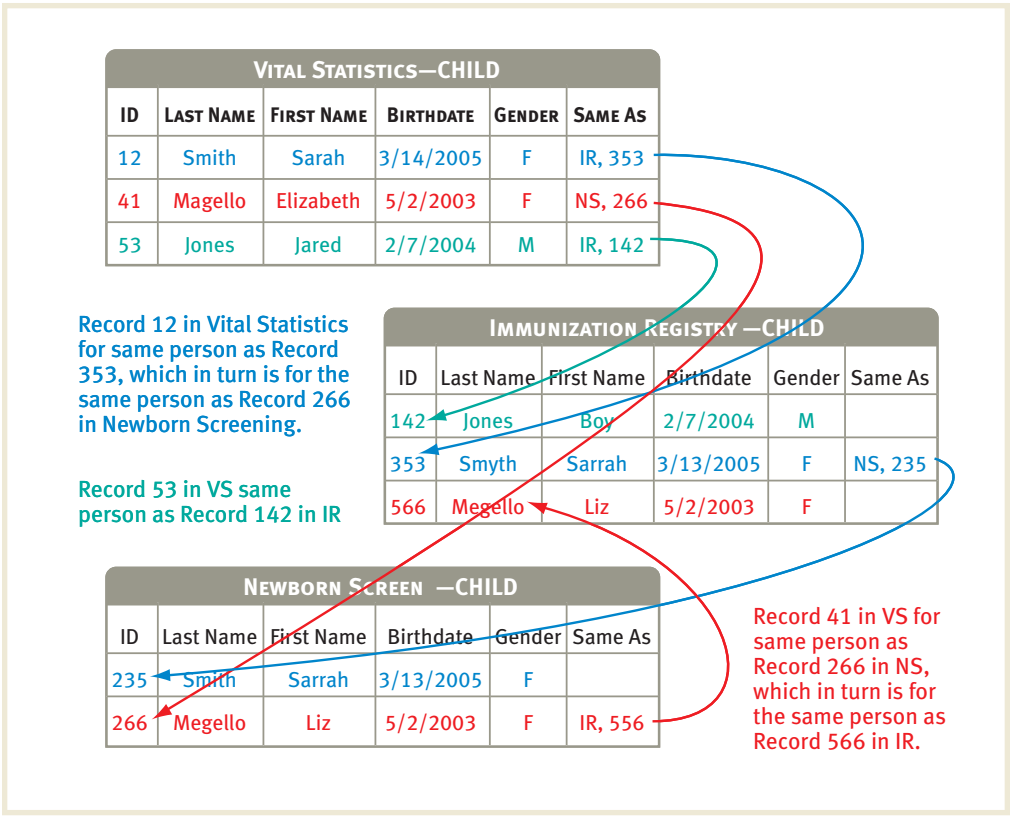


Figure 14: *Inter-record references in a system that integrates vital statistics system, an immunization registry, and a newborn screening system*

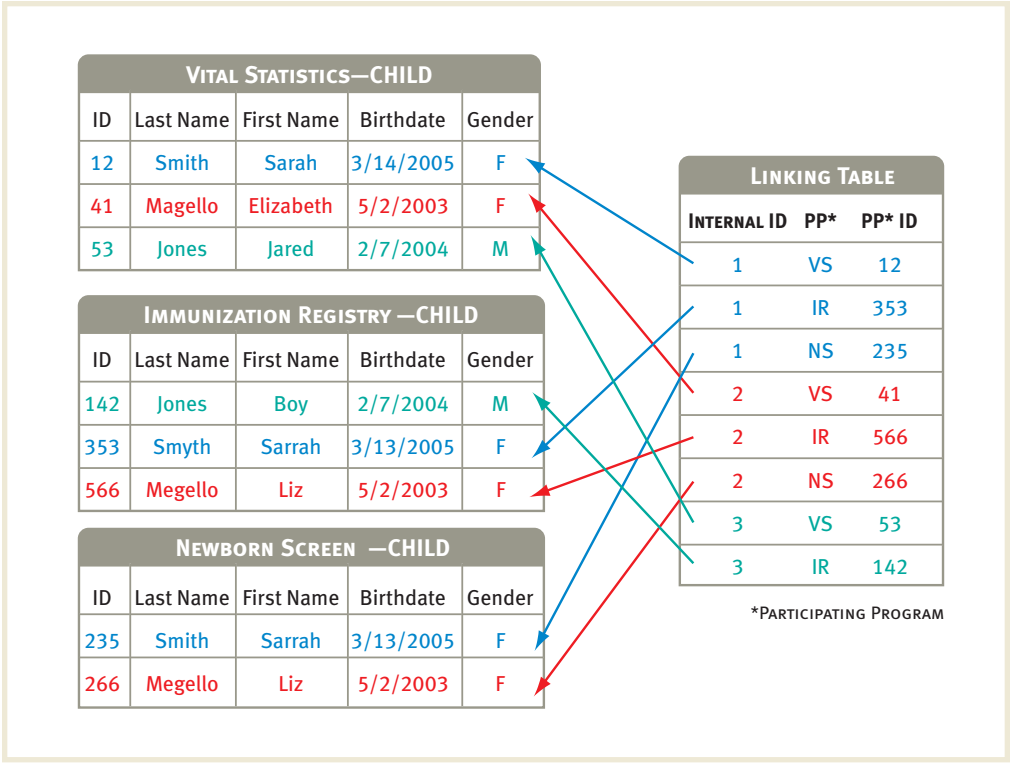


Figure 15: *Linking table in a system that integrates a vital statistics system, an immunization registry, and a newborn screening system*

One advantage of the linking table approach is that the integrated system doesn't have to retrieve all the records for a given person to find a specific piece of data. For example, consider the integrated system represented in Figure 15. If a data user from the newborn screening program wanted to find immunization information for Liz, then the integrated system would use just the linking record (2, IR, 566) to access that information. It would not have to access the vital statistics record for that person.

This approach is well suited for systems based on the central index or the arms-length information broker architectures.

Unique identifiers

A third technique for linking records is to use identifiers that are unique across the integrated system (Figure 16). The integration system assigns a unique ID to each person (i.e., set of matching records or individual, unmatched record) and gives this ID to all the participating programs. The participating programs store this ID as part of their person records.

This approach is actually an inherent part of identifier-based architecture, but can also be used with other architectures that don't use external global identifiers. To link with data across participating programs, the identifiers only have to be unique within the integrated system. They don't have to be external (i.e., known to the users or to the persons whose records are stored in the system).

Vital Statistics—CHILD

ID	Last Name	First Name	Birthdate	Gender	Same As
12	Smith	Sarah	3/14/2005	F	1
41	Magello	Elizabeth	5/2/2003	F	2
53	Jones	Jared	2/7/2004	M	3

Immunization Registry —CHILD

ID	Last Name	First Name	Birthdate	Gender	Same As
142	Jones	Boy	2/7/2004	M	3
353	Smyth	Sarrah	3/13/2005	F	1
566	Megello	Liz	5/2/2003	F	2

Newborn Screen —CHILD

ID	Last Name	First Name	Birthdate	Gender	Same As
235	Smith	Sarrah	3/13/2005	F	1
266	Megello	Liz	5/2/2003	F	2

Figure 16: *Global ID used for record linking in an integrated system of a vital statistics system, an immunization registry, and a newborn screening system*

VIII. Merging

This section explains a variety of merging techniques, prefaced by a discussion of the unique challenges that merging presents and the preparations that lead to effective merging as part of the deduplication process.

Merging (Figure 17) is the process of consolidating two or more records into one record, for any or all of these reasons:

- To create a unified view of a person's data.
- To remove single-source or multi-source duplicates (data cleaning).
- To consolidate overlapping records (a form of data cleaning).

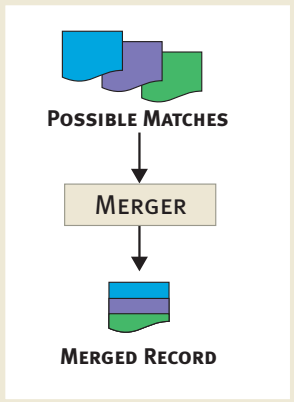


Figure 17: *Merging roles*

Merging can involve single-source duplicates, multi-source duplicates, and overlapping records. The processes and design decisions differ for each case.

For single-source duplicates, the participating program can handle merging internally, without involving the integrated system. The integrated system, however, can assist if the participating program doesn't provide the necessary tools. (For a discussion of merging tools, see *Discriminating characteristics for merging processes*, page 69.) Also, when merging single-source duplicates, the originating database can readily store the resulting records, and depending on the merging process, choose to delete or archive the original records.

For multi-source duplicates, the integrated system needs to play a significant role in the merging process. It may be necessary for users, independent of any of participating programs, to interactively direct the consolidation of data. Also, since the original records come from multi-sources, the integrated system cannot simply replace them with the merged records. Some options include:

- Storing the merged record in an ISD with links back to the original records.
- Sending the merged record back to sources so they can integrate any new or changed data back into the original records.
- Not storing the merged records but simply re-merging on-the-fly as needed. Obviously, this option helps only to create a unified view of a person's data and doesn't address data cleaning at all.

Challenges in merging

Effective merging requires a process owner who is accountable for the quality of the results. Without a clear process owner, merging can lead to inconsistencies that, if unmanaged and unresolved, will decrease the accuracy in the data within the integrated system.

Merging requires awareness of and attention to data quality, data variability, data inconsistency, and the potential to create duplicate records for an individual. Anomalies in data brought about by the "messiness" of real world data collection processes create

NOTE:
The term merging, as used in this document, should not be confused with the process of loading new data (i.e., from a data source external to the integrated system). Data loading, as with any addition of new records from participating programs, includes matching and record coalescing (i.e., linking, merging, or both).

For single-source duplicates, the participating program can handle merging internally, without involving the integrated system.

concern and problems when merging data. Dissimilarities in database schemes create yet another focus of concern. Poor data quality engenders loss of trust among users. Even when merging two identically structured databases with controlled vocabularies, problems have been found that can lead to misinterpretations (Lui and Shiffman, 1998).

A related challenge is controlling the effect of cumulative errors or inconsistencies. Over time, repetitive merging of the same records can lead to the loss of certain data and the whole becomes less than the sum of its parts. The more data sources involved, the worse these effects become and the more difficult it is to anticipate the problems.

Problem	Explanation and Considerations for Merging
Semantic heterogeneity	Seemingly similar fields from different data sources may have slightly different meanings that, if unrecognized, could cause data inconsistencies or loss of credibility. For example, two data sources may both include a field for an address. For one source, these might be intended to hold a physical address that could be used to locate an individual or check residency. The other source might use its address field only for mailings and verifying identity. If values from these fields were merged, the resulting merge record could be slightly incorrect and of less use to both data sources. A merging process needs to be aware of subtle semantic differences between seemingly similar fields.
Field-meaning shift	The meaning of a given field might drift over time, either intentionally or unintentionally. For example, the date field that originally meant the date of a point of service might gradually become the date of data entry for a point of service. A merging process needs to be aware of such shifts when they occur so it can properly interpret the semantics of a field value.
Incompatible code domain	Multi-source duplicates may have code fields that represent the same information but have different domains (i.e., sets of possible codes). For example, consider two data sources of person records that include ethnicity codes. Not only are the codes likely to be different but the two data sources might use a slightly different set and incompatible ethnicity categories. If code domains are truly incompatible, then a merging process may need to preserve the code value from all the records being merged, along with their sources so their meaning can be properly interpreted later on. (See Source Tagging in Merging Techniques, page 66.) Otherwise, if code domains are compatible, but different, a merge process simply needs to select a standard domain and map all others to it.
Impossible values	Over time, as programs and data structures change, some data fields may end up with values that are impossible or illogical (i.e., 99/99/99 for dates, -1 for birth weights, and numbers for names). Impossible and illogical values can find their way into the system through weak user interfaces, changes in database structures, programming errors, and field-meaning shift. A merging process needs to check for and know what to do about illogical or impossible values.
Meaningless values	Sometimes, a program requires information for a field, but the user doesn't know that information, or that information doesn't exist. To work around the requirement, the user enters a bogus or temporary value. For example, if a program requires a first and last name for child and a child doesn't have a first name yet, the user might simply enter "boy" or "girl." Such values are essentially meaningless with respect to the fields they are in. A merging process should try to recognize and properly handle meaningless values. Often, this means giving less precedence to meaningless values, so real values, if they exist, override them.
Extra information	Fields often include extra information that does not belong there, changing the intended meaning of the field. For example, a name first may include a message about the person, such as "John Smith (deceased)." The "(deceased)" entry is extra information. A merging process should try to recognize common or expected forms for extra information and attempt to handle them. For example, "John Smith (deceased)" could result in the "John Smith" name being merged with other name data, and the "(deceased)" being merged with other possible death information.

Table 10: *Common data problems*

Several of the merging techniques discussed below, such as *source tagging* and *stacking*, help reduce problems caused by cumulative errors.

Table 10 lists and explains some of the common data problems that need to be considered when planning or doing record merging. The purpose of this list is to stimulate thoughts about the kinds of inconsistencies and variability that might exist in a data source, and thereby lead to better merging processes. It is by no means an exhaustive list.

Effective merging requires a process owner who is accountable for the quality of the results.

Unmerging

A final challenge for a merging process is to allow for the unmerging of erroneously merged records. If the original records are just archived or deactivated in some way, then unmerging can be relatively straightforward. As long as no changes have been made to the merged record, the system just deletes the merged record and re-activates the old records. If changes have been made to the merged record, the system may have to allow a user to interactively undo the merging so the changes can be made to the original record. For example, if child record A and child record B were merged into a new child record C, and then some additional immunization records were added to C, the unmerging process would have to allow someone to specify whether the subsequent immunization records belong to A or B.

If the merging process does not preserve the original records, there are the following options for unmerging:

- Keep an audit trail of changes, including those caused by the merging process, so the unmerging process can roll back those changes.
- Provide an interactive tool for users to manually direct the unmerging process.
- Keep track of the origin(s) of each piece of data so the original records can be reconstructed. This is essentially the same as keeping the original records, but all of the information is part of the merged record.

Preparing for effective merging

Preparation is the key to effectively overcoming the challenges of merging and for making sure that record merging is correct and effective. Preparation activities can be rather involved, and should include the following:

1. Analyze data models for all participating programs

This involves reviewing (or reverse engineering) design documents that describe the data structures of each participating program. The analysis should look for:

- Data syntax (structure, format, data types, hidden languages, etc.)
- Data semantics (meaning)
- Temporal effects, drift in meaning (content) of data fields
- Why data fields are there in the database.

Data-model analysis can be a time-consuming process and may, in fact, be one of the most limiting factors in setting up an integration system.

2. Analyze data from all sources

Table 10 listed some common data problems. Proper preparation for merging needs to include a thorough analysis of the potential problems with data from all of the participating data sources. An automated script (e.g., a series of queries, computations, or other actions) may be used to check for impossible, illogical, or meaningless values. For example, a script could check person records to make sure that a person's death date, if there is one, is no earlier than that person's birth date. Another script could check to see if there are any names with meaningless values, such as "boy," "girl," or "unknown."

3. Analyze impact of merged data

Before implementing a merging process, it is important to understand the ramifications of both correctly merged and incorrectly merged data. For correctly merged data, the ramifications could involve increased concern about confidentiality and more complex data stewardship relationships. For incorrectly merged data, the ramifications can be much farther reaching, such as affecting the care and services to individuals that need them, and contributing to incorrect denominator and calculation functions in aggregated information.

Merging techniques

This section describes a variety of techniques that a merging process can use to meet the challenge discussed in Challenges in Merging page 63.

Source tagging

As a merger combines data from multiple sources into a single record, it can associate the original source of value with that data. This is called source tagging and can enhance the reliability and accuracy of subsequent matching, merging, and unmerging operations. Figure 18 illustrates how source tagging can be added to an existing database structure without dramatically altering every table in the database. In this simple example, the person and address tables were already in the database and remain unaltered. Only the "source tag" table was added. Each record in this table represents reference to the source of merged information (the merged data is in the original tables). The first three fields identify the record and field containing merged data. The fourth field identifies the source of that data and the fifth field indicates the date and time when the merging took place. Since the existing tables do not depend on the source tag table in any way, its overall impact on existing software is relatively low.

Stacking

Instead of trying to combine different values for a specific field, a merger can keep all of the values. ChoiceMakerTM refers to this as *stacking* (Borthwick, Buechi, & Goldberg), which can dramatically enhance subsequent matching operations. For example, if a merger could keep all distinct addresses and phones (with source tagging) found in a set of matching records, this information could help identify other matches for this person in the future.

Source tagging may be used in combination with stacking if one of the data sources (i.e., WIC or Medicaid) has a rule that the addresses used in its program cannot be changed by another program even if it has more current data.

Person

ID	Last Name	First Name	Birthdate	Birth Location	Gender
12	Smith	Sarah	3/14/2005	Weber County, UT	F
41	Magello	Elizabeth	5/2/2003	Cache County, UT	F
53	Jones	Jared	2/7/2004	Salt Lake County, UT	M

Person

ID	Person ID	Street	City	State
124	12	23 S. Washington Dr.	Odgen	UT
323	41	300 N. 553 E.	Logan	UT
643	41	2100 S. 300 W.	Smithfield	UT
995	53	100 North Main Street	Murray	UT

Existing Data Tables

Person

Table	ID	Field	Source	Timestamp
Person	41	First Name	Vital Records	12/2/2004 12:30 pm
Address	643	Street	WIC	7/15/2005 2:30 pm
Address	643	City	WIC	7/15/2005 2:30 pm

New Data Table for Source Tagging

Figure 18: *A simple example of how an independent table can be used for recording the sources of merged data*

Merge rules

Some mergers use customizable rules to determine how they will automatically combine data from multiple records. In general, these rules specify various kinds of data transformations that map inputs (matching records) to an output (a merged record).

The following list describes four basic types of rules and their capabilities:

- *Precedence rules*
  These rules assign precedence orders to multiple data sources for a specific piece of data. Consider an integrated system involving an immunization registry, vital statistics, and a newborn screening system. One rule could specify vital statistics, immunization registry, and newborn screening as the precedence order for names and birth dates. Then, if records from all three sources for a given child have slightly different names, the merge will keep the name from vital statistics as the most authoritative name. The other names can still be kept (i.e., stacked) but not as the most authoritative name.
- *Data-combination and conversion rules*
  These rules specify how the merge should combine or copy data. For example, in the integrated system discussed above, one rule might say that separate birth day, birth month, and birth year values from the vital statistics record are to be combined to

form the birth date in the merged record. These rules could also specify various data conversions, such as date and case conversions.

- *Redirection rules*
  These rules specify how and when data might get redirected to alternate fields. For example, redirection rules could say that if there is conflict in any of the name fields, then newborn screening and immunization registry names should be redirected to aliases.
- *Information-stacking rules*
  These rules specify when and how to do value stacking.

Merging rules can be customized for the idiosyncrasies of the data found in an integration system's data sources. In particular, the design of merging rules should be rooted in knowledge about field-meaning drift, incompatible codes, common impossible values, and meaningless values.

Relaxing or delaying error correction

It's not always possible for the merger to automatically combine the data from matching records. Sometimes the data is too ambiguous or conflicting, and requires human intervention to sort it out. In such cases, it is helpful if the merger can temporarily relax data constraints and delay the correction of the problem. This would allow the user time to research and correct the problem. Not allowing the user time to resolve problems might force the user to compromise the integrity of the data.

Keeping identifiers

One simple and important merging technique is keeping track of the identifiers of the original matching records. This can help in verifying the correctness of the merging and subsequent unmerging operations.

Interactive tools (examples of application integration)

A final, but very important, technique is to provide the user with interactive tools for verifying automated merging and resolving problems. Typically, interactive merge tools allow a user to review a matching set of records and interactively select which pieces of each record should comprise the merge result. Some of these tools provide sophisticated graphical user-interfaces with many conveniences, such as side-by-side multiple record displays and the dragging and dropping of value fields.

On-demand merging

One important question is whether to merge matching records and store until needed, or to leave them as separate records and merge them only when a user needs a consolidated view of the data. On-demand merging can offer more flexibility in the long-term but does not lend itself to conflict resolution. Generally, on-demand merging means that the application merger has to be able to deal with errors and inconsistencies as they appear (and again later when a user requests the same records.) At a minimum this means that the merger must possess a master vocabulary and rules that tell it what to do when it encounters semantic inconsistencies. Further, it has to be programmed to deal with inconsistencies without necessarily having any prior knowledge of what they might

be. The practical result is that on-demand merging requires large amounts of testing on real data to be sure most of the likely inconsistencies have been encountered and dealt with appropriately. The on-demand merging can be enhanced if the data in the disparate databases is required to go through a prescribed formatting process first. Such normalization can reduce processing errors and speed up merging computations.

Discriminating characteristics for merging processes

Merging processes vary significantly in sophistication and effectiveness. The following characteristics can help with the comparison of different processes and with making an informed decision about which ones might satisfy the needs of a specific integrated system:

- The degree to which the merger can be customized or tuned.
- The degree to which the merging process can relax data constraints and delay the resolution of a data problem.
- The degree to which the merger supports unmerging.
- The degree to which the merger supports periodic verification and evaluation.
- The degree to which the merger supports analyzing of data accuracy or relevancies.
- The degree to which the merger can enhance future matching by adding value to the merged data (e.g., data sources and estimated accuracy or confidence).
- The amount and quality of interactive tool support.

IX. Summary

The bottom line is that effective deduplication is critical to the success of an integrated health information system, but it is inherently complex. Solutions to deduplication involve matching, linking, or merging, and must consider how these activities work, when they take place, and which organizations are responsible for them.

These decisions are closely intertwined with the architecture of an integration system, and the nearly endless possibilities are difficult to sort out. Adopting or developing a solution requires awareness of the underlying issues, the key design choices, and the consequences of those choices as they relate to a specific integrated system. This conceptual framework provides a starting point for achieving this awareness, and a foundation for creating and sustaining effective deduplication strategies.

References

AHIMA MPI Task Force. (2004). Building an Enterprise Master Person Index. (AHIMA Practice Brief). *Journal of AHIMA 75 (1), 56A-D.*

Arzt, N. H., & HLN Consulting LLC. (2005). Frameworks and models for integration, *Response to request for information, development and adoption of a National Health Infomation Network, Department of Health and Human Services, Office of the National Coordinator for Health Information Technology, July 18, 2005*.

Borthwick, A., Buechi, M., & Goldberg, A. Key concepts in the ChoiceMaker 2 Record Matching System. Retrieved March 12, 2006, from **www.choicemaker.com/key_concepts.pdf.**

Connecting for Health. (2004). Achieving electronic connectivity in healthcare: A preliminary roadmap from the Nation's public and private-sector healthcare leaders. Retrieved March 12, 2006, from **http://www.connectingforhealth.org/resources/cfh_aech_roadmap_072004.pdf.**

Connecting for Health Working Group on Accurately Linking Information for Health Care Quality and Safety. (2005). Linking health care information: Proposed methods for improving care and protecting privacy. Retrieved March 12, 2006, from **http://www.connectingforhealth.org/assets/reports/linking_report_2_2005.pdf.**

Denning, D. E., & Denning, P. J. (1998). *Internet besieged: Countering cyberspace scofflaws*. New York: ACM Press.

Lui, J. C., & Shiffman, R. N. (1998). *Critical tedium: Data quality issues in merging two structured immunization databases*. Paper presented at the AMIA Annual Symposium, Miami Beach, FL.

Massachusetts Health Data Consortium Inc. (2004). Analysis of approaches to uniquely identifying patients in Massachusetts. Retrieved March 12, 2006, from **http://www.mahealthdata.org/ma-share/projects/communitympi/20040416_UPIpaper.pdf.**

McDonald, C. J., Overhage, J. M., Barnes, M., Schadow, G., Blevins, L., Dexter, P. R., et al. (2005). The Indiana Network For Patient Care: A working local health information infrastructure. *Health Affairs, 24* (5).

Salkowitz, S. M., & Clyde, S. W. (2003). De-duplication technology and practices for integrated child-health information systems. Retrieved March 12, 2006, from **http://www.phii.org/Files/dedupe.pdf.**

Stallings, W. (1999). *Cryptography and network security: Principles and practices*. Upper Saddle River, NJ: Prentice Hall.

Tanenbaum, A. S., & van Steen, M. (2002). *Distributed systems: Principles and paradigms*. Upper Saddle River, NJ: Prentice Hall.

U.S. Department of Education. (2005). Policy: Family Educational Rights and Privacy Act (FERPA). Retrieved March 12, 2006, from **http://www.ed.gov/policy/gen/guid/fpco/ferpa/index.html.**

U.S. Department of Health and Human Services, & Centers for Medicaid and Medicare Services. (2005). HIPAA General Information. Retrieved March 12, 2006, from **http://www.cms.hhs.gov/HIPAAGenInfo/.**

Case Examples

3

Case Examples

3

Introduction

The Unique Records Portfolio was developed by members of *Connections*, a community of practice of state and local public health practitioners developing or managing integrated child health information systems. The case examples in this chapter present a snapshot of the data integration approaches and deduplication practices of four *Connections* members' projects in Missouri, New York City, Rhode Island, and Utah.

Together, the cases provide a range of examples of integration system architecture and deduplication processes. They illustrate concepts presented in the *Portfolio's* Conceptual Framework, Metrics and Evaluation chapter, and Profile Questionnaire, allowing readers to understand more concretely what is described in the text.

Note: The page number references throughout this chapter refer to related text in the *Portfolio*.

What to Look for in the Case Examples

The examples provide a window to various policies and practices, including:

- The analysis and evaluation of prospective deduplication software.
- The steps for establishing and monitoring the processes.
- Some of the metrics for testing the efficacy and efficiency of deduplication.
- The decision process for setting quality parameters.
- Projects in New York City and Utah that focus on the interrelationship of the architecture and the deduplication processes.
- Projects in Rhode Island and Missouri (which have central databases) that provide greater detail on how the deduplication processes function.
- Variation in project size and complexity, with a range of birth cohorts, ethnic diversity, population density, urban/rural settings, and health department size.
- A range of program integration, from three to eleven. Some projects use data sources external to the public health programs themselves.
- The immunization program as the only common program being integrated, with vital records and newborn hearing screening implemented in three programs, and newborn dried blood screening in two programs and scheduled for a third.

Because of the diverse nature of the projects, parallels may not always be drawn across the projects. The table on the following page compares and contrasts projects.

Missouri	New York City	Rhode Island	Utah	Case example
• Software selection and evaluation • Software description • Selection of algorithms • Description of back-end matching process • Testing and evaluation • Manual review decisions • Metrics • Accuracy tolerance decisions	• Architecture selection (service-based) • Software description • Algorithm design • Detailed description of algorithm matching process and clue building • Testing and evaluation • Metrics • Probability threshold decisions	• Software selection and evaluation • Software description • Detailed description of probabilistic matching • Configuration and testing • Metrics—very detailed	• Architecture and strategies (service-based) • Software description • Detailed description of processes • Metrics	Case example topics
Missouri Health Strategic Architectures and Information Cooperative (MOHSAIC)	Master Child Index (MCI)	KIDSNET	Child Health Advanced Record Management (CHARM)	Project title
1993	2004	Mid 1990's	2000	Year started
Missouri	New York City	Rhode Island	State of Utah	Service area
Approximately 77,000	Approximately 125,000	Approximately 14,000	Approximately 40,000	Annual birth cohort
a. Vital Records b. Newborn Dried Blood Spot Screening c. Newborn Hearing Screening d. Immunization e. Childhood Lead Screening	a. Immunization b. Childhood Lead Screening	a. Vital Records b. Newborn Dried Blood Spot Screening c. Newborn Hearing Screening d. Immunization e. Childhood Lead Screening f. Newborn Developmental Risk g. WIC h. Early Intervention i. Family Outreach Program j. Birth Defects	a. Vital Records b. Newborn Dried Blood Spot Screening c. Newborn Hearing Screening d. Immunization e. Early Intervention	Participating programs
Central database (See pg.40.)	Central Index—Locator (See pg. 30.)	Central Database (See pg. 40.)	Arms-length Information Broker (See pg. 36.)	Integration architecture
Back-end (See pg. 51.)	Back-end upon record entry (See pg. 51.)	Back-end (See pg. 51.)	Supports both front-end and back-end (See pg. 51.)	Time of deduplication
Probabilistic and some deterministic	Probabilistic	Probabilistic	Rule-based	Matching algorithm
Interactive desktop application-Q/A developed in-house and being imbedded in a number of areas	Interactive desktop application; merges are communicated from central index back to program databases, which implement their own rules	Front-end desktop human review GUI to manage queue fed by backend probabilistic process; back-end merging for confirmed duplicates, combination of stacking and precedence rules	Rule-based, automated and interactive merging, with the ability to delay resolution of problems	Merging/data cleaning
SQL procedures	N/A	Back-end merging for matched records, combination of stacking and precedence rules	Rule-based, on-demand merging	Merging/data integration
Missouri Department of Health and Senior Services, Division of Community and Public Health	Citywide Immunization Registry (CIR) and Lead Poisoning Prevention Program (LPPP), New York City Department of Health and Mental Hygiene (NYC DOHMH)	Rhode Island Department of Health, Division of Family Health	Utah Department of Health	Organization responsible

Missouri Department of Health and Senior Services

Project Title	Missouri Health Strategic Architectures and Information Cooperative (MOHSAIC)
Focus of Example	• Software selection and evaluation • Software description • Selection of algorithms • Description of back-end matching process • Testing and evaluation • Manual review decisions • Metrics • Accuracy tolerance decisions
Service Coverage Area	Statewide
Participating Programs	Newborn hearing, dried blood spot screening, immunization, lead screening, vital records
Annual Birth Cohort	Approximately 77,000
Integration Architecture	Central database (See page 40.)
Time of Deduplication	Back-end (See page 51.)
Matching Algorithm	Probabilistic and some deterministic (See page 57.)
Merging/Data Cleaning	Interactive desktop application—Q/A developed in-house and being imbedded in a number of areas
Merging/Data Integration	SQL procedures
Organization Responsible	Missouri Department of Health and Senior Services, Division of Community and Public Health

Project scope and goals

To develop an integrated child health information system, including immunizations, newborn hearing, dried blood spot screening, and lead screening. Lead data are currently being added.

History

The Missouri Health Strategic Architectures and Information Cooperative (MOHSAIC) project was initiated in 1993 based on a departmental information systems strategic plan. Its first components were registration, appointment scheduling, immunization/TB skin testing data, and vaccine inventory. The project created a statewide secure "wide area network" that connected to a central database. It has been expanded over the years to include service coordination, regulated clients, surveillance (Missouri's equivalent of the NEDSS-Base System) Family Care Safety Register, TB cases, newborn screenings, laboratory results, and a data warehouse.

Architecture: Central database

Client demographic information is captured once, in an Oracle database, and shared by all users. As additional components are added, the data are included as part of the

client's record. The system interfaces electronically with Medicaid, Vital Records, and the state public health laboratory, and processes data with an external source file of billing data, *the Gateway EDI Clearing House*. Users access the system through a secure network or through Web applications. There is no central index because there is a central, shared database of people that functions as an index.

Deduplication software

- Limited front-end based on user search and selection.
- Back-end deduplication.
- Primarily deterministic with some variants.
- Self-developed SAS programs.
- Human review for close matches and for merging using a desktop-based GUI.

Deduplication software acquisition and evaluation

In FY '98 and FY '99, Missouri Department of Health and Senior Services (MODHSS) received a CDC grant for a targeted research project (Grant #U1W/CCU714725). One component of this project was to review matching software available to assist in including electronically reported vaccine doses in the central register. MODHSS staff reviewed several available applications. Based on this review, staff purchased and tested the ***AUTOMATCH*** probabilistic software application. MODHSS staff completed analysis of the results using this automated probabilistic matching software to determine which data elements most reliably indicated an accurate match. Frequent last name changes and name misspellings were—and continue to be—among the most common difficulties faced in accurately matching clients.

In 2002, MODHSS implemented a powerful data matching/deduplication software package known as DataFlux. Due to loss of funding for the DataFlux licensing fees, MODHSS is creating and tweaking a variety of algorithms to replace this powerful software. Depending on the source of data being added to MOHSAIC, different sets of algorithms are applied to the data.

Selection of algorithms

For example, MODHSS originally used vital records data to initiate the record in MOHSAIC. Recently, newborn screening data was added. This changed the sequence for receipt of electronic data files to initiate the MOHSAIC record. Newborn screening data is now available for inclusion in MOHSAIC before vital records are processed. Development is underway to create a new Web-based vital records system that will again impact the data files sequencing.

The newborn screening data included in the Neometrics system often has incomplete names. The Neometrics data system was developed under contract to Neometrics, a division of Natus, Inc., and is used by the State Public Health Laboratory. MODHSS uses other data elements such as mother's name, birth date, hospital, and child and mother's chart number.

The following back-end matching processes are used for these datasets. Please refer to the Conceptual Framework, weighted field comparisons (page 56) and Rule based matching (page 56) for further description.

1. **Mother's First Name, Mother's Last Name, Child's DOB, Child's Gender, Birth Order:** This key arrangement is similar to # 4 below (our traditional key). Here, combining mother's names with birth order serves as a good stand for the child's name, which is less prevalent in the Neometrics data.

   Hits for this key: 13,148

   Percent of Neometrics data matched at this point: 84%

2. **Birthing Facility (Hospital) ID, Child's Medical Record Number, Child Date of Birth:** This was moderately aggressive in current data but has great potential when Facility ID and Child Medical Record number are prevalent and recorded accurately. Enjoining the Child's Date of Birth circumvents concerns about the birthing facility reissuing the medical record number.

   Hits for this key: 3,796

   New hits for this key: 561

   Percent of Neometrics data matched at this point: 88%

3. **Birthing Facility (Hospital) ID, Mother's Medical Record Number, Child Date of Birth, Birth Order:** Comparable to above—substituting Mother's Medical Record Number and Order Born for Child Medical Record Number.

   Hits for this key: 4,137

   New hits for this key: 191

   Percent of Neometrics data matched at this point: 89%

4. **Child's First Name, Child's Last Name, Child's DOB, Child's Gender:** This is the traditional key used in our HMO process. It has about a 0033 frequency of duplication in birth records (i.e., low risk of false positives) and typically yields an aggressive match rate of 80 – 90 percent (dependent on data quality).

   Hits for this key: 3,975

   New hits for this key: 340

   Percent of Neometrics data matched at this point: 91%

5. **Child's Date of Birth, Birth Order, Mother's SSN:** When Mother's SSN is prevalent, this is a very aggressive key.

   Hits for this key: 10,636

   New hits for this key: 692

   Percent of Neometrics data matched at this point: 96%

6. **Address (scrubbed to significant numbers and name), Child's Date of Birth, Child's Time of Birth, and Birth Order:** Slight to moderate strength. Has possibility of small number of false positives. None have been manifest in trial data.

   New hits for this key: 162

   Percent of Neometrics data matched at this point: 97%

When processing data from another external source, such as the Gateway EDI billing clearinghouse, staff must depend on basic demographic data of first name, last name, and date of birth. Data coming into MOHSAIC from the Medicaid billing system includes the Medicaid number of the child, making it easier to guarantee an accurate match. Even in these cases when the Medicaid number matches, staff checks the child's first name and date of birth to ensure there has not been a keying error, such as entering a child's sibling's number by mistake or a simple typographical error.

MODHSS has made every attempt to minimize the need for manual review. The rigor required in the matching process is dependent on the purpose of matching. For example, if an outside agency gives us only basic demographic data for their clients and wishes to extract shot histories for those clients, the amount of manual review required varies with the quality of data elements we are given for comparison. If time is of the essence, staff may only declare a match if first name, last name, and date of birth are in agreement. If time allows, staff may match children with same first name and date of birth, and then manually review their last names. For example, staff may find a John P Hall and a John Paul Johnson-Hall born on the same date and living in the same county or having the same phone number.

Metrics

MOHSAIC is a central database that includes demographics for a client one time. It is a population-based system that includes all children born in Missouri since January 1, 1994, until the current date. Each service component added to the system uses the same demographics.

In the past, electronic submission of immunization data from outside systems has been limited. The primary means of populating the database was through manual data entry. An increasing number of managed care plans and a medical billing clearinghouse are submitting electronic data for inclusion in MOHSAIC on a regular schedule. The addition of electronic results for newborn dried blood spot, newborn hearing, and lead screenings has heightened MODHSS's awareness of the need to establish data-quality processes to monitor the database and detect and resolve duplicate records at the demographic and service levels.

Currently, the staff receives a monthly report of the number of clients 12 months of age or younger, the number of clients 24-35 months of age, the number of vaccines associated with each age group, the total number of clients registered in MOHSAIC, and the total number of vaccines for all ages. These monthly numbers allow staff to monitor the growth of the database. An increase in the number of clients 12 months of age and under was noted in August 2004 at the same time child blood lead screening results were con-

verted from the STELLAR database and merged into MOHSAIC. The information technology staff was informed and discovered a problem in the conversion program that was generating duplicate clients.

Determining the amount of duplication in Missouri data is difficult. The two largest population centers in Missouri lie directly on state borders. An unknown but significant number of children are born in Missouri but move to bordering states soon after birth or within the first year of life. MODHSS has found no mechanism for estimating how many children born in Missouri soon move or how many children born in other states immigrate to Missouri. Missouri's annual birth cohort is typically about 77,000. In reviewing the MOHSAIC data, staff may find 88,000 children born in a given year. Deduplication processes typically reduce this gap by several thousand children.

For each child, immunizations are carefully checked to ensure that they occurred after date of birth and that no shots from the same vaccine family have a date of service in close proximity to each other. Through continuous use and review of the data, appropriate metrics will be identified and implemented.

New York City Department of Health and Mental Hygiene

Project Title	Master Child Index (MCI)
Focus of Example	• Architecture selection (service-based) • Software description • Algorithm design • Detailed description of AI matching process and clue building • Testing techniques and evaluation • Metrics • Probability threshold decisions
Service Coverage Area	New York City
Participating Programs	Immunization, lead screening
Annual Birth Cohort	Approximately 125,000
Integration Architecture	Central Index—Locator (See page 30.)
Time of Deduplication	Back-end upon record entry (See page 51.)
Matching Algorithm	Probabilistic (See page 57.)
Merging/Data Cleaning	Desktop interactive application; merges are communicated from central index back to program databases, which implement their own rules.
Merging/Data Integration	N/A
Organization Responsible	Citywide Immunization Registry (CIR) and Lead Poisoning Prevention Program (LPPP), New York City Department of Health and Mental Hygiene (NYC DOHMH)

Project scope and goals

To facilitate matching and consolidating patient records across their systems, the Citywide Immunization Registry (CIR) and the Lead Poisoning Prevention Program (LPPP) launched the Master Child Index (MCI) system for the Department of Health and Mental Hygiene (NYC DOHMH) in January 2004. The MCI, which includes all of the children in both systems, allows LPPP and the CIR to share information using an assigned unique identifying number for each child.

Architecture: Central index, using a locator communication protocol

The architecture incorporates a central index driven by "locator" software.

1. MCI is the central index.
2. CORBA is the communications (locator) mechanism.

However, participating systems have been modified to invoke a set of core services in a service-based architecture to interact with the MCI. The service-based architecture, which uses CORBA, is the important part. Message types are proprietary.

Deduplication software

- Back-end deduplication upon record entry.
- Probabilistic with machine learning.
- Customized vendor product.
- Human review for close matches and for merging using a desktop-based GUI.

The MCI uses a specially designed matching system to match and merge records entering the database. This system is designed to replicate the human decision-making process and is composed of matching software and a set of highly-specific business rules.

Matching algorithms, type, specific implementation and performance

The MCI employs an artificial intelligence matching program built around a system of clues designed to replicate the human decision-making process.

The clues are attributes of each record, which argue for or against a "match" decision; they are assigned weights and are combined into an overall probability, indicating the certainty with which two records belong to the same child.

Clue building process

- Analyze data characteristics: To generate an appropriate set of clues, the characteristics of the data in the MCI participating databases—currently the Citywide Immunization Registry (CIR) and the LeadQuest (LQ) lead registry—were studied in depth.
- Identify data essential for human review: Input from human review staff was gathered to identify data essential when people make decisions about whether two records belong to the same child.
- Assign clue weights: The system assigns weights to the clues based on the human decisions, corresponding to their relative importance in the decision-making process. Clues that were strongly associated with DOHMH decisions received a high weight, whereas clues that were only weakly linked received a low weight. These clue weights were combined into an overall probability ranking. A higher ranking indicates a greater likelihood that the two records belong to the same child.

How clues are used

The MCI matching system contains 193 clues that use all available fields and combination of fields in a record. Examples of clues related to the first name field only are described below:

1. Do the first names match approximately using the well-known Soundex approximate name match algorithm[1], or using the less well-known NYSIIS algorithm[2] , or using the U.S. Census Bureau's Jaro-Winkler algorithm[3]?

[1] Soundex is an algorithm to code surnames phonetically by reducing them to the first letter and up to three digits, where each digit is one of six consonant sounds. This reduces matching problems from different spellings.
[2] The New York State Identification and Intelligence Algorithm (NYSIIS). In 1970 the NYSIIS project headed by Robert L. Taft published the paper "Name Search Techniques." In this paper, Taft compared Soundex with a new phonetic routine (NYSIIS) that was designed through rigorous empirical analysis. The NIST Dictionary of Algorithms and Data Structures defines it as an algorithm to convert a name to a phonetic coding of up to six characters.

2. Is this a common first name? A match on first name "JENNIFER" will be assigned a lower weight than a match on first name "VASSILIKI" because of the relative frequency of occurrence of these names in New York City's population.
3. Do the first and last names match when swapped in one of the two records? (Frequently the first and last names are reversed in reports submitted).
4. Are first names the same but genders different? If the name is gender-specific (e.g., "JENNIFER"), then the gender difference will be discounted.

Probability threshold decisions

DOHMH decided on 0.96 as the "match" probability threshold, and 0.70 as the "no match" probability threshold. Thus, an incoming record matching with an existing record with a probability of 0.96 or higher would then be judged a "match" and would merge to the existing record. If the incoming record matched with an existing record with a probability below 0.70, it would then be considered a "no-match" and created in the MCI as a new record.

Finally, an incoming record that matches an existing record with a probability within the 0.70 - 0.96 range would then be considered a "potential match," created as a new record, and sent to human review. It is worth pointing out that although a record is created, the system knows this record has not been fully assessed through human review and is not quite the same as a "permanent" record. While the decision to set the lower threshold at 0.70 was expected to result in a significant number of potential duplicate records left in the system, it was believed to reduce the number of records in need of human review to a manageable size. The very high threshold of 0.96 was chosen to minimize the risk of records of different children being merged.

All CIR and LQ records are evaluated by the matching system as they enter the MCI by being assigned a probability that will determine whether they will be merged, created, or sent to human review.

In some circumstances, however, the probabilistic decision is overridden by a small set of rules put in place to safeguard against false merging and forcing pairs into human review. For example, if a record in the database has more than five names, this record is sent to human review even if it matches an incoming record above the 0.96 threshold probability.

Testing techniques, evaluation procedures

DOHMH periodically evaluates the matching system's accuracy by comparing a sampling of decisions made by DOHMH staff and decisions made by the system. DOHMH also carefully weighs the tradeoff between accuracy and the number of records requiring human review. As a result, clues can be modified, added, or eliminated, and assignment

[3] Jaro-Winkler. A measure of similarity between two strings. The Jaro measure is the weighted sum of percentage of matched characters from each file and transposed characters. Winkler increased this measure for matching initial characters, and then rescaled it by a piecewise function, whose intervals and weights depend on the type of string (first name, last name, street, etc.). The Winkler comparator takes into account the fact that typographical errors occur more often toward the end of words, and thus gives an increase value to characters in agreement at the beginning of the string

of weights to the clues adjusted until an acceptable compromise between accuracy and human review is reached.

Metrics

During the testing phase, DOHMH staff settled on 98.96 percent accuracy and 3.79 percent human review. This was expected to result in 1.04 percent false negatives (duplicates missed) and 0.0 percent false positives (false merges) and an acceptable (low) volume of potential matches for human review. The accuracy of the MCI matching system was shown to be as high in operation, after implementation, as it was during the testing phase.

The **specificity** of the system is examined in production by selecting a sample of merged records and examining whether there are indications of incorrect merges. DOHMH data quality staff is engaged in the process of finding false merges in the MCI by identifying characteristics of records that indicate false merging and then further examining those records.

One example entails records that contain immunizations preceding one of the dates of birth (DOB) of the child, an indication that a potentially false merge has occurred. (A child can have more than one DOB when records with different DOBs are merged).

The system's **sensitivity** in production is also evaluated by assessing the percentage of known duplicates identified and merged. These analyses are based on sampling and the known size of NYC's annual birth cohort (approximately 125,000).

Outside of the MCI's matching engine, the MCI core services include a set of business rules (Matrix Rules) implemented to oversee the decisions of the matching program and safeguard against unusual merging activity. For example, the matrix rules currently prevent multi-way merging to avoid a falsely merged record to further deteriorate.

Rhode Island Department of Health

Project Title	KIDSNET
Focus of Example	• Software evaluation and selection • Software description • Detailed description of probabilistic matching • Configuration and testing • Metrics, very detailed
Service Coverage Area	Rhode Island
Participating Programs	11 programs (listed below)
Annual Birth Cohort	Approximately 14,000
Integration Architecture	Central database (See page 40.)
Time of Deduplication	Back-end (See page 51.)
Matching Algorithm	Probabilistic (See page 57.)
Merging/Data Cleaning	Front-end desktop human review GUI to manage queue fed by back-end probabilistic process; back-end merging for confirmed duplicates, combination of stacking and precedence rules
Merging/Data Integration	Back-end merging for matched records, combination of stacking and precedence rules
Organization Responsible	Rhode Island Department of Health, Division of Family Health

Project scope and goals

KIDSNET facilitates the collection and appropriate sharing of health data with health-care providers, parents, MCH programs, and other child service providers for the provision of timely and appropriate preventive health services and follow-up.

KIDSNET includes children born in Rhode Island, residing in the state, or being served by any health-care provider in the state. KIDSNET's 11 affiliated programs include: Newborn Developmental Risk, Newborn Blood Spot Screening, Newborn Hearing Assessment, Immunization, Childhood Lead Poisoning, Vital Records, WIC, Early Intervention, Family Outreach Program, and Birth Defects.

Deduplication software

Developed in the mid-1990's, KIDSNET originally employed a simple deterministic algorithm for matching incoming data to existing KIDSNET demographic records. By 2004, KIDSNET had accumulated a queue of more than 47,000 unmatched records. With its limited budget, the state embarked on a project to improve the matching process and to ultimately reduce the number of unmatched records.

In 2004, KIDSNET's unmatched record queue was analyzed, and surveys of matching methods and software options were conducted. Requirements were documented, and a

probabilistic matching, adding, and deduplication architecture for KIDSNET was designed. Freely Extensible Biomedical Record Linkage (FEBRL)[4], an open source package, was modified for use within the new framework. FEBRL makes use of a number of other algorithms which are referenced in the processes described below and more fully explained in footnotes. Probabilistic parameters were developed, and an extensive six-month testing process ensued. The process was placed into production in May 2004.

KIDSNET's new FEBRL-based process provides data standardization and Fellegi-Sunter[5], (a formal model for matching that uses the relative frequency of strings being compared) probabilistic matching of name and address data for the following electronic feeds:

- Immunizations
- Lead screening
- WIC (Women, Infants, and Children nutrition program)
- EI (Early Intervention Program)

In addition to the matching of incoming demographic data against existing records, KIDSNET also utilizes FEBRL to perform probabilistic deduplication of the existing records. FEBRL deduplication runs weekly, identifying potential duplicates in the database and feeding them to a new, custom-developed interactive merge tool for use by KIDSNET data managers. The tool allows data managers to analyze potential duplicates and resolve them quickly.

Probabilistic matching configuration and testing: Process and metrics

The process in Rhode Island to perform the initial KIDSNET matching configuration and testing, to perform ongoing periodic testing, and to perform periodic maintenance involves the following steps:

Initial configuration

Define initial configuration (calculate "u" and "m" probabilities; define basic blocking; calculate frequencies; define fields and string comparators).

Testing the configuration without blocking

1. Define a subset of data to perform deduplication or matching.
2. Remove all blocking from match configuration.
3. Set a very low output threshold from the match configuration to examine as many possible pairs as practical.
4. Perform a match run against the subset.
5. Manually examine every pair from the match output.
   - Specificity: Require 100 percent. No non-matching pairs should be found above the human review range (without deterministic "clues").

[4] Febrl is an open source software that does data standardization (segmentation and cleaning) and probabilistic record linkage ("fuzzy" matching) of one or more files or data sources that do not share a unique record key or identifier. It makes use of several other algorithms developed by other informaticists including: Fellegi-Sunter, The New York State Identification and Intelligence Algorithm and Jaro-Winkler.
[5] Fellegi-Sunter is a formal model for matching that uses the relative frequency of strings being compared.

- Sensitivity: Require 90 percent or higher. Ninety percent of all matching pairs should be above the lower human-review threshold.

The easiest way to meet these metrics is to increase the size of the human review range. Deterministic clues can also be used to help achieve the desired metrics. Other configuration adjustments, such as different string comparators, can be applied. The challenge of match configuration is to meet the metrics with the *smallest possible* human review range and the least number of deterministic "clues."

Testing the blocking configuration

1. Reinstate blocking.
2. Perform match run against the subset.
3. Compare results from previous run without blocking (see "Testing the configuration").
4. Note that missing matched pairs indicate a "hole" in the blocking strategy.
5. Adjust blocking as necessary.
6. Repeat.

Recommended annual configuration maintenance

1. Test with and without blocking.
2. Frequencies should be recalculated.
3. "u" (unmatched) and "m" (matched) probabilities should be recalculated.
4. Data standardization should be reviewed.

Probabilistic matching checklist

- Are address, name, and date data standardized?
- Is given name and surname frequency data taken into account by the matching algorithm? For example, in Rhode Island, some common names include Brown, Garcia, Johnson, and Rodriguez. These common names are 200-300 times more likely to appear in a database than uncommon names, such as Aboofazeli, Chabran, Czarnecki, or Guarnieri. Therefore, the matching algorithm should weigh matches on common names less than matches on uncommon names.
- Is blocking configured? Are multiple overlapping blocks being used? For example, Rhode Island combined the following four blocks:

  i. First letter of last name + First letter of first name + Gender

  ii. Gender + DOB

  iii. NYSIIS[6] of last name + Gender + DOB Month and Year

  iv. NYSIIS of last name + NYSIIS of first name + DOB Day and Month

[6] See Footnote [2] on page 82

- This blocking configuration provided Rhode Island with optimal speed and sensitivity.
- If using Fellegi-Sunter matching, are "m" and "u" probabilities being calculated for the datasets being matched? Are approximate string comparison algorithms being used?
  A typical configuration for names might be:
  Given name: Winkler String Comparator[7] component of Febrl: m = 0.995; u = 0.008
  Surname: Winkler String Comparator: m = 0.990; u = 0.043
- Probabilities should be *calculated*, not "borrowed" from other configurations.
- Deterministic "clues" may be used to force a possible match to human review. For example, a high-probability match with a gender difference; a high-probability match of one record against multiple records; or a match with indicators of multiple birth.
- Are probabilities assigned and records divided into three groups: not a match, match, and human review? Are the thresholds of these groups tested frequently?

[7] See Footnote [3] on page 83

Utah State Health Department

Project Title	Child Health Advanced Record Management (CHARM)
Focus of Example	• Architecture and strategies (service-based) • Software description • Detailed description of process • Metrics
Service Coverage Area	State of Utah
Participating Programs	Vital Records, Immunization, Newborn Hearing Screening, and later in 2005, Early Intervention, and Newborn Screening
Annual Birth Cohort	Approximately 40,000
Integration Architecture	Arms-length Information Broker (See page 36.)
Time of Deduplication	Supports front-end and back-end (See page 51.)
Matching Algorithm	Rule-based (See page 56.)
Merging/Data Cleaning	Rule-based, automated and interactive merging, with the ability to delay resolution of problems
Merging/Data Integration	Rule-based, on-demand merging
Organization Responsible	Utah Department of Health

Project scope and goals

Child Health Advanced Record Management (CHARM) is an integration project that has already integrated three child-health information systems in the state of Utah, and will soon integrate two more. The first three participating systems are Utah's immunization registry (USIIS), its statewide newborn hearing detection and intervention tracking system (HiTrack), and its vital statistics system (VS). The next two participating programs are a new early intervention tracking system (BTOTS) and newborn screening system.

Table 1 (page 90) lists some of CHARM's goals organized by five key issues: data access, integration, data stewardship, confidentiality, and extensibility.

Architecture

CHARM uses an integration system (CHARM-II) based on an *arms-length information-broker architecture* and, therefore, includes a central server and three agents (one for each participating program). The central server plays the matcher role; the central server and agents together play the linker role. The central server can play a merger role, if needed. (See page 36.)

Area	Goal
Data Access	Provide consistent, current, and authoritative information about any child born in Utah or receiving services from one of the participating health-care programs.
	Establish a catalog of approved data attributes that will be shared between programs.
	Provide access to share data in real time, as client-oriented activities are being performed (i.e., at the time of encounter).
	Allow users to match children as they are entered in a participating program's information system.
Integration	Allow any program that wants to participate to do so with minimal impact on the existing information system.
	Minimize the impact of real-time data sharing to the performance of a participating program's information system.
	Reduce or eliminate the re-entry of common demographic information that can be obtained from other participating programs.
	Allow participating programs to maintain and enhance their own information systems independent of CHARM or any other participating program.
Data stewardship	Ensure that participating programs retain stewardship of their own data.
	Allow a participating program to define which of its data will be shared with others and describe their intended use and meaning.
	Allow a participating program to define security policies that govern who has access to its data. Ensure that clients (parents or guardians of the children known to CHARM) can choose who can and cannot view the data within the system.
Security	Ensure that all program and department security policies can be enforced.
	Keep an audit of who accessed data and what they accessed.
Extensibility	Design CHARM to grow in scope and size (i.e., include children through age 18 and adults, and additional participating programs).

Table 1: CHARM Goals

Deduplication Software

Matching

CHARM's central server includes a rule-based matcher that can be used in either a front-end or back-end matching process. With a front-end process, a participating program can query an interactive matching service for potential matches before it adds a new record to its own information system. With a back-end process, a participating program adds new records to its own system, registers them with CHARM, and then allows a background service to search for possible matches with those records.

The matcher uses a special data repository to hold the demographic information on children registered with CHARM, including: child last, first, middle names; birth date; multiple-birth indicator; birth order; birth weight; death information; child's addresses; mother and father names and contact information (address, phone numbers, etc.); and medical event dates. It uses field stacking for names and addresses. The demographic data do not include program-specific identifiers and are only used for matching purposes.

Each of the matcher's rules represents an independent strategy and consists of a two-part condition and a match/no-match action. The first part of the condition is a set of field comparisons that the matcher can use to quickly select a set of candidate records. The second part is another set of field comparisons that the matcher uses to determine potential matches from that candidate set and corresponding confidence levels. When a result set includes just one high-confidence match, that record is considered an exact match and linking can proceed automatically.

Result sets with multiple high-confidence matches or one or more medium-confidence matches require human resolution. If the participating program is using a front-end matching process, the resolution can occur via interaction with the user. If the result set is from a back-end matching process, the result set is placed into a deferred-match queue for later resolution by CHARM staff.

The current rules were established through a series of tests on a subset of the real data. They are stored in a server database and can therefore be refined over time.

Linking

Consistent with arms-length information-broker architectures, CHARM uses linking as its primary form of record coalescing technique. Specifically, each agent keeps track of mapping its participating program's record ID's to internal ID's, called CHARM ID's. The CHARM ID's never leave the integration system, but logically link records for the same person. The matcher's demographic database keeps track of all CHARM ID's and is responsible for assigning new ID's when participating programs register new records with the system.

Merging

Even though linking is the primary record coalescing technique, CHARM does allow for merging in several situations. For example, using a batch-end matching process, CHARM can locate single-source or multi-source duplicates that were registered without checking for duplicates first. In such cases, the matcher's demographic data will include duplicate records. If these records are from a single source, an automated merger will combine the demographics and tell the source's agent to re-map the ID's for the duplicate records to the CHARM ID for the combined data.

The agent can also make the merge action known to the participating program so it can choose to merge or link the corresponding records in its own system. If the matches come from multiple sources, the merger simply combines the demographics and notifies the agents about re-mapping the participating program ID's to the correct CHARM ID's. The agents don't need to propagate that action to participating programs, as there is no action they can take.

Unique Records Profile Questionnaire

4

Unique Records Profile Questionnaire

4

Unique Records Profile Questionnaire

Introduction

The Profile Questionnaire assists integration information systems managers and their teams in categorizing the integration system and providing a baseline for ongoing quality improvement. It is a tool that can be used to document current processes and guide decisions on how best to organize deduplication processes.

The questionnaire contains prompts about the definitions used in the Conceptual Framework to ensure that uniform terminology is used to describe processes. By using common definitions, managers will be able to relate their project information to the information in the rest of the Unique Records Portfolio and will be able to use the tools more effectively to research and analyze approaches, validate them, and find solutions to problem areas.

The Profile Questionnaire is also available online in an accessible database that Connections will maintain until May 2007 at www.phii.org. The online version of this survey allows project teams to compare approaches, so participants can learn how others are handling deduplication. De-identified information from the online survey will also be used to examine trends to develop strategies and activities that will help projects improve data quality. Participants can choose to provide information for the research database but not have it accessible online. Options are available upon accessing the questionnaire online.

I. Definitions

While completing the survey, keep in mind the following definitions:

- Integration infrastructure refers to the technical architecture as shown in the figures in the Conceptual Framework, page 23.
- Integration processes include activities related to matching, merging, and linking data.
- Integrated systems are the output of the integration such as KIDSNET, CHARM, etc. (Case Examples, page 71.)
- Integration projects are the combined activities of an entity (e.g., a state) regarding its integrated systems.

II. Profile Questionnaire

1) Integrated system name or title:

_____

_____

2) Department, group, or team responsible for the integration project (and its parent organization):

_____

_____

3) List the key contacts for details related to the technologies (integration processes) and/or technical approaches to record deduplication.

Name:_____ Phone: _____ E-mail: _____

Name:_____ Phone: _____ E-mail: _____

4) List the key contacts for information about overall policies and procedures related to preventing duplicates at the source (data source quality, data entry quality, and staff training), and about resolving duplicates and merging records.

Name:_____ Phone: _____ E-mail: _____

Name:_____ Phone: _____ E-mail: _____

5) Where is your integration project's IT support located within the organization responsible for the project?

☐ In the same department and in the same part of the organization.

☐ In the same department, but not in the same part of the organization.

☐ In neither the department nor the same part of the organization.

Identify or describe where the IT support is based:

_____

6) What is your annual birth cohort? (births/year) _____

7) How many active individuals are in your integrated system (as of _____)? _____

8) How many total records are in your integrated system (as of _____)? _____

9) Which program's systems participate in your integrated system? (Check all that apply.)

Program name *or data source*	Current in production	In progress	Planned
☐ Vital Records	_____	_____	_____
☐ Newborn Dried Blood Spot Screening	_____	_____	_____
☐ Hearing Screening	_____	_____	_____
☐ Immunization	_____	_____	_____
☐ Lead Screening	_____	_____	_____
☐ Women, Infants, and Children (WIC)	_____	_____	_____
☐ Early Intervention	_____	_____	_____
☐ Medicaid	_____	_____	_____
☐ Birth Defects Registry	_____	_____	_____
☐ Family Services	_____	_____	_____
☐ CHSCN	_____	_____	_____
☐ Emergency Services	_____	_____	_____
☐ NEDSS	_____	_____	_____
☐ Other	_____	_____	_____

10) Which external data sources are included in your integrated system?

Program name *or data source*	Current in production	In progress	Planned
☐ Hospital discharge	_____	_____	_____
☐ Social services	_____	_____	_____
☐ Laboratory results	_____	_____	_____
☐ Preschool	_____	_____	_____
☐ Providers and clinics	_____	_____	_____
☐ Health plans	_____	_____	_____
☐ Eligibility	_____	_____	_____
☐ Preschool	_____	_____	_____
☐ Daycare	_____	_____	_____
☐ CHSCN	_____	_____	_____
☐ Head Start	_____	_____	_____
☐ Other	_____	_____	_____

11) Does your integrated system maintain a separate child or person index that acts as a "master" or "authority" for the deduplication process?

☐ Yes (Skip to question 14.) ☐ No

12) Which of the participating programs (listed in question 9) maintains a child or person index that acts as an "authority" in the deduplication process? (Check all that apply, or None.)

☐ Immunization
☐ Hearing Screening
☐ Vital Records
☐ Women, Infants, and Children (WIC)
☐ Medicaid
☐ NEDSS
☐ Newborn Dried Blood Spot Screening
☐ Lead Screening
☐ Early Intervention
☐ Birth Defects Registry
☐ Family Services
☐ Other _____
☐ None

13) Among the participating programs, which provides the most authoritative demographic (personal) information about individuals?

☐ Immunization
☐ Hearing Screening
☐ Vital Records
☐ Women, Infants, and Children (WIC)
☐ Medicaid
☐ NEDSS
☐ Newborn Dried Blood Spot Screening
☐ Lead Screening
☐ Early Intervention
☐ Birth Defects Registry
☐ Family Services
☐ Other _____

14) Is your deduplication process fully automated, manual, or semi-automated?
Note: In a ***fully automated deduplication process***, behind-the-scenes software plays the roles of matcher and linker. In a ***semi-automatic deduplication process***, the matcher role may be played, at least in part, by a person responsible for identifying duplicate or overlapping records. In a ***manual system***, people are responsible for matching, linking, and merging.

☐ Fully automated ☐ All manual ☐ Semi-automated (some manual)

15) If semi-automated, how do users interact with the software to conduct deduplication?

_____
_____
_____
_____
_____
_____

16) Does your overall deduplication process use front-end matching methods, back-end methods, or a combination approach? (Check all that apply.)
Note: ***Front-end*** and ***back-end matching***. In general, matching can occur as a user enters data into a system or afterward when the record is in the database. The first approach, called ***front-end matching***, can be interactive and takes advantage of additional knowledge that the user may have about an individual, especially if the individual is present. The second approach, called ***back-end matching***, can work on many records at the same time and may not require human interaction.

☐ Front-end: Batch or individual (record searching, matching, and merging occurs prior to entering new records into the database).

☐ Front-end: Data entry (record searching occurs prior to entering new records into the database and forces users to check the database before entering the record as new).

☐ Back-end (record searching, matching, and identification of possible duplicates occurs after entering new records into the database).

☐ Combination or "other." Describe: _____

17) If your deduplication process uses a front-end approach, answer the following:

a. What is the average percentage of records that duplicate existing records in the system as they are imported? _____%

b. Do you have a method for measuring that percentage?
☐ Yes ☐ No ☐ Don't know

c. Is your front-end record matching software based on deterministic (rule-based) record matching?
☐ Yes ☐ No ☐ Don't know

d. Is your front-end record matching software based on probabilistic matching?
Note: Typically, a probabilistic approach will return search results that list potential matches with some kind of score or ranking indicating the likelihood of a match for each record.
☐ Yes ☐ No ☐ Don't know

e. Is your front-end matching software based on machine-learning technology?
Note: Machine-learning technology requires that someone "train" the system on known matches prior to being used.
☐ Yes ☐ No ☐ Don't know

18) If you use a combination of matching methods, which method results in the fewest records requiring manual resolution?
☐ Deterministic ☐ Probabilistic ☐ Machine learning ☐ Don't know

19) If your integrated system uses back-end deduplication, answer the following:

a. What percentage of the records entered or imported into your integrated system are duplicates?
Note: If your integrated system uses a central database, this is the percentage of duplicates entering that database. If your integrated system involves multiple databases, this is the percentage of unlinked or uncorrelated duplicates across all participating databases.
_____%

b. Do you have a method of measuring the scope of your duplicate records problem?
☐ Yes ☐ No ☐ Don't know

c. What percentage of duplicate records in your integrated system does the back-end deduplication process ultimately identify?
_____%

d. Is your back-end matching software based on deterministic (rule-based) matching?
☐ Yes ☐ No ☐ Don't know

e. Is your back-end deduplication software based on probabilistic record matching?
☐ Yes ☐ No ☐ Don't know

f. Is your back-end deduplication software based on machine-learning technology?
☐ Yes ☐ No ☐ Don't know

20) Does your integrated system use any commercial or open-source software for record matching or deduplication?

☐ No

☐ Yes. Provide product names and vendors:

Product name: _____

Vendor / source: _____

Product name: _____

Vendor / source: _____

21) Which data elements (e.g., child's last name, birth date, mother's last name, etc.) does your deduplication software use in matching records? List data elements in order of importance (i.e., most heavily weighted to least.)

22) Did you test other data elements or combinations of data elements prior to arriving at your present configuration?

☐ Yes ☐ No

23) Describe your quality assurance procedures for checking incoming information from clinics and hospitals. For example, when adding or importing a child's record into the system, does your system require certain data elements, such as the child's last name and birth date? Do birth dates have to be complete, or can they be approximate (e.g., month and year)?

24) List any constraints, such as a prohibition on using social security numbers or other personal information.

25) Required Data Elements (Check all that apply. Add others as text at the bottom of this list.)

☐ Child first name ☐ Child last name ☐ DOB ☐ Child SSN ☐ Medicaid number

☐ Mother first name ☐ Mother last name ☐ Mother maiden/birth name ☐ SSN
☐ Mother Medicaid number

a. Do birth dates have to be complete? ☐ Yes ☐ No

b. Can they be approximate (e.g., month and year)? ☐ Yes ☐ No

c. Does your system validate the data entered in certain data elements to ensure that the data is valid (e.g., confirms that date of birth is in the past, not in the future)? ☐ Yes ☐ No

d. Does the gender field only allow defined values for gender that are selected from predefined choices ☐ Yes ☐ No **or** is gender entered by free-text entry? ☐ Yes ☐ No

e. Are numbers allowed in text fields? ☐ Yes ☐ No

f. Is text allowed in number fields? ☐ Yes ☐ No

Other data elements or Q/A checks:

_____
_____
_____
_____
_____

Constraints on data use:

_____
_____
_____
_____
_____

Metrics and Evaluation

5

Metrics and Evaluation

5

Metrics define what is to be measured.

Introduction

Metrics are measurements designed to assess, manage, and evaluate a process or project. The Metrics and Evaluation chapter establishes a need for metrics, describes application of metrics to the processes of deduplication, and provides examples of commonly used metrics for deduplication.

The Metrics and Evaluation chapter should be used with the Profile Questionnaire to determine whether a program has developed metrics to measure the processes of deduplication, and with the Self-Assessment Checklists to help a program establish and use metrics to monitor and evaluate data quality.

Section	Purpose and content	Page
I. Need for metrics	Discussion of measurement as an essential element to quantify and evaluate processes and projects.	100
II. Metrics as evaluation tools	Metrics within a programmatic logic model. Framing metrics within goals and objectives.	101
III. Applications of metrics to deduplication matching processes	A set of definitions and a detailed step-by-step explanation of how metrics are used specifically to measure the effectiveness of a deduplication matching process.	102
IV. Calculating metrics	A detailed description of how to calculate metrics and interpret and present the results.	105
V. Additional metrics for consideration	A list of key metrics that integration projects have used to identify areas where increased duplication may be occurring and to pinpoint problems with the data itself and not just the process.	107
VI. Uses and benefits of deduplication metrics	Consideration of the benefits of having specific metrics for deduplication for overall project success.	107

I. Need for Metrics

Integrated information systems managers should plan to establish a variety of objective measures for deduplication and continually apply them to determine how effectively and competently duplicate records are being reduced and resolved.

The success of an integrated health information system relies on correctly identified records, associated across multiple systems, for an individual. As described in the Conceptual Framework, deduplication is a set of processes—matching, merging, or linking (coalescing) records—used to accurately identify individuals and their individual records. Metrics are a system of parameters or methods of quantitative assessment of a process that is to be measured, along with the processes to carry out such measurement. Metrics define what is to be measured.

A variety of measures should be in place to quantify the efficacy and efficiency of deduplication. Integrated information systems managers should plan to establish a variety of objective measures for deduplication and continually apply them to determine how effectively and competently duplicate records are being reduced and resolved. Some of the Case Examples provide detailed information on how projects are defining and applying metrics to their deduplication and using them to evaluate the efficacy of their results (see Rhode Island and New York City Case Examples). Each of the four Case Examples is at a different stage in determining metrics specific to the project.

Understanding the effectiveness of the processes involved in deduplication will enable an information systems manager to plan and estimate needs and resources based on objective data. The manager develops a clear vision of what the overall deduplication strategy should accomplish, and the system's progress toward those expectations can be examined on a regular schedule.

II. Deduplication Metrics as Evaluation Tools

Metrics may apply to several steps within the development and evaluation of an integrated information system, including project management, technology architecture, application software, data structure, and outcome. A logic model for evaluation of a program (Figure 1) generally describes a sequence of activities for implementation and functions of the program.

Inputs	Activities/process	Outputs	Outcomes	Impact
Resources needed for process or activity (e.g., staff, software, hardware)	Activities or processes to achieve outputs (e.g., application of software, manual resolution of unmatched records)	Proximate products or results obtained from activities or process (e.g., report of number of deduplicated records)	Short-term programmatic changes (e.g., reduced staff time, user satisfaction)	Long-term programmatic and systems changes (e.g., comprehensive health services for children)
Programmatic Resources / Activities		Proximate and Distal Results		

Figure 1: *Description of steps in a logic model*

Metrics can be developed for each stage of the logic model. For the deduplication of records in an integrated information system, Figure 2 proposes a logic model focusing on the inputs, processes, and outputs.

Inputs	Activities/process	Outputs	Outcomes	Impact
Staff, software	Deduplication of records	% records matched	Staff time-saving	Reduced staff numbers
		Related to project outcomes	End-user satisfaction	Greater system utilization
			Related to project goals	Reduced missed-opportunities

Figure 2: *Example of deduplication metrics within an integrated information system logic model*

Metrics are tied to goals and objectives, with the more general goals representing outcomes, and supporting objectives representing the more proximate outputs. Data-centered metrics are data such as percent of successful matches. These metrics are generally objective and less subject to dispute and are measured at the level of outputs. For example, if a program objective is to have a 95 percent record match, then the output would be the percent of record matches. Evaluation is the process of capturing and synthesizing metrics to assess the degree to which project outputs and outcomes have been attained. Business outcome-centered metrics, such as staff time saved or expended as a result of the matching software, may be used to discuss staffing resources. By stating anticipated goals and outcomes, integrated information systems managers may then operationalize them through specific metrics.

Evaluation is the process of capturing and synthesizing metrics to assess the degree to which project outputs and outcomes have been attained.

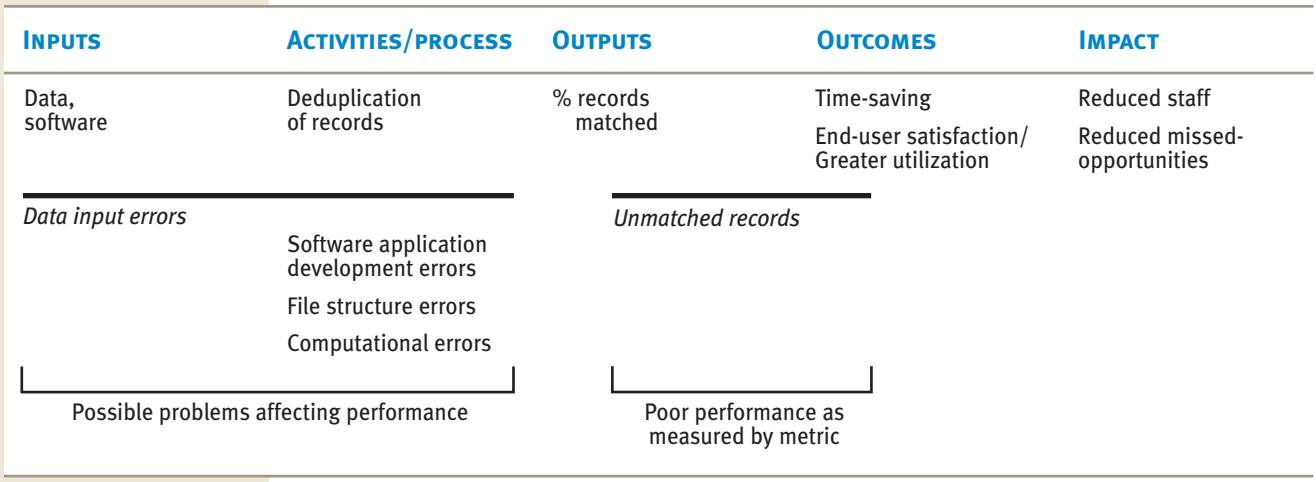


Figure 3: *Example of use of metrics for issue identification within an integrated information system*

For example, a metric may reveal that there is an increased number of duplicates in the system and may be used to assess whether problems exist in the data input or in process issues, such as software application development errors, file structure errors, or computational errors. In response, the integrated information systems manager may apply measures to the input process, either to reengineer the software application or refine computational algorithms. During any correction or recovery phase, the manager would also monitor timeliness of data release to ensure that appropriate resources are dedicated to each of the identified areas (i.e., project management, software, and hardware).

Establishing metrics for the processes of deduplication can aid in understanding system performance (Figure 3) and may be a valuable means of justifying the need for additional resources, identifying root causes of error, and understanding whether standard operations are in place.

III. Applications of Metrics to Deduplication Matching Processes

Data linkage and deduplication are a classification issue.

Data linkage and deduplication are a classification issue. Many different quality metrics for deduplication exist, but almost all of them are based on the same four terms related to a pair of records that are subject to a matching process. Each of these metrics is considered an output measure. It is important to understand these terms to correctly use the metrics that are derived from them. The Figure 4 matrix illustrates the four categories into which all pairs of records must fall after a matching process:

	DETERMINATION OF SOFTWARE	
ACTUAL	Match	Non-match
Match	TRUE POSITIVE	FALSE NEGATIVE
Non-match	FALSE POSITIVE	TRUE NEGATIVE

Figure 4: *Matching process categories*

- A **true positive** is a pair of records that are for and about the same person, and are correctly identified as such by the software.
- A **false positive** is a pair of records that are for and about two people, but are incorrectly identified as the same person by the software.
- A **false negative** is a pair of records that are for and about the same person, but are incorrectly identified by the software as two people.
- A **true negative** is a pair of records that are for and about two people, and are correctly identified as such by the software.

In practice, most matching processes have an additional category called "human review" (also known as "manual review" or "clerical review"). For the purpose of calculating metrics based on true and false positives and negatives, it is reasonable to assume that all pairs in the human review queue will be classified correctly (either as true positives or true negatives). The number of records requiring human review, however, is an important metric itself because of the staff time required to process them. For example, some projects calculate a "human review error" whereby two individuals review the same set of records and a third person reviews disagreements to determine whether an error exists or is simply a difference of opinion. The actual measurement is the average number of errors between the two raters. For optimal use of metrics in a data system, it is best to automate the metrics (those that can be automated) to identify changes (e.g., in data quality, duplication rate) as soon as they arise.

Measure	Algorithm
Sensitivity	$\frac{\text{true positives}}{\text{true positives + false negatives}}$
Specificity	$\frac{\text{true negatives}}{\text{true negatives + false positives}}$
Accuracy	$\frac{\text{true positives + true negatives}}{\text{true positives + false positives + true negatives + false negatives}}$
Precision	$\frac{\text{true positives}}{\text{true positives + false positives}}$
False positive rate	$\frac{\text{false positives}}{\text{true negatives + false positives}}$
Precision-recall break point	$\frac{\text{true positives}}{\text{true positives + false positives}} = \frac{\text{true positives}}{\text{true positives + false negatives}}$
F measure or f-score	$\frac{2(\text{precision x recall})}{(\text{precision + recall})}$

Figure 5: *Measures for common metrics used to assess deduplication processes*

Although terms such as "specificity" and "sensitivity" are used frequently, they are often understood in different terms by informaticians and epidemiologists. For the purposes of this document, these terms (Figure 5) are defined as follows:

- **Sensitivity** is a measure of the **completeness** of the results. The more matched pairs that are missed by the software, the worse the sensitivity will be. Sensitivity can be calculated as the ratio of true positives to all actual matches. In other words, it is the proportion of actual matches correctly classified by the software. Sensitivity is also known as **recall**.
- **Specificity** is similar to precision in that it also provides a measure of the proportion of false matches mixed in with the results, but, like accuracy, takes into account the total number of non-matches. Therefore, it is consistently very high regardless of the software implementation. It is calculated as the ratio of true negatives to all actual non-matches.
- **Accuracy** is a measure of the raw accuracy (correctness) of the software in classifying pairs as either matches or non-matches. Because of the high volume of non-matched pairs in most data sets, accuracy is typically in the high 90 percent range for most software implementations. It is calculated as the ratio of "true" record pairs to all record pairs. This measure is influenced by a larger number of true negatives. For example, classifying a large number of record pairs as non-matches can result in a high accuracy value.
- **Precision** is a measure of the purity of the results. The more non-matching record pairs that are incorrectly matched by the software, the worse the precision will be. Precision can be calculated as the ratio of true positives to all software matches. In other words, it is the proportion of software-identified matches that are actual matches, and it gives an idea of how many false matches will be mixed in with the actual matches. It is frequently incorrectly referred to as specificity.
- **False-positive rate** is the inverse of specificity and can be calculated as the ratio of false positives to all actual non-matches. The false positive rate includes the number of true negatives and therefore may result in misleading values due to a high number of non-matching records.

Sensitivity and precision are important not only for measuring the performance of matching software but also for identifying the probability thresholds that determine how a record pair is classified. In general, manipulating thresholds can increase sensitivity but only at the expense of precision; and vise-versa, as illustrated in Figure 6. The point where the lines for precision and sensitivity intersect (80 percent as shown in Figure 6) is called the **precision-recall break-even point.** This is the point where the precision becomes equal to recall (Christen and Goiser, 2006).

The **F measure or f-score** is the harmonic mean of the precision value and the recall value and will have high value only when both precision and recall have high values. This measure is used to identify the most appropriate compromise between precision and recall (Baeza-Yates & Ribeiro-Neto, 1999).

The importance of these common metrics cannot be understated. For example, false positives indicate falsely identified individuals, meaning an individual's identity could be linked to another individual's health records. The immediate effect of a false positive could mean additional records incorrectly matched to the original mismatched record. A

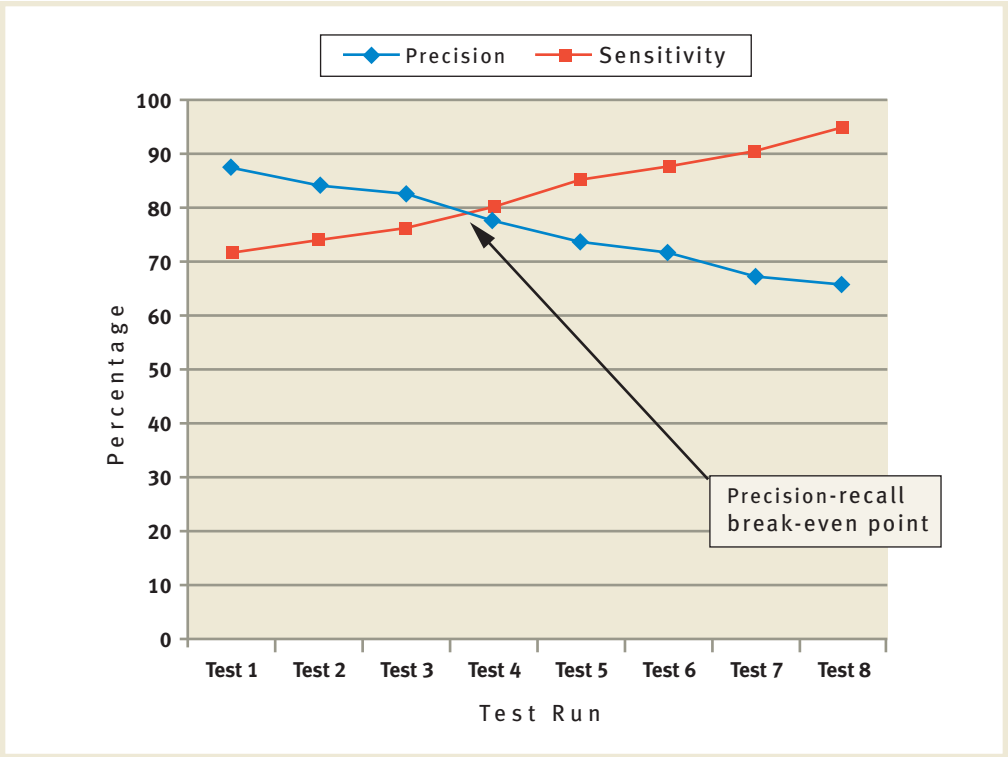


Figure 6: *Typical precision and sensitivity metrics for deduplication software*

more serious result of a false positive match is a lack of confidence in the information system by end users, such as physicians—and a possible missed opportunity for service.

A good deduplication system should minimize false positives.

As Paul Schaeffer, Director of the Master Child Index (MCI) of the New York City Health Department, states: "In New York City we are very, very careful about false positives. We have developed a deduplication system to minimize false positives as much as possible, even if it means not merging as many records. This demonstrates the seriousness of merging two different children's records. It is very difficult and costly to unmerge records, and there is a ripple effect; once two falsely merged records are brought together, they can find new false matches and the problem snowballs … A good deduplication system should minimize false positives."

IV. Calculating Metrics

Consider this example: An integrated child health information system with 100,000 children accepts an incoming data file from a large pediatric provider practice containing 500 child records. To compare every record in the provider file with every record in the integrated system, there will be a total of 50 million record pairs to examine (100,000 multiplied by 500). Assume that a software-based matching process identifies 200 matched record pairs. Next, assume that manual analysis of the results indicates that 35 of those 200 identified pairs are not actually matches. (If this were a larger data set, a preferred methodology would select a random subset of the identified pairs for manual analysis, and then extrapolate that analysis.)

	Matching software result	
Actual	Match	Non-match
Match	165 true positives	False negatives?
Non-match	35 false positives	True negatives?

Figure 7: *Example calculations for true positives and false positives*

With this information, half of the matrix can be populated as displayed in Figure 7.

Since the matching software identified 49,999,800 non-matched pairs (50 million total pairs minus 200 matched pairs), it is much more difficult to calculate the number of false negatives because of the sheer volume of non-matched pairs. Several techniques, however, seek out false negatives that will allow for an estimation of the overall figure without actually having to manually examine all the pairs. Popular techniques include:

- Relax or reconfigure blocking, or both, to seek additional matched pairs.
- Modify other matching parameters to seek additional matched pairs.
- Examine pairs manually within a certain probabilistic score below the matching threshold.

Once the false negative figure has been estimated, calculate the number of true negatives by subtracting the number of false negatives and true positives from the entire set of pairs as displayed in Figure 8.

	Matching software result	
Actual	Match	Non-match
Match	165 true positives	50 false negatives (estimate)
Non-match	35 false positives	49,999,750 true negatives

Figure 8: *Example calculations for false negatives and true positives*

Finally, we can calculate the major metrics using the numbers in the matrix in the metrics' formulas:

Precision	=	$\frac{\text{true positives}}{\text{all software matches}}$	=	$\frac{165}{165 + 35}$	=	82.5%
Sensitivity	=	$\frac{\text{true positives}}{\text{all actual matches}}$	=	$\frac{165}{165 + 50}$	=	76.7%
Accuracy	=	$\frac{\text{all "true" pairs}}{\text{all Record Pairs}}$	=	$\frac{49999750 + 165}{50000000}$	=	99.9%
Specificity	=	$\frac{\text{true negatives}}{\text{all actual non-matches}}$	=	$\frac{49999750}{49999750 + 35}$	=	99.9%

V. Additional Metrics for Consideration

Figure 9 lists key metrics that child health integration projects have used to identify areas in which increased duplication may occur. These metrics assist in also identifying possible problems not only within the processes of deduplication, but with the data itself.

Logic Model Phase	Metric	Measurement
Input	Comparisons with known or historical data source	• Number of new records without birth certificate number • Dates of records versus dates covered in batch files • Service data versus processed date • Number of incoming records versus number of records already in equals total new shots
	Categorize types of data that need manual resolution, for example the influence of certain ethnic groups on name conventions may require change to the programming algorithms or business rules changes.	Number of missing data elements in key fields
Process	Number of records requiring manual resolution; stratify by steps in process.	How many records processed / by step
	Time to resolve manual data resolution; stratify by number and type of personnel involved in manual resolution.	Number of transactions / hour
	Searches using query fields	• Number of tries to get a correct hit • Time log
	New records added without adequate search	User ID / number of duplicate records added
Outputs	Overall duplication rate	Number of duplicates over total
	System performance goal	Percent of duplicates for file acceptance: typically between two and five percent, depending on uses of file

Figure 9: *Metrics applied to the deduplication process within an integrated child health information system*

VI. Use and Benefits of Deduplication Metrics

The *Portfolio*'s Case Examples demonstrate some ways in which projects incorporate and use metrics in deduplication. (See Case Examples, page 71.) These projects demonstrate that designing processes to produce measurable outputs is an ongoing, iterative activity.

To include metrics in a project process, the team should:

- Think of metrics as critical to project management, strategic planning, system implementation, and evaluation of outputs.
- Build metrics into software requirements with goal targets.
- Insert requirements into internal or vendor contracts or, minimally, define the goals and objectives as expectations.
- Know what the program-specific needs are for deduplication.

To gain the most from metrics, project and evaluation staff should:

- Review metric reports on a periodic basis.
- Identify problems or shortcomings within the processes for deduplication and then engage the staff or developers responsible for the function.
- Close the information loop by making sure that there is follow-up by all parties involved in deduplication.
- Identify solutions and action steps.
- Initiate appropriate changes.

As Slaughter (1996) states in assessing the use and effectiveness of metrics in information systems: a case study: The information system function is under increasing pressure to demonstrate productive performance and to contribute to the [organization's] objective ... These pressures may work against a long-term perspective, discouraging the investments necessary to measure, evaluate, and learn to make continuous process improvements. Effective measurement of the process is essential in providing the information needed for process improvement.

Some of the benefits of establishing deduplication metrics include the ability to:

- Report successes and address weakness in the deduplication processes.
- Make adjustments to the deduplication processes.
- Manage deduplication software performance.
- Manage vendor performance as an outcome.
- Serve as a component of the overall software evaluation.
- Increase stakeholder buy-in to quality improvement activities.
- Manage costs and plan for deadlines.
- Manage resources: people, time, and money.
- Request additional funding based on measurable data.

Summary

The ability to resolve and remove duplicate records is of paramount importance to the credibility and usefulness of integrated information systems. Deduplication is often a sub-project within an integration project, and subject to the same planning, management, measurement, and evaluation requirements. Consequently, metrics specific to the processes of deduplication are a necessary component of an integrated information system project.

References

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. New York: ACM Press Series/Addison Wesley.

Christen, P., & Goiser, K. (2006). Quality and complexity measures for data linkage and deduplication.

In F. Guillet & H. Hamilton (Eds.), *Quality Measures in Data Mining, Studies in Computational Intelligence*: Springer.

Slaughter, S. (1996). *Assessing the use and effectiveness of metrics in information systems: A case study*. Paper presented at the ACM SIGCPR/SIGMIS conference on computer personnel research. Denver, Colorado.

Self-Assessment Checklists

6

Self-Assessment Checklists

6

Self-Assessment Checklists

Introduction

The Self-Assessment Checklists are tools to prompt managers of integrated information systems about important considerations and actions that affect deduplication. The Checklists can also help assess the readiness of the integration project for deduplicating data and provide guidance for ongoing quality assurance of the deduplication strategy. The Checklists are intended to be used internally to document where a project is and where it is going with integration and deduplication, and should be revisited periodically for project planning, staff training, and quality assurance.

The two checklists are:

Planning Integration and Deduplication—This checklist is for planning integration and deduplication strategies with specific reference to organizational and technical infrastructure and the data-sharing environment. Many of its questions are related to the challenges identified in the Conceptual Framework (see page 23). This checklist provides a guide for assessing organizational readiness for integration and for supporting deduplication.

Data Quality Assurance—This checklist is for performing ongoing quality assurance specifically for deduplication activities. It can be used to document and guide quality assurance processes for monitoring and correcting deduplication problems as well as for recording improvements.

Self-Assessment Checklist for Planning Integration and Deduplication

Purpose

This checklist is intended to help integrated information systems managers assess their organization's readiness for integration and supporting deduplication activities. The checklist is a tool to assist managers in identifying the questions to answer and documenting their integration approach to assist them in developing an effective deduplication strategy.

1. Organizational readiness for system integration

☐ a. Does your organization have any of the following?
- ○ Informatics master plan
- ○ Organization technology standards
- ○ Data quality assurance master plan
- ○ PHIN compliance strategy

☐ b. Have you identified any other existing integration projects within your organization? (If No, skip to question c.)

If Yes, what stage describes the project's status?
○ Planning ○ Development ○ Deployment

- ☐ Have the projects chosen a technical infrastructure?
- ☐ Do they have a deduplication strategy?
- ☐ Have they identified a software product or solution to deduplication?
- ☐ Are there opportunities to collaborate with existing integration projects?

☐ c. Have you identified external data sources for your proposed integration project?
- ○ If Yes, have you identified the personnel responsible for deduplication for each data source?

2. Participating public health program readiness

☐ a. Have you identified the programs that may be interested in participating in the integrated information system?

☐ b. Have you discussed the goals and expected benefits with each participating program?

☐ c. For each participating program, have you identified and documented the following?
- ○ Administrative contacts
- ○ Technical contacts
- ○ In general terms, the data that they could use from others and the data that they could share with others
- ○ How they intend to use the shared data
- ○ Timeframe in which they would like to start sharing data
- ○ Potential obstacles to sharing and integrating data
- ○ Privacy/confidentiality or consent issues
- ○ Staff and/or resource barriers
- ○ Technology barriers

3. Data readiness

☐ a. In reviewing and documenting the data models of the participating programs' existing systems:
- ○ Do they have data models, and are they current?
- ○ Are they documented in a common, understandable, and maintainable format, such as Entity Relationship diagrams, Unified Modeling Language diagrams, data dictionaries, etc.?
- ○ If not, what means will you use to determine whether and how to incorporate the data into an integrated system?

☐ b Have you reviewed the data in the databases for the following?
- ○ Inconsistent meanings in field values
- ○ Completeness (or incompleteness)
- ○ Hidden codes
- ○ Bad data
- ○ Identifier reliability
- ○ Key discriminators
- ○ Data frequencies

☐ c. Have you identified and documented what data are to be shared?
- ○ Have you considered how this may change in the future?

☐ d. If the integration system will use a common data model for shared information or for messages, have you identified and documented existing data maps against the common data model?
- ○ If so, can they be converted?

☐ e. If the integration system will include any kind of integrated system data (ISD) repository, have you identified the data that repository needs to include? (For ideas about what an ISD repository typically contains, depending on the integration-system architecture, see Conceptual Framework, page 29.)

☐ f. If the integration system will include an ISD repository, have you identified where the data will come from and how the repository will be populated? (See Conceptual Framework, page 29.)
- ○ Initial data loading
- ○ Incremental updates versus snapshot uploads versus continuous, real-time updates
- ○ Provisions for people who have opted out of data sharing in full or in part

4. Messaging and telecommunications readiness

☐ a. Have you researched existing or emerging messaging and data exchange protocols within your organization or within specific programs?
If Yes, do these include:
- ○ The use of standard message types (HL7)?
- ○ Communication technologies (Web services, CORBA, J2EE, etc.)?
- ○ Vocabularies (e.g., SNOMED, ICD-9, 10, ICF, LOINC, CPT/CVX, etc.)?

☐ b. Are you considering a portal with single sign-on for access to the integrated system?

5. Security readiness

☐ a. Is there an authentication/authorization infrastructure within the organization, either deployed or envisioned?
- ○ If Yes, who manages this activity?

☐ b. Have you reviewed the privacy implications of integrating data, and legislation related to disclosure, opt in versus opt out, etc.?

6. Organizational readiness for data sharing

☐ a. Have you formalized data sharing agreements within the organization or among the participating programs, or both?
- ○ Do they include data quality standards?

☐ b. Have you discussed the level of resources that would be available at the organization level for:
- ○ Resolution of potential matches of records across participating programs, if the architecture uses centralized back-end matching?
- ○ Merging requiring human intervention?
- ○ Technical support for integration system hardware and software?

☐ c. Have you discussed the level of resources that would be available with each program for:
- ○ Technical consultation?
- ○ Resolution of potential matches of records with the program?
- ○ Merging of records within the program?

7. Scalability

☐ a. Is the database robust enough to accommodate future growth?

☐ b. Can it be updated?

☐ c. Does it have "plug-and-play" capabilities with potential new data sources?

8. Support for system-wide analysis

☐ a. Is there a plan for quality checks?

☐ b. Is there a plan for evaluation?

☐ c. Have you included quality evaluation in the budget?

9. Project management

☐ a. Does the project have a project charter identifying the stakeholders and describing how the project will be governed?

☐ b. Does the project have a project plan?

☐ c. Is there a project manager to oversee the timeline of development and deployment?

☐ d. Have you planned for and budgeted for training and ongoing user help?

☐ e. Have you planned for future updates to accommodate future expansion?

☐ f. Do you have a plan for evaluating the information system?

10. Reliability

☐ a. Have performance benchmarks been established?

☐ b. Has the system been tested in pre-production and production releases?

☐ c. Is there a system for reporting and correcting performance problems?

11. Analysis of trade-offs among integration goals, resources, and system performance

☐ a. In designing your system, have you examined the trade-offs based on the degree and type of coordination required for data sharing? (See Conceptual Framework, page 48.)
- ○ Central versus distributed control
- ○ Incremental versus big-bang implementation
- ○ Strong versus weak information boundaries
- ○ Stewardship guidelines

☐ b. Have you examined the trade-offs based on the degree and type of coordination required among underlying data models?
- ○ Common data model versus coordinated data models
- ○ Lock-step versus independent evolution

☐ c. Have you examined the trade-offs based on the degree and type of coordination required for communication?
- ○ Common communication protocols
- ○ Availability of commercial solutions

Self-Assessment Checklist for Data Quality Assurance

Purpose

The Checklist for Data Quality Assurance provides a structure for performing ongoing quality assurance of the deduplication strategy. Documentation is the first step to establishing and performing a quality assurance process. Each time a data source is added, modified, or removed from an integrated information system, or a contact person is added, changed, or removed, the information and effective dates of such changes should be recorded. Documentation of quality assurance processes provides a road map for monitoring and correcting duplication problems and recording improvement and success.

1. Documentation

For each data source, have you documented:

- ☐ Contacts (administrative, technical, clinical, user)?
- ☐ Data models / data discontinued?
- ☐ Data elements / data definitions?
- ☐ Required fields?
- ☐ Communication?
  - ○ Protocols
  - ○ Frequency
  - ○ Size and trends of data input or data transfers
- ☐ Monitoring practices?
  - ○ Description of practices
  - ○ Data testing practices

2. Evaluation capabilities (Metrics)

Key questions:

- ☐ Have you assessed the database for errors?
- ☐ Have you assessed the database schemas for compatibility?
- ☐ Have you determined how you will deal with non-matches?
- ☐ Does your merge software allow the user to merge individuals?
- ☐ Does the merge software recommend merges of individuals based on user-established criteria?
- ☐ Does the merge software allow the user to merge tags?
- ☐ Does it allow the user to merge sources?
- ☐ Does the merge software allow / enable the user to attach a source citation indicating the source of the data prior to merge? If Yes, at what level (individual, encounter, etc.), and is the citation attached?

3. Quality-review capabilities

Does your project have the following types of reviews?

- ☐ Quality
  - ○ Of data output
  - ○ Of data input
- ☐ Metrics
  - ○ Sensitivity
  - ○ Specificity
  - ○ Accuracy
  - ○ Precision
  - ○ Does your project have a standard for the percent of tolerated duplicates after all processes?
  - ○ What percentage is your goal?

4. Follow-up capabilities

- ☐ Who is the contact person for feedback to the data source about:
  - ○ Technical issues?
  - ○ Quality issues?
  - ○ Clinical issues?
  - ○ Other issues?

5. Processes for resolution of data problems

- ☐ Has your project defined a process for resolving these data problems?
  - ○ Duplicates and overlays
  - ○ Records that can't be processed (as in the case of insufficient data)

6. Processes for change

- ☐ Does your project have a communication plan for:
  - ○ Planned updates
  - ○ Interruptions
  - ○ Back-up systems

7. Institutionalizing quality assurance

- ☐ Where is the quality assurance function located organizationally?
  - ○ Program management
  - ○ Technical management
  - ○ Staff level, program
  - ○ Staff level, technical
- ☐ Is there a quality assurance advisory or management group?
- ☐ Is there a means of communicating quality assurance problems and success to:
  - ○ Executive management?
  - ○ Program management?
  - ○ Technical management?
- ☐ Is communication in the form of:
  - ○ Regular reports of all quality assurance activities?
  - ○ Problem reports only?

Glossary

7

Glossary

7

ACCESS CONTROL. A security technology that selectively permits or prohibits certain types of data access based on the identity of the accessing entity and the data object being accessed.

ACCOUNTABILITY. The ability to trace the actions of people, organizations, programs, computers, etc.

ACCURACY. A measure of the raw accuracy (correctness) of the software in classifying pairs as matches or non-matches. Because of the high volume of non-matched pairs in most data sets, accuracy is typically in the high 90 percent range for most software implementations. It is calculated as the ratio of "true" record pairs to all record pairs.

ADAPTABILITY. The capability of being modified to suit different purposes or conditions.

ALGORITHM. A logical sequence of steps for solving a problem, often displayed in a flow chart that can be translated into a computer program. A formula or set of steps for solving a particular problem. To be an algorithm, a set of rules must be unambiguous and have a clear stopping point.

APPLICATION. A computer program or piece of software designed to perform a specific task.

APPLICATION INTEGRATION. The process of enhancing end-user software to present a unified view of the data.

ARCHITECTURE. Term applied to both the process and the outcome of specifying the overall structure, logical components, and the logical interrelationships of a computer, its operating system, a network, or other components.

AUDIT TRAIL. A security component of a computer system that maintains a log of user access, files accessed, and access times to help detect unauthorized use.

AUTHENTICATION. A process used to confirm the identity of a person or to prove the integrity of specific information. Message authentication involves determining its source and verifying that it has not been modified or replaced in transit.

AUTHORIZATION. Permission associated with accessing functions or subsets of data. Generally, an administrator will define the users who are authorized to access application functions or data.

AVAILABILITY. A high-level availability system allows the system to remain accessible, accurate, and secure in spite of failures.

BACK-END MATCHING. Matching that occurs after a record is already put into the database. Back-end matching allows users to enter new individuals into the system without initially worrying about duplication. The matching looks for duplicate and overlapping records behind the scenes at a later time.

BIOMETRICS. The electronic capture and analysis of biological characteristics, such as fingerprints, facial structure, or patterns in the eye. Through advancements in smart cards and cheaper reader prices, biometrics is catching on as a security alternative to passwords.

BLOCKING. A record-matching task of dividing records into groups to reduce the number of records needed to compare. Groups are based on an attribute such as the first letter of the last name, the birth year, or the gender of each record. The best attributes for blocking are evenly distributed in the data and not subject to a high degree of reporting error.

BROKER. An application system that acts as an intermediary between two collaborating systems or services.

BROKER SERVICES. Reads the business message that has been transformed to the canonical form and instantiates the appropriate workflow that will be used to process the business request.

BUSINESS RULE. A statement that defines or constrains some aspect of the business. It is intended to assert business structure or to control or influence the behavior of the business. The statement is usually in "if _ then" format that describes the appropriate next step to take given a variety of variables.

CENTRAL DATABASE. A central authority assembles and operates a database of consolidated records, including medical, service, or program data.

CENTRAL INDEX. Architecture that uses an index that contains information on the location of medical records for persons known to the system. The central index, also called a Central Repository MPI, Registry MPI, Enterprise MPI, or Index of Indices, needs to hold enough demographic information that it can be used to perform effective matching, but does not contain any medical or program data.

COALESCING. The linking or merging of records so users can readily access all the available data for a given individual, within the limits of confidentiality and inter-agency data-sharing policies. (See also Record Coalescing.)

CONFIDENTIALITY. The condition in which sensitive data is kept secret and disclosed only to authorized parties.

CONVERTER. A software element that transforms data from one form to another.

DATA FIELD. An area in a computer memory or program where information can be entered and manipulated.

DATA FREQUENCY. The number of times that the same data value occurs.

DATA INTEGRATION. The process of combining information from independent sources, such as vital records and newborn screening.

DATA INTEGRITY. A condition in which data has not been altered or destroyed in an unauthorized manner.

DATA MODEL. Describes the organization of data in an automated system. The data model includes the subjects of interest in the system (or entities) and the attributes (data elements) of those entities. It defines how the entities are related to each other (cardinality) and establishes the identifiers needed to relate entities to each other (primary and foreign keys). A data model can be expressed as a conceptual, logical, or physical model.

DATA SOURCE. Something or someone that creates new person records or provides new data to the system.

DATA USER. For an integrated system, a data user is someone or something that accesses information from one or more data repositories.

DATA WAREHOUSE. A database of information intended for use as part of a decision support system. The data is typically extracted from an organization's operational databases.

DATABASE. A set of related information created, stored, or manipulated by a computerized management information system.

DEDUPLICATION. The processes that match, link, and/or merge data.

DEDUPLICATION PROCESS. Identifying duplicate records and either linking or merging the records. Record matching and record coalescing are two fundamental processes for deduplication in an integrated information system.

DEDUPLICATION STRATEGY. A strategy for identifying duplicate records among separate data sources and linking or merging those records. A deduplication strategy should include at least four components:

- A set of policies and procedures guiding the operation of the integration system.
- A technical architecture that supports the policies and procedures.
- An operational plan or set of activities that address the core data quality goals of the integration system.
- A method of evaluating the deduplication processes to determine how effectively and competently duplicate records are being reduced and resolved.

DUPLICATE RECORDS, SAME SOURCE. Two or more records for the same person from the same source.

DUPLICATE RECORDS, MULTI SOURCE. Two or more records for the same person from different sources with similar types of data.

ENCRYPTION. The process of transforming plain text data into an unintelligible form (ciphertext) so that the original data either cannot be recovered (one-way encryption) or cannot be recovered without using an inverse decryption process (two-way encryption).

ENTERPRISE. A relative business unit. A public health enterprise may be the local public health department, a particular division within a large metropolitan health department, a state health department, or the entire local-state-and-federal public health system.

ENTERPRISE ARCHITECTURE (EA). An explicit, common, and meaningful structural frame of reference to enable efficient, consistent articulation of business objectives and information systems planning at all levels of an organization. EA development and evolution serves as a focal point for internal planning over time as well as the means to consistently articulate and extend organizational objectives and implications to business partners. (See also ARCHITECTURE and FEDERATED ARCHITECTURE.)

EVALUATION. The process of capturing and synthesizing metrics to assess the degree to which project output and outcome goals have been attained.

EXTENSIBILITY. The ability to economically modify or add functionality.

FALSE-POSITIVE RATE. The inverse of specificity. It can be calculated as the ratio of false positives to all actual non-matches. The false positive rate includes the number of true negatives and therefore may result in misleading values due to a high number of non-matching records.

FEDERATED ARCHITECTURE. A collection of database systems (components) to unite into a loosely coupled federation in order to share and exchange information. The term federation refers to the collection of constituent databases participating in a federated database. (See also ARCHITECTURE and ENTERPRISE ARCHITECTURE.)

FELLEGI-SUNTER. A formal model for matching that uses the relative frequency of strings being compared.

FIELD. An area in computer memory or program or on a computer display (e.g., a text field in an online form) where information can be entered and manipulated.

FIELD COMPARISON. An exact-field comparison allows users to provide possible values for some or all of the data fields and the matcher searches for records that have those values.

FIELD-MEANING SHIFT. The meaning of a given field might drift over time, either intentionally or unintentionally. For example, the date field that originally meant the date of a point of service might gradually become the date of entry for a point of service.

FIREWALL. A computer networking security device that controls access to network resources (e.g., computers and systems) using pre-defined security policies and rules.

FLEXIBILITY. The ability to support architectural and hardware configuration changes.

FORMAT. The structured arrangement of data fields and elements that make up a particular electronic transaction. The organization of data in such a way as to have a specific, agreed meaning to users.

FRONT-END MATCHING. Matching that occurs as a person enters data into a system, often interactively. Systems that support front-end matching allow users to look for potential matches prior to adding a new record into the system or during the process of adding a new record. If a match is found, it is used instead of adding the new record. The aim of front-end matching is to minimize the number of duplicate records that actually get into the database.

GLOBAL ID. System-wide identifier that uniquely identifies a person known to the system. A person's ID becomes a logical mechanism that links all the data for that person.

GLOBAL ID MANAGER. A software function (or a person) that handles the assignment of ID's to ensure that no two people receive the same ID.

GRAPHICAL USER INTERFACE. An interface to an application that allows users to do things by clicking on a visual screen, as opposed to typing commands. A GUI (pronounced gooey) features the following components: a pointing device (such as a mouse), icons, windows, and menus.

HIPAA (HEALTH INSURANCE PORTABILITY AND ACCOUNTABILITY ACT). The Administrative Simplification provisions of the Health Insurance Portability and Accountability (HIPAA) Act of 1996 establish national standards for electronic health care transactions and national identifiers for providers, health plans, and employers. HIPAA also addresses the security and privacy of personal health information.

IDENTIFIER. A set of numeric or alpha-numeric keys that uniquely identify an individual known to the integrated system.

IMPLEMENTATION. Implementation is the carrying out, execution, or practice of a plan, a method, or any design for doing something. Implementation is the action that must follow any preliminary thinking in order for something to actually happen.

IMPOSSIBLE VALUES. Over time, as programs and data structures change, some data fields may end up with values that are impossible or illogical (e.g., 99/99/99 for dates, -1 for birth weights, and numbers for names). Impossible and illogical values can find their way into the system through weak user interfaces, changes in database structures, programming errors, and field-meaning shift.

INCOMPATIBLE CODE. Multi-source duplicates may have code fields that represent the same basic information but have different meanings.

INFORMATICS. The science of information. It is often, though not exclusively, studied as a branch of computer science and information technology and is related to database, ontology, and software engineering. Informatics focuses on understanding problems and then applying information (and other) technology as needed.

INFORMATION INFRASTRUCTURE. The comprehensive information support structure, or core IT capacity, that enables achievement of broad objectives.

INFORMATION SYSTEMS (IS). The applications that enable the use of information technology (IT) to address specific business processes, in this case, the work of public health.

INFORMATION TECHNOLOGY (IT). The specific technical elements that enable electronic data management solutions.

INTEGRATION. 1: The process of bringing together related parts into a single system or a single view. To make various components function as a connected system. 2: Combining separately developed parts into a whole so that they work together. The means of integration may vary, from simply mating the parts at an interface to radically altering the parts or providing something to mediate between them.

INTEGRATED SYSTEM. An integrated system is a collection of data sources and data users that are connected via an integration infrastructure.

INTEGRATION INFRASTRUCTURE. A collection of software components, outside of the individual data sources, that support the integration processes. An integration infrastructure may include software matchers, mergers, data converters, etc.

INTEGRATION PROCESS. Any activity (manual or automated) that is needed for data integration. Key integration processes: Handling duplicate or overlapping data; presenting integrated data to users in a consistent and meaningful way; ensuring the security and confidentiality of shared data; and establishing data sharing agreements.

INTEGRATION PROJECT. An integration project involves building or adapting an integration infrastructure in support of specific integration processes to create an integrated system.

INTERFACE. As a noun, an interface is either: 1) A user interface, consisting of the set of dials, knobs, operating system commands, graphical display formats, and other devices provided by a computer or a program to allow the user to communicate and use the computer or program. A graphical user interface (GUI) provides its user a "picture-oriented" way to interact with technology; or 2) A programming interface, consisting of the set of statements, functions, options, and other ways of expressing program instructions and data provided by a program or language for a programmer to use.

INTEROPERABILITY. The ability of two or more systems or their components to exchange information and to use the information that has been exchanged. Interoperability enables health information systems to work together within and across organizational boundaries in order to advance the effective delivery of health care for individuals and communities.

ISD REPOSITORY. An integrated-system data (ISD) repository is a database within the integrated system, but not part of any particular data source. Depending on the architecture, the ISD repository may be called a central database, data vault/registry, master index, or a number of other terms.

INTERRECORD REFERENCES. Linking matching records using record IDs or keys. Within a single database, this can be relatively simple and effective. Given two matching records, the ID of the second is stored in the first record, typically in an internal linking field. Similarly, given three matching records, the first record references the second, and the second references the third. Adding another record to a set of matching records only requires adding a link from the last record of the existing chain to the newly found matching record.

LINKER. A mechanism that logically connects records that are determined to be for the same individual. It is not necessarily a piece of code.

LINKING. The act of logically connecting records that are determined to be for the same individual.

LINKING TABLE. A table of field values that compares the value in fields of multiple records and connects records that have the same value in the same field.

LOGICAL DESIGN. The process in which the database requirements for the system are described. This is the final step in the requirements development process, prior to physical design. The products of logical design provide guidelines from which the programmer can work.

MACHINE LEARNING. A machine-learning algorithm attempts to allow the deduplication software to customize itself. It does this through a training process in which pairs of records are fed into the system along with their true match/no-match status. For each training pair, the system attempts to compute its own match/no-match result based on its current settings. If it gets the right answer, it reinforces the current settings. If it gets the wrong answer, it tries to figure out what would have helped produce the right answer and alters its settings. By running lots of training data through the system, it can eventually tune its own configuration to correctly compute all answers. At this point, the system should be able to accurately match other pairs of records not in the training data.

MASTER PERSON INDEX (MPI). A system that coordinates client identification across multiple systems by collecting and storing ID's and person-identifying demographic information from source system (i.e., track new persons, track changes to existing persons). These systems also take on several other tasks and responsibilities associated with client ID management.

MATCHER. Someone (a person) or something (software) that tries to determine if two or more records are for the same individual.

MATCHING. The process of finding existing records that might be for the same person.

MATCHING, BACK END. Occurs after a record is already in the database. Back-end matching allows users to enter new individuals into the system without initially worrying about duplication. The matching looks for duplicate and overlapping records behind the scenes at a later time.

MATCHING, FRONT END. Matching that occurs as a user enters data into a system, often interactively. Systems that support front-end matching allow users to look for potential matches prior to adding a new record into the system or during the process of adding a new record. If a match is found, it is used instead of adding the new record. The aim of front-end matching is to minimize the number of duplicate records that actually get into the database.

MEANINGLESS VALUES. Sometimes a program requires information for a field, but the user doesn't know that information or that information doesn't exist. To get around the requirement and continue working, the user enters a bogus or temporary value. For example, if a program requires a first and last name for a child and the child doesn't have a first name yet, the user might simply enter Boy or Girl. Such values are essentially meaningless with respect to the fields they are in.

MEDICAL INFORMATICS. The application of information technology to health care.

MERGER. Something or someone that combines multiple records for an individual into a single record.

MESSAGE. A digital representation of information. A computer-based record.

METRICS (or measures). A system of parameters or methods of quantitative assessment of a process that is to be measured, along with the processes to carry out such measurement (Metrics, page 97).

MIDDLEWARE. Software systems that facilitate the interaction of disparate components through a set of commonly defined protocols. The purpose is to limit the number of interfaces required for interoperability by allowing all components to interact with the middleware using a common interface.

NORMALIZATION. The process of creating a uniform and agreed-on set of standards, policies, definitions, and technical procedures to allow for interoperability.

OVERLAPPING RECORDS. Two or more records for an individual from different sources with different types of data.

OVERLAYING RECORDS. Two or more records that appear to be for an individual, but are actually for different individuals.

PARSER. A function that recognizes valid sentences of a language by analyzing the syntax structure of a set of tokens passed to it from a lexical analyzer.

PARTITIONED CENTRAL DATABASE. Like a central database, a partitioned central database is a database of consolidated records, including medical, service, or program data. Instead of a single central database, however, the central authority maintains a collection of ISD repositories, called data vaults, typically one for each participating program.

PERFORMANCE. The ability to execute functions fast enough to meet requirements.

PERFORMANCE MEASURES. Quantitative measures of capacities, processes, or outcomes relevant to the assessment of a performance indicator (e.g., the number of trained epidemiologists available to investigate or the percentage of clients who rate health department services as "good "or "excellent").

PERFORMANCE STANDARDS. Objective standards or guidelines used to assess an organization's performance (e.g., one epidemiologist on staff per 100,000 population served or 80 percent of all clients who rate health department services as "good " or "excellent"). Standards may be set based on national, state, or scientific guidelines; by benchmarking against similar organizations; based on the public's or leaders' expectations (e.g., 100 percent access, zero disparities); or other methods.

PRIVACY. The right of an individual to control the dissemination and use of information that relates to the individual, or to have information about oneself be inaccessible to others.

PRECISION. A measure of the purity of the results (or to the epidemiologist, predictive value positive/negative.) The more non-matching record pairs that are incorrectly matched by the software, the worse the precision will be. Precision can be calculated as the ratio of true positives to all software matches. In other words, it is the proportion of software-identified matches that are actual matches, and it gives an idea of how many false matches will be mixed in with the actual matches. It is frequently incorrectly referred to as specificity.

PROBABILISTIC ALGORITHMS. Algorithms that take advantage of data frequencies based on the likelihood that an event will occur, expressed as the ratio of the number of favorable outcomes in the set of outcomes divided by the total number of possible outcomes.

RECALL. See SENSITIVITY.

RECORD COALESCING. The linking or merging of records so users can readily access all the available data for a given individual, within the limits of confidentiality and inter-agency data-sharing policies.

RECORD LOCATOR SERVICE (RLS). In a central index architecture, the component that actually performs matching. It can be either an integral part of the central index or a separate component.

RECORD MATCHING. The identification of previously unrelated records that represent the same individual.

REGISTRY. Directory-like system that focuses solely on managing data pertaining to one conceptual entity.

ROLE-BASED ACCESS CONTROL. Security environment in which users' rights to access or change information are controlled by the role or roles they fulfill within the organization—that is, what they do, rather than who they are.

RULE-BASED MATCHING. Rule-based matching algorithms are similar to weighted-field matching algorithms in that they can involve multiple field comparisons using a variety of advanced comparison functions. They don't compute confidence by summing the results of individual field comparisons. Instead, they apply a set of decision rules (i.e., IF‹condition›THEN ‹action› statements). The conditions consist of field comparisons, and the actions consist of "match" or "no-match" conclusions. If a rule's condition is true, then its action is taken.

SCALABILITY. The ability to support the required quality of service as load increases.

SECURITY. The ability to ensure that information is neither modified nor disclosed except in accordance with the security policy.

SECURITY AUDITS. Periodic evaluation of a system for vulnerability to potential security threats or breaches of confidentiality.

SECURITY ARCHITECTURE. A plan and set of principles for an administrative domain and its security domains that describe the security services that a system is required to provide to meet the needs of its users, the system elements required to implement the services, and the performance levels required in the elements to deal with the threat environment.

SECURITY POLICY. A set of rules and practices that specify or regulate how a system or organization provides security services to protect resources. Security policies are components of security architectures. Significant portions of security policies are implemented via security services, using security policy expressions.

SECURITY STANDARD. A set of requirements adopted or established to preserve and maintain the confidentiality of electronically stored, maintained, or transmitted health information.

SEMANTIC HETEROGENEITY. Seemingly similar fields from different data sources that may have slightly different meanings that, if unrecognized, could cause data inconsistencies or loss of credibility. For example, two data sources may both include a field for an address. For one source, these might be intended to hold a physical address that could be used to locate an individual or check residency. The other source might use its address field only for mailings and verifying identity. If values from these fields were merged, the resulting merged record could be slightly incorrect and of less use to both data sources.

SENSITIVITY. A measure of the completeness of the results (or to the epidemiologist, true/false positives/negatives). The more matched pairs that are missed by the software, the worse the sensitivity will be. Sensitivity can be calculated as the ratio of true positives to all actual matches. In other words, it is the proportion of actual matches correctly identified by the software. Sensitivity is also known as recall.

SIMPLE OBJECT ACCESS PROTOCOL (SOAP). Lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation specific semantics.

SOURCE TAGGING. As a merger combines data from multiple sources into a single record, it can associate the original source of value with that data. Thus source tagging can enhance the reliability and accuracy of subsequent matching, merging, and unmerging operations.

SPECIFICITY. Similar to precision in that it also provides a measure of the proportion of false matches mixed-in with the results, but, like accuracy, takes into account the total number of non-matches. Therefore, it is consistently very high regardless of the software implementation. It is calculated as the ratio of true negatives to all actual non-matches.

SSL (SECURE SOCKETS LAYER). The de-facto standard for executing secured transactions over the World Wide Web (e.g., SSL is in use when the padlock icon is displayed during secure online transactions). SSL protects confidentiality and data integrity through encryption and can also provide authentication of the parties involved in a transaction.

STANDARDS. Clearly defined and agreed-on conventions for the operation of specific computing operations, formats, and data elements.

STACKING. A technique of retaining historical data or multiple values for a field (i.e., address).

SYSTEM ARCHITECT. The system architect puts together the skeleton of an information system project. Depending on the specifications gathered by the business (or requirements) analyst, the system architect will choose to focus on ease of maintenance, application performance, compatibility with existing systems, or a combination. Each decision that the system architect makes has to be carefully considered because poor decisions made at the beginning of a project can have damaging effects later in the information systems development lifecycle.

SYSTEM SECURITY. The component of a security program that integrates physical and personnel security with procedures designed to protect the entity/system needing protection.

TCP/IP (TRANSMISSION CONTROL PROTOCOL/INTERNET PROTOCOL). Standards that are the basis for data transmission on the Internet and over a LAN (local area network) and WAN (wide area network).

TECHNICAL ARCHITECTURE. 1: Identifies and describes the types of applications, platforms, and external entities; their interfaces; and their services, as well as the context within which the entities interoperate. Serves as the basis for selecting and implementing the infrastructure to establish the target architecture. 2: The specific code plans to build an IT solution. The IT "blueprint" of the planned technical rollout.

TIMELINESS. A data quality measure that deals with the currency of the shared data and indicates that the data is available at the right or appropriate limited period during which an action, process, or condition exists or takes place.

UNIQUE IDENTIFIER. Typically a set of numeric or alpha-numeric keys, each the only one of its value that uniquely identifies an individual known to the integrated system.

VIRTUAL PRIVATE NETWORK (VPN). Secure and encrypted connection between two points across the Internet.

Notes

8

Notes

8

NOTES

Public Health INFORMATICS *Institute*

750 COMMERCE DRIVE, SUITE 400
DECATUR, GEORGIA 30030

1.866.815.9704 | WWW.PHII.ORG