NEDSS Logical Data Model
Overview and Users' Guide

Version 1.0

NEDSS Logical Data Model (NLDM) Overview and Users' Guide

Revision History

Date	Version	Description	Author

Table of Contents

1 1 INTRODUCTION 9

1 1 INTRODUCTION 9

1.1 PURPOSE OF THIS GUIDE 9
1.2 SCOPE OF THIS GUIDE 9
1.3 VOCABULARY: DEFINITIONS, ACRONYMS, AND ABBREVIATIONS 10
1.3.1 Introduction 10
1.3.2 NEDSS Program 10
1.3.3 NEDSS Base System (NBS) 11
1.3.4 Program Area Module (PAM) 11
1.3.5 Application Extension Domain 11
1.3.6 Health Level 7 (HL7) 11
1.3.7 HL7 Reference Information Model (RIM) 11
1.3.8 Entity and Class 11
1.3.9 Attribute 12
1.3.10 Named Attribute 12
1.3.11 Extensible Attributes 12
1.3.12 Extended Attributes 12
1.3.13 Table 12
1.3.14 Column 13
1.3.15 Row 13
1.3.16 Tuple 13
1.3.17 Instance 13
1.3.18 Occurrence 13
1.3.19 Name-Value Pair (NVP) 13
1.3.20 Use Code 13
1.3.21 Unique Identifier (UID) 13
1.3.22 Standard Identifiers 14
1.3.23 Local Identifiers 14
1.3.24 Database Management System (DBMS) 14
1.3.25 Operational Data Store (ODS) 14
1.3.26 Message-In (Msg-In) 14
1.3.27 Message-Out (Msg-Out) 14
1.3.28 NBS Data Mart 15
1.3.29 Schema 15
1.3.30 Acts 15
1.3.31 Work-Up 15
1.3.32 Investigation 15
1.3.33 Jurisdiction 15
1.3.34 Public Health Directory (PHDIR) 15
1.4 NLDM VERSIONS AND ITS FUTURE EVOLUTION 15
1.5 REFERENCES 16
1.6 MAJOR COMPONENTS OF THE NLDM 16

1.6.1 Introduction ... *17*
1.6.2 ODS ... *17*
1.6.3 Message-In (MSG-In) ... *17*
1.6.4 Message-Out (MSG-Out) ... *17*
1.6.5 NBS Data Mart ... *17*
1.6.6 Relationships of major NLDM components ... *18*
1.6.7 Message-In Data Store ... *19*
1.6.8 Operational Data Store ... *19*
1.6.9 Message-Out Data Store ... *19*
1.6.10 NBS Data Mart ... *19*
1.7 REPRESENTATIVE FLOW OF DATA THROUGH THE NLDM ... 20
1.7.1 Initial Data Migration of State Demographic data into the NLDM ... *20*
1.7.2 Conversion of Data from the Message-In to the Operational Data Store ... *20*
1.7.3 Registry Manager Review of Migrated Person Data ... *21*
1.7.4 Ongoing Receipt of Electronic Laboratory Results ... *21*
1.7.5 Review of Case Data ... *21*
1.7.6 Loading of the NBS Data Mart for Analysis and Reporting ... *22*
1.7.7 On-going Synchronization of an NBS Database with other State Databases. ... *22*
1.8 OVERVIEW OF THE PRIMARY NLDM DESIGN PRINCIPLES ... 22
1.9 BASIC STRUCTURE OF THE NLDM ... 23
1.9.1 Overview ... *23*
1.9.2 Entity Classes ... *24*
1.9.3 Locator Classes ... *25*
1.9.4 Entity-Act-Participation Classes ... *26*
1.9.5 Act Classes ... *27*
1.9.6 Observation Classes ... *28*
1.9.7 System Reference Code classes ... *29*
1.10 CODES ... 30
1.10.1 Functional codes ... *30*
1.10.2 Structural codes ... *30*
1.10.3 Attribute Name Identifiers ... *30*
1.10.4 Attribute Name Identifier Codes ... *31*

2 RELATIONSHIP OF THE NLDM TO OTHER DATA MODELS ... 32

2.1 INTRODUCTION ... 32
2.2 RELATIONSHIP TO HL7 REFERENCE INFORMATION MODEL (RIM) ... 32
2.3 RELATIONSHIP TO THE PUBLIC HEALTH CONCEPTUAL DATA MODEL (PHCDM) ... 32
2.4 RELATIONSHIP TO THE NBS INFORMATION MODELS ... 32
2.5 RELATIONSHIP TO STATE LOGICAL AND PHYSICAL MODELS ... 32
2.6 RELATIONSHIP OF THE NLDM TO SEPARATE MPI DATABASES ... 33

3 MAIN DATA MODELING PRINCIPALS FOLLOWED IN THE NLDM ... 35

3.1 INTRODUCTION ... 35
3.2 SUBJECT AREAS ... 35
3.2.1 Entities ... *35*
3.2.2 Locators ... *36*
3.2.3 Entity Act Participations ... *36*

3.2.4 *Acts* ... 36
3.2.5 *Observations* ... 37
3.2.6 *System Reference Codes & System Tables Subject Area* ... 37
3.3 GENERAL TYPES OF CLASSES ... 37
3.3.1 *Introduction* ... 37
3.3.2 *Core Classes* ... 38
3.3.3 *Supporting Classes* ... 38
3.3.4 *Associative classes* ... 38
3.3.5 *Super classes* ... 38
3.3.6 *Sub classes* ... 38
3.4 GENERAL TYPES OF RELATIONSHIPS ... 38
3.4.1 *Introduction* ... 38
3.4.2 *Super Class to Sub-Class* ... 39
3.4.3 *Participations* ... 39
3.4.4 *Associations* ... 39
3.4.5 *Relationships between a Class and Repeating Attributes of the Class* ... 40
3.5 IDENTIFYING, NON-IDENTIFYING, AND CATEGORY RELATIONSHIPS ... 40
3.5.1 *Introduction* ... 40
3.5.2 *Identifying Relationships* ... 40
3.5.3 *Non-identifying Relationships* ... 40
3.5.4 *Categorical Relationship* ... 41
3.6 KEYS ... 41
3.6.1 *Primary Keys* ... 41
3.6.2 *Foreign Keys* ... 41
3.6.3 *Indices* ... 41
3.7 CHANGE HISTORY CLASSES ... 41
3.8 ENTITY MERGE CLASSES ... 42
3.9 RECORD LEVEL SECURITY ... 42
3.10 SYSTEM MANAGED ATTRIBUTES ... 42
4 SUMMARY OF THE NLDM CLASSES ... 44
4.1 INTRODUCTION ... 44
4.2 SUMMARY OF THE ENTITY CLASSES ... 44
4.2.1 *Persons* ... 44
4.2.2 *Non-Person Living Subjects* ... 44
4.2.3 *Organizations* ... 44
4.2.4 *Entity_groups* ... 44
4.2.5 *Places* ... 44
4.2.6 *Materials* ... 44
4.2.7 *Roles* ... 44
4.2.8 *Entity ID* ... 45
4.2.9 *Person and Organization Names* ... 45
4.3 SUMMARY OF THE LOCATOR CLASSES ... 46
4.3.1 *Postal Locator* ... 46
4.3.2 *Tele Locator* ... 46
4.3.3 *Physical Locator* ... 46
4.3.4 *Entity-Locator Participation* ... 46

4.4 SUMMARY OF THE ACT CLASSES 46
4.4.1 Act *46*
4.4.2 Act ID *46*
4.4.3 Observation *47*
4.4.4 Public Health Cases *47*
4.4.5 Work-Ups *47*
4.4.6 Notifications *48*
4.4.7 Patient Encounters *48*
4.4.8 Interventions *48*
4.4.8.1 Substance Administrations 48
4.4.8.2 Procedures 48
4.4.9 Referrals *48*
4.4.10 Clinical Documents *48*
5 ATTRIBUTE TYPES 50
5.1 INTRODUCTION 50
5.2 NAMED ATTRIBUTES 50
5.3 EXTENSIBLE ATTRIBUTES 50
6 CODES FOR NAMED ATTRIBUTES 52
6.1 INTRODUCTION 52
6.2 STRUCTURAL CODES 52
6.2.1 class_cd *52*
6.2.2 cd *52*
6.2.3 Mood_cd *52*
6.2.4 Use_cd *53*
6.2.5 Type_cd *53*
6.3 FUNCTIONAL CODES 53
7 CODES FOR EXTENSIBLE ATTRIBUTES 55
7.1 INTRODUCTION 55
7.2 EXTENSIBLE ATTRIBUTES FOR OBSERVATIONS 55
8 SYSTEM REFERENCE TABLES (SRT) & TERMINOLOGY SERVICES 56
8.1 INTRODUCTION 56
8.2 SRT CLASSES 56
8.2.1 Codeset *56*
8.2.2 Code_relationship *56*
8.2.3 Code_Values *56*
8.2.4 Code_value_relationship *57*
8.2.5 Code_group *57*
8.2.6 Code_thesaurus *57*
9 NLDM SUPPORT FOR EXTENSIBILITY FEATURES 58
9.1 INTRODUCTION 58
10 RELATIONSHIPS OF THE NLDM TO NBS FUNCTIONALITY 58

10.1 INTRODUCTION 58
10.2 HIGH LEVEL OVERVIEW OF NBS FUNCTIONALITY 58
10.3 FUNDAMENTAL OPERATIONS ON NLDM CLASSES 61
11 RELATIONSHIP OF THE NLDM TO THE NBS MESSAGING SUB-SYSTEM.. 62
11.1 INTRODUCTION 62
11.2 MESSAGE-IN: RECEIVED MESSAGES 62
11.3 MESSAGE-OUT: OUT-BOUND MESSAGES 62
11.4 RELATIONSHIPS BETWEEN MESSAGE-IN AND MESSAGE-OUT. 62
12 EXAMPLES OF DATA REPRESENTATION IN THE NLDM 63
12.1 INTRODUCTION AND OBJECTIVES OF THE EXAMPLES 63
12.2 SCENARIO 1. REPRESENTING PERSONS WITH MULTIPLE NAMES, ROLES, AND LOCATORS 63
12.2.1 Scenario 1 Description 64
12.2.2 Scenario 1 Data Representation in NLDM classes 64
12.2.2.1 Entity 64
12.2.2.2 Person 65
12.2.2.3 Entity_ID 65
12.2.2.4 Person_Name 66
12.2.2.5 Locators 66
12.2.2.6 Entity_Locator_Participation. 67
12.2.2.7 Postal_Locator 67
12.2.2.8 Tele_Locator 67
12.2.2.9 Person Race-Ethnicity 68
12.3 SCENARIO 2: REPRESENTING ROLE RELATIONSHIPS AMONG ENTITIES 69
12.3.1 Scenario 2 Description 69
12.3.2 Scenario 2 Data Representation in NLDM classes 70
12.3.2.1 Entities 70
12.3.2.2 Persons 70
12.3.2.3 Person Names & Locators 71
12.3.2.4 Organizations 71
12.3.2.5 Non Person Living Subjects 71
12.3.2.6 Material 71
12.3.2.7 Entity_Group 72
12.3.2.8 Entity _ID 72
12.3.2.9 Roles 73
12.4 SCENARIO 3. REPRESENTING RELATIONSHIPS BETWEEN ENTITIES AND ACTS 75
12.4.1 Scenario 3 Description 75
12.4.2 Scenario 3 Data Representation in NLDM classes 75
12.4.2.1 Entity 76
12.4.2.2 Person 76
12.4.2.3 Organization 76
12.4.2.4 Material 77
12.4.2.5 Group 77
12.4.2.6 Place 77
12.4.2.7 Role 77

12.4.2.8 Act 78
12.4.2.9 Participation 79
12.4.2.10 Act_ID 80
12.4.2.11 Act_Relationship 81
12.4.2.12 Observation, Observation_Value, and Observation_Value_xxx 82
12.4.2.13 Patient Encounter 83
12.4.2.14 Work-Up 83
12.4.2.15 Notification 84
13 APPENDIX A – NEDSS PROGRAM AND NLDM GLOSSARY 86
14 APPENDIX B – CURRENTLY DEFINED CODED ATTRIBUTES AND VALUES 92

1

1 INTRODUCTION

1.1 Purpose of this Guide

This Guide provides an introductory overview of the CDC NEDSS Public Health Logical Data Model (NLDM), explains its key concepts and structures, and provides guidelines for its use. The complete, technical specification of the CDC NEDSS Logical Data Model (NLDM) is documented in the NLDM Class Model Diagram and the NLDM Data Dictionary. The class diagram provides a detailed specification of the NLDM's structure and content. The NLDM Data Dictionary provides a detailed textual description of each of the classes and attributes in the NLDM.

This guide may be read without access to the complete specification to gain a basic understanding of the model, but access to the class diagram in either the Rose file version or the excel image version would be helpful since the images copied into this document are not very easy to read given the number of classes.

This guide is intended for use by anyone planning to design a NEDSS compliant database, extend the NEDDS Base System to meet state specific requirements, design analysis or reporting tools for access to a NLDM compliant database, or design and implement an Element of the NEDSS Architecture. A basic understanding of relational or object oriented data modeling concepts is helpful but not required to follow the material in this Guide. The model uses Object Oriented (OO) design concepts (i.e. Class Models), but may be represented in a traditional entity-relational-diagram (ERD) as well. Section 3 provides a brief summary of the data modeling principles needed to understand the details in later sections.

The NLDM is one component of the NEDSS Architecture so the design of the NLDM reflects the requirements of this architecture. However, since it is sometimes difficult to make an architecture meaningful to readers, this Guide refers to an actual implementation of the architecture when describing the relation of the NLDM to a functioning system. These references are to the NEDSS Base System, but they apply to any system that is compliant with the NEDSS Architecture.
This Guide also serves as an introduction and index to the much more detailed NLDM Data Dictionary and Class Model (see References below).

1.2 Scope of this Guide

This Guide covers:

- The relationship of the NLDM to other NEDSS data models (section 2)
- The primary data modeling **concepts** and **approaches** used to develop the NLDM(section 3)
- A brief **description of each of the classes** (entities) in the model (section 4)
- The major **types of attributes** used in the model (section 5)
- The many types of **code systems** used (sections 6 & 7)
- The **System Reference Table (SRT**) sub-system, which implements the code systems (section 8)

- The **database extensibility features** which allow users to add attributes to the database and to NBS pages (screens) without programming or making structural changes to the physical database (section 9)
- A description of how the various classes in the data model relate to the **primary operations** (functions) provided by the NEDSS Base System applications (section 10)
- The relationship of the NLDM to the NEDSS Base System **messaging sub-system** (section 11)
- Detailed descriptions of examples of how data from various scenarios would be represented (stored) in the NLDM (section 12).

The major classes (classes are nearly synonymous with 'entities' and 'tables') are described briefly as well as the primary relationships among them. Specific attributes of classes are not described, except as they are used in the examples in Section 12. The attributes of each class are described in great detail in the NLDM Data Dictionary (See References). The scope is limited to the Logical level and does not address the Physical level because the physical levels differ depending on the particular DBMS used (e.g. Oracle vs. SQL/Server).

The physical design levels will be covered in a future NLDM Programmers' Guide.

1.3 Vocabulary: Definitions, Acronyms, and Abbreviations

1.3.1 Introduction

Vocabulary and Terminology are critical aspects of data modeling and data modeling has its own set of meanings for some commonly used terms. There are a large number of special terms used in this guide so this section provides a centralized set of definitions for those terms most important for an understanding of the NLDM. In addition, a glossary is provided in Appendix A that defines terms used in this guide as well as terms used in the overall NEDSS program.

The NEDSS Vocabulary Team is working continuously on defining the vocabulary for use in the NLDM and the NEDSS Base System. See Appendix B for a more complete summary of the current status of the results of the NEDSS Vocabulary Team on the vocabulary for coded attributes.

The vocabulary terms are organized in the following sections by related concepts. They are listed in the Glossary in alphabetic order.

1.3.2 NEDSS Program

The NEDSS Program is a collaboration of National, State, and Territorial agencies, along with allied professional organizations, to improve public health through the enhanced sharing and use of information. The NEDSS Program consists of many initiatives. These include working with various standards development organizations (SDO) to address data standards important to public health, developing and sharing best practices, sponsoring workshops and conferences where public health professionals can share best practice experiences, development of an architecture for IT support of public health, and development of a Reference Implementation of the architecture that may be optionally deployed and enhanced by public health agencies.

The two NEDSS technical initiatives that are most closely related to the NLDM are:

1. The NEDSS Architecture
2. The reference implementation of the NEDSS Architecture. This is a complete implementation of the NEDSS Architecture that is made available to the states as an operational system. The reference implementation is termed the NEDSS Base System (NBS).

The NLDM described in this Guide is a part of the NEDSS Architecture and is generally referred to as the NEDSS Logical Data Model (NLDM). One of the possible physical implementations of this NLDM is provided as part of the NEDSS Base System Option. The operation of the NBS depends on the use of specific implementations of the NLDM, commonly referred to as NBS Physical Data Models, (NPDM). Actual databases that implement the NPDMs are dependent on the actual DBMS used.

1.3.3 NEDSS Base System (NBS)

The CDC developed implementation of the NEDSS Architecture. See the NEDSS Technical Designers' Guide for detailed descriptions of the NBS Architecture, the system services and guidelines for extending the NBS to meet state and program specific requirements.

1.3.4 Program Area Module (PAM)

An extension to the NBS that provides functionality that is specific to one program area. That is, the NBS provides functionality that is common to all surveillance functions. In the future the NBS will be extended to include common public health functions in addition to surveillance. However, there is functionality that is specific to the investigation and treatment of specific diseases and there are sets of attributes (i.e. set of data elements) that are specific to particular diseases. Where these conditions exist, focused teams are formed to analyze and develop these disease specific functions as PAM extensions to the NBS.

1.3.5 Application Extension Domain

One aspect of the Extensibility feature of the NBS provides the capability to add fields to NBS pages and attributes to the database without the need for modifying NBS code or the underlying database tables. Specific additions are identified by 'application extension domains', (such as specific program areas, states, or state programs) so that the NBS session management software can properly invoke them.

1.3.6 Health Level 7 (HL7)

A standards development organization that develops standards for the exchange of health care information.

1.3.7 HL7 Reference Information Model (RIM)

A conceptual model developed by HL7 to facilitate common data model semantics, terminology, and structures

1.3.8 Entity and Class

These terms are common data modeling terms and may, for most purposes, be used synonymously. The term entity is used when one is using a relational data modeling approach and the term Class is used when one is using an Object Oriented approach. The term 'class' can

include behavior properties in addition to the structural ones, which are the properties that the Class concept have in common with the 'entity' concept. They refer to concepts or things that have properties or characteristics and that may have relationships to one another. For example, a person may be represented as an entity that has characteristics such as name and address. A person may also have relationships with other entities, such as organizations to which they belong or other persons with whom they have role-based relationships. For example, a person may be a patient of another person acting in the role of care provider. The characteristics of the entities are defined as 'attributes' and the values stored for the attributes are called 'attribute values'.

The HL7 RIM uses the Object Oriented modeling technique. However, they have named one of the classes 'Entity', which sometimes leads to confusion with the relational modeling concept of entity. This Guide remains consistent with the HL7 terminology as much as possible, so when the term Entity is used in the capitalized form it refers to a particular type of class, not to the data modeling concept of an 'entity'.

1.3.9 Attribute

An attribute defines one characteristic of an entity, such as the 'height' of a person. It is always used in the context of a particular class (entity)

1.3.10 Named Attribute

A named attribute is an attribute that has a pre-defined name that is used to identify a specific column in a database table. Values stored in that column always represent values of the named attribute.

1.3.11 Extensible Attributes

Extensible attributes are generic attributes that allow storing data on any property of the class (entity). This is accomplished by using two attributes to represent the property and its value rather than one as with Named Attributes. The first of the two attributes is the attribute that contains the name of the attribute as a value, the second attribute contains the actual value for the extensible attribute. These two attributes are commonly referred to as name-value pairs (or equivalently object-value-pairs). By allowing the name of the attribute to be variable, the Extensible Attribute may be used to represent and store the value of any characteristic of the class independent of the structure of the class and therefore may be added at any time.

One example of an Extensible Attribute is combination of 'cd' and 'obs_value_desc_txt' in the Observation Class. The attribute 'cd' contains the name component of the name-value pair and the attribute 'obs_value_desc_txt' contains the value part. See section 5.2 for more detail.

1.3.12 Extended Attributes

Extended Attributes are attributes that may be added to any class by the administrator of an NBS compliant system without changing the database structure or programming code. These attributes are also linked to newly defined fields on the application web pages. To the user of the system they are treated as Named Attributes but are actually stored in the database as Extensible Attributes. See section 9 for more details.

1.3.13 Table

The term 'table' is used to describe the physical implementation in a database of a class or entity.

1.3.14 Column

The term ‘column’ is used to represent the physical implementation in a database of an Attribute of a class or table.

1.3.15 Row

The term ‘Row’ refers to the set of data values stored for each column in a table for a particular instance of a class. Generally is synonymous with tuple. Sometimes the term row or tuple is used to refer to an actual instance of data for a row or tuple.

1.3.16 Tuple

The term, tuple, is used synonymously with the term row.

1.3.17 Instance

Instance has two somewhat related meanings in data modeling. One usage is a single occurrence of an entity, such as one specific person. It corresponds to one row in the table and represents all of the data stored for that class at that point in time. Another usage refers to one occurrence of a database. That is, all of the data for all tables in a database. For example, a state could have both test and operational instances of a database. In this Guide, unless the usage expressly states that the instance is of a database, the term ‘instance’ refers to an instance of a row in a table.

1.3.18 Occurrence

Occurrence is synonymous with ‘instance’

1.3.19 Name-Value Pair (NVP)

Name-Value Pair is a general term that describes a way of representing data where the name of the value is treated as data in the same way as the value is treated as data. The Name-Value-Pair technique is used in various situations, for example it is the technique used in XML. When used with relational technology, both the name and the value are stored as attributes of a class. For example, a person’s weight could be stored in a relational structure using an attribute named ‘weight’. The measurement of a person’s weight would be stored as the value of the attribute ‘weight’, e.g. ‘150’. Alternatively, the NVP approach would involve two attributes: one would contain the name of the attribute and the second would contain the value. Continuing the weight example, the name attribute in the table might be ‘observation type’ and the attribute value might be ‘observation value’. When data is entered for a persons’ weight they would enter ‘weight’ as the data value for ‘observation type’ and the actual measured weight for ‘observation value’.

NVP’s are used extensively in the NLDM and are described in more detail in section 5.2.

1.3.20 Use Code

A use-code is a special type of code that indicates the ‘usage’ of an attribute or set of attributes. For example, addresses may have ‘use-codes’ describing how the address is used, such as: home, employment, etc.

1.3.21 Unique Identifier (UID)

Unique identifiers (UID’s) are system-generated, specific numbers used internally to uniquely identify instances of entities. They are implemented as the primary key for the entity.

1.3.22 Standard Identifiers

Standard identifiers are identifiers that are assigned by widely accepted authorities, such as Social Security Numbers, Driver's License numbers, Medicaid numbers, etc.

1.3.23 Local Identifiers

Local identifiers are identifiers assigned by some authority that is not, necessarily, widely known. For example, an identifier assigned to persons by a local public health application would be considered a local identifier. Clearly, there isn't strict differentiation between Standard and Local identifiers. The practical distinction is that when one exchanges a Local Identifier one does not assume that the receiving person or system can associate it with the assigning authority. Common examples of local identifiers are the registration numbers assigned by the ADT systems in the various clinics of a large hospital. Resolution of local identifiers is usually accomplished using some form of Master Patient Index (MPI) or Patient Registry application. This basic MPI functionality is included as a service in the NBS.

1.3.24 Database Management System (DBMS)

A database management system is a software product that manages the storage of data on disks and provides a high-level programming interface to applications for accessing data stored within the DBMS. DBMSs also provide a set of tools to help manage and administer the database. When the database is based on relational technology the term Relational Database Management System (RDBMS) is sometime used.

1.3.25 Operational Data Store (ODS)

The operational data store is the set of tables that store the data during the period of time when the entities represented by the data are subject to some form of transactional activity. That is, when values for instances of the classes may be changed to support public health operational needs, for example values in a Case Report being updated during an investigation.

The term is typically used to distinguish this type of data store from a Data Warehouse where the data values are not changed and are used for analysis only. Once there is no longer a need to change values of data for a class, that data may be archived and deleted from the Operational Data Store since it is still available for analysis and reporting in the Data Warehouse.

1.3.26 Message-In (Msg-In)

The MSG-In is the set of classes into which data received via electronic messages or entered by partners using restricted use web forms is placed for validation before being stored in the ODS classes. For example, electronic lab results are received and translated from the message format into the relational format and stored in Msg-In. This data is then edited, validated, and checked for potential relationships to data already in the ODS before being moved from the Msg-In to the ODS.

1.3.27 Message-Out (Msg-Out)

The class where formatted messages are placed for routing by the message router to the named recipients.

1.3.28 NBS Data Mart

A set of classes whose structure is optimized to support data analysis as opposed to transactional activities. Changes to data in the ODS are copied to the NBS Data Mart on a periodic basis determined by the importance of the changes for inclusion in the analyses.

1.3.29 Schema

The term schema is used to describe the structure of a data model. It includes the classes (entities), the relationships between them, and sometimes the attributes for each class. It is most commonly used to describe the structure of a physical database, which is the way it is used in this Guide.

The term schema has recently been used for a similar construct when referring to the structure and organization of an XML message or structure. The construct is similar in that it defines for the XML content its structure and organization. In this Guide, the use of the term schema is limited to its use to describe database structures.

1.3.30 Acts

The RIM and NLDM contain two major super-classes: Entities and Acts. Acts represent health related activities that (usually) have a defined beginning and end time with corresponding duration.

1.3.31 Work-Up

The term Work-Up is used to capture the information that a public health worker records about the **process** of their investigation as well as to link to the observations they make about the subject of the investigation. For example, work queues, requests for information, appointments, notes, etc.

1.3.32 Investigation

The term 'investigation' is used to refer to the specific actions during a work-up where the investigator assesses the collection of information against one or more case definitions to determine whether the set of data meets a case definition.

1.3.33 Jurisdiction

The term jurisdiction is used to represent the organizations or groups within a public health agency that have responsibility for public health of a set of conditions within a specified geographical area. It is a key component of the NBS security sub-system.

1.3.34 Public Health Directory (PHDIR)

The PHDIR is a form of 'white pages' that allows a user to lookup the contact information for a public health worker to identify how to communicate public health information to them. For example, the NBS uses the PHDIR to find the information needs of a recipient to route a notification to them.

1.4 NLDM Versions and its Future Evolution

This is the first released version of the NLDM. This version focuses on the data needed to support the functionality of public health surveillance, but more specifically the application

functionality provided in the beta release of the NBS implementation of the NEDSS Architecture. The NLDM will be continuously enhanced both to address a wider range of public health needs and to remain consistent with evolutions of the HL7 RIM. These enhancements will be developed for review by the public health community and published periodically along with enhanced releases of the NBS. This version of the NLDM is focused primarily on surveillance of communicable diseases. Subsequent versions will add coverage of public health conditions in addition to communicable diseases, including: case prevention, case management, vital statistics, and other domains of public health.

The current version of the NLDM is version 1.07C. The numeric portion of the version identifier refers to the latest version of the HL7 RIM with which it is consistent. The alphabetic portion identifies successive versions with NEDSS specific extensions.

1.5 References

Each of the following references is available on the NEDSS Web Site.

For an overview of the functional and technical features of the NEDSS Base System:

- "The Base System: Implementation of NEDSS Standards and an Option for Use". This is a presentation given at the April, 2001 NEDSS Stakeholders' Meeting

For an overview of the technical architecture of the NEDSS Base System:

- "NEDSS Base System Design: Current Status". This is a brief, summary presentation given at the June 2001 CSTE Conference.
- "NEDSS Base System Technical Designers' Guide." This is a design reference that provides the detail necessary for a designer to understand the architecture and services in the NBS and to design their own extensions to it. (This document is currently under development.)

For details on the Public Health Conceptual Data Model and an extensive set of related references:

- "Public Health Conceptual Data Model, Premier Edition"

For the detailed description of the NLDM

- "NEDSS Logical Data Model Data Dictionary"

For a detailed diagram of the NLDM Class structure

- "NEDSS Logical Data Model"

1.6 Major components of the NLDM

1.6.1 Introduction

There are four distinct sets of classes in the NLDM that support four distinct aspects of NBS functionality:

1. Operational Data Store (ODS)
2. Message In Data Store (MSG-IN)
3. Message Out Data Store (MSG-OUT)
4. Optionally, an NBS Data Mart

1.6.2 ODS

The classes in the ODS hold the data that are used to support the day to day application functionality of an NBS like system, such as managing person demographic data, conducting investigations, etc.

1.6.3 Message-In (MSG-In)

The classes in the MSG-IN hold data that have been received via electronic messages and processed by the Messaging Sub-System before being validated and added to the ODS, such as electronically received lab results. Equivalently, data may be entered into the MSG-In tables by applications that have less security or trust than the NBS or NBS equivalent application, such as the restricted use Web applications provided by the NBS for use by public health partners that are not granted access to the ODS but are allowed to submit data to the system in a way equivalent to submitting it via an electronic message. For example, the NBS includes a web application that allows the entry and submission of lab results but does not allow viewing other data in the ODS, such as person information.

1.6.4 Message-Out (MSG-Out)

The classes in the MSG-OUT hold data that have been previously formed into a message structure that is part of the public health message family to be sent to a public health partner using the Messaging Sub-System, such as a public health notification to be sent to the CDC.

1.6.5 NBS Data Mart

The classes in the NBS Data Mart store basically the same information as in the ODS but in a structure that better supports analysis and reporting. The NBS Data Mart may be physical or virtual, that is the NBS Data Mart may be implemented as a set of Logical Views if the analysis workload does not warrant the creation and maintenance of a second database in addition to the ODS.

The following diagrams provide a high level overview of the NLDM components. Figure 1.1 shows the logical relationships between these major components and their relationship to the states' existing databases. Sections 1.6.1 through 1.6.4 describe these components in more detail. Section 1.7 then describes the typical flow of data between these major components.

1.6.6 Relationships of major NLDM components

-

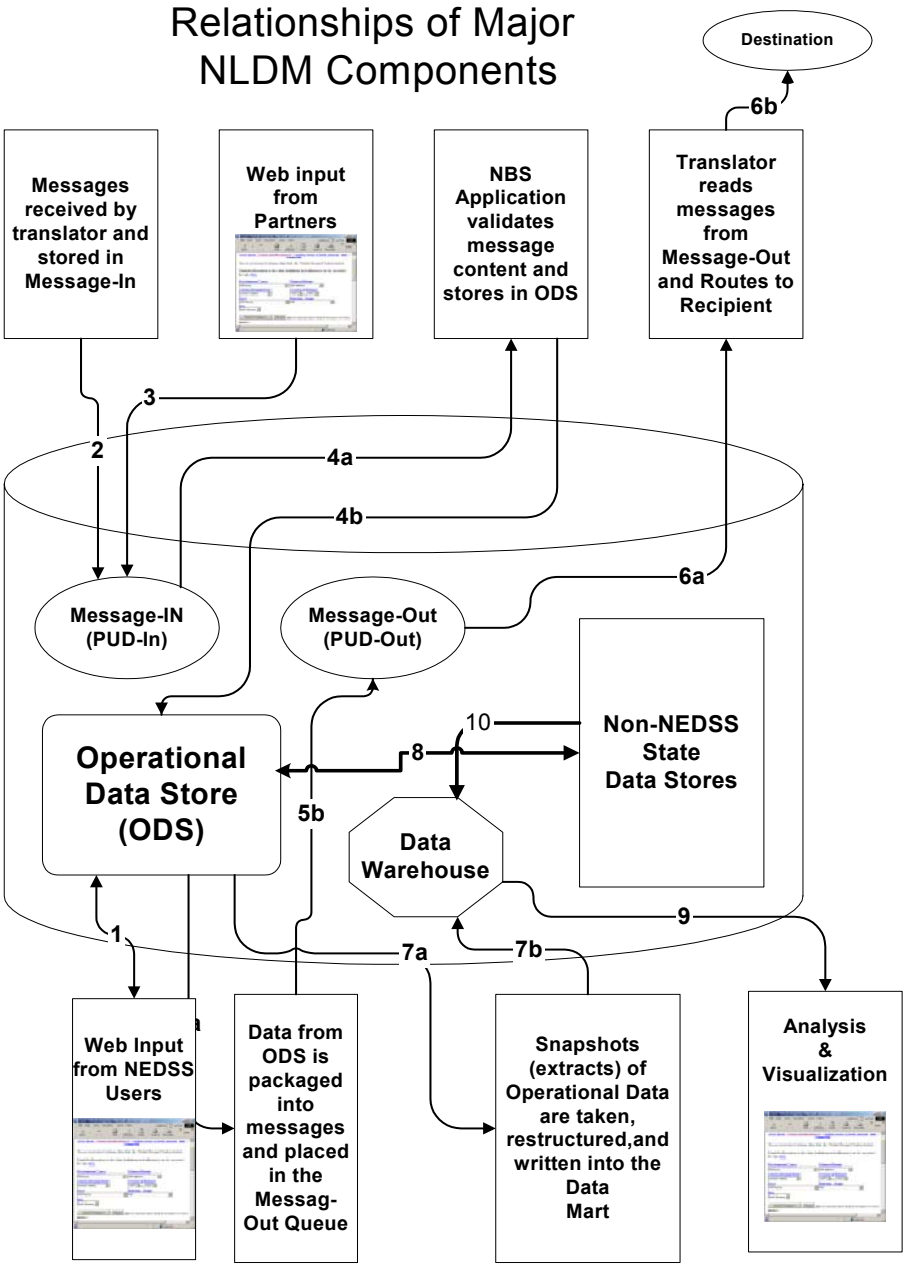


Figure 1.1 Major Components of the NLDM

1.6.7 Message-In Data Store

The Message-In component holds data that are received as electronic messages or are entered by Internet users who are not authorized to submit data directly into the ODS, such as laboratory technicians from partner labs. Electronic messages, such as lab results in HL7 ORU message format, are received by the router/translator, parsed, and written into the Message-In data store. The structure of the Message-In data store is similar to that of the ODS; it is maintained as a separate set of classes to ensure that data received by any means other than from NBS applications are not written directly to the ODS.

Path 2 in figure 1.1 illustrates the entry of data into the Message-In store from a received message. Path 3 illustrates the entry of data by users over the Internet that do not have access to the ODS.

1.6.8 Operational Data Store

The operational data store (ODS) is the main component in which data is stored and managed as public health workers perform their work. There are three main ways in which data are entered into the ODS:

- It can be read from the Message-In data store
- It can be entered directly by NBS applications that interact with users via the Internet
- It can be entered by NBS applications that retrieve it from other state databases.

Data that has been entered into the Message-In store need to be validated and assigned to a jurisdiction and program area before they are written into the ODS, this is shown in paths 4a and 4b.

Data is also written directly into the ODS by NBS applications that provide interactive sessions with users, as illustrated in path 1.

The third way data may be entered is directly from one or more of the state's other databases. This is simplest when the two databases are managed by the same DBMS as illustrated by path 8, though it may be done between dissimilar DBMS's using additional tools or custom applications.

1.6.9 Message-Out Data Store

Messages may be constructed from data in the ODS and sent by the translator/router tool, as illustrated in path 6. The most common example of this is the creation of Public Health Notification (PHN) messages for transmission to the CDC. The state may also place messages in this Queue for transmission by the Messaging router.

1.6.10 NBS Data Mart

The NBS Data Mart is a database specialized to support analysis and reporting. It is not used to manage data required to process updates or changes. Data management operations are supported in the ODS. Data is read from the ODS, restructured, and written to the NBS Data Mart. It is structured in such a way that analysis tools, such as SAS, can operate more efficiently and users

that have SAS datasets can access data in a more understandable structure Typically, the data is stored in the NBS Data Mart for longer periods of time than it is kept in the ODS

1.7 Representative Flow of Data through the NLDM

The following scenario (paragraphs 1.6.1- 1.6.7) shows a hypothetical flow of data through the NLDM to further clarify both the process and the purposes of the major components. The scenario is not necessarily valid because some intermediate steps have been left out that would be necessary to ensure consistency of the data.

Also, note that in this example we are treating the NLDM as if it were an actual database rather than a logical model. For the purposes of this example, one can assume that the PDM is identical to the NLDM. The use of eLink for data migration is not its intended primary use. Rather, its use in this way is included here to give an example of how its strengths can be taken advantage of.

1.7.1 Initial Data Migration of State Demographic data into the NLDM

We assume for this case that the DBMS used for the previous state system was one that is incompatible with the target DBMS. Rather than performing DBMS level conversions of the data from the database to be migrated, it is extracted and written to a comma-delimited ASCII file or HL7 message, depending on the capabilities of the program used to extract and send the data. The state uses its existing messaging product to read this comma-delimited file, create comma-delimited messages, and forward these messages to the NBS eLink translator.

The developer of the data migration software designs a mapping program for the eLink translator that maps the contents of the comma-delimited message to the classes in Message-In (Path 2). This translation involves changes to the codes used for the attributes and distribution of attributes to classes in a different way than was present in the source system. The eLink translation program accomplishes these translations.

This process of migrating data from one database to another using the messaging system is not limited to one-time data migrations. If the source is to be kept current and actively used, then data entered into the source system may be sent to the NBS system on a periodic basis using the messaging sub-system. However, see section 1.7.7 below for a possible alternative.

1.7.2 Conversion of Data from the Message-In to the Operational Data Store

Data received in messages do not have potential relationships established with data already in the ODS. Path 3 in the data flow shows how a NBS service accomplishes this translations, relates data in the incoming messages to other data already existing in the ODS, and enforces business rules defined for data in the ODS. Since in this example this is the first set of data being entered into the ODS, the only functions needed in this case are the application of business rules on the incoming data ensuring data consistency and validity. One step in this process is creating an entry in the Entity_id class that contains the primary identifiers for each record as used in the original system. These entries in the Entity_id class will be subsequently used for master patient indexing functions by the NBS. At this point the ODS now has the person information loaded

and the corresponding records have been removed from the Message-In classes. As with all database operations in the Message-In and ODS, a transaction log is kept of each such operation.

1.7.3 Registry Manager Review of Migrated Person Data

The person data in the NLDM may now be viewed using one of the web-enabled applications, such as the NBS Core Demographic Module's View Person function. At this time the user can check the accuracy and completeness of the loading process (Path 1). At this time the user can examine the system for potential duplicate records and merge confirmed duplicates using NBS applications.

1.7.4 Ongoing Receipt of Electronic Laboratory Results

Laboratory results may be entered into the system in at least three ways:

1. Electronically received results are loaded into the Message-In. (Path 2)
2. Results may be entered into the Message-In by labs that do not have access to the full NBS via web pages. (Path 3)
3. Results may be entered by public health workers directly into the ODS (Path 1) using standard NBS applications.

Data received electronically from external systems needs to be translated and mapped to the NLDM. Data input by users who do not have full access to the ODS also need to be validated before being moved to the ODS.

Besides applying business rules to the data as part of moving it from the message area to the ODS (path 4) these lab result records need to be associated or linked to other records in the ODS, such as the patient, the ordering care provider, the laboratory organization performing the tests, the state jurisdiction and program area that would be responsible for investigating the results, etc. If, for example, a patient record does not exist for the patient whose lab results are being entered, then a patient record is created. All of these operations are performed as part of Paths 4a & 4b.

1.7.5 Review of Case Data

As observations, such as lab results, are added to the system and associated with patients, the public health investigators can review the data (Path 1) and determine whether any of the observations suggest that a Case Investigation should be initiated. If not, the user can assess what data is needed to confirm or negate the condition as meeting a case definition and seek to collect the additional data. If so, (again using Path 1) the investigator creates a Work-up (see section 4.3.4) and evaluates whether the set of observations for a patient meet a case definition. If they do, then the investigator initiates a Notification (see section 4.3.5). The process involves the creation of a Case Notification message to be sent to the selected recipient (potentially another state PH agency or the CDC). The creation of this message involves the formatting of an XML message and the placing of this message in Message-Out (Path 5) for routing by the eLink translator and router (Path 6) to the recipients.

1.7.6 Loading of the NBS Data Mart for Analysis and Reporting

As is described throughout this Guide, the structure of the NLDM is highly optimized to support the capture of public health data and the investigation and case management activities. It is not optimized for analysis and reporting, though it may be used for it. Also, much of the data in the ODS is of a transient nature to support the investigation process, but is not of use for epidemiological analysis purposes. For these reasons the NLDM includes an optional NBS Data Mart. Data from the ODS is periodically extracted from the ODS, restructured to support analysis and reporting and written to the NBS Data Mart, (Path 7) for analysis and reporting. Data may be deleted from the ODS when it is no longer needed to support investigations, but data is normally left in the NBS Data Mart for long periods of time to support trend analysis and various research needs. Data in the NBS Data Mart may also be de-identified as needed to meet confidentiality requirements while still supporting epidemiological analysis and reporting, (Path 9).

1.7.7 On-going Synchronization of an NBS Database with other State Databases.

The example in section 1.7.1 demonstrated how the messaging sub-system could be used to move data into the NLDM database from a separate database where the DBMSs are dissimilar. Where the DBMS's are the same (e.g. both are Oracle), there are more efficient procedures available. When the 'Non-NEDSS State Data Stores' in the above diagram use the same DBMS, as the NLDM implementation, information from the two databases can be synchronized using standard DBMS stored procedures (Path 8). This is a highly effective way to make data entered into either database automatically available to applications accessing whichever database they were designed to access. The state can, of course, choose to implement the data transfer in one direction only.

1.8 Overview of the Primary NLDM Design Principles

The primary design goals for the NLDM are to:

1. Accommodate the current and future needs for the on-line entry of both independent observations as well as person-centric information relevant to public health, and eventually other data as well
2. Support the many types of relationships between the person and other entities and activities that that are discovered throughout the investigation and analysis processes
3. Provide for the electronic capture of health data from the site of origin that might indicate the occurrence of an event of interest to public health or to provide supporting evidence to a public health investigation (e.g. Birth records, discharge summaries, etc.)
4. Support the easy and rapid extension of the NLDM to support program and state specific data needs without having to make structural changes to the database
5. Support accurate and rapid sharing of data through the reliance on and adherence to shared data standards based primarily on national health care and other industry standards
6. Support the many types of unique state-specific requirements, such as unique identifiers and specialized codes
7. Maintain data quality by validating data received electronically and stored in the Message-In tables before moving it to the ODS, as well as data quality in the ODS using data quality consistency routines

8. Support full trace ability of changes to data by storing all changes to data in history tables
9. Support the ability to back out merged data so that the original records can be restored

To meet the goals associated with the maximum sharing of information (i.e. Goals 1, 2, 3, & 5), the primary structural code sets are based on the HL7 standards. This includes the HL7 Reference Information Model (RIM) as well as the HL7 supported code sets. Functional code sets (e.g. lab test types) are based on the many code sets established and maintained by various standards development organizations or locally defined.

1.9 Basic Structure of the NLDM

1.9.1 Overview

There are a large number of classes in the NLDM, but these can be organized into four main groups, or subject-areas: Entities, Locators, Participations between Entities and Acts, and Acts.

The Entity classes represent data about persistent subjects, e.g. persons, places, and things; whereas, Acts represent events or processes that are related to Entities, e.g. lab results, clinical observations, interview results, etc.

Each of these major subject areas are shown in figures 1.2 through 1.7 below for general use in this Guide and are described in more detail in section 4. A complete description of each class, its attributes, and relevant codes is provided in the NLDM Data Dictionary and the NLDM Class Diagrams.

1.9.2 Entity Classes

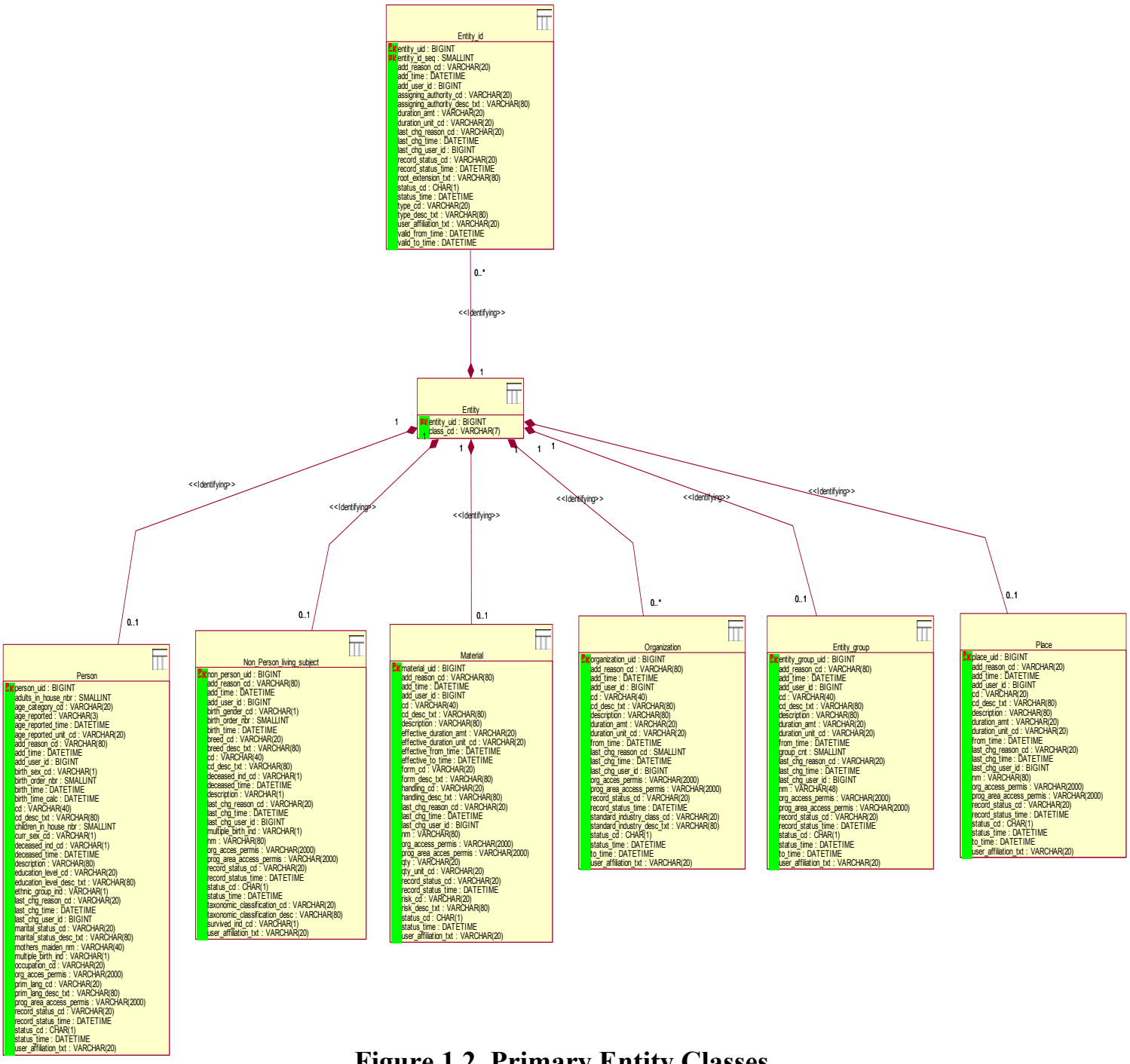


Figure 1.2 Primary Entity Classes

1.9.3 Locator Classes

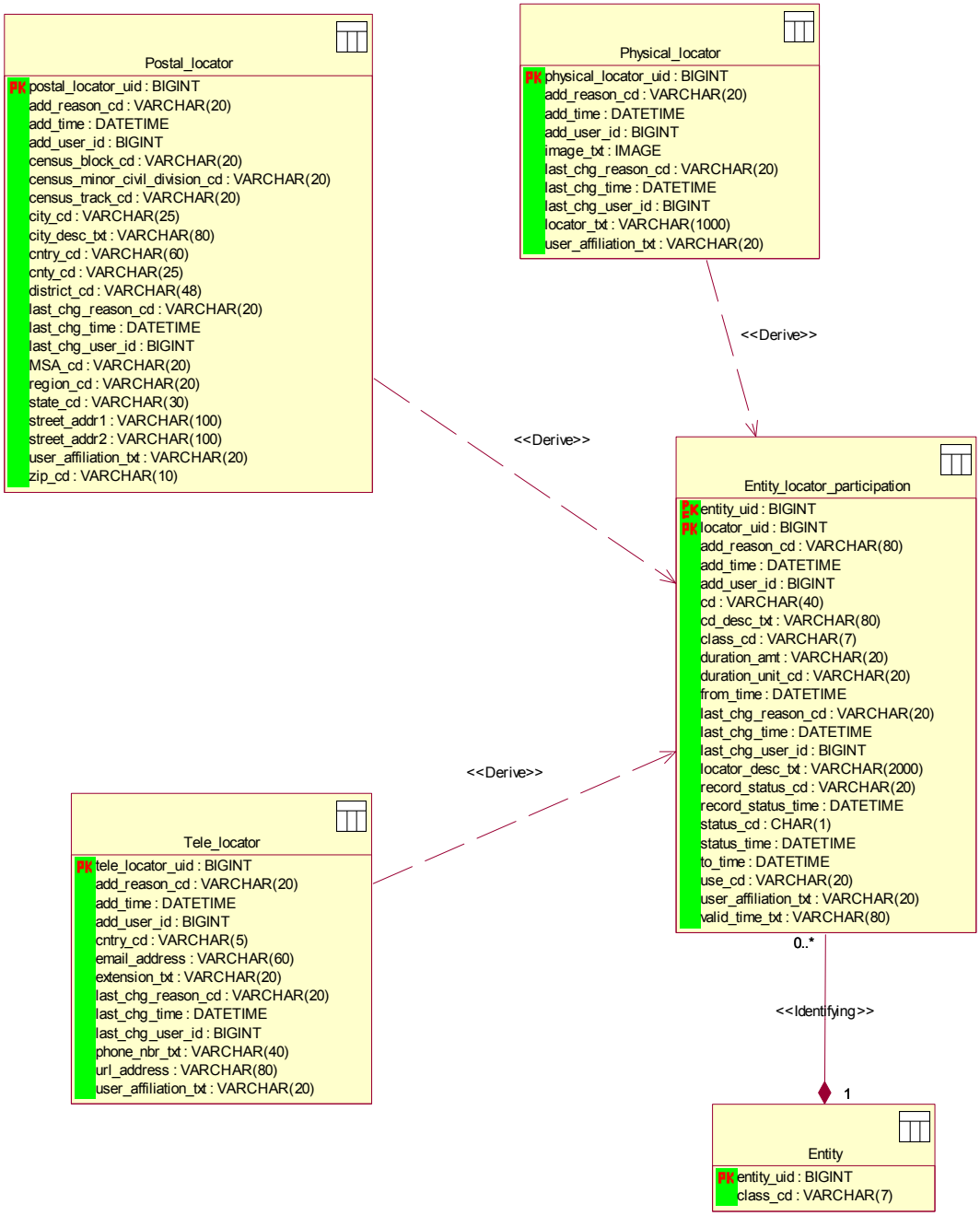


Figure 1.3 Primary Locator Classes

1.9.4 Entity-Act-Participation Classes

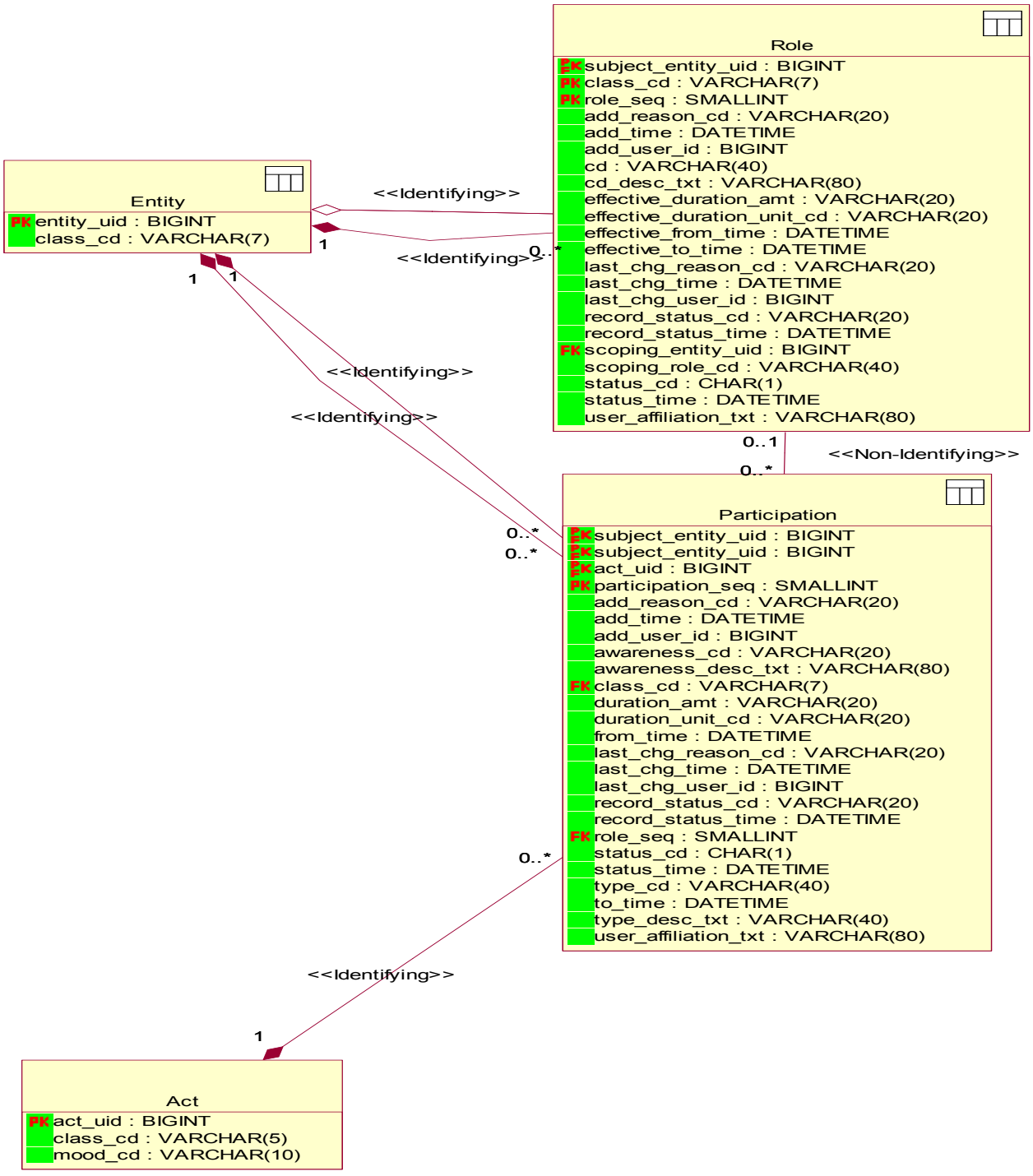


Figure 1.4 Entity Act Participation Classes

1.9.5 Act Classes

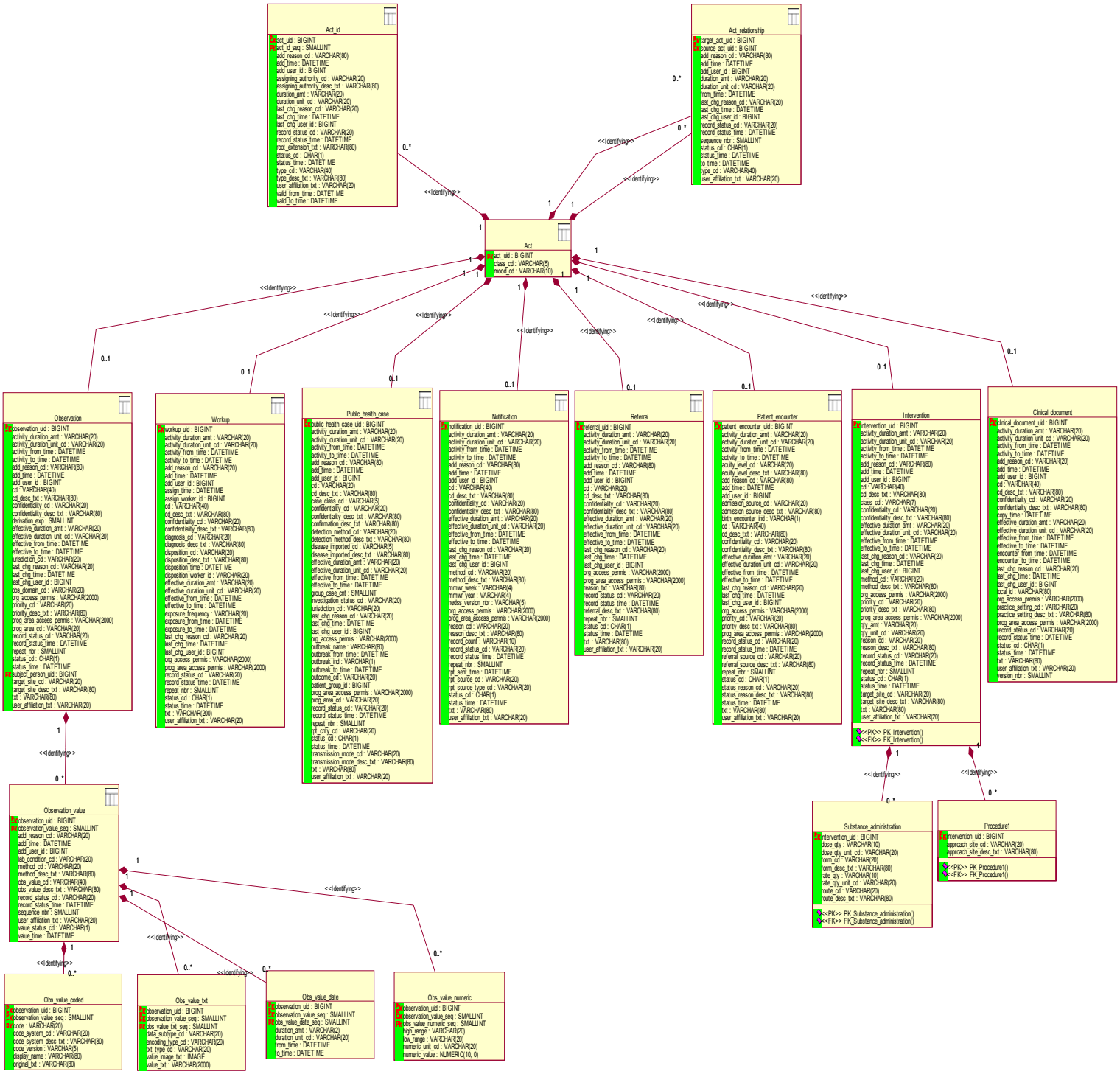


Figure 1.5 Act Classes

1.9.6 Observation Classes

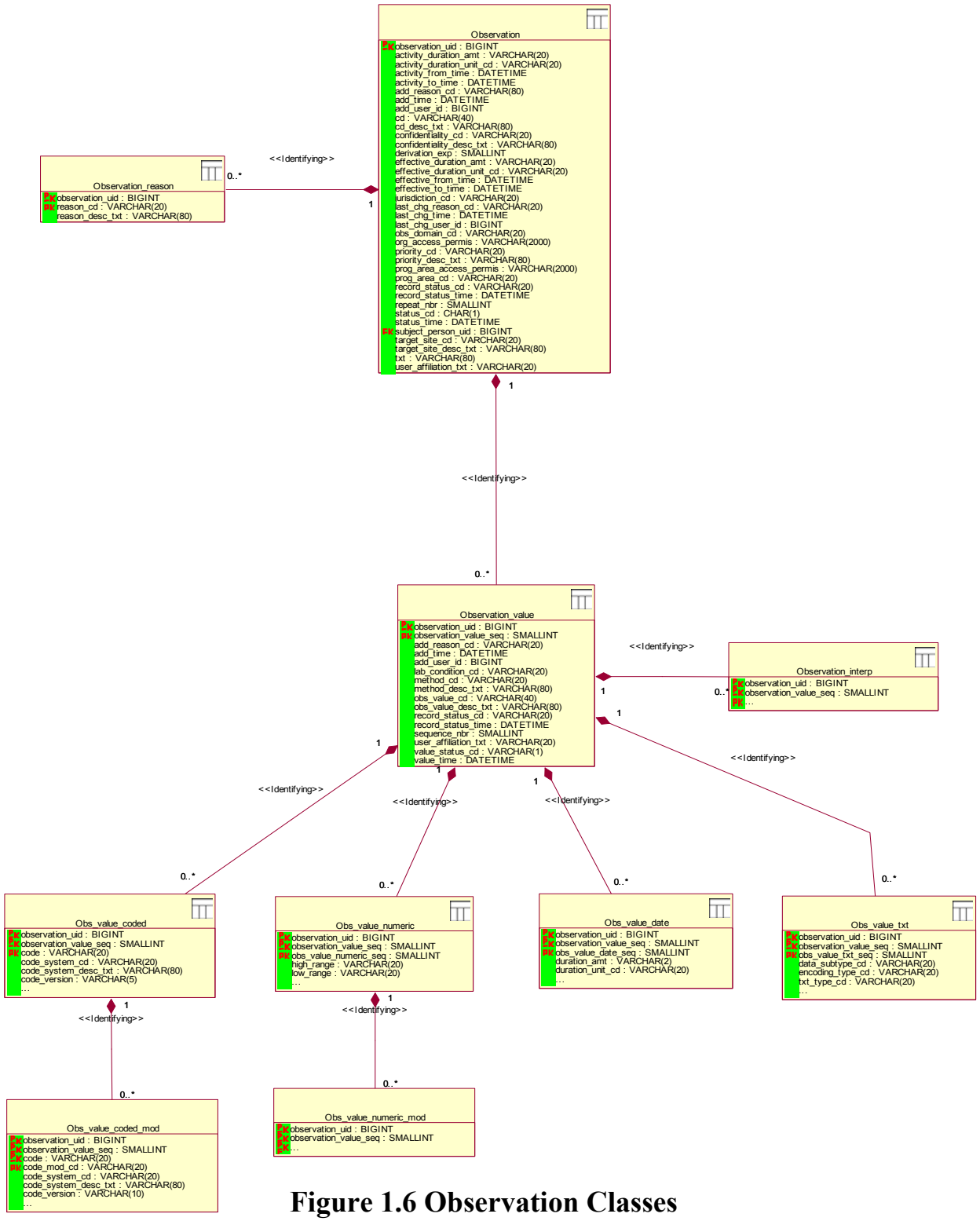


Figure 1.6 Observation Classes

1.9.7 System Reference Code classes

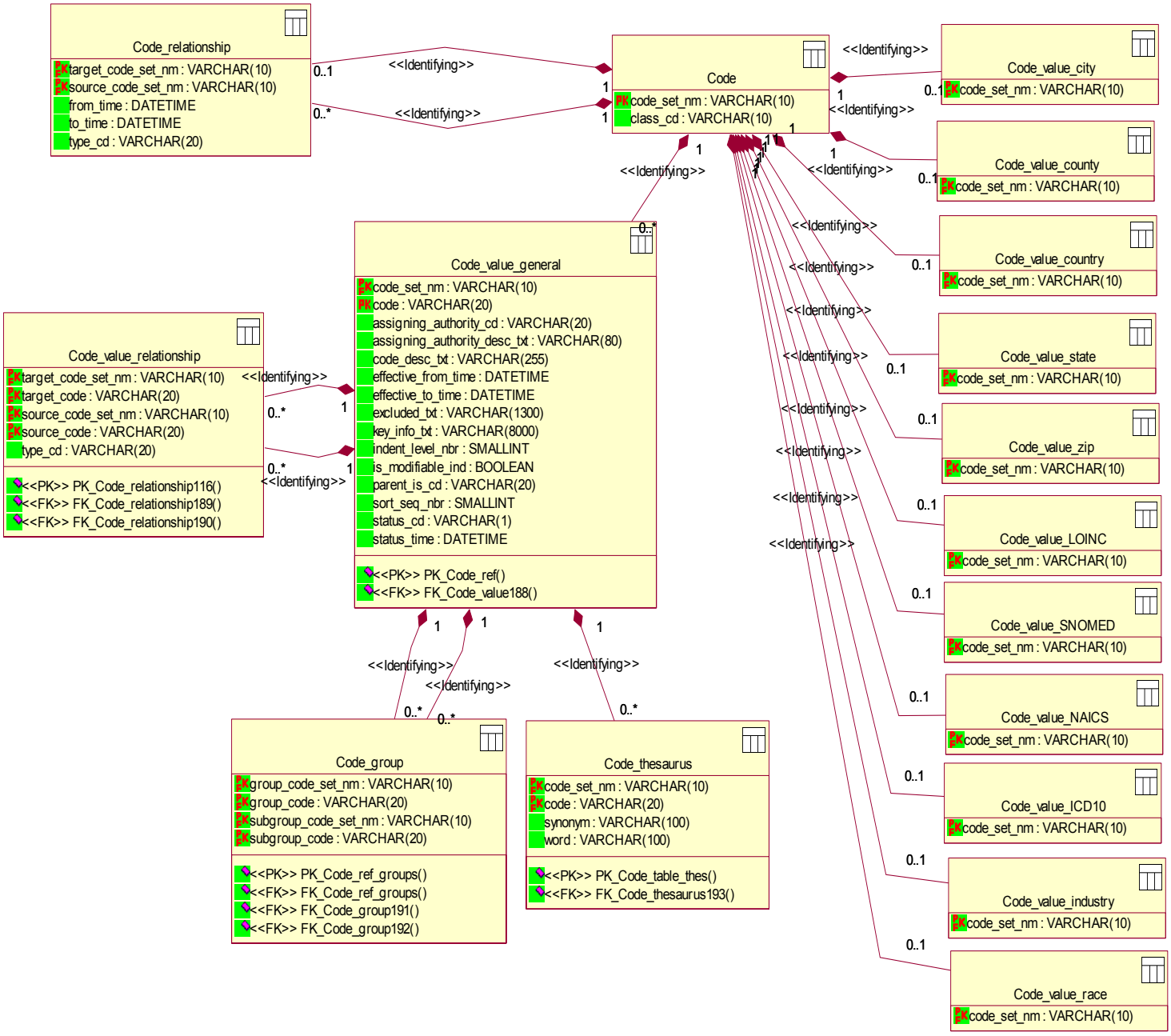


Figure 1.7 System Reference Code Classes

1.10 Codes

To support the flexibility, extensibility, and shared and common usage goals (i.e. primarily goals 4, 5, & 6 in section 1.8.1), the NLDM makes extensive use of the following types of codes:

1. Functional code for identifying standard or local codes for health related data
2. Structural codes for identifying relationships among the components of the model
3. Attribute Name Identifier codes for supplying the name part of data represented as name-value pairs (see section 1.2.21, Name-Value-Pairs
4. Attribute Name Identifiers for extending the concept of name-value-pairs by providing the capability for the states to extend the NBS by adding their own name-value pairs for any class.

1.10.1 Functional codes

Codes are primarily an alternative to using free text to describe an attribute or characteristic of a class. The advantages are that the codes have a shared, agreed upon meaning promoting the unambiguous exchange of data between systems and the shared interpretation of that data by users. The use of codes also facilitates the validation of data by the system when entered by the user. The difficulties are reaching agreement on the meaning of the codes across large communities of users and the migration of systems based either on proprietary (i.e. unique) codes or on largely free text.

The NLDM uses national and international codes wherever possible. The CDC is working with many other private and public health care agencies to extend and augment the code sets as needed to meet the public health data requirements.

1.10.2 Structural codes

Like the HL7 RIM, the NLDM makes extensive use of 'structural codes' to support the object oriented principle of inheritance. That is, to make each of the classes more useful by allowing each class to have one or more types codes rather than defining a wholly new class for minor variations in the properties of persons or things to be represented. The NLDM supports at least two levels of structural codes for each class. Inheritance provides a more compact and easy way to understand the data model by allowing sub-classes to possess all of the attributes and relationships of their super-class(es). In relational modeling this is usually defined as sub-typing.

1.10.3 Attribute Name Identifiers

Attribute name identifier codes provide the capability for states to add attributes to classes without changing the structure of the database. This is accomplished using the 'name-value-pair' technique.. The user specifies a code that defines the name of the attribute they want to add to the class and the system uses this code as an Attribute-name whenever a reference to the attribute's value is made. The NLDM uses Attribute Name Identifier codes in some of the classes, for example, 'cd' in the Observation class is an attribute name identifier code. The value

stored in the attribute identifier code represents the name of the attribute, for example the code CBC (Complete Blood Count) could be stored as a value in the 'cd' attribute name identifier. See section 7 for details on how Attribute Name Identifiers can be used to extend the data that can be collected, stored, and analyzed in the NBS.

1.10.4 Attribute Name Identifier Codes

Just as values for attributes can be stored either as free text or as codes, values for Attribute Name Identifiers can be stored as codes. In the example, above 'CBC' is an Attribute Name Identifier code for the attribute name identifier of 'cd'. For another example, the attribute 'preferred sex' could be stored either as a named attribute in a table with values stored for that attribute in each row of the table or using the name-value pair approach the name of the attribute could be represented as a Attribute Name Identifier and the value for the attribute could be stored as the value part of the name-value-pair, e.g. 'Attribute Name Identifier value'

The same technical approach is also used to address the problem that there are a potentially huge number of attributes associated with typical health care observation reports, e.g. lab results and clinical reports. Typically, a report entity would be defined with all potential observations represented as attributes while in most cases only a few actual observations are made out of the possible set. The NLDM uses the name-value pair technique to solve this problem as well as the need to allow for the representation of observation types that have not been predefined. See section 5.3 below for more details on this feature also as well as some specific examples of how this technique is used.

2 RELATIONSHIP OF THE NLDM TO OTHER DATA MODELS

2.1 Introduction

The NLDM is a member of a family of related data models. The following sections briefly describe each of these. More detailed information on each can be found in some of the documents described in the References section.

2.2 Relationship to HL7 Reference Information Model (RIM)

The HL7 RIM is a conceptual model that addresses many areas of health care. The NLDM is a logical model that is: 1) consistent with the HL7 RIM, 2) includes only those classes that are needed for public health, and 3) extends the RIM with public health classes and attributes that are not in the RIM. Many of these extensions will be proposed for inclusion in future versions of the RIM.

The Public Health Conceptual Data Model (PHCDM) and NLDM are both derived from the HL7 RIM. Not all properties (e.g. classes and attributes) of the RIM are included in the PHCDM or the NLDM since many of the RIM properties are not commonly used in public health, such as Admission/Discharge/Transfer or payment claims. The RIM and PHCDM are both conceptual models, whereas the NLDM is a logical model. The primary differences between the conceptual and logical models is that the NLDM is more specific and includes a complete set of attributes and corresponding data types for each class. For example, a primary use of the PHCDM is to assess whether a particular public health concept can be represented in the data model, whereas the NLDM specifies how it would be represented.

2.3 Relationship to the Public Health Conceptual Data Model (PHCDM)

The PHCDM was designed prior to the NLDM and is also based on the HL7 RIM. It shares with the NLDM the three properties itemized in section 2.2. Being a conceptual model, the PHCDM uses the more abstract data types used in the RIM as opposed to the more detailed and specific data types used in the NLDM. The PHCDM was first published in July, 2000. As of the publication of version 1.07C of the NLDM, the PHCDM had not yet been officially updated, consequently, there are classes and attributes in the current version of the NLDM that do not appear in Premier Edition of the PHCDM. The PHCDM is being updated and a future version will be published.

2.4 Relationship to the NBS Information Models

The NBS information models define the display properties of data in the NLDM as well as some of the business rules associated with each of the data elements. Since this information is primarily specific to an implementation of the NEDSS Architecture, the NBS information models are not addressed further in this Guide.

2.5 Relationship to state Logical and Physical Models

The NLDM is designed to be implemented as a schema within the state's existing database environment. That is, the physical tables based on the NLDM classes can be implemented as

tables within an existing set of tables within a state's database schema. This approach facilitates the sharing of data both between applications and between the database tables directly since the DBMS built-in utilities have the schema information needed to operate directly on both sets of tables. However, a state can elect to install the NBS database as a separate schema or a separate database instance if they choose. For example, if a state uses the Oracle DBMS, they can include the NBS schema (table definitions) within the same set of definitions for their existing state databases, or they can create a separate schema for the NBS database and run it in the same Oracle instance, or they can create the NBS database as a separate schema and run it in a separate instance of the Oracle DBMS.

2.6 Relationship of the NLDM to separate MPI Databases

Master Patient Index, MPI, functionality is an important component of any public health information system and the NLDM is designed to be inter-operable with external MPI products and implementations, as well as provide basic MPI functionality services. MPI systems generally provide means to identify patients using multiple demographic attributes and various types of identifiers, provide ways to match patient records given approximate values for some subset of the potential attributes, and provide a cross index of patients based on their identifiers in different systems. These features are included in the NBS, but the NLDM also provides the capability to interface to a states' existing MPI system if already present.

The preferred approach to integrating external MPI's has three levels:

1. Implement two-way synchronization of the contents of the states' MPI and an implementation of the NLDM structures so that the data does not need to be shared via programmatic interfaces.
2. Provide access to the MPI via an application programming interface that can be implemented in the NBS modules that manage an implementation of the NLDM
3. Provide a context management structure (that is a structure that contains values for entity attributes, such as a patient's identifiers used in two different systems, e.g., a state's registry system and the NBS). This allows passing data from the MPI to the NBS applications managing an implementation of the NLDM via a variable string appended to the HTML message created by the MPI or other application using the MPI.

Approach one is important so that data contained in the MPI can be added to the NBS Core Demographic Data, CDD, structures without re-entry and, thus, be made available to the NBS modules without needing to retrieve it each time they are needed from the MPI and vice versa.

Approach two is important so that the user is not forced to switch between the NBS user interface and the MPI interface each time a search or match is attempted.

Approach three complements approach 2 and can be used as an alternative when the MPI is accessed via a web browser interface or windows interface that supports the HL7 Context Management Architecture (CMA) standard. It allows the passing of data between the MPI and the NBS either as parameters of the HTML message if a browser is used or via a broker service if a windows interface is used.

3 MAIN DATA MODELING PRINCIPALS FOLLOWED IN THE NLDM

3.1 Introduction

The NLDM class diagrams and Data Dictionary provide detailed specifications of the results of the design process, that is the 'what' that resulted from the design effort. To support a better understanding of the model it is important to understand the 'how' and 'why' of the many design decisions that went into achieving the final design. This section provides a quick overview of those design principals.

The following sections describe the approaches taken to the main design aspects of any logical data model:

- Subject Areas
- Types of Classes used
- Types of Relationships among Classes
- The Types of Keys used
- The use of History classes to maintain access to changed values of any class
- The Merge and De-Merge capabilities
- The role based security applied to objects (entity instances/table rows)
- The system managed attributes of each class

3.2 Subject Areas

Subject areas are logical groupings of components of the total NLDM used primarily to simplify viewing and describing the model. The grouping of any particular sub-set of components of the model into a subject area is arbitrary and is largely determined by the particular needs of the user. Subject areas do not exist in a physical sense in the database, they are strictly a logical way of viewing parts of the database on a diagram so that topics specific to those areas can be discussed without having the diagram complicated by all of the other components of the model.

The following subject areas are used in this Guide:

1. Entities
2. Locators
3. Acts
4. Participations
5. Observations
6. System Reference Codes and System tables

(Note: The following lists of classes within each subject area do not include the history tables.)

3.2.1 Entities

The Entities subject area includes the following NLDM classes:

- Entity
- Entity_id
- Entity_id_merge_history

- Person
- Person Name
- Person Ethnic Group Modifier
- Person Race
- Person Race Modifier
- Non-Person Living Subject
- Organization
- Organization Name
- Entity_Group
- Place
- Material
- Role

3.2.2 Locators

The Locators subject area includes the following classes:

- Entity-Locator Participation
- Postal Locator
- Tele Locator
- Physical Locator

3.2.3 Entity Act Participations

The Entity Act Participation subject area includes the following classes:

- Entity
- Role
- Participation
- Act

3.2.4 Acts

This subject area includes the following activities, i.e. Sub-classes of the Act super-class:

- Observation
- Public Health Case
- Work-Up
- Notification
- Patient-Encounter
- Intervention
- Substance administration (sub-class of Intervention)
- Procedure (sub-class of Intervention)
- Referral
- Clinical Document

3.2.5 Observations

While the observation class is part of the Acts subject area it is also frequently discussed as a subject area by itself because it has a rather complex set of sub-classes, uses a large number of both structural and functional codes, and is the most heavily used class in the NLDM.

The Observations subject area includes the following classes:

- Observation
- Observation_Value
- Observation_Interp
- Observation_reason
- Obs_value_coded
- Obs_value_coded_mod
- Obs_value_txt
- Obs_value_date
- Obs_value_numeric
- Obs_value_numeric_mod

3.2.6 System Reference Codes & System Tables Subject Area

The System Reference Codes classes contain data about all of the codes used for all attributes in the NLDM.
The System Reference Codes Subject area includes the following classes:

- Code_set
- Code_value_general
- Code_value_xxxxx, where xxxxx are multiple specific code sets, e.g. zip codes
- Code_set_relationship
- Code_value_relationship
- Code_value_xxxxx_relationship, where xxxxx are multiple specific code sets, e.g. zip codes
- Code_set_group
- Code_set_xxxxx_group, where xxxxx are multiple specific code sets, e.g. zip codes
- Code_set_thesaurus
- Code_set_xxxxx_thesaurus, where xxxxx are multiple specific code sets, e.g. zip codes
- Extended_page_definitions
- Extended_class_definitions

3.3 General types of classes

3.3.1 Introduction

Various classes in the NLDM serve different purposes. The primary types of classes differentiated by purpose are: Core, Supporting, Associative, Super, and sub.

3.3.2 Core Classes

In the NLDM classes that represent distinct objects are referred to as Core classes. These classes always have a unique id (UID) as the primary key. Examples include Person, Organization, Observation, etc.

3.3.3 Supporting Classes

Supporting classes provide attribute extensions to Core classes by supporting one to many relationships, such as multiple race codes for a person. These classes will have as the primary key the primary key of the Core class and either a code representing one of many values for the attribute (e.g. person race) or a sequence number to make non-coded attribute values unique

3.3.4 Associative classes

Associative classes represent many to many relationships between two Core classes, such as between two persons, between a person and a locator for that person, or between a person and an act.

3.3.5 Super classes

Super classes serve to group classes that share attributes in common and to provide a way to form associations between groups of classes without having to form separate associations for each type within the group. The members of the groups organized by super classes are called sub classes. The Entity class is a super-class. It holds attributes common to all sub-classes of Entity and serves to represent relationships among the sub-classes and between the sub-classes and other classes. For example, the Entity class is used to represent the relationships between Persons and Organizations and their locators through the Entity_Locator_Participation class. The Entity super class is also used to represent relationships between any entity type and any act type.

3.3.6 Sub classes

Sub-classes are specializations of a super class that are distinguished by a class code.

3.4 General Types of Relationships

3.4.1 Introduction

Relationships describe how the classes (entities) in the model can be related to one another. The NLDM makes extensive use of class relationships and understanding the many types of relationships is key to understanding the NLDM. Based on the approach used in the HL7 RIM and the NEDSS Architecture, there are 5 general types of relationships represented in the NLDM:

1. Super Class to Sub-class
2. Associations
3. Participations
4. Repeating Attributes
5. Extended Attributes

3.4.2 Super Class to Sub-Class

This type of relationship allows the specification of classes as being sub-types of another class. This means that each type of sub-class has all of the same attributes and associations as the super-class, but also has some attributes specific to its sub-class. Each instance of a super-class must also exist as an instance of a sub-class and will be assigned the same UID as the primary key.
This relationship supports:

- Ease in understanding how sub-classes are similar and different
- Simplification in specifying the associative relationships between the sub-classes of a super class and other classes by providing one associative relationship between the super-class and the target class of the association. For example, the associative relationship between Entity and Role is used to represent the roles that any of the sub-classes of Entity (e.g. Person) can take on.
- Compactness in the model specification

The primary examples of super class to sub-class are the sub-classes of the super class, Entity, which include: Person, Entity_Group, Organization, Non-person living subject, Material, and Place.

3.4.3 Participations

Participations describe the relationships between instances of one core class and instances of a different core class. Participations are defined using associative classes. For example, participations of an entity in an act are defined using the Participation class. Participations can be many to many relationships. The most common example of this are the Participations of patients, care providers, and investigators in Observation and Work Up Acts. The patient (as a person entity) participates with the Role of subject of the observations and the Work Up. The care provider (as a person entity) participates with the Role of care provider in the observation as the provider of it. The investigator (as a person entity) participates with the Role of investigator in analyzing the observation and conducting the Work Up.

3.4.4 Associations

Associations describe the reciprocal relationships between two instances of the same class or between a core class and supporting classes. For example, the relationship between two persons, a patient and their care provider, can be represented as Role association with one person defined with a role of patient in the Role class and a scoping role in the Role class of care provider pointing to a second entry in the Person Class (scoping role is a role that defines the relationship of the entity's role, e.g. if the entity's role is 'patient' then the scoping role would be 'care provider).

Similarly, a relationship between two Acts can be defined using the Act-Relationship class. For example, an Observation sub-class of Act can be associated with a Case Report sub-class by defining the observation as supporting the Case. In this example, an investigation could have

started off by the investigator receiving a lab result suggesting a condition. The lab result was entered into the system. Subsequently, the investigator opens a Work Up and after conducting further observations determines that a Case does exist. As the investigator links the various observations to the Work Up and ultimately to the Case these linkages are stored in the Act-relationship class. This modeling approach allows for the independent existence of many types of events without having dependencies between them known before they can be stored. It also allows any act to be related to multiple other acts. So, for example, a particular lab result could be associated with two ongoing investigations by two different investigators.
Associations can be one to many or many to many relationships.

3.4.5 Relationships between a Class and Repeating Attributes of the Class

Repeating attribute relationships are used to represent attributes for a class that can occur more than once for any instance of the class; for example, a person can have many addresses, names, identifiers, etc. These are examples of standard modeling techniques for 'one to many' relationships.

3.5 Identifying, Non-identifying, and Category relationships

3.5.1 Introduction

These relationship terms describe ways of representing the relationship types described above in terms of the keys of the class. These terms describe the relationships between classes based on the way key values are shared or used. The key relationships in the NLDM Class Diagram are labeled as being either identifying or non-identifying. (Note. The particular tool used to create the NLDM Class Diagram does not distinguish between Identifying and categorical relationships, so categorical relationships are labeled as identifying relationships.)

3.5.2 Identifying Relationships

In an identifying relationship, the primary key of the parent class is used as part of the key of the child class. For example, the primary key of the Person class, person_UID, is the first part of the primary key of the Person_name class. Identifying relationships are used to represent Associations, Participation, and repeating attribute types of relationships.

3.5.3 Non-identifying Relationships

In a non-identifying relationship, the primary key of the parent is not included in the primary key of the child. Instead, the relationship is indicated by including the primary key of the parent class as a foreign key attribute in the child class.

3.5.4 Categorical Relationship

In a categorical relationship the primary keys of both classes are identical; categorical relationships apply to super-class/sub-class relationships. For example the primary key of the subclasses of Entity are identical to the primary key of the corresponding Entity class.

3.6 Keys

Keys are attributes that are used to both store attributes of the class and to provide efficient access to particular records based on the values stored in the keys. This efficient access is provided by the DBMS through the use of special indexing techniques.

3.6.1 Primary Keys

The NBS application architecture is object oriented even though it uses a relational DBMS for data storage. Primarily for this reason, the primary keys for each of the core classes are system assigned unique numeric identifiers rather than meaningful alphanumeric identifiers, such as person name. One of the key reasons for this choice is that (for technical reasons) the primary keys on a class cannot contain null (empty or blank) values and one of the design goals for NEDSS is to minimize such constraints on the NLDM. So, for example, the NLDM allows storing information about a person when all that is known, for example at the beginning of an investigation, is an address or phone number.

3.6.2 Foreign Keys

Foreign keys are attributes of one class that are primary keys in another class. They support the ability to directly link from the class in which the key is a foreign key to the class where the key is the primary key.

3.6.3 Indices

Indices are special structures built to support rapid access to records based on particular values of one or more attributes. In some respects they are more flexible than keys because they can be used even when some instances of a class do not have values for the attributes. Since this is common in public health, the NLDM uses indices extensively.

3.7 Change History Classes

For each Core class there is a corresponding History class that maintains the values for all attributes before any changes were made. Each time a change is made to a Core class a row is added to the History class for that class that includes the original values for all attributes along with identifying information of the user who made the changes as well as the program area the user was working within when the change was made. These history records can be viewed either to track changes to attributes or to view the values that were in effect at a given date and time.

Besides the obvious advantage of recording the details of changes to core classes, this facility also supports maintaining program specific versions of Core records when this is absolutely necessary. The contents of the Core classes represent the values approved by the Registry Manager role or those last entered by anyone with change permissions to the respective class. If a situation arises where two groups require some different values for the same core class, these can be maintained in one of the class history records.

3.8 Entity Merge Classes

The NBS functionality supports merging the data values of two Entity records and all associated records if it is determined that the two entities in fact represent the same Entity. For example, if the information in two different person records is determined to actually represent the same person. This information is stored in the Entity_ID_Merge_Hist class. This class also maintains the data needed to reconstruct the original two sets of records from the merged set if it is subsequently determined that the data really did represent two different people.

3.9 Record level Security

The NEDSS Base System includes a sophisticated authorization security mechanism that limits access by a user to only those objects in the database that they are authorized to access. The security model has three dimensions: organizational jurisdictions, program areas, and business objects. (See the NBS Architecture Overview and Technical Designers' Guide for more detail). Each user of the system has two security profiles stored in their individual profile: an authorization profile and a default permissions profile. The authorization profile defines their authorizations to objects in the system. The permissions profile defines the permissions granted to other users to objects that they create. The permissions profile of the user is stored in each object in the database when it is written or changed on behalf of that user.

Whenever a user attempts to access an object in the database that user's authorizations profile is compared with the permissions profile stored with the object to determine whether their authorizations match the permissions. For example, if a case report were written with a program area access permission of TB+HIV, only a user with either TB or HIV program area authorizations would be allowed to access the record. The NBS includes an operation to allow the changing of the permissions on an object.
Each occurrence of a record in any class has two security attributes written to it:

- Org_access_permis
- Prog_area_access_permis

These attributes contain coded lists, which indicate which jurisdiction and program areas have permission to access the record. Each user of the system has assigned to them an Authorization Profile, which indicates which records belonging to which jurisdictions and program areas they have authorization to access. These profiles are established when each user's UID and password are assigned by the security administrator and can be changed by the security administrator.

3.10 System Managed Attributes

Each instance of a class has a set of attributes that are managed by the system rather than entered and managed by the users. These include:

- Add_user_id
- Add_time
- Chg_user_id
- Chg_time
- Org_access_permis

- Prog_area_access_permis
- Record_status_cd
- Record_status_time

Each of these attributes is used for data quality control, security access control and monitoring, and data status maintenance.

4 SUMMARY OF THE NLDM CLASSES

4.1 Introduction

This section provides a brief description of the major classes in the NLDM. The full description of the classes and the attributes within each is provided in the NLDM data dictionary. However the data dictionary document is very detailed and is over 300 pages long. Therefore this section provides a summary of each of the classes in the NDLM.

4.2 Summary of the Entity classes

Entity is a super-class, that is, it is a class that contains properties common to all of its sub-classes. The sub-classes are: Persons, Non-Person Living Subjects, Organizations, Entity_Groups, Places, and Materials. Each of these classes corresponds directly to an HL7 RIM sub-class (except for Entity_Group).

4.2.1 Persons

Persons represent individual human beings. Each instance of a person can have multiple associated identifiers and locators. Each person can have many roles, such as patient, case investigator, state epidemiologist, primary care providers, etc.

4.2.2 Non-Person Living Subjects

Non-Person Living subjects represent all non-human organisms, such as bacteria, birds, domestic pets, viruses, etc.

4.2.3 Organizations

Organizations are formally defined groups of people and other organizations.

4.2.4 Entity_groups

Entity_groups are informally defined collections of people (or other Entities). Groups are typically defined for investigation, intervention, reporting, or survey purposes where the group definition describes the population to be reported on or surveyed and the attributes define the common characteristics of members of the group.

4.2.5 Places

Places represent buildings or locations identifiable by name, room numbers, postal addresses, etc. or geographic areas identifiable by physical latitude and longitude or other physical or geographic identification, e.g. the Rocky Mountains.

4.2.6 Materials

Materials represent any inanimate, physical substance not classified as a place. Typical public health examples include specimens, vaccines, medications, etc.

4.2.7 Roles

The Role Class allows the representation of the different types of roles that an entity can play. For example, a person could have the roles of: parent and state epidemiologist or a person could

have the roles of patient and food handler. The Roles association class allows storing any number of different roles for an Entity.

A person's roles in relationship to other entities can also be represented. For example, a person could have the roles of patient of a particular care provider and employee of a particular organization. The Role class includes an attribute named 'scoping_UID' which can be used to store the UID of the entity that is being related to the subject UID via the role relationship. For example, to represent the care provider of the patient the care provider's UID would be stored in the 'subject_UID' attribute and the patient's UID would be stored in the 'scoping_UID' attribute.

The entries in the Role class can be extended by indicating the specific acts that the person participates in under this role. This is done by linking the person to the act via both the Role entry and an entry in the Participation class. See the explanation of the Participation class below.

4.2.8 Entity ID

As with Person Name, the Entity ID class allows representing a multiple and variable number of identifiers for any Entity along with information specific to each such identifier, such as the organization issuing the particular ID (assigning authority) and the time period over which the ID is valid. For example, types of person identifiers could include: social security number, chart number, identifiers used in local systems, etc.

The Entity ID class provides a form of master patient index since the assigning authority can be anything the state chooses, such as existing state systems. (This usage is illustrated in section 12). For example, if the state has several systems in which information about persons is stored, the person identifiers used in each of those systems can be stored in the Entity Id class associating the external system person identifier with the NBS person identifier. Typically, these identifiers from the source systems would be loaded as part of the data migration step or as part of a Master Patient Index synchronization process. They can also be entered via the NBS interactive applications.

These linking identifiers can be used interactively and by any means of application-to-application or database-to-database sharing. For example, if someone wants to look up a person when they have the id assigned to that person from a separate system, they can use the NBS person lookup function supplying the identifier from the external system. Similarly, when stored procedures are used to synchronize two databases the identifiers from the non-NBS database for a person are loaded into the Entity Id class along with the id of the sending database. When an update of this record is sent by the NBS database to the non-NBS database the NBS stored procedure can supply the identifier used by the non-NBS database. The Entity Id class can keep multiple identifiers used in one external database and identifiers for any number of external databases.

4.2.9 Person and Organization Names

The NLDM represents person and organization names as separate classes rather than as attributes of Person or Organization so that a variable number of names can be stored for any instance of a person or organization. Each of the different names can also have a use code associated with it

that indicates how the name is used. For example, the NLDM allows the user to represent and differentiate a persons legal, preferred, stage, and nick names. There is an entry in the Person Name class or Organization Name for each instance of a different type of name.

4.3 Summary of the Locator classes

Locators are an especially important type of information for public health, particularly for investigations and interventions. This is one of the areas where the PHCDM and NLDM differ significantly from the HL7 RIM. As with names and identifiers, the NLDM allows storing multiple types of locators and for each type of locator allows storing multiple instances when used for different purposes. The model represents locators as independent classes rather than just as attributes of an Entity so that, for example, a user can identify all Entities associated with a particular address (e.g. the address of a fast food restaurant where an outbreak occurred.).

4.3.1 Postal Locator

The Postal Locator class allows the representation of any number of addresses for a person, organization, or place. The various addresses are differentiated by a 'use code', such as residence, employment, temporary living, birthplace, etc.

4.3.2 Tele Locator

Like the Postal Locator, the Tele Locator allows the representation of multiple telephone numbers based on the same set of 'use codes'. It further supports multiple types of telecom addresses: phone, fax, and email.

4.3.3 Physical Locator

The physical locator class represents physical and geographical locators for an Entity.

4.3.4 Entity-Locator Participation

The Entity-Locator Participation class allows any entity to be associated with any number of postal, physical, or telecom locators. It also allows any number of entities to be associated with the same locators so that the user can determine what other entities are associated with a given locator.

4.4 Summary of the Act classes

4.4.1 Act

Act is the super-class of all acts. Like Entity, it provides a single class by which relationships among act sub-classes and between act sub-classes and Entity classes can be represented.

4.4.2 Act ID

There is an Act ID class for acts that serves the same function as the Entity ID does for Entity. For example, a lab result would be assigned a unique id for its instance in the NLDM, but it may also be identified by a code assigned by the testing laboratory and by the system that placed the order with the lab. The Act_id class allows storing the values of each of these identifiers along

with information about the source of the identifiers. This allows flexible cross-referencing of an Act in the NLDM with the identifiers of the Act as stored in other systems.

The Act ID class provides a form of master act index (similar to the master patient index kept for Entities, see section 4.2.8), since the assigning authority can be anything the state chooses, such as existing state system. (This usage is illustrated in section 12). For example, if the state has several systems in which information about acts is stored, the event/act identifiers used in each of those systems can be stored in the Act Id class associating the external system Act identifier with the NBS Act identifier. Typically, these identifiers from the source systems would be loaded as part of the data migration step or as part of a master act index synchronization process. These identifiers can also be entered using an NBS application.

These linking identifiers can be used interactively and by any means of application-to-application or database-to-database sharing. For example, if someone wants to look up an act when they have the id assigned to that act from a separate system, they can use the NBS act lookup function supplying the identifier from the external system. Similarly, when stored procedures are used to synchronize two databases the identifiers from the non-NBS database for an act are loaded into the Entity Id class along with the id of the sending database. When an update of this record is sent by the NBS database to the non-NBS database the NBS stored procedure can supply the identifier used in the non-NBS database. The Act Id class can keep multiple identifiers for a single act used in one external database as well as identifiers from any number of external databases.

4.4.3 Observation

The Observation class represents one or more specific observations about an entity along with information about the source of the report. Observations can be of many types, for example, lab reports, clinical reports, hospitalization history, etc. Any given observation can contain any number of observation values. For example, a lab report of a CBC test could have multiple values for the various components of the test. The specific types of tests and results being reported are recorded as ‘observation types’ to allow flexibility, extensibility, and economy of storage. See section 5.3 below for a more detailed description of this feature.

4.4.4 Public Health Cases

This class represents the set of observations, interpretations, diagnoses, etc. made with respect to a particular case definition, as well as other relevant data associated with determining whether an observed condition meets a pre-defined Case Definition. These case definitions could be CSTE-CDC defined case definitions <see www.cdc.gov/epo/dphsi/phs.htm> or state defined conditions of interest.

4.4.5 Work-Ups

Work-ups are an extension of the concept of a case investigation. The work-up provides a linkage between all of the observations and other data associated with any investigation. The work-up can be against a case definition or for some other purpose. It allows the investigator to record the process and status of the investigation; that is, data that assist in the investigation but that are not normally reported as a conclusion of the investigation, as is normally done with the

Public Health Case data. For example, it allows the storage of work queues and outstanding requests for additional information from various partners, e.g. a primary care provider.
A desirable outcome of many types of investigations is not just the confirmation and reporting of the occurrence of a case, but also the prevention of related cases, such as in STD investigations.

4.4.6 Notifications

Notifications are formalized communication of public health data. One example is a nationally notifiable disease (NND) notification from states to the CDC. The NEDSS Architecture extends this concept to support notifications between various state organizations and jurisdiction as well as between state public health organizations and their partners. The exchange of these notifications is accomplished using messages.

The CDC is designing a family of public health messages that address message needs not covered by current HL7 message definitions.

4.4.7 Patient Encounters

Patient Encounters represent events where the patient is present or in direct communication while observations are being made. For example, a public health clinician could record a patient encounter during which clinical and laboratory observations were made, e.g. blood pressure measured and a blood sample taken.

4.4.8 Interventions

Interventions are a super class introduced in the PHCDM to represent the data common to both procedures and substance administrations. In the near future, the Intervention class will be further extended beyond surveillance to cover the major public health goals of prevention and case management.

4.4.8.1 Substance Administrations

This class represents events where a patient is administered a material substance via a specified route; for example, a vaccination given subcutaneously or a drug given orally.

4.4.8.2 Procedures

Procedures represent a wide range of activities involving some physical change or manipulation of the patient; such as a surgical operation or skeletal adjustment.

4.4.9 Referrals

Referrals represent any activity whereby one care provider or investigator refers a patient to another care provider or investigator for any reason.

4.4.10 Clinical Documents

Clinical documents are semi-structured representations of any type of healthcare act or service. As such, it is not so much a healthcare act as are the other classes in this subject area as it is a

way of storing textual descriptions of these other acts (or services). That is, it allows classifying and identifying a document by a set of header attributes that are represented in the NLDM as attributes and the storage of the document as an electronic file within an attribute of the Class. It is an implementation of the HL7 Clinical Document Architecture with level one heading elements represented as Named Attributes.

5 ATTRIBUTE TYPES

5.1 Introduction

One distinguishing and complicating property of health related information is that there are a large number of potential attributes that can be recorded for any entity or occurrence of an event and, typically, only a few are relevant and actually made during any particular encounter or case. It is also true of health related data that it is very dynamic and there is a recurring need to define and manage new attributes. To address these requirements, the NLDM uses two types of attributes, Named and Extensible.

5.2 Named Attributes

Named attributes are attributes that are identified and specified prior to the creation of the database tables and the application code that process them. Named attributes are the types that are most widely known and used. They correspond to the column names in relational tables and have specified formats, edit rules, and business rules defined in advance and implemented by a combination of code in the presentation, application logic, and database tiers. All of the named attributes for a class are included within each row of a class and are accessed as part of a single read or write to the corresponding table. For example, in the Person class 'birth_time' and 'marital_status_cd' are named attributes. The values for named attributes are stored as a single set of values as a row in a table.

Named attributes are used when:

- The value(s) the attribute can be assigned do not change over time (e.g. birth_time), or if they do change only the most recent value is of interest (e.g. education_level_cd). That is, there is typically only one value of the attribute for a given instance of the class.
- The values for the attribute are regularly or usually collected for most instances of the class (e.g. a person's name vs. CBC test results)
- The need for the attribute is known prior to the creation of the database schema
- The attribute is needed as a foreign key
- The attribute is accessed frequently and, therefore, must be accessed efficiently for adequate response time

5.3 Extensible Attributes

Extensible attributes can be used when the conditions for a Named Attribute are not met. That is when:

- The values the attribute can be assigned change over time and it is important to keep track of the changes (e.g., a person's blood sugar level). That is, there are typically multiple values of the attribute for any instance of the class.
- The values for the attribute are only occasionally collected for most instances of the class (e.g., the dates of a person's surgical operations)
- The need for collecting values of the attribute was not known at the time of database creation.
- The attribute is not needed as a foreign key
- The attribute is not accessed frequently relative to the object that it is an attribute of.

Extensible attributes represent data in a much different way than do Named attributes. Whereas for named attributes, all of the attributes for a given instance are contained as column values in a single table row, for extensible attributes a separate row in the table is created for each occurrence of a value for the attribute for an entity. . The information is stored as a name-value pair; that is the identifier for the attribute is stored as data rather than referencing an attribute name in the structure for the class. For example, laboratory test results are stored in the NLDM as name-value pairs for any person since lab results meet the first two conditions listed above for extensible attributes; that is:

- The values the attribute can take change over time and it is important to keep track of the changes (e.g., a person's blood sugar levels). That is, there are typically multiple values of the attribute for any instance of the class.
- The values for the attribute are only occasionally collected for most instances of the class (e.g., the dates and times of a person's blood sugar levels are not recorded for most patients.)

This allows both the dynamic naming of attributes (since the names are treated just like data) and allows storing only the information for attributes that actually exist for the instance of the class. The name of the attribute can be treated as a coded value or as free text. If it is a coded value, then the attribute name must match a set of attribute names that are stored in the System Reference Tables. Note that this predefinition is not the same as the pre-definition associated with a named attribute where the name needs to be part of the data definition language used to create the physical table. Since it only needs to be entered into the System Reference Table, the user can add new name-value pairs to an extensible attribute at any time.
Extensible attributes are most prominently used in the Observation class. In this class the attribute, 'cd', identifies the name of the attribute being stored.

Detailed examples of the use of Extensible attributes are described in section 12, 'Examples of data Representation in the NLDM'.

6 CODES FOR NAMED ATTRIBUTES

6.1 Introduction

The NLDM relies extensively on the use of standardized codes for many reasons, including: comparability of entered data, speed of data entry, shared understanding of entered data, etc. Of the approximately 700 Named Attributes in the version 1.07C version of the NLDM, approximately 450 of them have sets of code values defined. There are approximately 100 unique code sets that are used for these 450 coded named attributes, that is many of the codes are found in multiple classes; for example, status_code, duration_unit_cd.

The standard codes used in the NLDM can be categorized into two groups: those that support **the technical structure** of the NLDM and those that support the **functional needs** of public health workers.

6.2 Structural Codes

Some of the codes used to support the class structure of the NLDM are partially defined by HL7 in the HL7 RIM conceptual model. These code sets are extended by the NLDM and can be further extended by individual states. (See the section, System Reference Tables, for more detail on extending the NLDM Code Sets). These codes apply to most of the classes in the model and understanding their use is essential for understanding how to use the NLDM.

The primary structural codes are:

- class_cd
- mood_cd
- use_cd, nm_use_cd
- type_cd
- cd

6.2.1 class_cd

Class_cd is primarily an attribute of the super-classes, e.g. Entity and Act, and indicates for the type of sub-class this particular instance of the super-class is. For example, an Entity with UID of 100 might be a Person subtype, whereas an Entity with UID of 34 might be an Organization subtype. The class_cd's are the top level of the inheritance hierarchy for the classes. For example, currently defined class_cd's for the Entity super-class include: Person, Organization, Non-Person Living Subject, Entity_group, Material, and Place.

6.2.2 cd

The 'cd' code is an extension of the class_cd concept and allows for the finer differentiation of sub-classes.

6.2.3 Mood_cd

The mood_cd is an attribute of the Act class and is inherited by all Act sub-classes. It is used to indicate the overall context for any Act.
Currently defined mood_cd's are:

- Event an activity that has occurred
- Order an activity that has been ordered
- Definition an algorithm for describing an activity; a definition of a service (master.)
- Intent an intention or plan to perform an activity
- Appointment a planned activity for a specific time and place.
- Appointment Request A request for the booking of an appointment.

Any act sub-class may have one or more entries differentiated by the mood_cd. For example, a CBC lab panel (class_cd = Observation, cd=CBC) could have an entry with mood_cd of 'definition' which provides a description of the panel, multiple entries with mood_cd's of 'order' which represent orders for CBC panels, and multiple entries with mood_cd's of 'event' representing actual CBC test/result combinations.

6.2.4 Use_cd

Use_cd represents a type of relationship between two classes where the one class uses the second class in a particular way. For example, Persons and Postal locators have independent existences. One person might have a 'use_cd' of 'attends school at' a particular postal location, whereas another person might have a 'use_cd' of 'works at' for the same postal location. A person may have relationships to multiple postal locators each differentiated by the person's particular use of those locators. Similarly, a person may have multiple names each differentiated by a 'use_cd', such as, legal, nick, preferred, stage, etc.

6.2.5 Type_cd

Type_cd's are used the same way 'cd' are used, but are used for classes that do not have a super-class to sub-class relationship. That is, the type_cd's do not have a two level hierarchy.

See the NLDM Data Dictionary for a detailed description of each of these codes.

6.3 Functional Codes

To facilitate sharing of data with other health care partners, the NEDSS Program is adopting as many of the coding standards established by standards development organizations as possible. These code standards include, but are not limited to:

- Health Level 7 (HL7)
- Federal Information Processing Standard (FIPS)
- International Standards Organization (ISO)
- US Health and Human Services
- Office of Management and Budget (OMB)
- Logical Observation Identifiers, Names, and Codes (LOINC)
- Systematized Nomenclature of Medicine (SNOMED)
- International Classification of Diseases (ICD)
- Etc.

The process of selecting the correct codes from these national standards and specifying their potentially unique meaning within the public health context and identifying those public health concepts for which a suitable standard does not yet exist is a complex task. CDC has a group working on a continuous basis to define the codes and the relationships among the codes to be used in the NLDM..

A description of the currently defined code sets is provided in Appendix B of this Guide and in the NLDM Data Dictionary.

7 CODES FOR EXTENSIBLE ATTRIBUTES

7.1 Introduction

As described in section 5.2, Extensible attributes include two components. One component represents the name of the attribute. The other represents the value of the attribute. The term name-value-pair is used to describe this basic approach. When name-value-pairs are stored in relational databases each part has to be represented by a separate attribute, one that contains the name of the attribute and the other the value. Thus, there may be two different sets of codes for extensible attributes: one associated with the name and another with the values. The codes used for the value part of the name-value pair in an Extensible attribute may be the same as for the Named attribute types, or new codes may be defined and entered into the System Reference Tables.

The name part of the pair may be either coded or free text, though the usefulness of free text names is quite limited since the group choosing the text then only knows the meaning of the attribute. Most names are based on code sets so that business rules may be associated with the value based on the code and the general meaning of the name and value is shared based on a shared knowledge of the code. For example, the meaning of the lab result value, '5.3', is much more meaningful if it is associated with the name code of 'CBC/WBC' than with the free text of 'white'.

7.2 Extensible Attributes for Observations

There are numerous code sets that may be used for the name categories, such as LOINC and SNOMED. The NEDSS Vocabulary team is actively selecting and designing these codes and results will be published in the forthcoming, "NLDM Terminology and Vocabulary Guide".

8 SYSTEM REFERENCE TABLES (SRT) & TERMINOLOGY SERVICES

8.1 Introduction

As noted above, the NLDM uses a large number of code sets to ensure the common representation of data. Most of these codes are based on international, national, and NEDSS Program standards, but allow for program area and state extensions. Since there are so many code sets and since even the national standard codes are continuously being revised, management of the code sets is a complex task. The NLDM includes a two level code management process. The national, international, HHS, and CDC code sets in the SRT are maintained in a code management system at CDC. . This version of the SRT data contains the complete range of codes for each of the code sets. For example, it includes all of the LOINC codes even though only a subset of these codes is normally used in public health. A version of the SRT management tool is included with each NBS implementation to support state level extensions to the code sets. Each state can manage their own codes sets using this tool. They can load the portions of the large code sets that they want either from the CDC version or directly from the various standards development organizations. They may also use this tool to add their own local extensions to the standard code sets.

The size of the potentially relevant code sets for health related activities is extremely large. To support access to the widest possible range of terminology and vocabulary without burdening each state instance of the system reference tables with infrequently used codes, the NBS provides a Terminology Service which allows the user to connect to the Terminology Services maintained at the CDC whenever their local copy of the SRT does not contain the vocabulary references they need. Codes selected from the CDC Terminology Services may be optionally loaded into the local SRT. See the NBS Architecture Overview and Technical Designers' Guide for more detail on the Terminology Services.

The structure and content of the System Reference Table classes are shown in section 1.8.7 above.

8.2 SRT Classes

8.2.1 Codeset

The Codeset class contains entries for each of the different domains or categories of codes. For example, zip codes, country codes, race ethnicity codes etc.

8.2.2 Code_relationship

This class provides the capability to specify relationships between sets of codes defined in Codeset

8.2.3 Code_Values

This set of classes contains the defined code values for each of the sets of codes defined in Codeset. For example, all of the codes define for the code set of 'person_race'. For the code

sets that contain large numbers of code values, separate classes are used. There is also a class that contains the code values for all of the code sets that are not represented by separate code value tables. This class is named 'Code_value_general'.

The code value classes that are specific to one code set are:

- Code_value_city
- Code_value_county
- Code_value_country
- Code_value_state
- Code_value_zip
- Code_value_loinc
- Code_value_snomed
- Code_value_naics
- Code_value_icd09
- Code_value_icd10
- Code_value_industry
- Code_value_race

8.2.4 Code_value_relationship

This class provides the capability to represent relationships between two codes in any code set.

8.2.5 Code_group

This class provides the capability to form hierarchical groupings of codes within and across code sets.

8.2.6 Code_thesaurus

This class provides the capability to define codes as being synonyms and to define words as being synonyms of defined codes.

9 NLDM SUPPORT FOR EXTENSIBILITY FEATURES

9.1 Introduction

Since specific implementations of the NEDSS Architecture must support functionality unique to each state, extensibility is a primary design feature of each component of the architecture. The NEDSS architecture supports two basic technologies for extensibility. The first technology supports the addition of fields to pages and the management of the data entered for these new fields without requiring application programming or changes to the database structure. The second technology supports the addition of fields to the user interface and to the database as well as supporting the addition of new screens to the UI and adding unique business logic and work flow to the functionality of the Base System.

The first approach is supported using the Extended Attribute techniques described in section 5.3. This approach is described in sections 9.2 and 9.3 below.

The second approach is implemented by the development of additional modules that are invoked by the Base System without needing to make changes to the Base System code. These new modules may use all of the system services provided with the Base System and the physical database instantiation of the NLDM This approach is supported by the Session and Context Control sub-systems of the NBS. Descriptions of these sub-systems and guidelines for their use are provided in the NBS Architecture Overview and Technical Designers' Guide.

10 RELATIONSHIPS OF THE NLDM TO NBS FUNCTIONALITY

10.1 Introduction

The NLDM is designed to support the functionality planned for the NEDSS Base System as well as program area and state area functional extensions. The initial releases of the NBS will focus on public health surveillance. Subsequent releases will address additional public health functional needs.

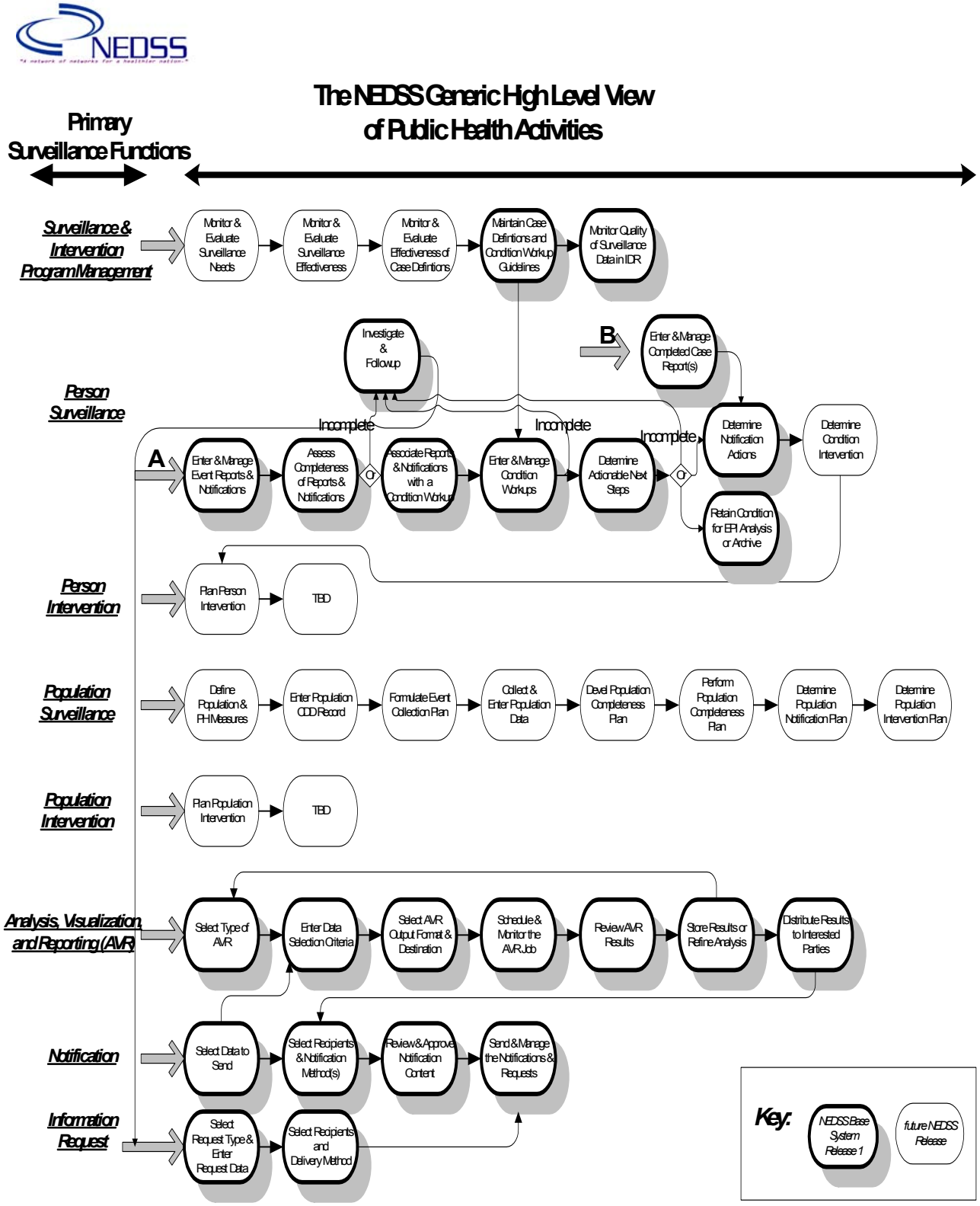
10.2 High Level Overview of NBS Functionality

Figure 10.1 provides a high level overview of the functionality of the NBS. The shaded functions will be addressed by the initial releases of the NBS.

The NLDM provides the basis for supporting these functions while providing a path for supporting much wider functionality in the near future. The fundamentals of surveillance are supported by the NLDM as follows:

- Events of public health interest are collected and stored as observations.
- The observations may be related to multiple types of entities with multiple role relationships defined between each entity instance and the Act as described in section 4.
- Investigators collect related sets of observations on a person or population as a work-up
- Assessment of the data in the work-up is done against a Case Definition to determine whether the set of observations constitute a Case.

- If the Case is a Nationally Notifiable one, a public health notification message is sent to the CDC
- If the Case warrants notifications to other public health workers, those notifications are sent
- The whole set of collected data is available for epidemiological analysis


The NEDSS Generic High Level View of Public Health Activities
Primary Surveillance Functions
Surveillance & Intervention Program/Management
Monitor & Evaluate Surveillance Needs
Monitor & Evaluate Surveillance Effectiveness
Monitor & Evaluate Effectiveness of Case Definitions
Maintain Case Definitions and Condition Workup Guidelines
Monitor Quality of Surveillance Data in NDR
Person Surveillance
Investigate & Follow-up
B
Enter & Manage Completed Case Report(s)
A
Enter & Manage Event Reports & Notifications
Assess Completeness of Reports & Notifications
Incomplete
Or
Associate Reports & Notifications with a Condition Workup
Enter & Manage Condition Workups
Incomplete
Determine Actionable Next Steps
Incomplete
Or
Determine Notification Actions
Determine Condition Intervention
Retain Condition for EPI Analysis or Archive
Person Intervention
Plan Person Intervention
TBD
Population Surveillance
Define Population & PHI Measures
Enter Population CDD Record
Formulate Event Collection Plan
Collect & Enter Population Data
Devel Population Completeness Plan
Perform Population Completeness Plan
Determine Population Notification Plan
Determine Population Intervention Plan
Population Intervention
Plan Population Intervention
TBD
Analysis, Visualization and Reporting (AVR)
Select Type of AVR
Enter Data Selection Criteria
Select AVR Output Format & Destination
Schedule & Monitor the AVR Job
Review AVR Results
Store Results or Refine Analysis
Distribute Results to Interested Parties
Notification
Select Data to Send
Select Recipients & Notification Method(s)
Review & Approve Notification Content
Send & Manage the Notifications & Requests
Information Request
Select Request Type & Enter Request Data
Select Recipients and Delivery Method
Key:
NEDSS Base System Release 1
future NEDSS Release

10.3 Fundamental Operations on NLDM classes

The functions or workflows illustrated in figure 10.1 are implemented as a sequence of operations on the Core classes in the NLDM. As part of the extensibility and interoperability design goals of the NEDSS Architecture, the NLDM is designed so that it may be accessed equally well by either NBS compliant applications or by external applications using the set of transaction messages defined within the Public Health Message family.

This common access is supported by a set of operations that may be performed on the core classes. The following lists summarizes the operations that may be performed on the core classes:

- Search (by multiple parameters)
    - Exact Match
    - Approximate Match
- View
    - Current
    - Change History
    - Merge History
- Add
- Edit (Update)
- Undo Changes
- Logically Delete
- Undo Logical Delete
- Archive
- Delete
- Validate (e.g. against a Case Definition)
- Compare (2 or more records)
    - All Attributes
    - Differences only
- Merge (2 records)
- De-Merge
- Match (identify potential duplicates)
- Reset Status
- Re-Set Access Permissions
- Encrypt/Decrypt
    - Selected fields
    - Entire record

The presentations and documents listed in the reference section provide detailed descriptions of the functionality of the NBS.

11 RELATIONSHIP OF THE NLDM TO THE NBS MESSAGING SUB-SYSTEM

11.1 Introduction

The NBS Messaging Sub-System is used to exchange electronic messages between public health partners. For example, a state public health department may receive electronic lab reports from laboratories or selected discharge summaries from hospitals and send notifications to other public health departments or to the CDC. The messaging sub-system provides functionality similar to commonly used email, but with some significant differences. First, the messages and their transport are highly secured. Also, the messages are highly structured so that they can be processed by applications.

Since the NEDSS Architecture defines the Messaging sub-system as being able to be installed without the NBS, the NLDM Class Diagrams and Data Dictionary for Message_In and Message-Out are packaged separately from that of the core NLDM.

11.2 Message-In: Received Messages

When messages are received, they are translated from their native message structure (e.g. an HL7 2.3 pipe-delimited electronic laboratory result reporting message (ORU)) into a set of classes in the IDR referred to as the MESSAGE_IN. The structure of these classes is very close to the structure of the classes in the ODS. Since messages received electronically have not been validated the way data entered via NBS applications are, received messages are first stored in these staging classes so that the data contents can be validated before being moved to the ODS. As part of the validation processes, the contents of the messages are assigned to a jurisdiction and program area for review and potentially linked to Entities (e.g. persons and organizations).

11.3 Message-Out: Out-bound Messages

The Message-Out table contains messages that have been formatted according to a pre-defined message structure, for example, a message structure defined within the public health message family or by HL7. The messages contain an identifier for the recipient derived from the PHDIR or the local NBS Directory.

11.4 Relationships between Message-In and Message-Out.

The relationship between these two components is described in section 1.5, Major Components of the NLDM.

12 EXAMPLES OF DATA REPRESENTATION IN THE NLDM

12.1 Introduction and Objectives of the Examples

The following sections describe scenarios showing how various sets of data would be represented (stored) in a database based on the NLDM. The scenarios are not intended to be representative in the completeness of the data stored for each class, rather the emphasis is on showing:

- a comparison of the use of Named vs. Extensible attribute types and the functional richness of the Extensible attributes
- the use of structural and functional codes
- how role relationships among Entities are represented
- how relationships between Entities and Acts are represented
- how multiple types of identifiers for classes are represented
- how the Extensible Attribute types may be used to efficiently store a wide range of pre-defined and ad-hoc defined attributes and their values

These examples are intended to be representative only and are not complete in many of the details. In particular only a few attributes for each class are included. The examples also show that the user only needs to enter some of the attribute values since the system determines many of the structural codes and identifiers based on the application context and current state of the database. For example, the user never needs to enter any of the sequence codes or unique id's since these are assigned and used by the NBS to maintain uniqueness. It is also the case that most of the values shown in the associative classes are taken from values entered for the core classes; that is, the user only enters the data for the core classes and the system combines the user entered data to create the data values for the associative classes.

Future versions of this Guide will include additional examples that show the use of all of the other classes and additional scenarios showing how different sets of data would be represented in the NLDM.

12.2 Scenario 1. Representing Persons with multiple Names, Roles, and Locators

One of the unique complexities of Public Health investigation is identifying and tracking individuals when information about them is received from different sources using different identifiers; for example different names and addresses. Similarly, people have different types of roles in public health related activities and these roles may change over time.

The following scenario shows how the NLDM can be used to meet these requirements. This particular scenario demonstrates a rather complex set of requirements that would probably occur infrequently in the real world of public health, but shows in one example how any one or more of these requirements would be represented for any given situation.

12.2.1 Scenario 1 Description

The scenario involves the case management of a person confirmed to have an STD and who reports a history of sex with strangers.

- Given name = Joseph Michael Snow White
- Nick name = Joey D
- Preferred name = Joey Snow
- Residence = 123 Adams, Atlanta GA from 7/1/2001
- Previous Residence = 222 Peachtree, Apt 10 to 6/1/2001
- Residence Phone = 404-404-1234
- Residence email = joey@aol.com
- Employment = 34 West Avenue, Dekalb
- Employment phone = 404-404-3333
- Employment from = 9/10/99
- Usually found Even = 124 Adams, Atlanta GA
- Usually found Days = Adams and Peachtree
- Usual Phone = 404-456-1234
- SSN =123-12-1234
- GA Drivers # = 5
- LA Drivers # = 6
- GA Medicaid # = 8
- GA PH System A = JS78
- GA PH System B = 12345
- Race = Italian
- Race = Arab
- Race = French
- Birth Date = 1961
- Birth sex = male
- Age = 34 on 4/3/95

12.2.2 Scenario 1 Data Representation in NLDM classes

Note in each of the scenarios, only the attributes within each class that have values are represented. Also, note that the NLDM is a fully normalized model. So that, for example, attributes that can take on multiple values are represented in separate classes, such as name. However, any physical implementation of the model would include de-normalizations so that the most important instances of a multi-valued attribute would be stored as part of the Core Classes. For example, while each of a person's names would be stored in the Person_name class, the preferred or most frequently used would also be stored in the Person class for efficiency and ease of use purposes.

12.2.2.1 Entity

The Entity super-class contains the unique system Id for the person along with the class_cd for his sub-class; that is, PSN for person. The unique ID for the person is the same in the super-class as in the sub-class, Person. This allows us to link all entity sub-classes to other classes through single association classes with entity rather than having association classes between each sub-type and all other classes.

ENTITY_UID	CLASS_CD
1	PSN

ENTITY_UID: is a unique number assigned by the system for each instance of an Entity and the instance of the corresponding sub-class.

12.2.2.2 Person

The Person class contains the attributes for which there is only one possible or likely occurrence, e.g. Birth sex. It links to classes that contain multiple values for an attribute, e.g. addresses and names. The class below shows the attributes from our examples that are stored in the Person class. It is linked to a class that has multiple entries for name, one entry per type of name as described in section 12.2.2.4 below.

(Note that there is not always a strict rule determining whether an attribute may have only one or possibly multiple values, for example occupation. Many people hold two or more jobs in different occupations groups. The decision to model an attribute as a single or multi-valued one is based on the extent to which it is important to the system to keep track of the multiple values. Notice also that whether it is important or not is not a universally agreed upon fact. Refer to Section 5, Attribute Types for more detail)

For this reason, the NLDM supports local extensions, as described in the section named NLDM class Extensibility Features. For example, if a particular state or program area needs to keep track of multiple occupations this can be accomplished. A request may also be submitted to have the NLDM extended to accommodate this requirement as part of the Base System.

PERSON_UID	STATUS_CD	CD	BIRTH_SEX_CD	REPORTED_AGE	REP_AGE_TIME	BIRTH_TIME
1	A		M	34	4/3/95	1961

PERSON_UID: is a unique number assigned by the system for each instance of an Entity and the instance of the corresponding sub-class of Person.

The status_cd of ‘A’ indicates that this record is ‘active’. A status_cd of “I” would indicate the record was inactive.

12.2.2.3 Entity_ID

Entity_ID is an important and powerful class for public health. It allows representing any number of different types of identifiers for entities. Some of these types are part of the NEDSS Base System Standards, but the user may add their own also.
For this scenario we have identified 5 standard identifiers (SSN, GA Driver’s License, LA Driver’s License, and GA Medicaid # and 2 locally defined identifiers that are used to record the

identifiers for this person in two of the public health department's other application systems (GA PH System A and GA PH System B).

ENTITY_UID	ENTITY_ID_SEQ	STATUS_CD	ROOT_EXTENTION_TXT	ASSIGNING_AUTHORITY_CD	ASSIGNING_AUTHORITY_DESC_TXT	TYPE_CD	TYPE_DESC_TXT
1	1	A	5	GADR	GA Dept Revenue	DLNUM	Drivers License Number
1	2	A	123-12-1234	US SSA	US Social Security Administration	USSSN	Soc Sec Num
1	4	A	8	GAMCD	GA Dept Health	MEDCAIDNUM	GA Medicaid Number
1	5	A	JS78	GA PH Sys A	GA Dept Health	GASYSA_ID	GAPH A Identifier
1	6	A	12345	GA PH Sys B	GA Dept Health	GASYSB_ID	GAPH B Identifier

ASSIGNING_AUTHORITY_CD: Code that designates the authority that assigned the Entity identifier
ASSIGNING_AUTHORITY_DESC_TXT: Textual description of the entity identifier assigning authority.
ROOT_EXTENTION_TXT: Text representing the entity identifier assigned to an entity by an assigning authority.
TYPE_CD: Code that designates the type of entity identifier assigned. Types include: drivers license number, social security number, etc.
TYPE_DESC_TXT: Textual description of the entity identifier type

12.2.2.4 Person_Name

The Person_Name class contains an entry for each type of name. The name type is indicated by the 'name_use_cd' which indicates the usage of the name; e.g. Legal, Nick, or Preferred. Note that the structure allows capturing two last names.

PERSON_UID	NM_USE_CD	PERSON_NAME_SEQ	STATUS_CD	LAST_NAME	LAST_NAME_2	FIRST_NAME	MIDDLE
1	L	1	A	Snow	White	Joseph	Michael
1	NK	2	A	D		Joey	
1	AL	3	A	Snow		Joey	

NM_USE_CD: A code indicating the reason for which the name is used. Includes the following: display (the name normally used), license (encompassing birth certificates, school records, degrees and titles, licenses, etc.), artist (encompassing stage names, pseudonyms/writer names), indigenous/tribal, and religious. "L" = Legal Name, "NK" = Nickname, "AL" = alias..

12.2.2.5 Locators

As with names, the NLDM supports representing multiple occurrences of different types of locators, the types being distinguished by the 'use_cd'; for example, residence and work. It also allows specifying and representing less standard uses of locators; for example, 'may usually be found at'. The NLDM also supports representing the time periods applicable to each locator so that, for example, one may identify prior residences when it is important to know where a person was located at some time in the past.

The Locator subject area consists of 4 primary classes: Postal_Locator, Tele_Locator, Physical_Locator, and Entity_Locator_Participation. This separation of the entity from the locator types provides the capability to maintain information about locators independent of a particular entity. This supports the capability of being able to identify all persons associated with a given locator; for example all persons working at a given address or all persons living at a given address. The locator classes also support recording physical and geographical locations, but these are not represented in this scenario.

12.2.2.6 Entity_Locator_Participation.

This class represents the relationship between an entity and the postal and tele locators by which they may be located. The UIDs of the locators point to the corresponding entries in the Postal and Tele locator classes.

ENTITY_UID	LOCATOR_UID	STATUS_CD	Class_CD	USE_CD	CD	FROM_TIME	TO_TIME
1	100	A	PST	PR	House		1-June-2001
1	105	A	TEL	PR	Phone		1-June-2001
1	101	I	PST	PR	Mailing	7-June-2001	
1	102	A	PST	WP	Office	10-Sep-1999	
1	106	A	TEL	WP	Fax	10-Sep-1999	
1	103	A	PST	UFE	Dorm		
1	107	A	TEL	UFE	Phone		
1	104	A	PST	UFD	Office		

USE_CD: Indicates the use that an entity makes of a location. For example, residence, place of work, mailing location, network, birth, death, etc. "PR"= Primary Residence, "WP" = Work Place, "UFE" = Usually found days, "UFE" = Usually found evenings.
CD: Code that indicates the type of location. For Postal locations, this may include the type of location (e.g., residence, office, restaurant, hospital, daycare center, ship, prison, nursing home, etc.). For telecommunication locations, this may include the type of telecommunications equipment (e.g., phone, fax, cell phone, answering service, etc.).

12.2.2.7 Postal_Locator

POSTAL_LOC_UID	STREET_ADDR_1	STREET_ADDR_2	CITY_CD	STATE_CD
100	123 Adams		ATL	GA
101	222 Peachtree	Apt # 10	ATL	GA
102	34 West Ave		DKB	GA
103	124 Adams		ATL	GA
104	Adams & Peachtree		ATL	GA

12.2.2.8 Tele_Locator

TELE_LOC_UID	PHONE_NBR	EMAIL
105	4044041234	JOEYD@AOL.COM

106	4044043333	
107	4044561234	

12.2.2.9 Person Race

The NLDM supports representing multiple races for a person to accommodate recent reporting requirements.

PERSON_UID	RACE_CD	RACE DESC TEXT	RACE CATEOGRY CD	RACE CATEGORY_DESCRIPTION_
1	2114-7	ITALIAN	2106-B	WHITE
1	2129-5	ARAB	2106-B	WHITE
1	2111-3	FRENCH	2106-B	WHITE

RACE_CD : Code for the person's race (e.g., American Indian/Alaskan Native, White, African American, Asian, Hawaiian/Pacific Islander). The attribute repeats in order to record the multiple racial categories to which a person may belong. “2114-7”: =Italian, “2129-5” = Arab, “2111-3” = French.

12.3 Scenario 2: Representing Role Relationships among Entities

A critical emerging requirement in public health surveillance and case management is the capability to represent and subsequently analyze the many types of relationships that may exist not only between persons but also between persons and other types of entities. These types of relationships, called Roles, are represented in the Role class. The role class supports three primary functions:

1. Associates one or more roles to an Entity (e.g. Person)
2. Associates two or more entities together based on a role relationship between them
3. Optionally, may include a Role when representing the participation of one or more entities in an Act.

A person may have many different types of roles; for example, participant in a survey, patient in an outbreak, patient, primary care provider, case worker, state epidemiologist, etc. By extension the relationship between two persons may be represented in the Role class with one person being the subject and the other person being the target. For example, one person may be related as a 'patient' to another representing the 'primary care provider'; or one person may be related as the 'subject of an investigation' conducted by another person whose role is that of 'public health investigator'.

Additionally, roles can be represented between different types of Entities; that is between:

- Persons
- Non-Human living organisms
- Organizations
- Groups
- Materials
- Places

A single, unique identifier in the Entity super-class represents each occurrence of an instance of one of these classes. This single, unique UID is used for any instance of a class wherever that instance is referenced, for example in the Locator, Role, and Participation classes.

12.3.1 Scenario 2 Description

This scenario shows how the following relationships that Joey Snow has are represented in the NLDM along with additional relationships among some of the entities with which Joey has relationships:

1. Joey is being treated by Dr. Daniels
2. Joey belongs to the Peachtree Healthcare HMO
3. Dr. Daniels has taken a blood sample for testing from Joey
4. Joey is part of public health survey, 'Partners', sexual habits of bi-sexual males
5. Dr. Rudolf is the state epidemiologist directing the study
6. Helen Smith is the case worker conducting the survey
7. Martin is a close friend of Joey's.
8. John is also a close friend of Joey's
9. Dr. Rudolf is Helen's supervisor
10. Both Dr. Rudolf and Helen Smith work in the Atlanta Public Health Department

11. The blood test has been completed and positive results have been detected.
12. John belongs to the Peachtree Healthcare HMO
13. Martin belongs to the Peachtree Healthcare HMO

12.3.2 Scenario 2 Data Representation in NLDM classes

12.3.2.1 Entities

The Entity class holds a unique ID for each entity, indicates the type of sub-class that each entity belongs to, and is used to represent associations that each Entity occurrence has with other entities and with Acts (which are described in the following section).

ENTITY_UID	CLASS_CD
1	PSN
2	PSN
3	ORG
4	MAT
5	GRP
6	PSN
7	PSN
8	PSN
9	PSN
10	ORG
11	NLIV

Key to Entities:
Joey
Dr. Daniels
Peachtree Healthcare HMO
Blood sample from Joey by Dr. Daniels
Partners Survey
Dr. Rudolf
Helen Smith
Martin
John
ATL PHD
Treponemal pallidium (Syphilis)

CLASS_CD : This is the most general classifying attribute of the entity class. This code indicates, at a high level, what kind of entity is intended using a terminology defined by the NLDM. Entity class values include codes for person, place, material, etc. In fact each specialization of entity within the model is identified by a class code. However, other classes of entity may be defined, but not included as model specializations when they do not have distinct attributes. "PSN" = Person, "ORG" = Organization, "MAT" = material, "GRP" = Group, :"NLIV" = Non-Person living subject

12.3.2.2 Persons

Note in the class below that attributes are entered in the Person class only for the subject of the investigation, Joey. There are other entries in the person class, but the system does not require that attributes be populated. This allows identification of an entity and relating it to other entities or acts with minimal required information. Constraints such as required fields are specified when the system is installed and tailored for a particular state.

PERSON_UID	STATUS_CD	CD	BIRTH_SEX_CD	REPORTED_AGE	REP_AGE_TIME	BIRTH_TIME
1	A		M	34	4/3/95	6/22/1991
2	A					
6	A					
7	A					
8	A					
9	A					

Key to Entities:
Joey
Dr. Daniels
Dr. Rudolf
Helen Smith
Martin
John

CD: Entity type code is the detailed typing attribute of the entity class and all of its subclasses. This code indicates what kind of entity is intended using some terminology. The terminology is hierarchical. At the top is a high-level category (such as is represented by the Entity Class Code). Beneath this layer there exist multiple, frequently external, domains that reflect much more fine-grained typing.

BIRTH_GENDER_CD: Code for the individual's sex at birth. Includes Male and Female.

12.3.2.3 Person Names & Locators

The Person_Name class would have additional entries for each of the new Persons, but this would not include any additional functionality over the display of the various person names shown in Scenario 1, it is not repeated here. The same applies for the various types of Locators that could be associated with each new Person

12.3.2.4 Organizations

Organizations are represented in a separate sub_class with attributes particular to the sub_class. As with the Person class, multiple names and locators may be stored for any organization and each is distinguished by a use_cd. These are not represented in this scenario since the usage is the same as for Persons as described in scenario 1.

ORGANIZATION_UID	STATUS	CD	ORGANIZATION_NM_TXT
3	A	HMO	Peachtree Healthcare HMO
10	A	PHD	Atlanta Public Health Dept

CD: This is the main classifying attribute of the organization. This code indicates what kind of organization using a code depicting administrative and functional structures places, such as reference lab, hospital, etc "HMO" = Health Maintenance Organization, "PHD" = Public Health Department.

12.3.2.5 Non Person Living Subjects

The bacteria discovered in Joey's blood sample is represented in a sub_class named Non_Person_Living_Subject. The following class shows how the primary attributes of the bacteria would be represented.

NON_PERSON_UID	STATUS	CD	NM	TAXONOMIC_ CLASSIFICATION_CD
11	A	BACT	*Treponemal pallidium* (Syphilis)	30-33-78

CD: This is the main classifying attribute of the Non-Person living subject class. This code indicates what kind of Non-person is meant using a code from one of several coding systems, such as living subjects (typed by animal and plant taxonomies). "BACT"= Bacteria.

TAXINOMIC_CLASSIFICATION_CD: A code representing the taxonomy of the living subject.

12.3.2.6 Material

The blood sample taken by Dr. Daniels from Joey is represented as a material with a cd of BLD and a Name of Blood Specimen.

MATERIAL_UID	STATUS	CD	FORM_CD	NM
4	A	BLD	JLDS	Blood Specimen

CD: This code describes the kind of material. In this example, "BLD" = Blood.
FORM_CD: the material. This includes the typical state of matter (solid, liquid, gas) and, for therapeutic substances, the dose form.

NM: : Name of the material. This is important in special cases such as the name of a lake, an amusement park, or a cruise ship.

12.3.2.7 Entity_Group

The public health survey being conducted by Dr Rudolf and Helen Smith is represented as a Group sub-class. The sub-class defines the population properties and other characteristics of the survey.

ENTITY_GROUP_UID	STATUS	CD	NM	FROM_TIME	TO_TIME
5	A	PHSUR	Partners Survey	Sep-00	Aug-01

CD: Code for the type of entity group. Examples include groups such as families, Rotary Club members, girl scouts, retired persons, persons with heart disease, alcoholics, persons vaccinated against measles, persons who are chronic typhoid carriers, or patients on a given floor or ward of a hospital. In this example, "PHSUR" = Public Health Survey.

NM: A name of the entity group.

12.3.2.8 Entity _ID

The Entity_Id class shows the additional entries that were made for identifiers for the new entities. All entities were entered with an ID assigned by an existing application, GA SysA, so that they may be automatically linked between an NBS and that local system. In addition, the social security numbers were entered for each of the Person Entities.

ENTITY_UID	ENTITY_ID_SEQ	STATUS_CD	ROOT_EXTENSION_TXT	ASSIGNING_AUTHORITY_CD	ASSIGNING_AUTHORITY_DESC_TXT	TYPE_CD	TYPE_DESC_TXT
1	1	A	5	GADR	GA Dept Rev	DLNUM	Drivers License Num
1	2	A	123-12-1234	US SSA	U Security Admin	USSSN	Soc Sec Num
1	3	A	6	LADR	LA Dept Rev	DLNUM	Drivers License Num
1	4	A	8	GAMCD	GA Dept Health	MEDCAIDNUM	GA Medicaid Number
1	5	A	JS78	GA PH Sys A	GA Dept Health	GASYSA_ID	GAPH A Identifier
1	6	A	12345	GA PH Sys B	GA Dept Health	GASYSB_ID	GAPH A Identifier
2	1	A	22222	GA PH Sys A	GA Dept Health	GASYSA_ID	GAPH A Identifier
2	2	A	123-12-1111	US SSA	US Social Security Admin	USSSN	Soc Sec Num
3	1	A	33333	GA PH Sys A	GA Dept Health	GASYSA_ID	GAPH A Identifier
4	1	A	44444	GA PH Sys A	GA Dept Health	GASYSA_ID	GAPH A Identifier

5	1	A	55555	GA PH Sys A	GA Dept Health	GASYSA_ID	GAPH A Identifier
6	1	A	66666	GA PH Sys A	GA Dept Health	GASYSA_ID	GAPH A Identifier
6	2	A	123-12-6666	US SSA	US Social Security Admin	USSSN	Soc Sec Num
7	1	A	77777	GA PH Sys A	GA Dept Health	GASYSA_ID	GAPH A Identifier
7	2	A	123-12-7777	US SSA	US Social Security Admin	USSSN	Soc Sec Num
8	1	A	88888	GA PH Sys A	GA Dept Health	GASYSA_ID	GAPH A Identifier
8	2	A	123-12-8888	US SSA	US Social Security Admin	USSSN	Soc Sec Num
9	1	A	99999	GA PH Sys A	GA Dept Health	GASYSA_ID	GAPH A Identifier
9	2	A	123-12-2222	US SSA	US Social Security Admin	USSSN	Soc Sec Num
10	1	A	10101	GA PH Sys A	GA Dept Health	GASYSA_ID	GAPH A Identifier
11	1	A	20202	GA PH Sys A	GA Dept Health	GASYSA_ID	GAPH A Identifier

12.3.2.9 Roles

Having established the identities of the entities for this scenario, we can now describe how the Role class is used to represent the many potential relationships among these entities. These relationships are shown in the following sub-set of a Role class. The column 'key' is not part of the actual class, it is included here only to tie the descriptions of the roles to their representation in the class. The attribute, ENTITY_UID, is the subject of the relationship and the attribute, SCOPING_ENTITY_UID, is the object of the relationship. The attribute, ROLE_SEQ, indicates the distinctive instance of this particular role relationship for the subject UID. The attribute, CD, indicates the type of role for the subject relative to the object. For example, a subject might have the role of 'patient' to a care provider or 'member' of an HMO. The model supports specifying the time period during which a particular relationship occurred, but these are not filled in for this example. The explanations of each relationship are provided below the class.

Note that it is not necessary to indicate a Role for any entity. This is optional and would be entered by the user only when there is a functional advantage in doing so.

Key	ENTITY_UID	ROLE_SEQ	SCOPING_ENTITY_UID	CLASS_CD	CD FROM_TIME	TO_TIME
1	1	1		PATIENT		
2	2	1	1	PHYSICIAN	CARE_PROVIDER	
3	2	2	4	PHYSICIAN	PHLEBOTOMIST	
4	3	1	1	HEALTHCARE PAYOR	INSURER	
5	3	2	8	HEALTHCARE PAYOR	INSURER	
6	3	3	9	HEALTHCARE PAYOR	INSURER	
7	4	1	1	SPECIMEN	BLOOD_SPECIMEN	
8	5	1	1	HEALTHCARE_STUDY	HAS_PARTICIPANT	
9	5	2	8	HEALTHCARE_STUDY	HAS_PARTICIPANT	
10	5	3	9	HEALTHCARE_STUDY	HAS_PARTICIPANT	
11	6	1	5	EPIDEMIOLOGIST	DIRECTOR	
12	7	1	5	CASE WORKER	CONDUCTOR	
13	6	2	7	EPIDEMIOLOGIST	MANAGER	
14	8	1	1	FRIEND	SEX_PARTNER	
15	9	1	1	FRIEND	SEX_PARTNER	
16	10	1	6	PUBLIC_HEALTH_FACILITY	EMPLOYER	
17	10	2	7	PUBLIC_HEALTH_FACILITY	EMPLOYER	
18	11	1	4	ORGANISM	INDENTIFIED	

The following class provides a description of the above relationships

Key	Description
1	Joey has the role of patient (UID=1)
2	Dr. Daniels (UID=2) is a physician who is the primary care provider for Joey (SCOPING_UID = 1)
3	Dr. Daniels (UID=2) draws is a physician that draws Joey's blood specimen (SCOPING_UID = 4)
4	Peachtree Healthcare HMO (UID=3) is a healthcare carrier that insures Joey (SCOPING_UID = 1)
5	Peachtree Healthcare HMO (UID=3) is a healthcare carrier that insures Martin (SCOPING_UID = 8)
6	Peachtree Healthcare HMO (UID=3) is a healthcare carrier that insures John (SCOPING_UID = 9)
7	The material (blood specimen -UID=4) was drawn from Joey (SCOPING_UID=1)
8	The group study (the 'Partner Survey') has a participant named 'Joey' (SCOPING_UID=1)
9	The group study (the 'Partner Survey') has a participant named 'Martin' (SCOPING_UID=8)
10	The group study (the 'Partner Survey') has a participant named 'John' SCOPING_UID=9)
11	Dr. Rudolf (UID=6) is the epidemiologist that is the director of the Partner Survey (SCOPING_UID=5)
12	Helen Smith (UID=6) is a case worker in charge of conducting the 'Partner Survey' (SCOPING_UID=5)
13	Dr. Rudolf (UID=6) is an epidemiologist that manages Helen Smith (SCOPING_UID=7)
14	Martin (UID=8) is a friend and a sex partner of Joey (SCOPING_UID=1)
15	John (UID=9) is a friend and a sex partner of Joey (SCOPING_UID=1)
16	Atlanta Public Health Department (UID=10) is the employer of Dr. Rudolf (UID=6)
17	Atlanta Public Health Department (UID=10) is the employer of Helen Smith (UID=7)
18	An organism (UID=11) has been identified in Joey's specimen (SCOPING_UID=4)

12.4 Scenario 3. Representing Relationships between Entities and Acts

A major difficulty in health care informatics is representing the many different ways that people are involved in a health care activity. There are many types of activities and the various roles that people may play in each type vary widely. The solution that HL7 adopted and that is incorporated in the NLDM is to create a Participation class, which (as explained in section 3.3) is a way to represent the relationships between multiple entities. In this particular instance we use the Participation class to represent the relationships that one or many Entities (e.g. Persons) have with a particular Act. For example, for a particular laboratory test result the following different roles could be associated with specific individuals identified as persons and linked to the lab result through the Participation class:

- Subject from whom the specimen was taken
- Care Provider who ordered the test
- Lab technician that performed the test
- Lab organization that processed the test
- Epidemiologist that interpreted the results
- Case investigator that reviewed results with the subject

The following scenario illustrates how these important relationships between types of entities (e.g. persons, organizations, etc.) and health related activities may be represented.

12.4.1 Scenario 3 Description

The following scenario describes the following set of Observations made on Joey:

1. Dr. Daniels uses the blood sample taken during scenario 2 and orders another STD test on Joey's specimen from a national lab, NatLab.
2. NatLab performs the test and records a positive result. (Note: this is external to the NBS. This information is not recorded in NBS until step 4 below).
3. Based on STD reporting guidelines, NatLab sends a copy of the results to ATL PHD as well as Dr. Daniels.
4. The ATL PHD routes the positive STD lab result to Helen Smith who enters the lab result as an observation on Joey.
5. Helen sends a notification to Dr. Rudolf informing him of the suspected new case
6. Based on the results, Helen Smith initiates an Investigation of this event.
7. With Dr. Daniels' permission, Helen schedules an interview visit (patient encounter) with Joey
8. During the interview Helen completes a sexual history questionnaire noting that Joey has had regular, unprotected sexual encounters with Martin and John at Harry's Bar.
9. Based on Helen's interview results and the positive STD test, Helen sends a NND STD Case Report to the CDC.
10. Helen schedules interviews with Martin and John.

12.4.2 Scenario 3 Data Representation in NLDM classes

12.4.2.1 Entity

This scenario adds three Entities to the Entities super-class: the NatLab testing lab and Harry's bar.

ENTITY_UID	CLASS_CD
1	PSN
2	PSN
3	ORG
4	MTR
5	GRP
6	PSN
7	PSN
8	PSN
9	PSN
10	ORG
11	NLIV
12	ORG
13	PLC
14	CDC

Key to Entities:
Joey
Dr. Daniels
Peachtree Healthcare HMO
Blood sample from Joey by Dr. Daniels
Partner Survey
Dr. Rudolf
Helen Smith
Martin
John
Atlanta Public Health Department
Treponemal pallidium (Syphilis)
NatLab
Harry's Bar
Centers For Disease Control

12.4.2.2 Person

This scenario does not add any Persons to the Person class. It is copied here from scenario 2 for the readers' convenience.

PERSON_UID	STATUS_CD	TYPE_CD	BIRTH_GENDER_CD	PREFERRED_GENDER_CD	REPORTED_AGE	REP_AGE_TIME	BIRTH_TIME
1	A		M	F	34	4/3/95	1961
2	A						
6	A						
7	A						
8	A						
9	A						

Key to Entities:
Joey
Dr. Daniels
Dr. Rudolf
Helen Smith
Martin
John

12.4.2.3 Organization

Corresponding to the NatLab addition to the Entity class, NatLab is added to the Organization sub-class with the same UID, e.g. 13.

ORGANIZATION_UID	STATUS	CD	NAME
3	A	HMO	Peachtree Healthcare HMO

10	A	PHD	Atlanta Public Health Dept
12	A	LAB	NatLab
14	A	FHA	CDC

12.4.2.4 Material

Materials do not play a role in this scenario.

12.4.2.5 Group

Groups do not play a role in this scenario.

12.4.2.6 Place

Harry's Bar is added as a place so that activities may be associated to the their place of occurrence and so that the investigation may be linked to one of the places associated with people contracting STD.

PLACE_UID	STATUS	CD	NAME
13	A	PRES	Harry's Bar and Grill

CD: This is the main classifying attribute of the place. This code indicates what kind of place is meant using a code depicting places, such as hospital, park, lake, etc. In this example, "PRES" = Public Restaurant.

12.4.2.7 Role

As described above, any Entity may have zero, one, or many roles associated with it. In addition, an Entity may have these roles included in the participation relationship between itself and Acts. So that, for example, a person Entity may be the care provider in one set of acts and the patient in a different set of acts. Being able to capture and represent the different roles that different types of Entities play in the many types of Acts tracked in public health is critical. The associations between Entities and Acts that are specified in the Participation class may be either directly between the Entity and the Act or may be represented from the Entity to the Role to the Participation and then to the Act classes depending on whether the Role relationship is important to include in the association between Entity and Act.

The following class shows the Role entries for this series of scenarios. The entries to the Role class resulting from scenario 3 start at Key element number 31.

Key	ENTITY_UID	ROLE_SEQ	SCOPING_ENTITY_UID	CLASS_CD	CD	FROM_TIME	TO_TIME
1	1	1		PATIENT			
2	2	1	1	PHYSICIAN	CARE_PROVIDER		
3	2	2	4	PHYSICIAN	PHLEBOTOMIST		
4	3	1	1	HEALTHCARE PAYOR	INSURER		
5	3	2	8	HEALTHCARE PAYOR	INSURER		
6	3	3	9	HEALTHCARE PAYOR	INSURER		
7	4	1	1	SPECIMEN	BLOOD_SPECIMEN		
8	5	1	1	HEALTHCARE_STUDY	HAS_PARTICIPANT		
9	5	2	8	HEALTHCARE_STUDY	HAS_PARTICIPANT		
10	5	3	9	HEALTHCARE_STUDY	HAS_PARTICIPANT		
11	6	1	5	EPIDEMIOLOGIST	DIRECTOR		
12	7	1	5	CASE WORKER	CONDUCTOR		
13	6	2	7	PHYSICIAN	MANAGER		
14	8	1	1	FRIEND	SEX_PARTNER		
15	9	1	1	FRIEND	SEX_PARTNER		
16	10	1	6	PUBLIC_HEALTH_FACILITY	EMPLOYER		
17	10	2	7	PUBLIC_HEALTH_FACILITY	EMPLOYER		
18	11	1	4	ORGANISM	INDENTIFIED		
				Added during scenario 3			
19	12	1	4	HEALTHCARE_FACILITY	PERFORMING_FACILITY		
20	13	1	1	PUBLIC FACILITY	SERVICE PROVIDER		
21	13	1	8	PUBLIC FACILITY	SERVICE PROVIDER		
22	13	1	9	PUBLIC FACILITY	SERVICE PROVIDER		

Description of Role entries:
(Note: the descriptions of the Acts (events) are provided in the section on the Act classes)

Key	Description
19	NatLab (UID=12) performs the lab test on the sample (UID=4)
20	Harry's bar (UID=13) provides service to Joey (UID=1)
21	Harry's bar (UID=13) provides service to Martin (UID=8)
22	Harry's bar (UID=13) provides service to John (UID=9)

12.4.2.8 Act

The Act super-class provides the UID for an act, stores the type for the subclass, links the Act to the Act_id and Participation classes. It includes an HL7 RIM attribute termed the "mood_cd" which represents the context or meaning of an Act. In this example, a distinction is made between the order of an Act and the performance of the Act; that is the ordering of a lab test and the scheduling of a Patient Encounter vs. the performance of the lab test and the conducting of the interview during the Patient Encounter.

This scenario includes four types of Acts: Observations, Notifications, Work-Ups, and Patient Encounters.

ACT_UID	CLASS_CD	MOOD_CD
3000	OBS	ORDER
3001	OBS	EVENT
3002	NOT	EVENT
3003	WUP	EVENT
3004	ENC	APT
3005	ENC	EVENT
3006	OBS	EVENT
3007	OBS	EVENT
3008	NOT	EVENT
3009	ENC	APT
3010	ENC	APT

Key

Lab tests are ordered
Entry of lab results
Notification of lab results to Dr. Rudolf
Work-up based on lab result
Interview Scheduled with Joey
Interview with Joey Conducted
Record observation of sexual contact with Martin
Record observation of sexual contact with John
Notification to CDC of Case
Interview scheduled with Martin
Interview scheduled with John

CLASS_CD: This is the most general classifying attribute of the act class. This code indicates, at a high level, what kind of act is intended using a terminology defined by the PHCDM. Act values include case, notification, and intervention. In fact each specialization of act within the model is identified by a class code. However, other classes of act may be defined, but not included as model specializations when they do not have distinct attributes.

MOOD_CD: The activity mood code determines the meaning or context for the activity. The activity (corresponding to a verb in natural language) may be conceived as **an event that happened (event)**, an ordered service (order), a possible service (master), an algorithm for describing an event (definition), a planned act for a specific time and place (appointment), and **a goal (intent)** of health care. Each of these is a different mood. The mood code is critical to the design of this model. Without it, the model described here would be at least three times as big, in order to distinguish between the following: a) The definition of the act (e.g., a case or test definition); b) Health-related activities that are planned; c) Scheduled health-related activities; d) Health-related activities that have already occurred or been performed.

12.4.2.9 Participation

The Participation Class represents the associations between Entities and Acts. Multiple Entities of the same and different classes may be associated with an Act. The following table shows representative examples of these associations:

SUBJECT _ENTITY_UID	ACT_UID	PARTICIPATION_SEQ	TYPE_CD	ROLE_SEQ	KEY
1	3000	1	Subject	1	Joey is the subject patient of the lab order
2	3000	1	Ordering Provider	1	Dr Daniels ordered tests on Joey
2	3000	2	Specimen Procurer	2	Dr. Daniels obtains Joey's specimen for tests
4	3000	1	Specimen	1	Joey's specimen is used for testing
1	3001	1	Subject	1	Test results recorded on patient Joey
4	3001	1	Specimen	1	Specimen is used for testing
12	3001	1	Performer	1	NatLab performed testing on Joey's specimen
7	3002	1	Performer	1	Helen performs Notification activity (to Dr. Rudolf)
7	3003	1	Performer	1	Helen performs Workup act
1	3003	1	Subject	1	Joey is the subject patient of the workup

7	3004	1	Performer	1	Helen schedules interview with Joey
7	3005	1	Performer	1	Helen performs interview with Joey
1	3006	1	Subject	1	Joey is the subject patient of the interview event
7	3006	1	Performer	1	Helen records interview observation
1	3007	1	Subject	1	Joey is the subject of the interview event
7	3007	1	Performer	1	Helen records interview observation event
10	3008	1	Performer	1	Helen performs notification act (to CDC)
1	3008	1	Subject	1	Joey is the subject of the notification event
7	3009	1	Performer	1	Helen schedules interview with Martin
7	3010	1	Performer	1	Helen schedules interview with John

TYPE_CD: Identifies the particular kind of Participation that an Entity performs in the Act. In practice, there are very many different participation types whose names and responsibilities vary. The number and kinds of involved participants also depend on the special kind of service. The "Participation Type" vocabulary domain defines a few orthogonal axes along which Participation types may be defined more regularly. In this example, "Subject" is the subject of the act; "Performer" is the Performer of the act

12.4.2.10 Act_ID

Some of the Acts entered into the NBS system were also entered into a separate local system and the identifier assigned by that system (in this example, 'GA PH Sys A') was entered when the Acts were entered into NBS. These local identifiers are stored in the Act_ID class for each of the ACT_UID's.

ACT_UID	SEQ_NUM	TYPE_CD	ASSIGNING _AUTHORITY_CD	ROOT_EXTENSION_ TXT
3000	1	Placer Number	GA PH Sys A	22222
3001	1	NBS_ID	NBS	348220
3001	2	Placer Number	GA PH Sys A	346346
3001	3	Filler Number	GA PH Sys A	6893453
3002	1	GASYSA_ID	GA PH Sys A	339922
3003	1	GASYSA_ID	GA PH Sys A	99882233
3005	1	GASYSA_ID	GA PH Sys A	23929223
3006	1	GASYSA_ID	GA PH Sys A	223333
3007	1	GASYSA_ID	GA PH Sys A	1820739
3008	1	GASYSA_ID	GA PH Sys A	22392203

TYPE_CD: Code that designates the type of activity identifier (for example, placer number, filler number, lab reference number, etc.).

12.4.2.11 Act_Relationship

Just as different Entities may be related to one another through the Role associative class, the relationships between Acts may be specified in the Act_Relationship class. In this case, the UID's of each of the Acts are specified and the type of relationship between them as the type_cd.

Key	SOURCE_ACT_UID	TARGET_ACT_UID	TYPE_CD	COMMENT
1	3001	3000	PERTAINS	Order_Result
2	3002	3001	PERTAINS	NOTFI_OF_OBS
3	3001	3003	PERTAINS	RELATED_OBS
4	3004	3003	PERTAINS	SCHEDULED_INTRV
5	3005	3003	PERTAINS	INTERVIEW
6	3006	3005	PERTAINS	OBS_FROM_INTERVIEW
7	3006	3003	PERTAINS	RELATED_OBS
8	3007	3005	PERTAINS	OBS_FROM_INTERVIEW
9	3007	3003	PERTAINS	RELATED_OBS
10	3008	3001	PERTAINS	CASE_SUPPORTING_LAB_OBS
11	3008	3006	PERTAINS	CASE_SUPPORTING_SELF_REPORT_OBS
12	3009	3006	PERTAINS	FOLLOWUP_ON_OBS
13	3010	3007	PERTAINS	FOLLOWUP_ON_OBS

Key	Description
1	The lab results entered (act=3001) are from the order placed (act=3000)
2	The Notification (act=3002) is of the observation (act=3001)
3	The observation (act=3001) is associated with the Work-Up (act=3003)
4	The scheduled interview (act=3004) is part of the Work-Up (act=3003)
5	The interview (act=3005) is part of the Work-Up (act=3003)
6	The observation (act=3006) was made during the interview (act=3005)
7	The observation (act=3006) is part of the Work-Up (act=3003)
8	The observation (act=3007) was made during the interview (act=3005)
9	The observation (act=3007) is part of the Work-Up (act=3003)
10	The notification (act=3008) includes the lab result observation (act=3001)
11	The notification (act=3008) includes the interview observation (act=3006)
12	The scheduled interview (act=3009) is to follow-up on observation (act=3006)
13	The scheduled interview (act=3010) is to follow-up on observation (act=3007)

1

TYPE_CD: Determines the meaning of a relationship between two Acts. Each of its values implies specific constraints to what kinds of Act objects may be related and in which way. Refer to the USAM specification document for defined act relationship types and examples of their use. The type code reflects the nature of the relationship that exists between two or more associated health-related activities. The possible values include "comprises", "causes", "pertains" and "is associated with". An example of "comprises" relationship is a case definition that is comprised of laboratory tests, symptoms, and other qualifying criteria. An example of a "causes" relationship is a case notification causes a case investigation. An example of an "is associated with" relationship is an outbreak and the associated cases. "Pertains" is used to designate a non-specific relationship from one item of clinical information to another. It does not judge about the role the pertinent information plays.

12.4.2.12 Observation, Observation_Value, and Observation_Value_xxx

Of the ten Acts in this scenario, four are Observation sub-classes. The Observation sub-class consists of:

1. Observation Class which represents the type of Observation being made, such as a CBC (complete blood count)
2. Observation_Values, which represents the value or values resulting from the observation.
3. Depending on the format of the reported observation values, one of the following extension_classes: Obs_Value_Codes, Obs_Value_Txt, Obs_Value_Date, Obs_Value_Numeric.

The lab test order is represented in the database as an Act (UID=3000) with a mood_cd of 'order'. The entry in the Observation class indicates the type of test ordered along with miscellaneous data. (Note: Normally, tests ordered by primary care providers as in this scenario would not be entered into the NBS database. The entry of an order would, in practice, be more common for tests that are ordered for processing by the local public health lab).
The results of the lab test entered into the Observation_Value class and the extension of this class for the specific format of the result; in this case the results are in free text, 'Positive'.

The other two observations are results from the interview questions asked during the patient encounter. Two responses were given to the question of the frequency of 'unprotected anal intercourse' with named partners (the identification of the partners is represented in the Role entries associated with these observations through the Participation class entries). In this example, the observations (UID's = 3006 and 3007) were entered as coded responses of 'Frequent'.

Observation

OBSERVATION_UID	SUBJECT_PERSON_UID	CD	CD_Desc_txt
3000	1	Chlamydia/ RPR	Chlamydia trichomatis antibody and RPR (for Order)
3001	1	Chlamydia/ RPR	Chlamydia trichomatis antibody and RPR (for Result)
3006	1	INTER_QUES	Interview Question
3007	1	INTER_QUES	Interview Question

CD: A code specifying the kind of action (e.g. physical examination, serum potassium, patient encounter, financial transaction, etc.). The Observation.cd specifies the act conceptually using a code from one of several, typically external, coding systems depending on the class of act, such as observations (LOINC), procedures (e.g., SNOMED), medication treatments (e.g., UMLS), etc.

Observation_Value

OBSERVATION_UID	OBSERVATION_VAL_SEQ	OBS_VALUE_CD	OBS_VALUE_DESC
3001	1	Chlamydia	Chlamydia trichomatis antibody
3001	2	RPR	Rapid Plasma Reagin (Syphilis)
3006	1	UNPR_ _SEX	Unprotected Sex

3007	1	UNPR_ SEX	Unprotected Sex

OBS_VALUE_CD: which observation results are reported (e.g. physical examination, serum potassium, patient encounter, financial transaction, etc.). The Observation_value.obs_value_cd specifies the act conceptually using a code from one of several, typically external, coding systems depending on the class of act, such as observations (LOINC), procedures (e.g., SNOMED), medication treatments (e.g., UMLS), etc.

OBS_VALUE_DESC: Textual description of the observation value type code.

Obs_Value_Txt

OBSERVATION_UID	OBSERVATION_VAL_SEQ	OBSERVATION_VAL_TXT_SEQ	VALUE_TEXT
3001	1	1	Positive
3001	2	1	Positive

OBSERVATION_UID	OBSERVATION_VAL_SEQ	OBSERVATION_VAL_TXT_SEQ	VALUE_TEXT
3006	1	1	FREQUENTLY WITH MARTIN
3007	1	1	FREQUENTLY WITH JOHN

12.4.2.13 Patient Encounter

In this scenario, the context for the interviews is represented as Patient_Encounters. This context is used both as part of the investigation workflow recorded in the Investigation Work-Up class and to provide a link for each of the observations made during the interview.

PATIENT_ENCOUNTER_UID	ACTTIVITY_FROM_TIME	ACTIVITY_DURATION UNIT_AMT	CD	CD_DESCRIPTION_TEXT
3004	11-Sep-01: 1pm		INTRV	Sexual History Interview
3005	11-Sep-01: 1pm	45 minutes	INTRV	Sexual History Interview
3009	15-Sep-01: 3pm		INTRV	Sexual History Interview
3010	19-Sep-01: 9am		INTRV	Sexual History Interview

ACTIVITY_DURATION_UNIT_AMT: The duration amount provides an indication of the time period during which the patient encounter is valid. The "biologically relevant" time for a health related activity. In this example, the interview (act UID=3005) lasted 45 minutes.

12.4.2.14 Work-Up

The investigation Work-Up provides a view into all data related to an investigation and allows for the collection of temporary workflow information associated with the process of conducting the investigation as opposed to the results of the investigation which are represented as Observations, Public_Health_Cases, and Notifications.

WORKUP_UID	CD	CD_DESC_TXT	ACTIVITY_FROM_TIME	ACTIVITY_TO_TIME
3003	STD	STD CASE WORKUP	12-Sep-01	

12.4.2.15 Notification

The Notification class is used to record both the traditional NND notifications as well as informal notifications among public health workers. The use of the informal notification allows the sender and receiver to directly link to associated data in the database rather than having the indirect references included in letters, notes, emails and phone conversations.

NOTIFICATION_UID	CD	CD_DESC_TXT	MMWR_WK	MMWR_YR
3002	APHD	NOTITIFICATION TO APHD		
3008	CDC	NOTIFICATION TO CDC	36	2001

13 APPENDIX A – NEDSS PROGRAM AND NLDM GLOSSARY

Acts
The RIM and NLDM contain two major super-classes: Entities and Acts. Acts represent health related activities that 1) (usually) have a defined beginning and end time with corresponding duration, and 2) involve one or more Entities

Application Extension Domain
One aspect of the Extensibility feature of the NBS provides the capability to add fields to NBS pages and attributes to the database without the need for modifying NBS code or the underlying database tables. Specific additions are identified by 'application extension domains', (such as specific program areas, states, or state programs) so that the NBS session management software can properly invoke them.

Attribute
An attribute defines one characteristic of an entity, such as the 'height' of a person. It is always used in the context of a particular class (entity)

Class
Class and entity are common data modeling terms and may, for most purposes, be used synonymously. The term entity is used when one is using a relational data modeling approach and the term Class is used when one is using an Object Oriented approach. They refer to concepts or things that have properties or characteristics and that may have relationships to one another. For example, a person may be represented as an entity that has characteristics such as name and address. A person may also have relationships with other entities, such as organizations to which they belong or other persons with whom they have role-based relationships. For example, a person may be a patient of another person acting in the role of care provider. The characteristics of the entities are defined as 'attributes' and the values stored for the attributes are called 'attribute values'.

The HL7 RIM uses the Object Oriented modeling technique. However, they have named one of the classes 'Entity', which sometimes leads to confusion with the relational modeling concept of entity. This Guide remains consistent with the HL7 terminology as much as possible, so when the term Entity is used in the capitalized form it refers to a particular type of class, not to the data modeling concept of an 'entity'.

Column
The term column is used to represent the physical implementation in a database of an Attribute.

Data Mart
Similar to a Data Warehouse but contains a subject area specific subset of the data.

Data Warehouse

A set of classes or tables (either physical for logical) whose structure is optimized to support data analysis as opposed to transactional activities. Changes to data in the ODS are copied to the Data Warehouse on a periodic basis determined by the importance of the changes for inclusion in the analyses.

Database Management System (DBMS)
A database management system is a software product that manages the storage of data on disks and provides a high-level programming interface to applications. DBMS's also provide a set of tools to help manage and administer the database. When the database is based on relational technology, the term Relational Database Management System (RDBMS) is sometime used.

Entity
A term used to describe person, thing, or concept about which information is stored. Fundamentally entity is synonymous with Class when describing only the structural properties of a Class as opposed to both the structural and behavioral properties .

Extended Attributes
Extended Attributes are attributes that may be added to any class by the administrator of an NBS compliant system without changing the database structure or programming code. These attributes are also linked to newly defined fields on the application web pages. To the user of the system they are treated as Named Attributes but are actually stored in the database as Extensible Attributes. See section 9 for more details.

Extensible Attributes
Extensible attributes are generic attributes that allow storing data on any property of the class (entity). This is accomplished by using two attributes to represent the property and its value rather than one as with Named Attributes. The first of the two attributes is the attribute that contains the name of the attribute as a value, the second attribute contains the actual value for the named attribute. These two attributes are commonly referred to as name-value pairs. By allowing the name of the attribute to be variable, the Extensible Attribute may be used to represent and store the value of any characteristic of the class.

One example of an Extensible Attribute is combination of Observation:'cd' and Observation_value:'obs_val_cd'. The attribute 'cd' contains the name component of the name-value pair and the attribute 'obs_val_cd' contains the value part. See section 5.2 for more detail.

Health Level 7 (HL7)
A standards development organization that develops standards for the exchange of health care information.

HL7 Reference Information Model (RIM)
A conceptual model developed by HL7 to facilitate common data model semantics, terminology, and structures

Investigation (public health)

The term 'investigation' is used to refer to the specific actions during a work-up where the investigator assesses the collection of information against one or more case definitions to determine whether the set of data meets a case definition.

Instance
Instance has two somewhat related meanings in data modeling. One usage is a single occurrence of an Entity, such as one specific person. It corresponds to one row in the table and represents all of the data stored for that class at that point in time. Another usage refers to one occurrence of a database. That is all of the data for all tables in a database. For example, a state could have both test and operational instances of a NBS database. In this Guide, unless the usage expressly states that the instance is of a database, the term instance refers to an instance of a row in a table. Instance is synonymous with 'occurrence'.

Jurisdiction
The term jurisdiction is used to represent the Geopolitical area served by a public health system. It is a key dimension of the NBS security sub-system.

Local Identifiers
Local identifiers are identifiers assigned by some authority that is not, necessarily, widely known. For example, an identifier assigned to persons by a local public health application would be considered a local identifier. Clearly, there isn't strict differentiation between Standard and Local identifiers. The practical distinction is that when one exchanges a Local Identifier one does not assume that the receiving person or system may associate it with the assigning authority.
Common examples of local identifiers are the registration numbers assigned by the ADT systems in the various clinics of a large hospital. Resolution of local identifiers is usually accomplished using some form of Master Patient Index (MPI) or Patient Registry application.

Message-In
The Message-In is the set of classes into which data received via electronic messages or entered by partners using restricted use web forms are placed for validation before being stored in the ODS classes. For example, electronic lab results are received and translated from the message format into the relational format and stored in the Message-In. These data are then edited, validated, and checked for potential relationships to data already in the ODS before being moved from the Message-In to the ODS.

Message-Out
The class where formatted messages are placed for routing by the message router for sending to the named recipients.
.
Named Attribute
A named attribute is an attribute that has a pre-defined name that is used to identify a specific column in a database table. Values stored in that column always represent values of the named attribute.

Name-Value Pair (NVP)

Name-Value Pair is a general term that describes a way of representing data where the name of the value is treated as data in the same way as the value is treated as data. The Name-Value-Pair technique is used in various situations, for example it is the technique used in XML. When used with relational technology, both the name and the value are stored as attributes of a class. For example, a person's weight could be stored in a relational structure as an attribute named 'weight'. The measurement of a person's weight would be stored as the value of the attribute 'weight'. Alternatively, the NVP approach would involve two attributes: one would contain the name of the attribute and the second would contain the value. Continuing the weight example, the name attribute in the table might be 'observation type' and the attribute value might be 'observation value'. When data are entered for a person's weight, they would enter 'weight' as the data value for 'observation type' and the actual measured weight for 'observation value'.

NVP's are used extensively in the NLDM and are described in more detail in section 5.2.

NEDSS Base System (NBS)
The CDC developed implementation of the NEDSS Architecture. See the NEDSS Technical Designers' Guide for specifics.

NEDSS Program
The NEDSS Program is a collaboration of National, State, and Territorial agencies, along with allied professional organizations, to improve public health through the enhanced sharing and use of information. The NEDSS Program consists of many initiatives. These include working with various standards development organizations (SDO) to address data standards important to public health, developing and sharing best practices, sponsoring workshops and conferences where public health professionals may share best practice experiences, development of a Architecture for IT support of public health, and development of a Reference Implementation of the Architecture that may be optionally deployed and enhanced by public health agencies.

The two NEDSS technical initiatives that are most closely related to the NLDM are:
1. The NEDSS Architecture, sometimes just called the NEDSS Architecture. 2. The reference implementation of the NEDSS Architecture. This is a complete implementation of the NEDSS Architecture that is made available to the states as an operational system. The implementation is termed the NEDSS Base System (NBS).

The NLDM described in this Guide is a part of the NEDSS Architecture and is generally referred to as the NEDSS Logical Data Model (NLDM). One of the possible physical implementations of this NLDM is provided as part of the NEDSS Base System Option. The operation of the NBS depends on the use of specific implementations of the NLDM, commonly referred to as NBS Physical Data Models, (NPDM). Actual databases that implement the NPDMs are dependent on the actual DBMS used.

Occurrence
Refers to once instance of a business object or class, for example, one particular individual in the Person class. Occurrence is synonymous with 'instance'

Operational Data Store (ODS)

The operational data store is the set of tables that store the data during the period of time when the entities represented by the data are subject to some form of transactional activity. That is when values for instances of the classes may be changed to support public health operational needs, for example values in a Case Report being updated during an investigation.

The term is typically used to distinguish this type of data store from a Data Warehouse where the data values are not changed and are used for analysis only. Once there is no longer a need to change values of data for a class, that data can be archived and deleted from the Operational Data Store since it is still available for analysis and reporting in the Data Warehouse.

Program Area Module (PAM)
An extension to the NBS that provides functionality that is specific to one program area. That is, the NBS provides functionality that is common to all surveillance functions and in the future the NBS will be extended to include common public health functions in addition to surveillance. However, there is functionality that is specific to the investigation and treatment of specific diseases and there are sets of attributes (i.e. set of data elements) that are specific to particular diseases. Where these conditions exist, focused teams are formed to analyze and develop these disease specific functions as PAM extensions to the NBS.

.

Row
The term Row refers to the set of data values stored for each column in a table for a particular instance of a class. It represents the data stored for one instance of a class or entity. Generally synonymous with Tuple, instance, and occurrence.

Schema
The term schema is used to describe the structure of a data model. It includes the classes (entities), the relationships between them, and sometimes the attributes for each class. It is most commonly used to describe the structure of a physical database, which is the way it is used in this Guide.

The term schema has recently been used for a similar construct when referring to the structure and organization of an XML message.

Standard Identifiers
Standard identifiers are identifiers that are assigned by a widely accepted authority, such as Social Security Numbers, Driver's License numbers, Medicaid numbers, etc.

Table
The term table is used to describe the physical implementation in a database of a Class or Entity.

Tuple
The term, tuple, is used synonymously with the term 'row'. Less specifically it is also used to refer to an instance or occurrence of a row.

Unique Identifier (UID)
Unique identifiers (UID's) are system-generated, specific numbers used to uniquely identify instances of entities. They are implemented as the primary key for the entity.

Use Code
A use-code is a special type of code that indicates the 'usage' of an attribute or set of attributes. For example, addresses may have 'use-codes' describing how the address is used, such as: home, employment, etc.

Work-Up
The term Work-Up is used to capture the information that a public health worker records about the **process** of their investigation. For example, work queues, requests for information, appointments, notes, etc. It also provides links to the results of the work-up, including: observations, public health cases, interpretations, notifications, etc.

14 APPENDIX B – Currently Defined Coded Attributes and Values

Appendix B is contained in a separate file named, NLDM DG Appendix B Code Sets.