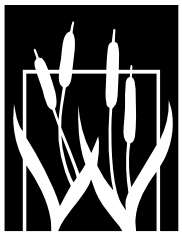

Whitemarsh
Information Systems Corporation

Modeling Data and Designing Databases

Whitemarsh Information Systems Corporation
2008 Althea Lane
Bowie, Maryland 20716
Tele: 301-249-1142
Email: Whitemarsh@wiscorp.com
Web: www.wiscorp.com

Table of Contents

1. Objective

The objective of this ***Whitemarsh Short Paper*** is to present an approach to the comprehensive, efficient and effective process of creating data models through mission, database domain, and localized entity-relationship modeling. The paper then concludes with an overall process and check list for designing databases that is supported by a reference to a different Whitemarsh paper. ***Whitemarsh Short Papers*** are available from the Whitemarsh website at:

http://www.wiscorp.com/short_paper_series.html

At first, it might seem strange to separate the process of modeling data from the process of designing databases. Not only are these topics really different, but if the first is not done before the second, then not only will you have a bad database design, but you will also have a high probability that the databases are not-integrated, are redundant, and that definitely possesses conflicting semantics. These are the inverses of desired characteristics.

Simply stated a schema and its contained tables represent a completely encapsulated set of data specifications within the "hard boundaries" of its schematic (hence, it's within a schema). In contrast, a model of data is simply a collection of entities, attributes and relationships among the entities across one or more subject areas. Data models have no real hard-boundaries, only abstract boundaries. Data models are supposed to represent the semantics of natural collections of data specifications. Schemas are specifications of semantics of data specifications contained within a database that, in turn, lead to actual instances of stored data. A data model is a data-type collection specification. A database is a data-instance-collection specification. Database models are then the use and re-use of various data-type collections.

Case Study

A school system needed to build a database application (an application where the database is the core component). The problem domain was to track individuals who were being trained to be Health-Room Medical Assistants (HMA). The school system has about 140,000 students, exists within an urban setting, reports to an urban government that exists within a U.S. State.

There are about 210 health rooms across the school system, and whenever the RN (Registered Nurse) is absent from a school, an HMA has to handle the dispensing of medicine, medical record keeping, and first aid. The HMA can only be authorized to perform health-related activities through a "delegating" RN. The HMA "works under" the delegating RN's license.

Schools principals are responsible for selecting HMA candidates. They must pass a reading, writing, and arithmetic test. If the test is failed, the HMA candidate cannot come back to the HMA training class for 90 days. The HMA nominee must “sit” for a 40-hour class. There are three tests during the class, and a practical demonstration-based exam at the end. If any test is failed, the HMA candidate is “expelled” and has to retake the entire course starting with the entrance test.

Once an HMA has passed the tests they are deployed. Thereafter, there are follow-up

reviews and direct observation every 45 school days. After the first 45-day period, the HMA candidate can apply to the State to become a certified HMA. If after certification, the HMA fails a follow-up review and/or any direct observation, the supervising RN must report that failure to the State Board of Nursing. The Board of Nursing may decertify the HMA. If decertified, the process begins again starting with the entrance test. The rationale for all this is that giving the wrong drugs or dosage to a student, or failing to properly document medical treatments is a serious matter that might cause harm or even death to a student. This, the State's Board of Nursing takes seriously

The task was to build the HMA database application against these requirements. A nurse supervisor of the school system's student health organization chose MS/Access to build the application based on guidance from the school system's IT department. Since the nurse supervisor was told that MS/Access is really simple, and since the nurse supervisor was a HMA subject matter expert, she just fired up Access on her computer and started to type in the database's design: One table, of course. As she discovered something she forgot, she just modified the database's one table design, one column at a time.

After about nine months, the one-table database in MS/Access was complete. She started to type in data for about 300 records of HMAs in various stages of the certification process. Of course "she" herself was required to be present in order to understand and explain each and every row of data and all the implied relationships among the columns. There were multiple status codes, some of which are related to others. There was no history.

During a review of the database's design, and in light of the requirements stated above, questions were asked about the underlying process, test re-takes, test failures, observations, re-certifications, how HMA moves from one school to the next are recorded, how changes in RN delegation are handled, and the automatic refreshing of names, addresses, and schools from the school system's databases. None of these questions had been accommodated within the database's design.

A question was then asked, "Where's your model of the data?" She stated, "Oh, right here." She showed the MS/Access database table. She was not aware that a model of the data is different from a database's design. After this review, the next two hours were then spent figuring out the model of the data. At the end of this first data model design session about 15-20 entities were identified and related. At this point she knew that these requirements were beyond what she knew how to do with Access. What was starting to emerge was the need for: 1) a data model that would mirror the real requirements, and 2) an application system to manage the user interface, underlying processes, data entry, errors, relationships, editing, workflow, and reporting.

What should she have done? That's the subject of a short paper. The paper was written as if she had all the necessary tools to accomplish all the steps. Did she have these tools? Would the school system expend the resources to make "simple, little applications" like these practically possible?

A good, sophisticated tool set like Clarion for Windows (www.SoftVelocity.com) make both client/server and Internet-based applications such as these very practical and cost effective to accomplish.

2. Topics Covered

The topics in this paper include:

- How to discover the data model
- Where to put your data model
- How to build a database design

3. How to Discover the Data Model

The overall process of modeling data is provided in Figure 1. Instantly you may be asking, "Why are functions and organizations involved in modeling data?" Won't that lead to a stove-pipe functional specification? The answer to those questions is simple: If you don't know, in a general way, what your going to do with the data, nor who's going to be using it, then how can you know if you've got the right model?

However, if you treat the functions and organizations as "hard" outside boundaries then yes, a stove-pipe database design might result. If, however, organizations and functions merely serve as contextual sources then no, while the data model design is focused, a stove-pipe database will not necessarily result.

Some suggest, in contrast, to using functional and organizations contexts that you just cover a whole wall in a 50x 30 room with butcher block paper, put a rectangle in the upper left corner, write "Customer," and then wait for divine intercession. Practitioners of this approach have grown old and retired waiting for the revelations. In addition, such a strategy often results in exploded budgets and employment termination. Clearly, these are not desired outcomes.

A careful review of the process model in Figure 1 shows that neither "schema" nor "database" (other than database domains) is specified. That's because it's a process to model data. Thereafter, the data model can be employed to help create database schemas that are, of course, an essential requirement for database-centric applications. Provided here are summary descriptions of these data modeling processes, and reasons why they are performed. Detailed descriptions including specifications of the data model products that are built and their interrelationships are all provided in Section 7, References.

3.1 Missions

Missions are the idealized descriptions of what the enterprise is all about, ***not*** what is done on a day to day basis with respect to a specific narrow function. Identification and specification of the relevant missions are critical. In the case study, the focus was on the database's design to accomplish known processes. There was no overall understanding of how these collections of processes fit into the overall mission of the school system.

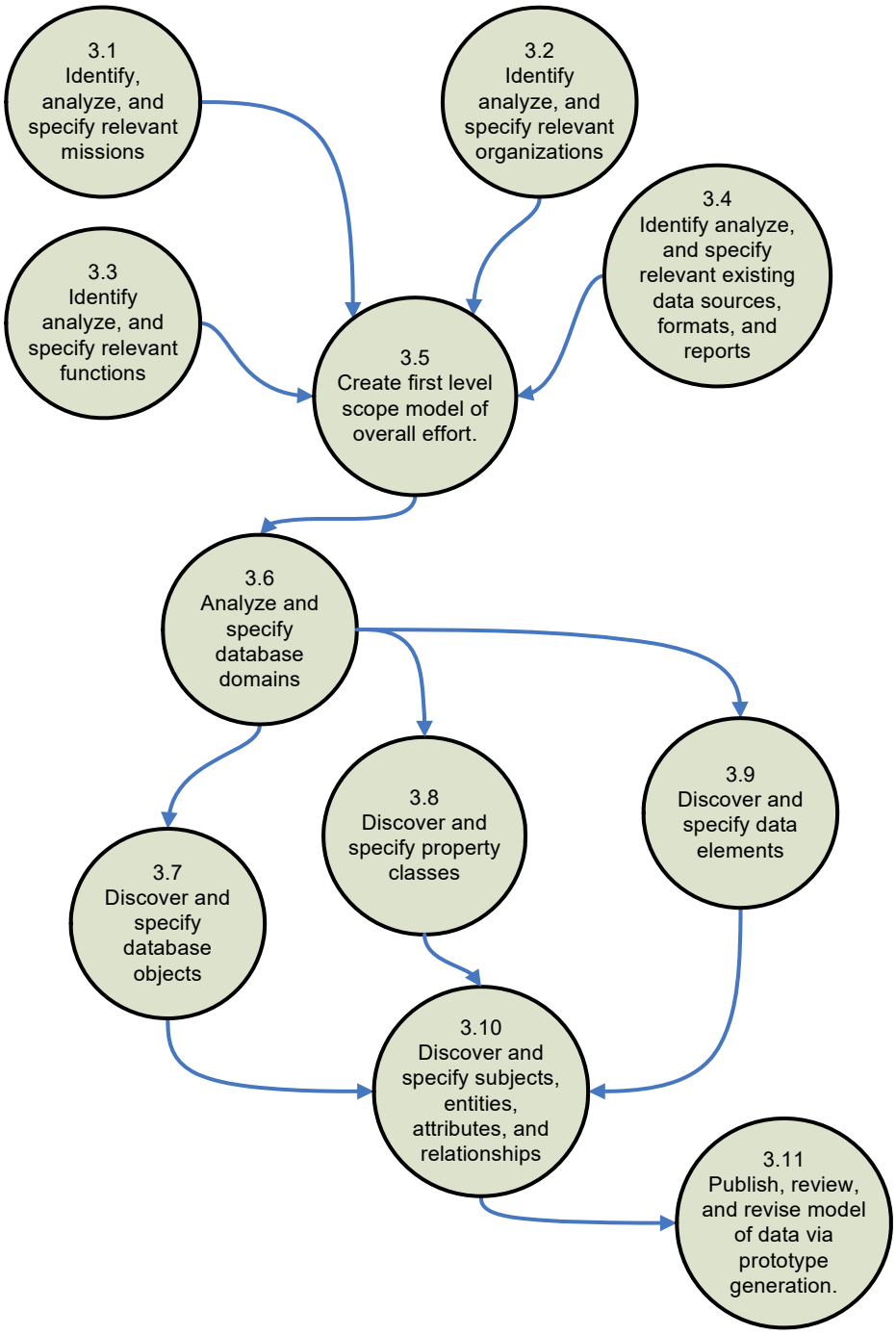

Figure 1. Data modeling process flow.

Because the nurse supervisor was focused only on the database's design, she missed the opportunity to identify all the different objects (in the broadest sense of the word, object) that should have been involved. Her focus was on the HMA only. Mission isn't the sole answer, however. Rather, missions are the context within which the data model answer is derived.

3.2 Organizations

Organizations accomplish aspects of missions with databases, information systems and functions. In the case study, the enumeration of the various organizations serves to identify the groups that have to be involved in getting, maintaining, and using the information. For example, HMAs may be transferred from one school to the next. Knowing that, and knowing that probably the last thing on the mind of the transferred HMA is to notify someone in the school system's student health organization of the transfer, how to know the current HMA's school is a real problem. The school system's HR organization should have the most current assignment for a person. Consequently, a feature of the resulting HMA system would be to access, on a weekly basis, the HR system's database, and validate, HMA by HMA whether the current HMA assigned location is the correct one. If it is, then OK. Otherwise, that school would have to be flagged as no longer having an available HMA.

Again, organization, like mission, is not the answer to knowing the full set of requirements. Rather, it too is a contextual element and provides the identification of groups that have to be involved in the full set of requirements for a complete solution.

3.3 Functions

Functions are the human-based procedures performed by persons as they accomplish the various missions of the enterprise from within different enterprise organizations. This is where the actual activities are specified that need to be performed. Setting out and defining functions cause the identification of areas of data, data elements, states, exceptions and the like involved in the effort. Functions are all identified within the scope of the mission-organizations that result from the previous step. These functions directly lead to the set of processes, automated and manual, that are needed in some way for a complete HMA solution. Common functions would include for example:

- HMA candidate registration
- HMA class formation
- HMA entrance test administration
- HMA instructor assignment
- HMA class room acquisition
- HMA attendance recording

- HMA class test administration
- HMA medical records management
- HMA practical test administration
- Certification application
- HMA infraction reporting

3.4 Prior Data and Reports

Prior data and reports generally refer to the set of all data that must be involved in the application, and by inference also in the data model's design. In the case study, there is Board of Nursing requirements for HMA applications, transactions, and reporting. There are also data requirements to support the individual HMA person such as biographic data, assignments, tests, scores, and HMA progress states. All these data need to be identified and analyzed within the context of the missions, organizations, and functions as they directly lead to subjects, entities, attributes, and of course, relationships.

3.5 Scope Model

The above four components: missions, organizations, functions and prior data are all brought together to convey one overall understanding of just what the mission is, which organizations are involved, what activities are performed, and just what data is involved in accomplishing the effort.

Once the scope model is complete, then from a non-IT point of view, the specification is not only complete, it should also be able to be reviewed for missing items and functionality. If any are found, then they should be corrected before proceeding. Correcting these errors here is very inexpensive as no databases are yet designed nor has any software been built. Once the scope model is as complete as it can be, then, the next step, database domains, can proceed quickly.

3.6 Database Domains

Database domains are text or list based noun-intensive statements. Database domain paragraphs are the precursor step to entity-relationship models. Well done, each sentence in a Database Domain's text becomes a multi-entity and relationship statement. Here, there's no intended granularity to an entity. Some entities may end up being complex (i.e., database objects), others become named-classes of properties, and some just data elements.

From the point of view of the case study, database domain statements example statements include:

- School system has health rooms.
- HMAs have delegating RNs
- Schools have certified HMAs
- School system delivers HMA classes
- HMA classes have certified teachers
- School system employees are enrolled in HMA classes
- School system HMA enrollees have passed (or failed) HMA tests
- Candidate HMAs apply for State Certifications
- State Certified HMAs must re-certify every two years from date of original certification

Once database domain sentences and paragraphs are complete, the various entity relationship diagrams are created, subject by subject. Relationships can be made between entities in different subjects to provide a more uniform model of the data.

Once drawn, the entity-relationship models should be validated by reviewing the subject-verb-object relationships and cordialities with subject matter experts. If any statement is missing, now is the time to add it.

At this point, since the data and its context is known, the model of the data can begin in earnest. This is accomplished by dealing with the various database domain components, that is: Database objects, property classes, and data elements. Once the data model is complete, the database's design can be accomplished.

3.7 Database Domain Component: Database Objects

Database objects are collections of entities (which, in turn, often become collections of tables within a database's schema). Full database object specification includes:

- A data structure of a table collection including columns.

- Table-based add, delete, and update processes to ensure correct atomic processes.

- Database object value-states that represent whole state-transforms of the database object's collection of table rows.

- Database object information system that accomplishes the value-state transform specifications.

But for the purposes of this paper, only the first component is accomplished. For this component, only a database Object Class diagram needs be created. This diagram is almost always

hierarchical that contains a root-table with the name of the database object.

There after, as the business information system is constructed, the database object models are completed for a coherent understanding of all the processes needed to ensure an database and business information system integrity. Database objects are extensively described within documents via references in Section 7.

In the case study, the database objects are likely School System, Schools, Teachers, Employees, and HMA enrolles. The are database objects because each likely contains multiple property classes such as names, addresses, current and prior assignments, education, positions, locations, and the like.

3.8 Database Domain Component: Property Classes

A property class is a collection of data-based properties. Each property represents a single value. From the database domain statements, the "HMA class" probably a property class, which, in turn, becomes an entity as it merely contains Class Identification, Class Start and End Date, Class Location, etc. "Certification Date" is a data element as it represents an atomic non-derived value.

Similarly, the HMA test is likely a property class because it would only contain an ID, Data of Test, HMA Enrollee Id, and Test Grade. State Certification would be property classes for the same reason.

3.9 Database Domain Component: Data Elements

During the process of creating attributes within entities, each attribute should be mapped to enterprise-wide data elements as a way to maximize semantic uniformity across differently named attributes that represent the same data element, and also to ensure a uniformity of value domains. Once data elements are created and mapped to attributes, "where-used" reports can document maximum semantic integrity.

In the case study, Certification Date is the data element, and the columns, Certification Date, and Renewal Certification Date are specializations of the data element that are mapped to it.

3.10 Data Models

If the steps above are accomplished, the data model for the HMAs can begin. It should be clear that much more data will have been discovered and defined than needs to be in the HMA

database. For example, for most of the database objects merely need to be the root table needs to be represented in the data model mainly because that table becomes a reference table from an authoritative data source. If this effort is part of an overall larger modeling effort, many of the subjects, entities, attributes, and relationships will already have been created and stored in a metadata database (e.g., the Whitemarsh Metabase). Having these available from a metadata repository such as the Metabase System greatly speeds these steps. The remaining step is to actually create a database schema of just the relevant data specifications and then proceed through prototyping to then validate the data model.

In the case study, a very quick set of entities was discovered, and included for example:

- HMA Instructor
- School System Employee
- HMA Class
- HMA Test
- HMA Test Score
- In-school Observation
- Health Room
- HMA Assignment
- State Certification Record
- Supervising Registered Nurse

3.11 Validation through Prototyping

This step, validation through prototyping, is critical because up to this point the solution has been almost entirely paper-based. Even if data models are in a metadata repository, they are still untested designs. Data model design testing requires a process model through which the data model can be exercised to know that it represents the necessary data in an easy-to-use way.

The first step is to make a database schema. This is accomplished by importing collections, or subsets of collections of entities into a database's schema. Once represented as a database schema, a code generator that ingests the schema can materialize an actual database-centric application.

Once the application is generated, it can be electronically "shaped" to match a desired behavior model, and then demonstrated to users. User feedback is essential because that's when you hear, "Where's the XYZ data?" Reviewing something on paper is just not sufficient. Once a review is done, and the comments are all logged, just throw the generated application away. If there are changes to the data models, then make the changes and regenerate the prototype application. Once a series of cycles is performed, eventually the application and the data model will become stable. If there are six such cycles, all within a two-month period, then that's like implementing at version six versus implementing at version one, and then expending two or three years achieving the same design. For sure, this strategy is faster, cheaper, and far more

effective.

In the case study, the database's schema was able to be quickly created because all the entities, attributes and relationships existed in the metadata repository. A schema with all the tables was created in one afternoon. It was reviewed. The application was generated about two hours later. Another day was spent pruning the application prior to demonstration. Then, there were six sessions with relevant subject matter experts set up; one every two weeks. That was more than enough time to cycle back in the changes to the data model, the database's design, and the generated application.

Figure One of the early data model graphics for the HMA data model is

4. Where to Put Your Data Model

The data models should be placed into a metadata repository that is integrated and nonredundant. The model of such a metadata repository is similar to the one in Figure 2. Each box represents a category of metadata and is sometimes called a meta-entity. This is a very high level subset representation of the metadata repository data model. The diagram shows that the obvious places where the results from steps 3.1 through 3.9 are stored. Step 3.10 produces the metadata for the subject, entity, and attribute meta-entities. Relationships among entities are not shown. Step 3.11 causes the creation of the metadata for the business information system, module, database table, and column meta-entities. Relationships among tables are not shown. Figure 3 presents one of the early data model graphics for the HMA.

It is sometimes stated that metadata repositories are too expensive and take too long to implement. Thus, they are an impractical solution. This is not true because if you are an IT data management professional, then a metadata repository capable of storing all the metadata identified in this paper and performing all the functions this paper describes can be purchased for less than one-day's consulting fee. A fully sophisticated code generator can be purchased for less than three-days of consulting fees. The products exist. The way and the strategy exist.

If these tools are used on this project, then they are also available for other projects. If properly used, these tools enable data integration and reuse such that subsequent projects are of higher quality, lower risk, lower cost and the involved staff are much more productive.

There is no downside to increasing the sophistication of the workers and their work tools. The sophistication of solutions increases, the ability of non-IT staff to accomplish work of this nature increases, and the overall data integration and reuse greatly expand. Again, there is no downside to employing more sophisticated tool sets.

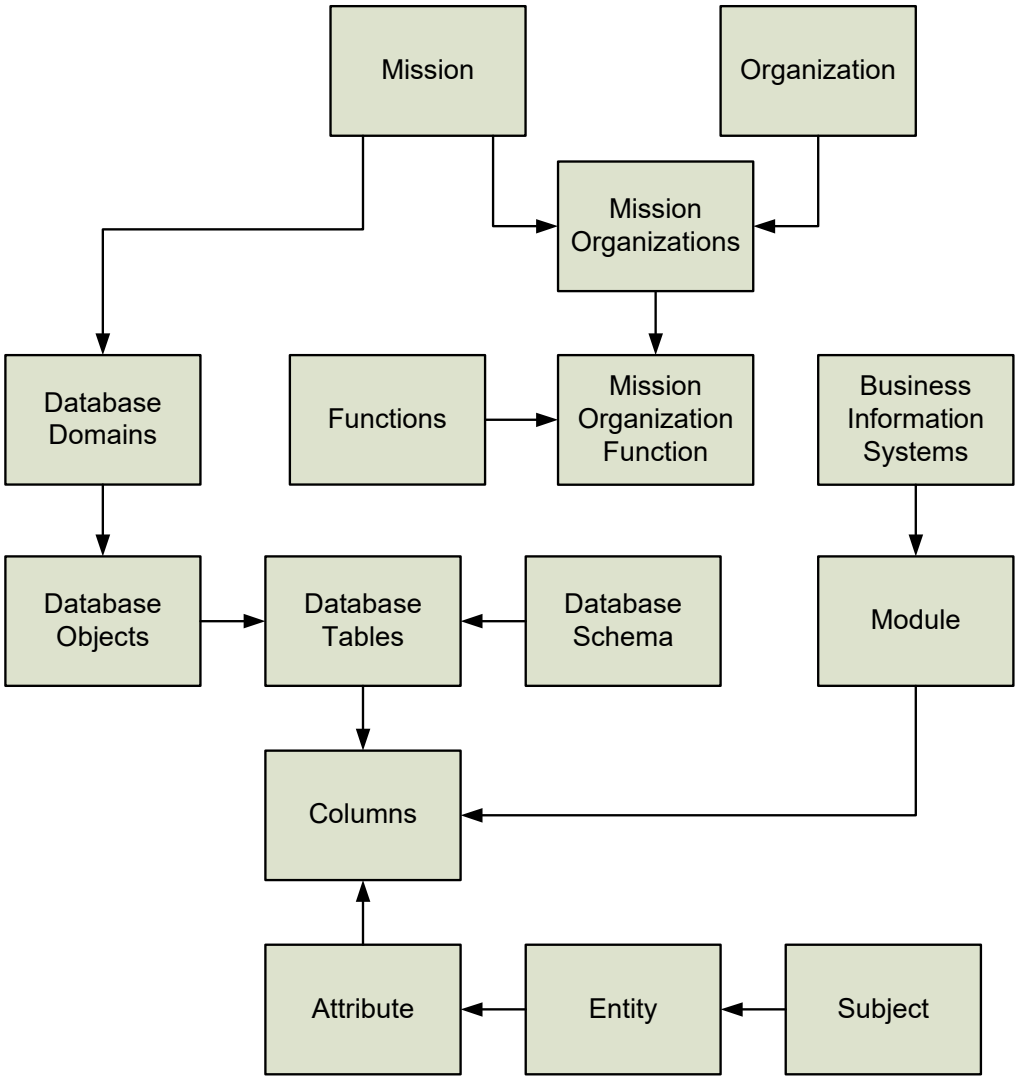

Figure 2. High level representation of metadata repository subset.

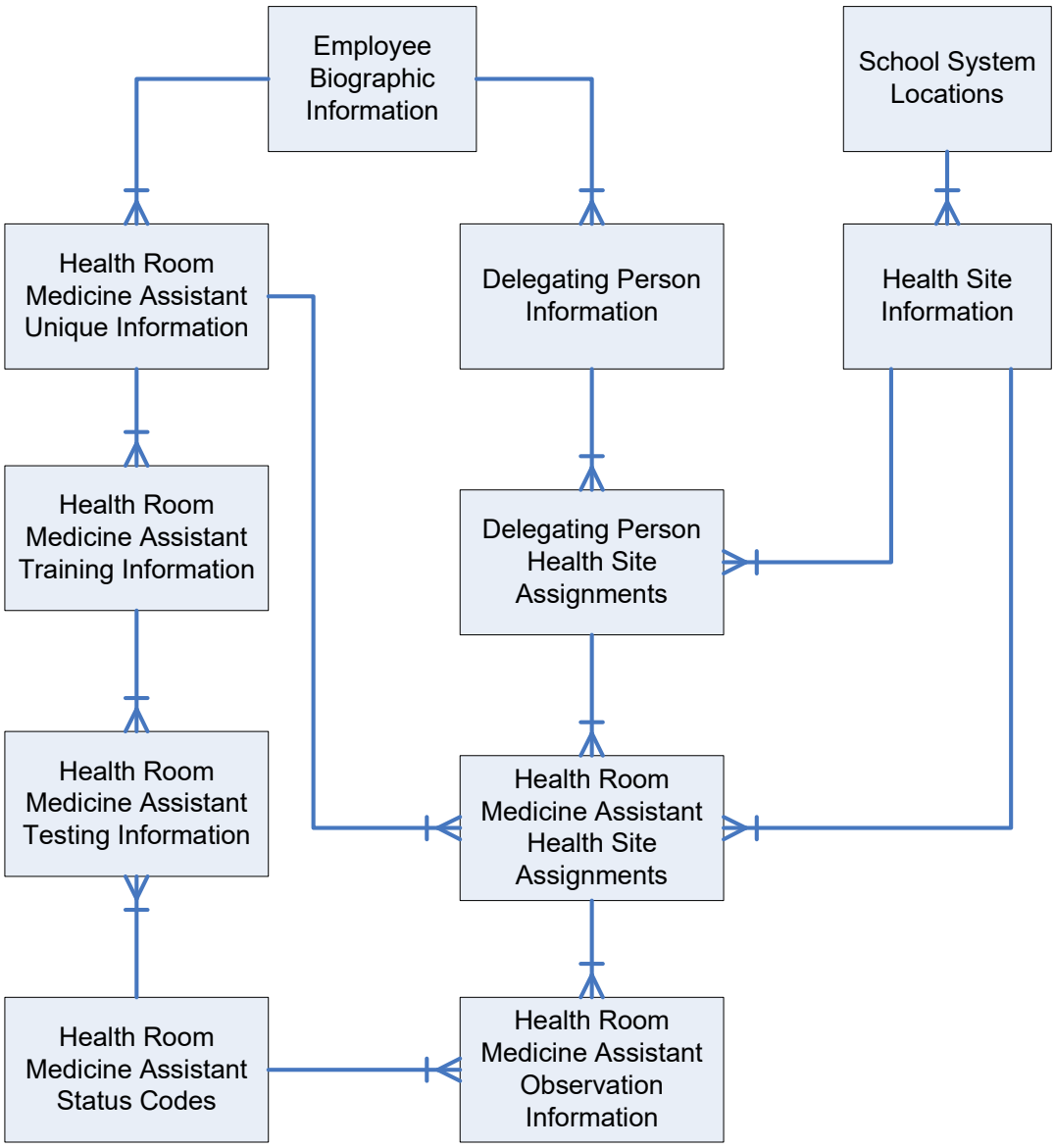

Figure 3. Health Room Medical Assistant entity-relationship diagram

5. How to Build a Database Design

The database's design starts after Step 3.10 is complete. Existing at that point, is a functionally acceptable database design that conforms to business policy requirements. It is most likely not complete from an Information Technology point of view, however.

The complete iterations of database design include:

- Ensuring that the database schema meets business policy requirements are satisfied. (Steps 3.1 through 3.10)
- Incorporating database structures to properly reflect historical data.
- Adding database columns to support comprehensive audits.
- Installing security devices to protect against data and process misuse and theft.
- Inclusion of database administrator special columns and processes.
- Creating generalized versus specialized data structures to support multiple use data.

Once a database schema has been created and the database application completed, there maybe other iterations such as:

- Tuning database structures to ensure adequate performance.
- Accommodating different flavors of SQL syntax for different database management systems (DBMS).
- Installing special key structures instead of business-fact-based keys.
- Accommodating the special physical database requirements of different DBMS.
- Modifying database structures to support special or intensive analyses and reporting.

Finally special iterations of a database's design may be needed for:

- Ensuring that the database is "non-stop" and always optimized.

- Accommodating the needs of client/server, the Internet, or Service Oriented Architectures, or combinations of all three.

These iterations are all addressed in a Whitemarsh paper, Iterations of Database Design that is identified in Section 7, References.

In terms of the case study, software systems like Clarion for Windows can interact with MS/Access databases through ODBC and JDBC. So, if the original objective is to have a simple, efficient, and an effective database-centered application to manage the HMA area, this objective is achievable. If the right tool-set for the HMA application is employed, the entire application could be accomplished within two calendar months.

6. Conclusions

The practical application of the points made in this paper include:

- A data model is not the same as a database design.
- The activities for modeling data are an integral part of traditional requirement's analysis and design.
- Metadata resulting from the activities in all the steps of Section 3 should be stored in a metadata repository.
- Database design begins in earnest after the database has been functionally validated through prototyping.
- From an architecture and engineering point of view it is both invalid and counter productive in terms of semantic integration, nonredundancy, and enterprise interoperability to begin real business system design and implementation until after a data model has been throughly validated.

If sophisticated processes, methodologies, and tool sets s are used on projects such as this, then there will be greater data integration on this project and on subsequent projects. The overall results will be of higher quality. There will be fewer under specified processes, bad or conflicting edits, schools without certified HMAs, and the like. Finally, because the developing staff is all professionals, many with advanced degrees, these sophisticated processes, methodologies and tool sets are well within their capabilities. There is no downside to increasing the sophistication of the workers and their work tools.

7. References

The following references to Whitemarsh materials provide a more detailed exposition practical application of the significant content of this paper.

The following documents are available free from the Whitemarsh website:

Paper	URL
Iterations of Database Design	http://www.wiscorp.com/iterations_of_database_design.pdf
Data Management Conferences	http://www.wiscorp.com/dama2002.zip http://www.wiscorp.com/dama2003.zip http://www.wiscorp.com/wrad2000.zip
Comprehensive Metadata Management	http://www.wiscorp.com/ComprehensiveMetadataManagement.pdf
Metabase Overview	http://www.wiscorp.com/metabase.zip
Metabase System	http://www.wiscorp.com/metabasesystem.html
Metabase System Request	http://www.wiscorp.com/freembrqst.html
Data Modeler Architecture & Concept of Operations	http://www.wiscorp.com/metabase/MetabaseDataModelerArchitectureandConceptofOperations.zip
Reverse and Forward Engineering Users Guide	http://www.wiscorp.com/metabase/MetabaseReverseAndForwardEngineeringUsersGuide.zip