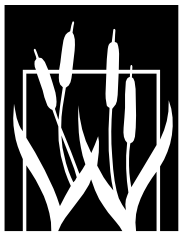

Whitemarsh
Information Systems Corporation

Managing Value Domains

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 paper is to present the Whitemarsh approach to value domain management. Included in this overall objective is an enterprise-wide definition and management of reference data including value management and mapping. Implementation of value domains in a database environment can also vary in two major ways. That is, a value domain class, e.g., Region with values such as New England can be through various means. First as a column, Region, with a DBMS enforced set of explicit values such that the use of any other value causes an error. The second major way is the creation of a table such as Regions that would have a row of data for the New England region, and the use

2. Topics Covered

The topics in this paper include:

- What is a Value Domain?
- Discovering Value Domains
- Engineering Value Domains
- Value Domain Value Management
- Mapping between value domain sets
- Assigning Value Domains in Metabase Data Models
- Deploying Value Domains in Databases

3. What is a Value Domain?

A value domain is a set of values within a specific domain that is or is expected to be restricted. Examples are:

- The set of middle initials, that is, the 26 letters in the alphabet.
- The amount of interest that can be charged on a loan, that is, from 0.0 to 7.99
- The enumeration of U.S. State names, that is, Alabama.
- The enumeration of all valid Social Security Numbers for a set of employees.

In the first example, the letters are “naturally” restricted by making the data type, Character and the length of one. In this case, no special rules would have to be created. In the second case, the value domain is the set of all two-decimal-place numbers between a low value and a high number. In the third case, there would be a very specific list of values that either matched correctly or would be deemed as wrong. The fourth case would be the set of all Social Security Numbers for an existing set of employees. Not only must the Social Security Number be

"correct," that is, not contain letters, and all three sets of number codes within the Social Security Number correctly constructed, but also that the set of Social Security Numbers must be drawn from the existing set of a company's employees.

There are other special cases of value domains such as Boolean where the values are True or False. There are also exclusionary value domains such as "all single characters except A, E, I, O, or U." There could also be exclusionary ranges such as values from 0.0 to 3.0 and from 4.0. This would make the values from 3.01 to 3.99 illegal.

Value domains should be squarely based on policy. The values are then value-based representations of allowed policy. Rejected values are representations of disallowed policy.

4. Discovering Value Domains

Discovering value domains can be an adventure of limitless dimensions. This "adventure" normally happens when value domains and value domain management are not established from the very start that would prevent the storage of illegal values in files and databases. In one extreme example there were four whole pages of listings of discovered "yes" and "no" value variations from across a large collection of files all supposedly from the same application class. A number of the unique instances included values for "null," unknown," "don't know," "please ask, " and even "uncertain." So much for a crisp, clean value domain policy. In another example, a single database had nine discrete values for Gender. Why nine? Where did the odd gender come from? In an even more interesting example, a column that had a date value was allowed to have values up to 40, as in February 40th.

The first step in the value discovery adventure is a thorough policy analysis. Every column that is to have a restricted value domain needs to be examined to determine the values. Each value should be grounded in some policy. Once the sets of allowed values are determined, reviewed and approved by policy makers, a determination must be made of the existing set of valid values across all databases and files. The examination of the values, record by record can be accomplished by commercially available software packages designed for just this purpose. Once all the values are determined, reactions to the illegal values need to be determined.

In general, every database table column ultimately has a restricted value domain, except, for example a table's primary key. That doesn't mean however that the domain of values for all columns should be examined to determine the value domain, nor that its value domain should be managed. The only columns that should be examined are those whose values are employed for sorting, statistics, and/or control breaks. It's unlikely that an enterprise would make decisions based on an employee's name. Not so for the employee's gender, ethnic status, or age. Other columns that quickly come to mind are those related to states, statuses, determinations, and the like. If decisions are made on the basis of a column's value then its value domain is most likely restricted and should be managed.

Once the set of columns for value domain management has been determined, their source data elements need to be discovered because columns derived from the same data element should

almost always share the same value domain, and the value domains should be managed at the data element level. A good example is completion codes. Two examples are: Employee Application Completion Code, or an Employee Advanced Training Completion Code. If the value domains for the two columns are the same, both may be derived from the same data element. But if the value domains are significantly different, there would likely be different quantities of valid value, and the meanings for the values would certainly be different.

In the case of Employee Application Completion Code, the valid values may merely be Null, Yes, or No. Null means "not known." The meanings for "Yes" and "No" are obvious. In contrast, the valid values for Employee Advance Training Completion Code might be Null, Pass, and Fail. Pass might also be further refined into "Excellent," "Good," and "Adequate."

In the second case, because the quantities of valid values are different and the meanings are different, the two columns would have been derived from different data elements. In this second case, the value domains would be named differently and managed independently.

In the first case where the quantity of valid values is the same, the meanings are the same, but the actual values are different, the source data element would likely be the same. A subsequent task might be to either map the value domains, one to the other, or to actually change the values in one of the sets of data records to the values of the other columns.

Regardless of whether there is a common data element or not, the value domain must:

- Be named,
- Be set within the context of the ISO Standard 11179 for Data Element metadata,
- Have its values and value meanings enumerated, and,
- Be assigned to a data element if the two columns are semantic derivatives of the same data element, or assigned directly to the two different columns.

If the value domain is assigned to the common data element, the value domain would be automatically inherited by the two derive columns. In a sophisticated treatment of value domains, a general set of values could be assigned to a data element, and a more restricted or specialized set of values could be assigned to a particular column.

5. Engineering Value Domains

From above, value domains can be assigned to data elements and then inherited by columns, or they can be assigned to columns directly. Additionally, in sophisticated environments a generalized value domain can be assigned to a data element while a restricted or specialized value domain would be assigned to a column. In the Whitemarsh metabase value domains or restricted value domains can also be assigned to attributes and DBMS columns. In the case of the

metabase, the most generalized value domain would be assigned to the data element and the most restricted one would be assigned to the DBMS column. In between would be the value domain assignment to attribute and to columns. This would form an assignment hierarchy starting with data elements from the ISO 11179 data element metadata model, then attributes from conceptual data models (in Whitemarsh these are called Specified Data Models), then columns from logical data models (in Whitemarsh these are called Implemented Data Models), and finally, DBMS columns from physical data models (in Whitemarsh these are called Operational Data Models).

6. Value Domain Value Management

Value domains can consist of discrete or continuous values. In the first case, the list of all United States state names is an example of discrete values. In the second case, continuous values, the set of all Fahrenheit temperatures, say from minus 50 through to a positive 130 might be the allowed set of temperatures for the continental United States. This is sometimes called a range. Ranges must also be inclusive and/or exclusive of its end values. Even if no decimal places are allowed for the temperatures they would still be considered as continuous.

Another way of engineering is the use of NOT, as in "all alphabetic characters but NOT A, E, I, O, and U. Combinations of value domain expressions could be allowed as well such as A through J, but NOT D and E, and all letters from M through Z. The ultimate objective is to employ a definition language that is expressively complete and able to be processed by a software system that would determine that errors have not occurred. An erroneous statement would be A through Z but NOT G and H. A language that is universally recognized and is complete for value domain engineering is ISO/ANSI standard SQL.

7. Mapping between Value Domain Sets

Once the sets of all values are specified within value domains and are assigned to data elements through DBMS columns, there has to a strategy for mapping among specific values. This requirement exists because not all files and databases will contain the same value sets even if the value sets have the same meaning.

For example, if there was one set of values for temperatures in the United States in Fahrenheit and another in Europe for Centigrade there would have to be some value domain value translation process.

Another class of requirement would be for a database that had gender values of 0 and 1 for male and female but another database that had a set of gender values 1 and 2 also for male and female.

A more complicated mapping would be the mapping of a few values to many values. For example, course achievement grades of A to values 90 through 100, or B from 80 to 89, and so

on. If these mappings are accomplished in application programs, these would have to be specified and implemented in exactly the same way in every programming language and application. Clearly, that is both very error prone and not very efficient. It is far more effective to have these values defined with a DBMS and enforced automatically and uniformly.

8. Assigning Value Domains in Metabase Data Models

The metabase has the following facilities for managing value domains:

- Conceptual Value Domain
- Value Domains
- Value Domain Values
- Value Domain Value Mapping

8.1 Conceptual Value Domains

Conceptual Value Domains are the conceptual basis for value domains. Figure 1 illustrates the metadata model for Conceptual Value Domains. Examples include numbers, dates, descriptions, names, and the like. Some of the concepts could be complex such as numbers containing integers, decimals, or complex numbers. Other concepts such as dates could consist of century, year, month, and day of month. These conceptual value domains are not value domains because they are not related to any specific policy-based item about which there would be specific policy-based representative values.

Other concepts might address measurements. Measurements of "what" is the value domain. Measurements are just broad concepts. A measurement might be represented as integers or decimals. In this case, the conceptual value domain would be a complex combination of integers or decimals and measurements.

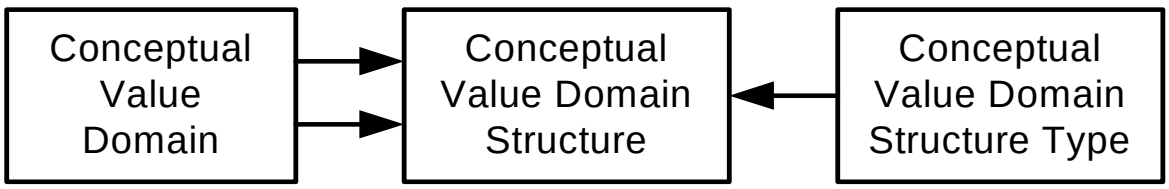

Figure 1. Conceptual Value Domain.

A conceptual value domain is recorded in the table, Conceptual Value Domain. If it is related to another conceptual value domain in either a parent-child or peer relationship, that relationship is recorded in Conceptual Value Domain Structure. The type of relationships is recorded in Conceptual Value Domain Structure Type. A type of relationship might be "contains." Each "arrow" is a one-to-many relationship. The two relationships between Conceptual Value Domain and Conceptual Value Domain Structure represent "contains" and "is contained in." Because of these two relationships, networks are able to be represented.

8.2 Value Domains

Figure 2 provides the metadata model for defining value domains. A value domain is targeted to a specific type of policy that is to be represented through values. Examples are Status, Gender, Department Code, and the like. Value domains are an employment of a conceptual value domain within either a simple or complex context.

Figure 2 shows that value domains can be configured in complex networks. This enables complex realities to be represented. In the metabase, a simple concept might be length. A complex measure might be measure, where in there are physical measures. Physical measures can be dimensions such as lengths, widths, and heights. Other physical measures could be color, density, and the like. All of these are value domains that are all related to a conceptual value domain.

Special to value domains is the existence of a data type. The data type provides for the boundaries of the types of value domain values that can be represented. The value domain types are best drawn from the ISO Standard ISO Standard 11404 on Language Independent Data

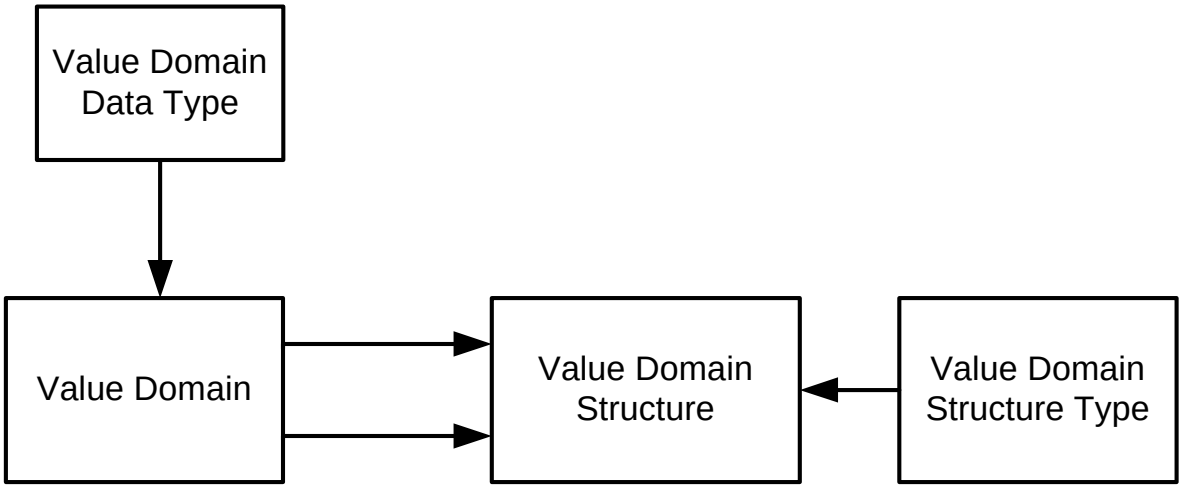

Figure 2. Value Domains

Types. There are two other data types in the metabase. The first is the SQL data types, and the second is the DBMS data types. SQL data types are those allowed by the ISO/ANSI standard SQL. Each of these data types is mapped to an ISO 11404 data type. Similarly, DBMSs like Oracle or DB2 have data types that govern the values for DBMS columns. These data types are mapped to the SQL data types. These three data types, value domain, column, and DBMS column enable complete management.

The relationships able to be recorded in value domains are the same as those for Conceptual Value Domains with the addition of the relationship between Value Domain Data Type to Value domain.

8.3 Value Domain Values

Figure 3 provides the metadata model for value domain values. This is just a simple listing of the values below each value domain. The value domain value's screen has to be sophisticated to record all the cases that are listed in Section 5, Engineering Value Domains. Once a complete set

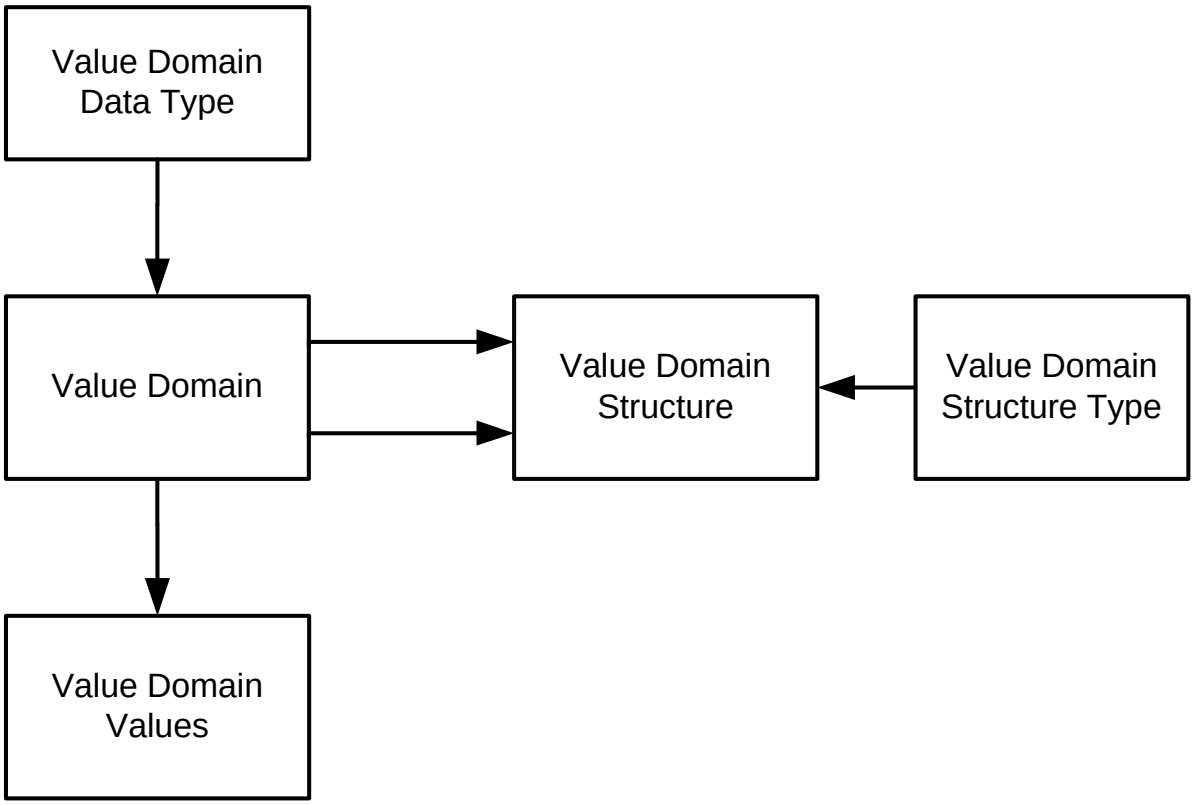

Figure 3. Value Domain Values.

of value domains is entered, a process must be able to be invoked at the Value Domain table level that would examine all the value domains to ensure that no inappropriate values are recorded.

8.4 Value Domain Value Mapping

Figure 4 provides the metadata model for value domain values. Value domain values metadata structure is similar to that of Value Domains and Conceptual Value Domains. A difference is that the Value Domains Structure Type can be hierarchical to support both general and special cases of value domain mapping. An example might be "Is equivalent to." This would then enable different values to be related to each other as in the case of Gender for Male being a 0 or an M and the values for Female being 1 and F. In this specific example there would be two different value domains: Gender Numeric Code and Gender Alphabetic Code. These two would be related to each other through the common "parent" Value Domain, Gender. The metabase software knows that the two genders, Gender Numeric Code and Gender Alphabetic Code are related to each other, thus their associated value domain values are able to be related to each other.

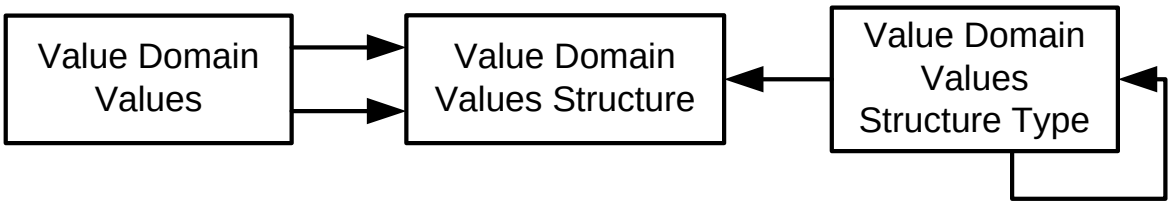

Figure 4. Value Domain Value Mapping.

8.5 Value Domain Metabase Summary Value Mapping

Figure 5 presents the overall value domain metabase implementation for value domains. The following should be noted:

- Value domains are NOT mapped directly to data elements, attributes, columns, or DBMS Columns. Rather, they are mapped through an instance of Value Domain Structure. That is preserves the context within which the value domain exists.

- Similarly, Conceptual Value Domains are mapped to Value Domains through Conceptual Value Domain Structures for the same reason.

- Value Domain Structures are mapped to Data Elements, Attributes, Columns, and DBMS Columns. This creates the potential for significant ambiguity. Ambiguity is however prevented through sophisticated assignment processes that ensure that only subordinate value domains are assigned as the assignment process proceeds from Data Element down through to DBMS Columns.

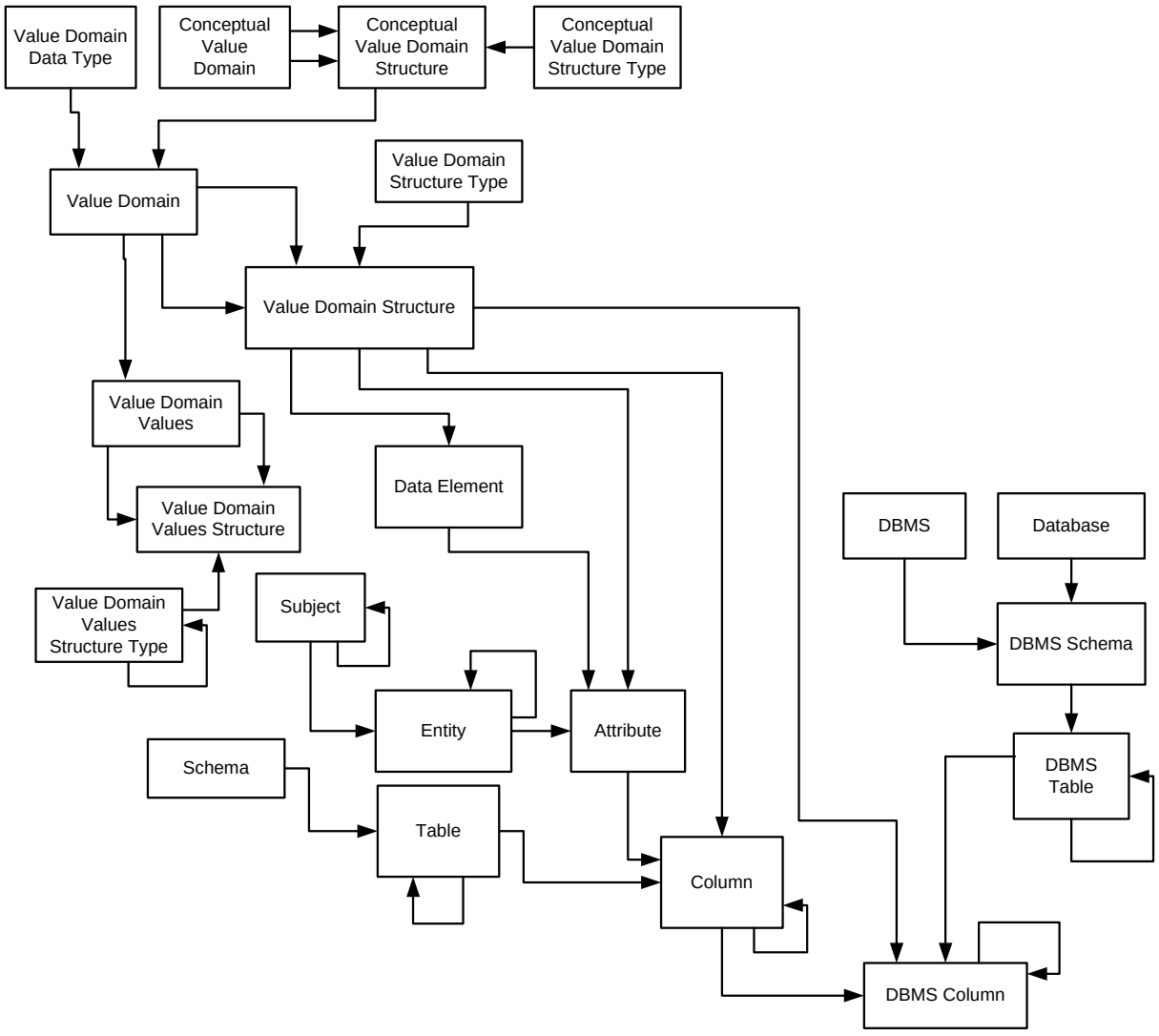

Figure 5. Complete Value Domain Meta Model.

Because of this sophisticated metabase implementation, value domains are able to be defined centrally and then managed across the complete set of enterprise databases. The metabase, as easily seen in Figure 5 can report where value domains are assigned within different schemas, tables, and columns.

9. Deploying Value Domains in Databases

Just because value domains are identified, discovered, specified, and defined into the metabase does not mean that they are, by magic installed in all your databases and applications. That final step is accomplished through two basic alternatives, which are:

- Explicit statements in SQL clauses
- Factored tables

If a value domain is for gender, then domain values should really only be two: Male and Female. Maybe an argument could be made for Unknown as in the case of Saturday Night Live's androgynous Pat. In the case, however, where the values are quite stable and few, the only really appropriate solution is the creation of the appropriate SQL data definition clause that automatically enforces the values.

If there are ranges of values that remain generally stable, then column constraints and even stored procedures can be written from the information contained in the metabase's value domains. Again these are all schema-based objects.

If there are many values of the discrete valid-value kind, like the names of U.S. States, postal codes, Country Codes and the like, the most appropriate solution is to create a factored table that has its name closely match the factored value, such as Country Codes, Postal Codes, and so on. The columns would be an ID for the row's primary key, the code value, e.g., AL for Alabama, a long value column, e.g., Alabama, a column for a description, and then several columns for data like Date of Last Update. For each column that is factored, that is, Country Codes, the column would be replaced with a column like Country Code Id that would act as a foreign key to the factored table, which in this case would be Country Codes.

While not currently available in the Whitemarsh metabase as of early 2007, all these value domain capabilities are quite easy to program and will be provided in a future metabase version. Once programmed, the alternatives would be solution choices on the metabase's SQL Schema's data definition language screen. Once the option was chosen, the metabase SQL DDL generation process would create all the necessary languages to directly implement the value domain enforcement capabilities.

10. Conclusions

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

- Value domains should be squarely based on policy formulation.
- Value domains should be managed from the outset so as to avoid unending discovery "adventures" that are time consuming and costly.
- Value domains need to be centrally defined and managed across all databases and files in a top-down, centrally-planned, but implemented on a project by project basis manner.
- Value domains should be defined and managed if they are the basis for decisions, selections, and control breaks.
- Value domains should be set within an ISO 11179 metadata structure.
- The SQL basis for implementing value domains should be automatically generated from a metabase.

11. 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
Whitemarsh Data Modeler, Architecture and Concept of Operations	http://www.wiscorp.com/MetabaseDataModelerArchitectueandConceptofOperations.zip
Metabase User Guides	http://www.wiscorp.com/MetabaseUserGuides.zip
Comprehensive Metadata Management	http://www.wiscorp.com/ComprehensiveMetadataManagement.pdf

The following documents are available for Whitemarsh Website Members. The URLs that follow provide descriptions of the pages. Members should log in and proceed to the appropriate page, e.g., Enterprise Database, find the book, paper, or course and perform the download.

Paper	URL
Data Management Program - Metadata Architecture For Data Sharing	http://www.wiscorp.com/wwmembr/mbr_products_edb.html
Achieving Data Standardization Achieving Data Standardization Long Talk Achieving Data Standardization Short Talk Achieving Enterprise Wide Data Semantics Standardization Data Standardization Work Plan The Data Standardization Problem	http://www.wiscorp.com/wwmembr/mbr_products_dq.html