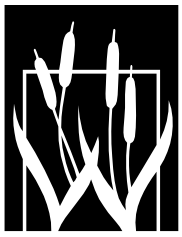

Whitemarsh
Information Systems Corporation

Managing Data Names, Value Domains Abbreviations, and Definitions

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 creating data names through the use of managed collections of semantic hierarchies. ***Whitemarsh Short Papers*** are available from the Whitemarsh website at:

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

Supporting the Whitemarsh approach are facilities from within the ISO standard 11179. The ultimate objective is a comprehensive, non redundant, and integrated strategy for the definition of data that can be employed in structured databases. The Whitemarsh approach also addresses value domain management, and automatic abbreviations and definitions, which are also components of a complete strategy.

2. Topics Covered

The topics is this paper include:

- The data naming problem
- The Core Whitemarsh Architecture
  - Name Part Strategy
  - ISO 11179 Data Element Metadata

3. The Data Naming Problem

What's in a name? Most commonly, the idea is that a name should convey the essence of the object being named. Sounds simple, and to some degree it is. For example, when you make a reservation on *Travelocity* for a trip to a distant city you end up with a "Reservation Number." But is it a number? Almost always, the reservation number either contains some "letters" or consists of only letters. So, then should it be called "Reservation Letters?" If you did then nobody would know what you're talking about. This same situation exists with other common everyday items like Phone Number. An example is 1-Dial-A-Prayer. Where are the numbers?

Over the years strategies have arisen to create names. The most common one was created in the middle 1970s. It has the form,

<Prime Word> <Modifier>s <ClassWord>

This scheme was very popularly used for about 30 years. Under this scheme the prime word was most often used for the entity or table name. There were then one or more modifier words that conveyed an implied restriction on the data being named. Finally there was a class word that related to the form or intended use of the actual data values.

In isolation, the three-part naming strategy had appeal. But like the reservation number or even the telephone number above, it doesn't hold up well under scrutiny. For example:

Social Security Number

First a Social Security Number is not a number. Rather, it is three sets of codes. Second, the entire Social Security system is in financial difficulty (insolvent by 2017?), so it's certainly not secure. Finally, some people say that dealing with this organization is not a very "social experience."

Another example is the Phone Number. Is it really the number of the phone, that is, the phone's serial number? Hardly. Rather, it is a set of at least four codes. Country Code, Area Code, Exchange, and some serialized number. There may also be an "extension" that follows the entire phone number. Is "Phone" the prime word? Since there are no other words, then it would seem so. But to say that a phone number is a number is hardly correct. What is the meaning of an average phone number? Or, the sum of phone numbers? Can you subtract two phone numbers?

While the issue here appears to be simple, it really isn't because data names are not just used within well defined contexts or within contexts that everyone knows well. Data names often do not infer the real value. Nor are data names used to infer just "what" the data is. Rather, data names are often over-loaded with additional words that describe for example:

Over-loaded Data Name Component	Example
Context of the data	Customer, Person, Course, Reservation, or Skill
Implied range of associated values	North East, Minimum, Highest, Estimated, Final, First, or Last
Type of the data	Integer, Text, Date
Inferred use of the data	Count, Amount, Date, Sum, or Average
State of the data	Active, Inactive, in-progress, or terminated

The name should be a shorthand definition for the fact, and the value should be a surrogate representation of the state of that fact. For example, a person's name is not the person, but a surrogate often used to represent the person. A picture serves a similar purpose. Someone's age is just not just a number. Rather it is a years-based chronological representation of the quantity of years since someone was born. It is interesting to note that two persons conceived at the same time can have different ages based on their birth date. So, age is not an absolute number representing the quantity of years from conception but the quantity of years from the specific event, birth. The fact being named has to at least address:

- An overarching concept within which that fact exists,
- A name employed as a surrogate for the fact within the concept,

- The context of the fact,
- Restrictions and/or modifications affecting the value domain representation of the fact,
- The intended use of the concept, and finally,
- The value(s) employed to represent the fact including the value's common meaning.

That is a lot of information to pack into a business fact's name. In actuality, too much.

Is this much ado about nothing? For hundreds or thousands of years the answer was almost certainly yes. But since "globalization," facts travel at the speed of light around the globe, across institutions, and between cultures. Facts therefore almost never remain within their name and meaning generating contexts, nor are facts only shared by a few persons within their "gestational" context.

A simple example serves that point. A global foods' company needed to have regional identifiers for its sales brokers. Each region could only have one broker. A number was needed. Handy, and already known by the food brokers was the food broker's federal tax-identification number. It was national, unique, and every business had one. So, the federal tax-id number was employed as the regional identifier. So long as that broker never moved from one region to the next the number was perfectly reasonable. However, one such broker moved from a very small region to a very large one. Sales for that small region were seen to skyrocket beyond all expectations. Large commissions were paid the small region's manager. And no commissions were paid to the large region's manager. Why did all this misrepresentation happen? Because the sales were reported by broker number, and that very same number was used as the regional identifier. While there were all sorts of data modeling and data naming errors here, none of these surfaced until the broker moved. "Globalization" occurred. The fact's value "moved" out of its originating context. Similar examples abound. For example, the use of Social Security Numbers as account numbers.

Other fact naming and meaning errors are also common. For example, in one reporting system for a government agency, the identification of specific end-of-month reports were bound into the "date" of the end of month report. Thus, some end-of-month dates went up to the "39th" of the month. In another situation, Gender, a one digit code was expanded to then include age-bracket and handicap status. After all, there were 10 single numeric values, so why waste them?

With a backdrop of all these examples, the ultimate objective of this short paper is to present a business fact naming and meaning strategy that is reliable, repeatable, and discriminating across a reasonable set of business fact domains.

4. The Core Whitemarsh Architecture

The core of the Whitemarsh architecture for managing data names, value domains, abbreviations and definitions is founded on two components:

- A revision of the fundamental <Prime Word> <Modifier>s <ClassWord>

- Use of the ISO Standard 11179 for Data Element metadata

4.1 Name Part Strategy

The three-part naming strategy is changed to the following:

<business domain> <semantic modifier>s <common business name> <data use modifier>s

A key difference is that there is no prime word. It is replaced by a more abstract concept called business domain. For example, employee, person, finance, and the like. There is a common business name, and there can be multiple "class words" that is, data use modifiers. In the examples above, the common business name for Social Security Number is just that. So too is the common business name for Reservation Number and Telephone Number.

The reason for not having a prime word is because so much of the tradition and the actual instructions required the prime word to be the entity's or table's name. When this occurred, the named business fact could only be an attribute of "that" entity, or a column of "that" table. Analysis has shown that the semantics of a well-engineered business fact is deployed 30 to 100 or more times across an enterprise in various forms, contexts, or localized names. So, if there are 100 databases, of 200 tables each, and 15 columns for each table, then there are 300,000 columns. If prime word is employed to represent the table, then users are "condemned" to explicitly name and define 300,000 columns. At just one hour each, that's $29 million dollars of effort.

By eliminating the prime word you are freed from that extreme amount of wasteful work. Another key difference is the use of multiple "data use modifiers." Data use modifiers are similar in concept to the traditional class word. However, under the Whitemarsh approach there can be multiple such words. By allowing multiple such words then the name that is actually constructed

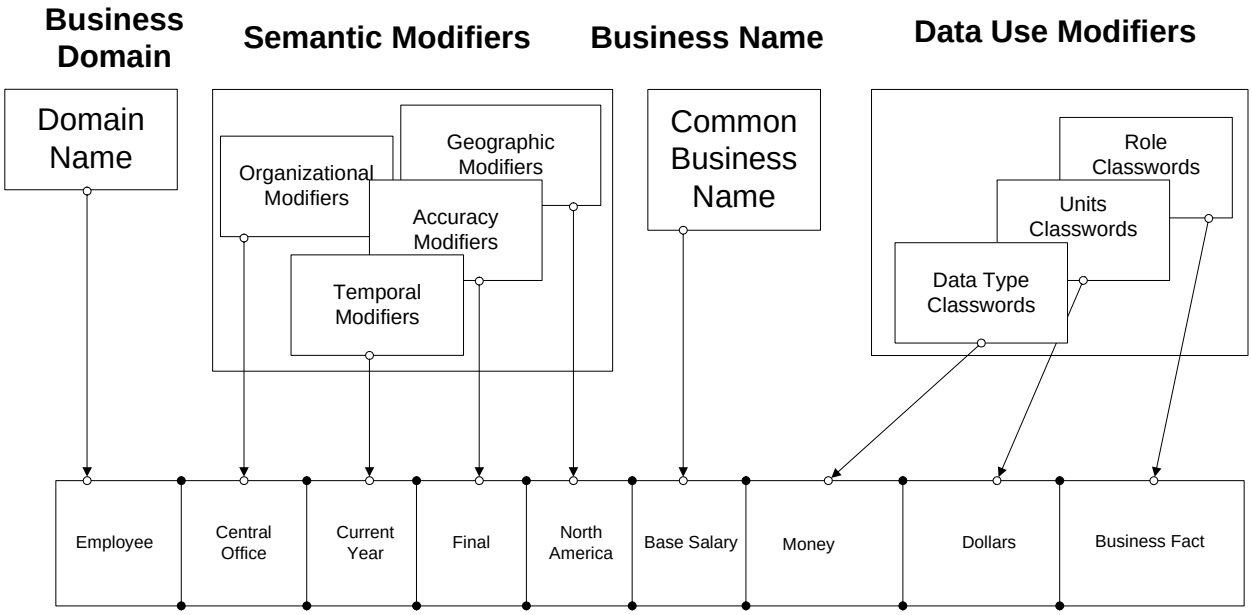

Figure 1. Business fact name components.

can be more expressive of all the different contexts inferred. Figure 1 shows the obvious benefit. A reading of the constructed name provides ample evidence as to the business fact's complete meaning. That is, the business domain within which it resides, the semantic modifiers or conditions related to its value and meaning, and in regards to the value itself, the representation of the value (money), that it is in dollars, and that the role of the constructed name, a business fact.

In actuality, what is actually "stored" are not the literal strings, e.g., Employee, but the foreign key based identifiers to those literal strings. When the software generates the complete name string, the result is simply:

Employee Central Office Current Year Final North American Base Salary Money Dollars Business Fact

Clearly, that's quite long. So you can also allocate to the constructed name a user based name such as "Salary." The key here is not what you call it, i.e., Salary, but what it actually is, that is,

Employee Central Office Current Year Final North American Base Salary Money Dollars Business Fact

And, more importantly, that long string is supplemented by all the various modifier abbreviations and definition fragments. Consequently, with this strategy there is:

- Automatic naming
- Automatic definitions, and
- Automatic abbreviations.

The semantic modifiers and the data use modifiers are just allocated through tag-based pick lists schemes within the software. Clean, simple, effective, and very efficient.

A key side benefit of this strategy is that users can now search and find collections of business facts based on any semantic part, that is, based on business domains, common business names, any value within the use context modifiers and data use modifiers.

4.2 ISO 11179 Data Element Metadata

The ISO Standard, 11179, Data Element Metadata, was constructed during the 1990s and early 2000s. It had two versions. The second and current version was completed in 2002. The standard's Part 3, the metadata model, is excellent. The ISO 11179 standard actually portrays a conceptual data model which then has to be brought into third normal form so that it can be implemented in a DBMS.

The key value from the ISO 11179 standard is that it sets "data element" into its right context. A data element is a context independent business fact semantic template. Simply put that means that the entirety of the semantics of a data element are then able to be deployed in as many different context dependent "containers" as necessary. Figure 2 illustrates this point.

This figure shows a number of different "containers." For example, process has one or more variables, report has one or more report columns, file has one or more fields, table has one

or more columns, entity has one or more attributes, and XML schema has one or more XML Tag Names. The semantics of a data element can be mapped to any, and in some instances, all of these different "semantically contained" representations of the data element. This is precisely why a well-engineered data element has its semantics used over and over, a 100 times or more.

Practically, that means that a properly engineered environment can greatly increase the velocity though which business facts are defined, and can greatly increase the level of interoperability across databases because of the data element's semantic reuse. If the U.S. Department of Defense had adopted this engineering strategy, then its 12-year effort to achieve data standardization through the "8320.1" program would have been totally successful rather than judged a total failure. Hundreds of millions of dollars were wasted because there was no understanding and infrastructure within the DoD to support Figure 2. In the example cited above there would only be 3,000 data elements. That's just 1.5 staff years or about $300,000 to define them all. Once the data elements were defined, the entire set of 300,000 columns could have been automatically created, named, abbreviated, and defined.

ISO 11179 is just not about data elements. It is also about higher levels of generalizations that then enable data elements to also be set into contexts. Figure 3 presents a set of database tables for implementing the key semantic components of the Part 3. In this figure there are

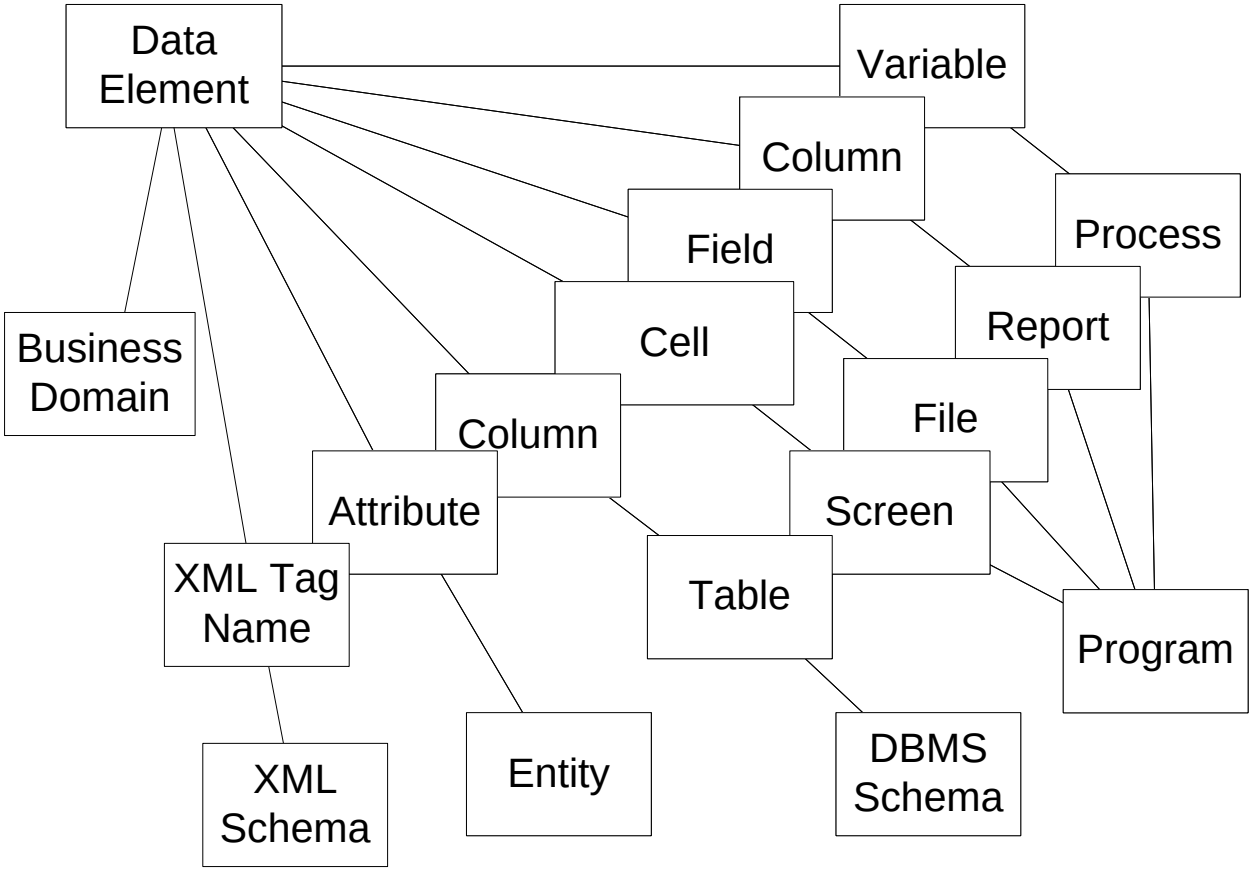

Figure 2. Data element semantics deployments.

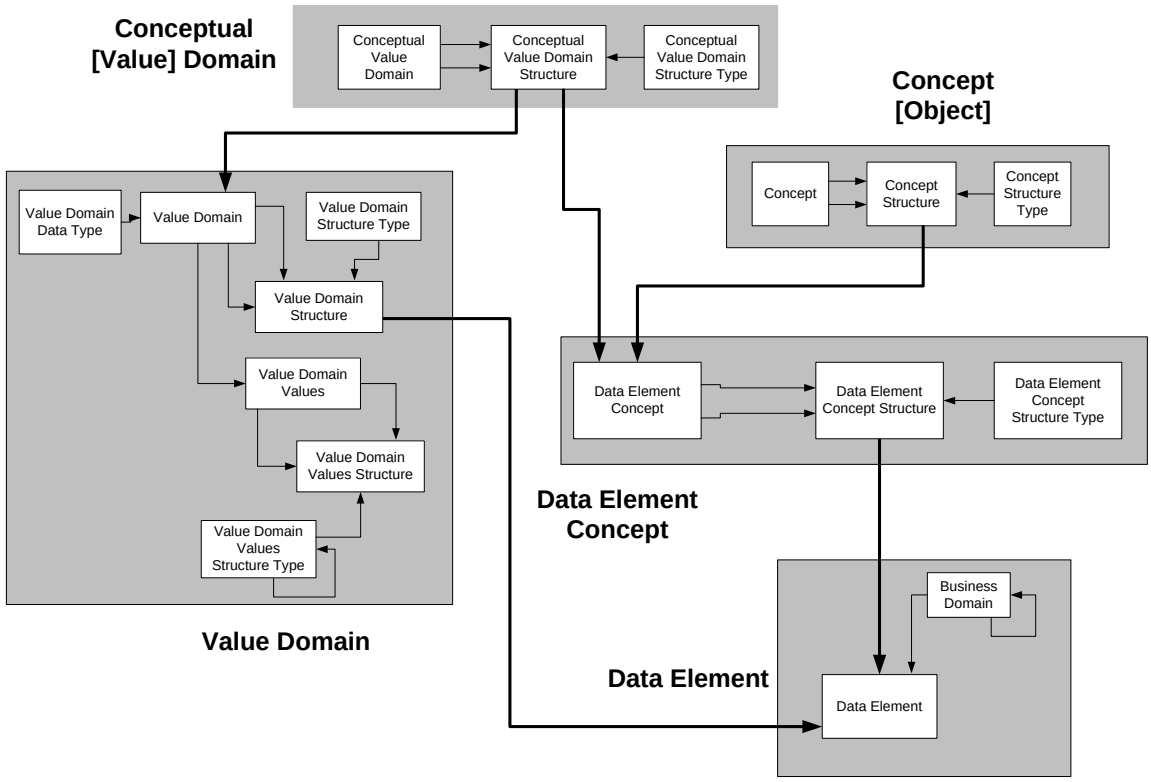

Figure 3. Key semantic components of ISO 11179 Part 3.

collections of tables for Concept, Conceptual Value Domain, Data Element Concept, Value Domain, and Data Element. Concepts and conceptual value domains are the highest level of generalization. Within each there can be any quantity of contained generalization levels. A "marriage" of a concept with a conceptual value domain produces a data element concept.

Concepts can also be of arbitrary complexity. A data element concept, when "married" with a specific value domain, then causes the creation of a data element. Data element is thus not the "top." Rather, it is the lowest level of generalization within ISO 11179.

Figure 4 shows a simple use of 11179 for the creation of the data element, Annual Salary. First there is the concept, Individual Award. Then there is the conceptual value domain, Monies. The "marriage" of the two produces the data element concept, Personal Compensation. There can be many other data element concepts within this specialization. When the data element concept, Personal Compensation, is "married" to the value domain, say Dollars, then the result would be the data element, Annual Salary. The data element, Annual Salary, is then an available set of business fact semantics can be deployed within different contexts, and even may have been subsequently specialized (within the context of an attribute or column) through different use context semantics. For example, there might be a column in a table called Average Annual Salary. That would be a semantic modifier on the fundamental data element, Annual Salary. Or there might be another column called Previous Year Annual Salary. This too would be fundamentally based on Annual Salary. It's just a different kind of specialization.

When the metadata exists within the meta entities associated with ISO 11179 are combined with the metadata associated with the creation of business fact names, an entirely new

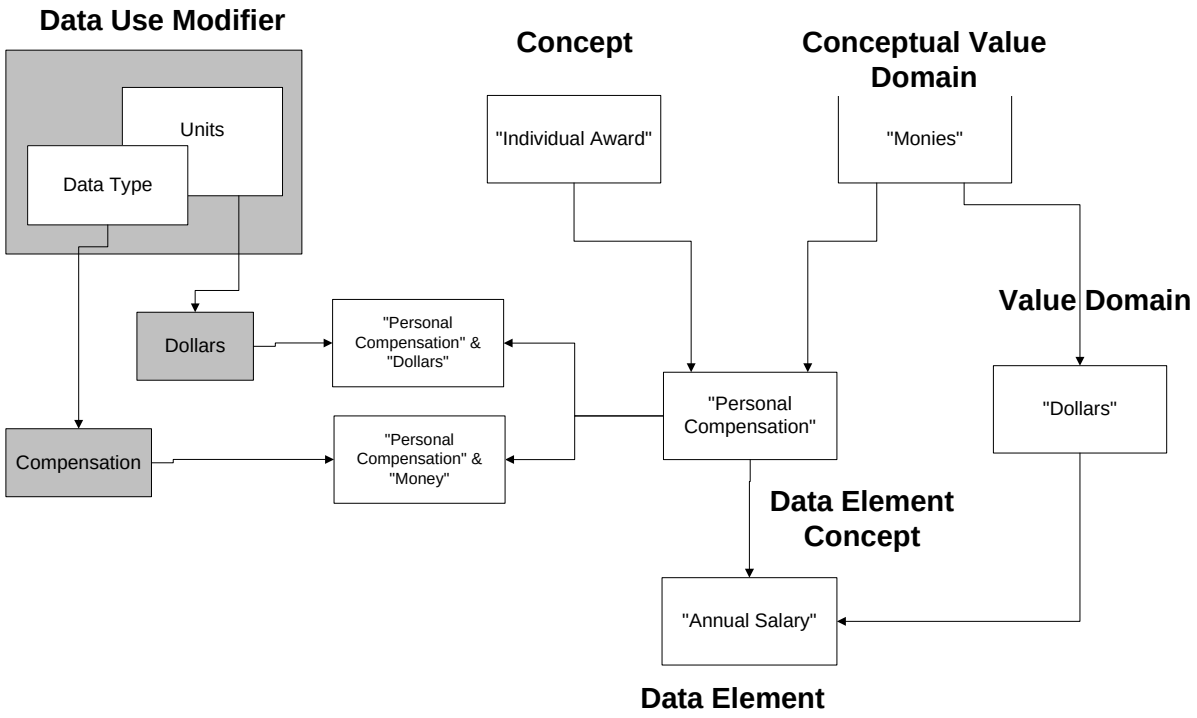

Figure 4. Example of the development of the data element, Annual Salary.

level of semantic richness and flexibility emerges. Figure 5, illustrates the combination. In this example, there is the data element concept, Compensation, which is associated with the data use modifiers of dollars and compensation. That means that every associated data element specialization is also a form of compensation in the units, dollars. A specialization of compensation is Salary. It has the data use modifier, business fact. Now, the data element, Salary, has its semantics set within the database table, Employee, to then form the column, Hourly Wage. Note that the column additionally has the semantic modifiers of United States, Current Year, Final, and Accounting. Taken together, the column, Hourly Wage has the following semantics associated with it:

- Dollars
- Money
- Business Fact
- Salary
- United States
- Current Year
- Final
- Accounting
- Employee

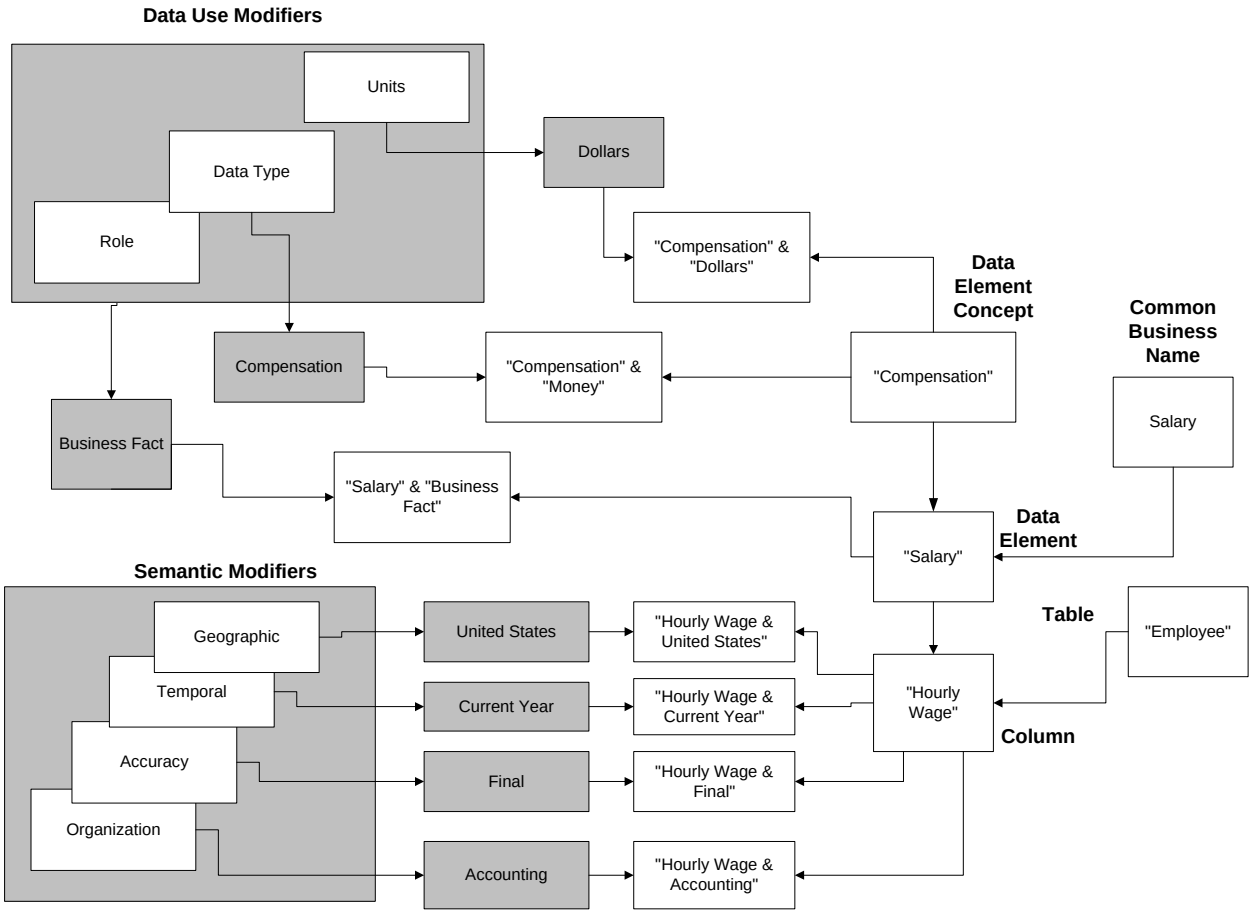

Figure 5. Database table column, Hourly Wage.

Associated with each of these are a definition and a set of abbreviations. The software is able to automatically compose the complete name, that is,

United States Current Year Final Accounting Hourly Wage Compensation Dollars Business Fact

which is simply known to the user as Hourly Wage. But when the user would want to know what Hourly Wage really means, the repository responds with all the associated semantics. Again, clean, simple, effective, and very efficient.

As stated above, a data element exists as the “marriage” of a data element concept and a value domain. Not shown in Figure 4 are the set of tables that then enable users to control the exact values, or ranges of values that are allowed/disallowed to be represented by any use of a data element within an attribute, column, or DBMS column. In a similar way, value domains can be directly associated with attributes, columns, or DBMS columns. Such allocations are allowed to occur if and only if the value domains are more restrictive than that allocated to the data element. Additionally supported is the ability to map one value domain to another so that value

domain evolutions can be tracked across time. These allocated value domains can then be used by application systems and/or database management systems (DBMS) to control data record inserts and updates.

5. Conclusions

Data element identification, allocation, standardization, and definition are probably considered the worst, most tedious activity within data management. While the approach presented in this paper is not complete nirvana, it greatly simplifies the data standardization process and makes it much more efficient.

The objective of this short paper was to set fourth a comprehensive, non redundant, and integrated semantic strategy for the definition of data that can be employed in structured databases. That has been done with a scheme that is reliable, repeatable, and discriminating across a reasonable set of business fact domains. Further, the solution is set within an automated setting that can be scaled and distributed across an enterprise.

This overall strategy is carried though the entire data modeler component of the Whitemarsh metabase. Semantic parts are allocated where they occur first. They are carried down through the levels of specialization through inheritance unless “replaced” by a more specialized variation. This greatly enhances the velocity and efficiency of data element creation and data element semantics deployment within any one or more of variables within a process, report columns within a report, fields of a file, columns from a table, attributes of an entity, or XML Tag Names of a XML Schema. Additionally, the strategy includes a one-to-many relationship between Data Elements to Attributes to Columns to DBMS Columns. These additionally greatly enhance the ability to have a define-once use many time culture.

6. 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
Comprehensive Metadata Management	http://www.wiscorp.com/ComprehensiveMetadataManagement.pdf
An Old Saw That Just Won’t Cut	http://www.wiscorp.com/old_saw.pdf
A Column by Any Other Name	http://www.wiscorp.com/a_column_by_any_other_name.pdf
Metabase Overview	http://www.wiscorp.com/Metabase.zip

DAMA 2001 Data Standardization Talk	http://www.wiscorp.com/dama2001datastandardizationtalk.zip
DAMA 2002 Data Standardization, The Problem	http://www.wiscorp.com/dama2003.zip
DAMA 2003 Data Standardization, The Solution	http://www.wiscorp.com/dama2003.zip
Whitemarsh Data Modeler, Architecture and Concept of Operations	http://www.wiscorp.com/MetabaseDataModelerArchitectueandConceptofOperations.zip
Metabase User Guides	http://www.wiscorp.com/MetabaseUserGuides.zip

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
Work Breakdown Structures	http://www.wiscorp.com/wwmembr/mbr_products_dp.html
A Column By Any Other Name is Not a Data Element Paper A Column By Any Other Name is Not a Data Element Presentation A Column By Any Other Name is Not a Data Element Presentation - Short Course Achieving Data Standardization Achieving Data Standardization Long Talk Achieving Data Standardization Short Talk Achieving Enterprise Wide Data Semantics Standardization An Old Saw That Just Wont Cut An Old Saw That Just Wont Cut - Software Implementation Report	http://www.wiscorp.com/wwmembr/mbr_products_dq.html

Paper	URL
Analysis of the DISA 8320 Data Standardization Approach Data Standardization Work Plan The Data Standardization Problem