**The Data Administration Newsletter**
**www.tdan.com**

**Robert S. Seiner**
**Publisher**

**Primary Menu**
**Register On-Line!**
**Whats New?**
**Job Classifieds**
**Links & More**
**Contact**

**Interested in Sponsorship?**
**Hit Here**

**Keep an eye here for Announcements!**

# AN OLD SAW THAT JUST DON'T CUT

**Michael Gorman - Whitemarsh Information Systems, Corp.**

## Introduction

The old saw at issue is the three part paradigm for data element names: prime word, modifier[s], and class word. This old saw just don't cut it. Never did, and never will. So, stop wasting your time and money and choose one that does.

## Old Saw Tooth One: Data elements are synonyms for table columns, screen cells, entity attributes, or report fields.

Whenever you review a relational table, you're likely to say, "The data elements are:..." Ditto when you review a screen, an entity or a report. So, the first tooth on the Old Saw that doesn't cut it is that the data element is a synonym for a table's column, a screen's cell, an entity's attributes or a report's field. Rather, a data element is the meta data representation of the semantics for the column, for the cell, for the attribute or for the field. The data element is thus not their synonym, but their common, abstracted meta data representation.

For example, when you say, "Where are the First Names," you are likely to be answered, in the Person table, or the Biographic screen, the Employee entity, or on the Personnel report. So, if they all contain First Name, how can Person First Name and Biographic First Name, Employee First Name, and Personnel Report First Name be the same? Clearly, they can not. Rather they all possess "containerized" representations of the First Name's semantics. Hence, data elements are not that which is contained, rather, they are abstracted common meta data representations of those contained meta data representations of business fact instances.

When you take this leap of common sense, you are immediately freed from the almost impossible, thankless, and worthless job of trying to identify and define every column, cell, or field that has or will ever exist. Because, if you believe that data element is the synonym for columns, cells, or fields then the effort to define these four is exactly that: impossible, thankless, and worthless. If you believe otherwise, then you have to totally discount the confession of a senior U.S. Defense Department official who admitted to having wasted $55 million on just that effort! In essence this U.S. Defense Department agency built a huge landfill of discordant, unintegrated semantics of columns. The only thing that keeps them from accepting the complete futility of their work is the mistaken belief that they have actually defined about 20,000 "data elements." And, since admitting the utter senselessness would be too painful, they just continue.

Data elements thus become context independent business facts that may be used in many different contexts, e.g., tables, screens, entities, and reports. Given this REAL definition, the effort to define data elements is finite, manageable, predictable, and cost effective. In general, across an enterprise wide sea of columns, cells, attributes, and fields, the ratio is at least 10 to 1. The broader the area, the larger the ratio. With this new definition, the overall definition effort is cut by 90%. Not bad!

Along with the first tooth on the old saw that don't cut is the concept of PRIME WORD. It's appropriate for columns, cells, attributes, and fields but totally inappropriate for data elements. Primary business domain or subject area is appropriate, however.

## Old Saw Tooth Two: Data Elements Don't Have Names

The second tooth is that just don't cut it is that data elements don't have names! If in fact, they do. They each have a common business name. If however you accept the "traditional" construct for a data element, i.e., Prime word + Modifer[s] + Class word then you MUST believe that data elements don't have names. The reason you MUST believe they don't is because under the traditional approach, a data element's name is said to be a concatenated representation of precisely chosen word strings, each of which is chosen to represent a specific part of the overall set of semantics. Given you believe that to be true, then how do you explain, Employee Social Security Number?. If you examine each word, then Employee must be the prime word, Number must be the class word, and Social and Security must be two distinct and "nested" modifiers, each of which conveys semantic meaning. One thing you know for sure is that those characterizations are untrue. The same, forced, delusional analysis would also apply for example to Telephone Numbers and Part Numbers.

So either you are forced to proudly accept that exhibitions of forced, delusional, analysis is something to brag about, or you must accept that opposite: that data elements do have names that are NOT necessarily characterizations of their semantics. Rather, they are their commonly known names, such as Social Security Number (that is a three part numeric code separated by hyphens) or Part Number (that commonly contains multiple embedded codes, some of which contain letters), or is a Telephone Number (that like Social Security Number is a multi-part code, but has either hyphens, a set of parentheses and hyphens, or under the new ISO scheme is a four part numeric sequence separated by periods).

**Old Saw Tooth, Three: That Prime Word, Modifer[s] and Class word are Part of the Data Element's Name**

Given that we have shown above that a data element has a common business name and that it is NOT prime word + modifer[s] + class word, then what happens to Prime Word, Modifer[s], and Class word? Well, they are useful and they do describe some important semantics of the context dependent business facts they represent.

There is an additional problem that arises when the prime word, modifier[s] and class word become melded into the name string: they lose their individual semantic identity and distinguishability. That's especially so when abbreviations are required, or when wholly different words must be used to accommodate different languages and cultures. Simply put, each is a different meta attribute within the complete set of meta attributes that fully define either the data element or the context dependent business fact.

First, however, there is a distinction between that which contains and that which is contained. Data elements are the former, and columns, cells, attributes and field are the latter. All of which, are meta entities within an overall meta model for information technology. It's just that data elements largely stand alone while columns, cells, attributes and field are properly contained in tables, screens, entities, and reports, respectively. Collectively, these later four are context dependent meta data about business fact value instances, while data elements are context independent meta data about context dependent business facts. Because of this difference, each has a slightly different set of meta-attributes. Table 1 presents these differences.

| Meta Attributes for Data Elements and Context Dependent Meta Entities | | |
|---|---|---|
| **Meta Attribute or Semantic Part** | **Context Independent Data Element** | **Context Dependent: Table Column, Screen Cell, Entity Attribute, Report Field, etc** |
| Business Domain | Yes | No |
| Common Business Name | Yes | Yes |
| Prime word | No | Yes |
| Modifier subclass[es] | No | Yes |
| Class word subclass[es] | Yes | Yes |
| Generated Policy Basis Description | Yes | Yes |
| Computer Data Type | No | Yes |
| Data Structure | Yes | Yes |
| Required Uniqueness | No | Yes |
| Relationship Function | No | Yes |

**Table 1:** Meta Attributes for Data Element and Context Dependent Business Facts

As can be seen, Prime word is a proper meta attribute of a context dependent component but not the context independent component, data element. This same difference holds as well for modifier and class word. Other meta attributes from Table 1 above are common. For example, data structure is now really important since SQL/3 is no longer relational. An SQL/3 column within a single row may represent single values, multi-values, groups, repeating groups, nested structures, BLOBS,

CLOBS, REF-types to other-site objects, and abstract data types with both data structure and methods. So much for the relational model...

**Old Saw Tooth Four: Modifiers are from a single homogeneous set**

Modifier has been commonly thought to be derived from a single homogenous pool. While that may have been true for a single database within a single project, it is not true for enterprise-wide suites of databases that cross countries and cultures. Table 2 contains a minimum set of modifier classes and examples of the value sets for each.

| **Modifier Classes and suggested contained values** | | | |
|---|---|---|---|
| **Temporal** | **Accuracy** | **Geographic** | **Organizational** |
| last | estimated | world | world-wide |
| first | projected | hemisphere | business unit |
| latest | revised | north American | region |
| earliest | initial | United States | district |
| current | actual | mid-Atlantic | territory |
| this year | | Maryland | |
| last year | | Bowie | |

**Table 2.** Examples of Modifier Classes

**Old Saw Tooth Five: There is only a choice of one class word**

Class word has always been thought to be drawn from a single choice list. That however has never been true despite our being "forced by tradition" to believe it. For example, suppose there is a price table, where PRICE is the primary key column. Is price the identifier? Yes. Is price an Amount? Yes. How can that be if we force ourselves to make only one class word choice. How can Social Security Number be an Identifier, but not a Code, and certainly not it's already existing class word choice, Number?

The way out of this "absurd to be in" situation is to recognize that there are multiple classes of class word, just like there are multiple classes of modifiers. Table 3 offers suggestions in this area. This categorization and example sets are not complete by any means.

| **Class Word Types and Example Value Sets** | | |
|---|---|---|
| **Data Type** | **Role** | **Unit** |
| Date or date component | Identifier component | Day |
| Code | Factor | Case |
| Text | Flag | Aisle |
| Weight | Indicator | Pallet |
| Dimension | Identifier component | Transaction |
| Money | Rank | Percent |
| Integer | Business fact | Inches |

**Table 3.** Examples of Class Word Classes

**A Saw with Teeth That Cut**

When all these meta attributes are fully valued for either a context independent or a context dependent component, a full set of semantics can be printed AS IF THEY WERE the name of the data element or the context dependent component. Figure 1 illustrates just such a scheme. It is critical however NOT to be misled into believing that these brought together strings ARE the component's name. The name may be contrived wholly differently and by another set of rules

altogether. For example, the column represented in Figure 1 may just be SALARY within the Employee table.

Additionally, all these value selection are really just foreign key value instances within their respective meta attributes of the meta-entity, COLUMN. Their originating meta-entity, e.g., PRIME WORD, or ORGANIZATIONAL MODIFIER, etc, not only have the real values, but also important information about the real value such as who authorized it, when, why, different accepted abbreviations (none, medium, short, ultra-short), full definition, and definition fragment.

Of especial value are the definition fragments. These are the components that can be brought together to then AUTOMATICALLY define all context business facts. That gives a real boost to both the 90% reduction in the amount of work to define context dependent facts and also to the standardized, but automatic definition of both context independent and context dependent business facts.

Now, is that a saw that cuts, or what?

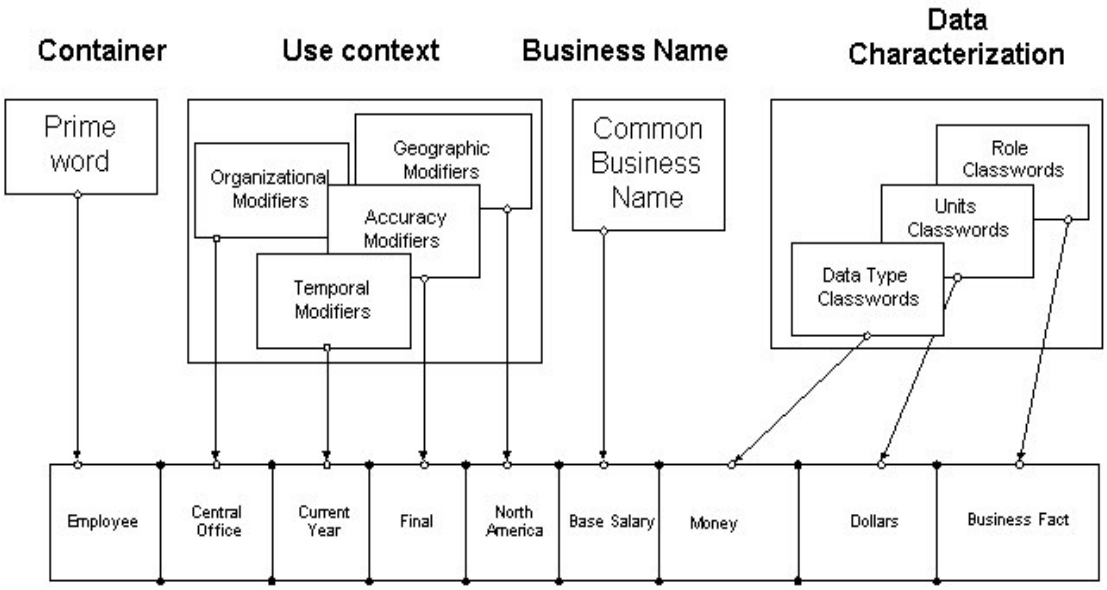


**Figure 1.** Fully Selected Semantics for Employee Base Salary Business Fact

*Michael M. Gorman, President of Whitemarsh Information Systems Corporation, has been involved in database and DBMS for almost 30 years. Mr. Gorman has been the Secretary of the ANSI Database Languages Committee, X3H2 for 19 years. X3H2 standardizes SQL. A full list of Whitemarsh's clients and products can be found on the web site, www.wiscorp.com. The goal of the web site, WisWeb, is to make data management books, courses, methodologies, software, and metrics available to the database community through electronic publishing and downloading. WisWeb memberships are very reasonable and are designed for the individual, the ISD organization, universities/colleges, and professional training organizations.*

*Whitemarsh Information Systems Corporation*
*2008 Althea Lane*
*Bowie, Maryland 20716-1518 USA*
*Phone: +1.301.249.1142*
*FAX: +1.301.249.8955*
*Email: mmgorman@wiscorp.com*
*WWWeb: http://www.wiscorp.com*

**[The Article Archive]**

**The Data Administration Newsletter**
**Robert S. Seiner - Publisher and Editor - rseiner@tdan.com**