Whitemarsh
Information Systems Corporation

Managing the Meaning of Data

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, nonredundant, and integrated semantic definition of data as it is employed in structured databases. ***Whitemarsh Short Papers*** are available from the Whitemarsh website at:

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

Why a formal approach is needed, should not be a subject for debate or even question. A "google-search" on the phrase "data quality" produces 411,000 hits. A quick review shows that almost all the hits refer to documents about data quality problems. Data quality problems are rooted in discordant semantics, which are, the rules for meaning and usage. The following are examples of the most common data quality problems:

> *A VP demands a consolidated customer sales report across product lines across different divisions. He wants it by the end of the week. Energetically you dig into the request just to discover that a sale is defined differently across the different products and across the divisions. Some have it as gross, some net before taxes, some after taxes. Worse yet, there are codes everywhere. Some have the same code name but there are different value sets with different meanings. Worse even still, you find that some record average daily sales, some by the sale, and some others you just cannot figure out. Of course the VP also wants it consolidated by customer. Right, what's a customer?*

Other examples include:

- Having or not having dashes in social security numbers
- 0 and 1 for Gender, and 1 and 2 for Gender. (Value domain mismatch)
- Mike Gorman vs. Michael M. Gorman (Same person, different names errors)
- March East Region Sales = Sales for March of NE Division + Sales of March of SE Division. But, the March Sales of NE Division is Net After Expenses, while the March Sales of SE Division is Total Monthly Sales.

The challenge is not whether there are data quality issues, nor how to fix them, but how to design these data quality issues out of the IT process from the very beginning. Not only does make fixing errors faster, it also makes IT system development faster and cheaper to evolve and maintain. In a 1995 United States Air Force study it was determined that the Air Force was spending about $167 million every year creating and maintaining extracts-transforms-and-loads (ETL) systems that transfer data from one system to the next. While some such ETL systems cannot be avoided, every dollar spent has essentially zero value.

To address the complete meaning of data, this paper briefly presents:

- Whitemarsh metabase models
- Whitemarsh metabase data models
- Data element models
- Specified data models
- Implemented data models
- Operational data models
- View data models

2. Whitemarsh Metabase Models

The Whitemarsh Metabase is a CASE/Repository system that runs under the control of SQL engines. The metadata databases can reside on Unix or Windows operating system platforms. The Metabase system is thus both inherently multi-user and able to be distributed. Further, reporting from the Metabase can be through Crystal Reports which makes data access and reporting able to be ubiquitous. The client side of the Whitemarsh metabase operates only under Windows operating systems.

The metabase contains the following models, which collectively tell a story:

Whitemarsh Metabase Models	Description
Missions, Organizations, Functions, Positions	The essential missions that define the very existence of the enterprise, and that are the ultimate goals and objectives that measure enterprise accomplishment from within different business functions and organizations. The organizations that are accomplishing specific aspects of missions with specific databases, information systems and through specific functions. The functional procedures performed by groups in their achievement the various missions of the enterprise from within different enterprise organizations. The positions staffed by persons who perform the functions within the context of organizations to achieve enterprise missions.
Information Needs Analysis	The information (a.k.a. query results or reports) needed by various organizations in their functional accomplishment of missions and what databases and information systems provide this information.

Whitemarsh Metabase Models	Description
Resource Life Cycle Analysis	The key resources (facilities, materiel, staff, etc.) of the enterprise including how are they sequenced, interrelated, and how are they supported through databases and information systems.
Database Object Classes	The large data-based concepts, associated processes and states that are needed to squarely represent the needs and transforms of business activity. These are then transformed into the database tables, information systems, and embedded processes that comprise the implemented IT environment.
Business Information Systems	The business information systems including where are they, how they are related to mission, organization, function, and databases, and finally the ability to identify and show the impact on these business information systems when policy (a.k.a., data) is required or changed.
Data Elements	The critical business fact templates that are employed over and over again in various forms, documents, data structures, and database tables. These are bound by semantic meanings and restricted value domains to effectively define the semantics for all data captured, stored and manipulated by the enterprise.
Specified Data Model	The standardized data structure templates for events, persons, financial transaction types, addresses, etc. that are employed in different contexts within databases of all different classes that then capture the execution of enterprise policy through well established procedures.
Implemented Data Models	Specific designs for databases representing activities and functions critical to the persistent memory of the enterprise as it carries out its critical activities in support of resources such as staff, finance, facilities, purchasing, inventory, transportation, marketing, and the like.
Operational Data Model	Specific DBMS created designs of databases that either represent different physical designs of databases required by computing environments, processing demands, and the like.
View Data Model	The specifications of the interfaces between databases and the application information systems that capture, manipulate, and report data.

This short paper focuses on the Whitemarsh metabase's five "data" data models, that is, Data Elements, Specified, Implemented, Operational and View data models. These are collectively shown in Figure 1. Focus is on these five models because they define the meaning and allowed uses of data within databases and across the enterprise.

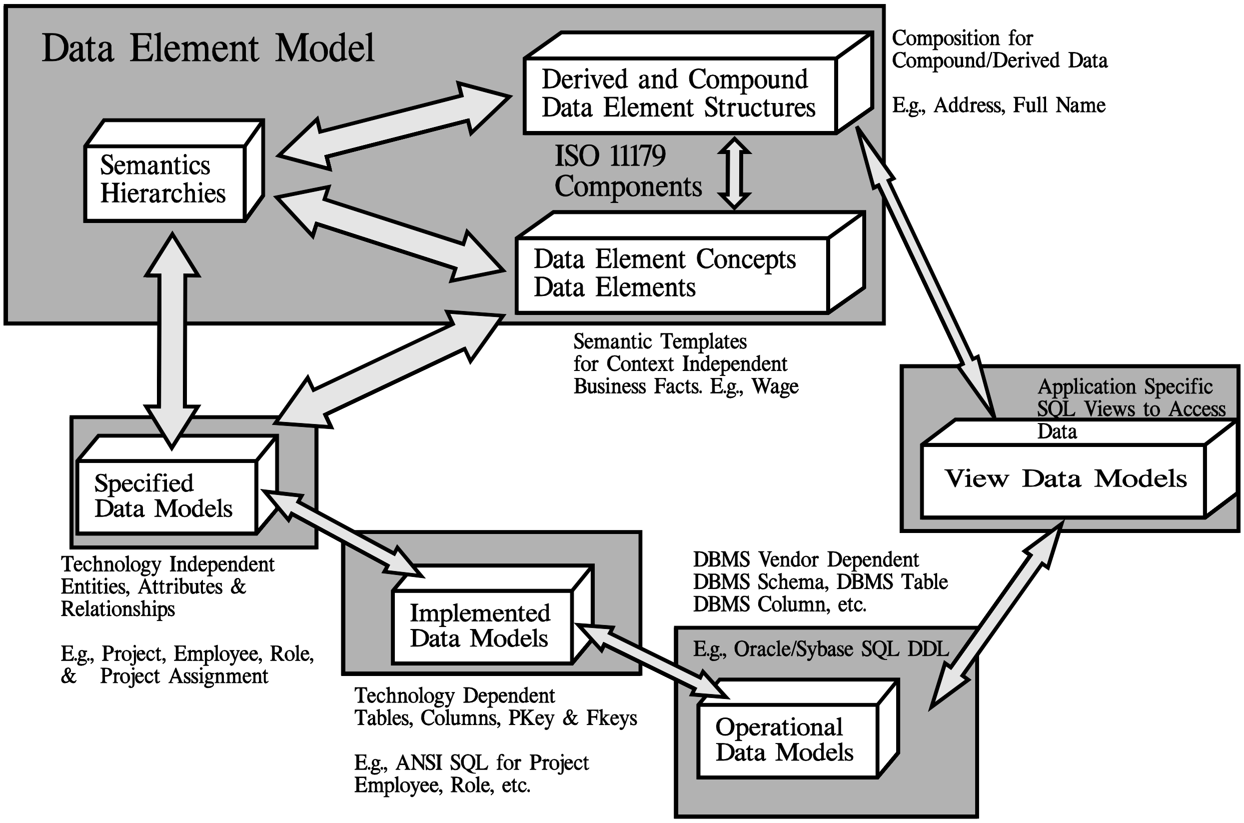

Figure 1. Whitemarsh Metabase's "Data" Data Models.

3. Whitemarsh Metabase Data Models

As stated, this short paper focuses on the Whitemarsh metabase's five data data-models, that is, Data Elements, Specified, Implemented, Operational and View data models because these models--collectively--define the meaning and allowed uses of data within databases, and across the enterprise.

4. Data Element Models

The data element model really consists of three parts: Semantic Hierarchies, Derived and Compound Data Element Structures, and the ISO 11179 Components including Data Element Concepts and Data Elements.

A very key concept here is that a data element is used as a semantic template (a rubber stamp of business fact semantics) many, many times across all the other data models. It is common for a data element to be semantically deployed into different contexts 30 to 100 times.

Semantic hierarchies are a key component. They are like taxonomies of data concepts such as temporal (first, last, next, or annual, quarterly, etc.), or geographic (e.g., worldwide, United States, or New England), or precision (e.g., estimated, exact, forecast, or final). When these semantic hierarchies are used to provide the "words" that comprise a data element's name, then there can be automatic definitions and abbreviations.

The data element metadata model also supports the ISO standard 11179, thus there are layers of metadata supporting data elements including value domains, value domain value mappings, and data type assignments. These all greatly support the management of the meaning of data. Thus, semantics standardization designs the data quality problems out of IT right from the beginning is the right approach. It is both efficient and effective.

Data Element models can be created by the data administration groups or data resource management groups as some are now called, and be available for use by designers and developers.

5. Specified Data Models

Specified data models, also known as conceptual models of data, enable subject matter experts to define standard data structures that represent the data required for key policy execution surrounding assets, facilities, abstract and real property, financial transactions, human resources, and events. Once these data concepts are modeled, they can be used over and over as standardized data structures in database data models.

These conceptual models of data consist of subjects, entities, and attributes. Every attribute is required to be mapped to a data element. Hence all attributes are derived from standard semantics.

Semantic hierarchies from the data element model can also be used to form the words that comprise the name of an attribute. If semantic words are assigned to a data element, they are automatically inherited by the attribute. If a more restrictive semantic is to apply to an attribute it can be assigned. The attribute's name is then automatically adjusted. The semantics of the attribute are thus all those associated with the attribute, its entity, and those it inherits from its data element from within the data element model. This is a real management of meaning.

Because of this strategy, attribute definitions and also abbreviations can be largely automated. Given that attributes are mapped to value domains, then their value sets are either automatically restricted by the inherited value domain of the attribute's data element, or restricted further through the assignment of a more restricted value domain.

6. Implemented Data Models

Implemented data models, also commonly known as logical data models, are data models of database schemas. The schema is thus a "container" for the tables, and tables are containers for the columns. This is in contrast to the Specified Data Model wherein the subjects are just abstract collections of entities. That is why in the Specified Data Model relationships are able to cross subjects, but in the Implemented Data Model relationships are not allowed to cross schemas. In short, schemas are "physical barriers."

These Implemented Data Models of data consist of schemas, tables, and columns. A schema and its collection of tables are "uses" of collections of entities and their attributes. An actual table is thus able to include a whole entity, part of an entity, multiple entities, or an entity multiple times. For example, the Employee Table might have a partial Person Biographic entity, an Address entity, and then multiple uses of a Telephone entity. The table below illustrates this situation. Missing from the Person entity might be Birth City, Birth Country, and the like.

Employee Table			
Table Column	**Subject**	**Entity**	**Attribute**
Employee Id	Information Technology	Unique Identifiers	Identifier
Employee First Name	Person	Person	First Name
Employee Middle Name	Person	Person	Middle Name
Employee Birth Date	Person	Person	Given Name
Employee Gender	Person	Person	Sex
Employee Street 1	Address	General	Line 1
Employee City	Address	General	Municipality
Employee State	Address	General	State Or Province
Employee Zip	Address	General	Postal Code
Employee Home Phone	Electronic Connectors	Telephonic	Telephone Number
Employee Cell Phone	Electronic Connectors	Telephonic	Telephone Number

Employee Table			
Table Column	**Subject**	**Entity**	**Attribute**
Employee Office Phone	Electronic Connectors	Telephonic	Telephone Number
Employee Emergency Phone	Electronic Connectors	Telephonic	Telephone Number

Semantic hierarchies from the data element model can also be used to form the words that comprise the name of a column. If semantic words are assigned to a data element, they are automatically inherited by the column. But, if a more restrictive semantic is already assigned to the column's attribute then the column automatically inherits those of the attribute. Finally, a more restrictive set is able to be assigned to the column. In that case, the column's name is automatically adjusted. The semantics of the column are thus all those associated with itself, its table, and then its attribute, and finally, those it inherits from its data element. Again, a real management of meaning.

The value of this approach is significant: Define once, use many times. This strategy enables database designers to employ standardized collections of fact specifications in the building of database schemas. Again, this goes a long way to managing the meaning of data. Every column in every table is mapped to an attribute, which in turn are mapped to data elements, which in turn are mapped to upper level semantics about concepts and that have their value domains restricted by prior agreement.

Again, because of this approach, column definitions and abbreviations can be largely automated. Given that columns are mapped to value domains, then their value sets are either automatically restricted by the inherited value domain of the column's attribute, or restricted further through the assignment of a more restricted value domain.

7. Operational Data Models

Operational Data Models, also commonly called physical data model are analogous to Implemented Data Models (logical data models) except that they are tuned specifically for particular DBMSs (e.g., Oracle, DB/2, Sybase) and for particular computer hardware situations. Physical databases are those that actually "contain" real data and that interface with real business information systems.

These Operational Data Models of data consist of DBMS schemas, DBMS tables, and DBMS columns. Operational data models are not just transformations of logical data models.

While Operational Data Models are also containers, the relationship between implemented and operational is just not one of implemented (one) too operational (many). A common style of a database is the data warehouse which is most often the melding of data from multiple source databases into a single database. If there are multiple source data models at the implemented level, then the one data warehouse (the target) data model is in a many-to-one relationship. Additionally, there could be multiple physical design variations of a particular data warehouse. Thus, the relationship might also be one-to-many. Hence, like in the many-to-many relationships between Specified Data Model and Implemented Data Model there is also a many-to-many relationship between Implemented Data Model and Operational Data Model.

The value of this approach is as significant as before: Define once, use many times. This strategy enables database designers to employ standardized collections of fact specifications in the building of DBMS schemas. Again this goes a long way to managing the meaning of data. Every DBMS column in every table is mapped to a database table column, which in turn is mapped to an attribute, which in turn are mapped to data elements, which in turn are mapped to upper level semantics about concepts.

Thus, because of this approach, DBMS column definitions and abbreviations can be largely automated. Given that DBMS columns are mapped to value domains, then their value sets are either automatically restricted by the inherited value domain of the DBMS column's Implemented Data Model table column, or restricted further through the assignment of a more restricted value domain. Again, even more management of meaning.

8. View Data Models

View Data Models are the interface between databases and the application information systems. Views can be mapped to different database schemas thus enabling a given application program to access multiple databases operating under different DBMSs.

A very key value of View Data Models is that the view column name can be different from the DBMS column name from which it is derived. Thus, view column names can be set to those of it ultimately inherited data element. So too can the definition of the view column be inherited from the data element. This can be automated.

9. Conclusions

Managing the meaning of data is of vital interest. Connectivity interoperability (that is, I hear you) is not sufficient. Understanding interoperability is essential. The time and resources necessary to broker understanding-based data interoperability takes too long and is too subject to changing technologies. Instead there must be a top-down facilitation of creating standardized data elements and standardized concept-based data structures that can be employed over and over in different database schemas.

Additionally, with the advent of enterprise resource packages (ERP) from vendors such as SAP or Oracle, the ability to have all the enterprise application data conform to each other through brokering is just not practical in any timely manner. And, if an enterprise did conform, wouldn't it be completely captive by the ERP vendor? A better approach is to have standardized semantics through the approach described in this paper and to then have the ERP package read and write from a single enterprise-semantics based interchange database. That way the enterprise could establish a single strategy to manage the meaning of all its data and accomplish that strategy in an efficient and effective manner.

In case this all seems like a significant quantity of work, the current approach, one-off definitions of database columns, is profoundly more expensive and time consuming. In the United States Department of Defense, a quick estimate shows that there are about 308 million database columns. At 15 minutes each for just a definition, that's $7.7 billion dollars. Added to that is all the one-off analysis and design time, and the estimated $350,000 per year create and management a reasonable sized interface between systems. In 1995 the Air Force was spending $167 million per year on those interfaces.

10. References

The following documents are available free from the Whitemarsh website:

Paper	URL
Data Modeler Architecture and Concept Of Operations Reverse and Forward Engineering Guide Metabase Module User Guides	http://www.wiscorp.com/metabase_demo.html
Comprehensive Metadata Management (short paper).	http://www.wiscorp.com/featured_papers.html
DAMA 2002 - Metadata Architecture for Enterprise Wide Data Sharing - Problem Specification DAMA 2003 - Metadata Architecture for Enterprise Wide Data Sharing - Problem Solution	http://www.wiscorp.com/DatabaseDesignInformation.html

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 - Data Standards Architectures And Implementation Data Management Program - Engineering Data Management Program - Metadata Architecture For Data Sharing Data Management Program - Tag And Post Vs Data Standardization	http://www.wiscorp.com/EnterpriseDatabase.htm
Iterations of Database Design	http://www.wiscorp.com/DatabaseDesign.htm
Data Is Executed Policy	http://www.wiscorp.com/DatabaseProjects.htm
A Column By Any Other Name is Not a Data Element Presentation (several paper and courses) Achieving Data Standardization (several papers, a book and courses) Achieving Enterprise Wide Data Semantics Standardization An Old Saw That Just Wont Cut An Old Saw That Just Wont Cut - Software Implementation Report Analysis of the DISA 8320 Data Standardization Approach Data Standardization Work Plan The Data Standardization Problem	http://www.wiscorp.com/DataQuality.htm