

# The Data Administration Newsletter (TDAN.com)

[Robert S. Seiner - Publisher](#)

[Main Menu](#)

[New Articles](#)

[Article Archive](#)

[Special Features](#)

[What's New](#)

[Contact the Publisher](#)

## COMPREHENSIVE METADATA MANAGEMENT

Written by Michael Gorman - Whitemarsh Information Services Corp.

[\[Return to the Article Archive\]](#)

### 1.0 Rationale for Comprehensive Metadata Management

*No one would ever question why a business needs it's finance books. Well, the metadata repository is the business's information systems' books. If you cannot run a good business without the former, you cannot run good information systems environment without the latter.*

A significant portion of the time and costs associated with resolving the Year 2000 problem can be directly attributed to a lack of a quality metadata environment within information systems organizations. The fact that one information system organization within an enterprise had virtually no Year 2000 problem while another organization within that same enterprise was running their information systems shop "24x7" to develop and install Y2K solutions was no accident. The former had a long history of metadata management and the later thought metadata was a wasted overhead expense.

In the development of large data processing projects dealing with enterprise-wide, indispensable business functions, documentation of the design requirements and resulting information system specifications is seldom accomplished such that it is timely, accurate, or complete. That is disastrous for the following three reasons:

- Only the momentous facts that are remembered are recorded.
- As systems are specified, the lower-level design details are redundantly developed, often in conflicting manners.
- As system components are maintained, the efforts are crippled because of the undocumented business knowledge that is essential to understanding the component.

Collectively, the entire set of business information system specifications up through requirements and extending into the "data" that defines, structures, and models the activities of the enterprise are metadata. This paper addresses the need for a comprehensive metadata management environment that is woven into the very fabric of database information system specification, implementation, operation, and evolution so as to successfully specify, design, implement, operate and maintain complex information technology components of enterprises. In support of this objective, the paper addresses the following topics:

- What Is Metadata?
- Architecture Framework for a Metadata Management System
- Metadata Management System Use Scenarios

## 2.0 What is Metadata?

The quick answer, of course, is that metadata is data about data. However, that's too cute. More formally the string, metadata is divided into *meta* and *data*. Meta in the Oxford Dictionary means, "something of a higher or second-order kind." The word, data, however is not employed within this paper in its strictest sense, that is, a data item like *Birth date = 03/22/1941*, but in more general sense so as to include unstructured data like text and diagrams.

For the purposes of this paper, the scope of metadata is restricted to Information Technology. Consequently, metadata are the materialized artifacts that define the requirements for, the specifications of, design of, or even executing characteristics of an IT system, or component of that system. "System" here is used in a very broad context. Thus, included within the scope of systems are databases, application systems, and their technology environments. Therefore, metadata is all that which is one or more levels of abstraction removed from the actual databases, applications, or their technology environments. In a computing environment, metadata would therefore include:

- Requirements
- Functional descriptions
- Work plans
- Database designs through to schema DDL (data definition language)
- Application system designs possibly through to computer program source code libraries
- Technology environment designs through to actual installation artifacts

But within this context, would not include:

- Actual databases with data records of employees, invoices, products, and customers
- Executing application systems
- Operating systems and other systems software such as DBMS and Web browsers
- Telecommunications Networks
- Computers

These are not metadata because they are "real," while the previous list represents artifacts about the reality. But once the information system is executing, metadata may be created that describes the characteristics of the operating environment. That class of metadata would include for example:

- Computer system execution schedules

- Computing resource consumption requirements
- Quantity of records in particular files
- Quantity of users by time of day for particular processes
- Job completion and/or error messages

### 3.0 Architecture Framework For a Metadata Management System

From the previous sections, the mission of a comprehensive metadata repository is to provide metadata in response to at least the following needs:

- *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 procedures performed by groups in their achievement the various missions of the enterprise from within different enterprise organizations.*
- *The organizations that are accomplishing what aspects of missions with what databases, information systems and through which functions.*
- *The key Resources (facilities, materiel, staff, etc.) How are they sequenced, interrelated, and how are they supported through databases and information systems.*
- *The information (a.k.a. query results or reports) is needed by various organizations in their functional accomplishment of missions and what databases and information systems provide this information.*
- *The data is needed by functional proponents, how is it defined within data architectures and databases and how and where are those databases deployed and then used by business information systems in support of mission accomplishment.*
- *The context independent semantic templates of data elements and how are these configured into models of data (the consequence of policy execution) determined as needed by functional experts in support of enterprise missions, and how are these specified data model requirements configured into implemented databases that ultimately operate within various organizations as they perform the functions needed by enterprise missions.*
- *The business information systems, where are they, how are they related to mission, organization, function, and databases. What is the impact on these business information systems when policy (a.k.a., data) is required or changed.*
- *The identification, allocation, and scheduling of database and information system projects within the context of Resource Life Cycles. Resulting project schedules are then able to be accomplished in a business resource defined sequence to best achieve enterprise missions through business functions and organizations.*
- *The identification, estimation, and then monitoring of database and business information system projects from within the context of well established metrics, and templates work breakdown structures, and deliverables.*

### 3.1 Metadata Management Data Architecture

It is not sufficient to merely infer and then list all the infrastructure work products that must be produced and managed, or to use a collection of CASE and modeling tools such as Erwin that do not derive their data from a single metadata repository. If it were sufficient, then the existing set of a Database project documents, Power Point presentations, and Erwin data models would have been adequate. What must be done is that the work products must be cast into database tables and completely interrelated into an integrated database that is commonly called a metadata repository. From the missions above, the following high level metadata object classes follow:

Application View Data Models	Information Needs and Characterizations
Business Calendars	ISO 11179 Data Element Metadata
Business Events	Logical Data Models
Business Information Systems	Missions
Conceptual Data Models	Organizations
Database Domains	Persons and Roles
Database Object Classes	Physical Data Models
Databases	Projects Management
Functions	Resources and Life Cycles

### 3.2 Metadata Management Application Architecture

The metadata management system flows from the enumerated set of metadata object classes within the Data Architecture section above required for comprehensive metadata management. Each subsystem within a comprehensive metadata management environment would operate on a subset of the metadata that exists within the enterprise-wide metadata database. A brief description of each inferred metadata subsystem follows:

An *Application View Data Model* metadata system component enables the generation, inventory, and maintenance of the views that an application system has of the databases to which it is loading data, retrieving data and/maintaining data. This metadata then enables a good knowledge of the uses of various databases within the various business cycles and calendars through the execution of the business information systems that contain the views. Since application views are interrelated with business events and calendars it is then possible to view database processing within the context of business information systems, business calendars, and/or business cycles.

A *Business Calendars* metadata system component enables the creation and interrelationship of the various business calendars that govern the accomplishment of business information systems within various business cycles.

A *Business Events* metadata system component enables the identification and interrelationship of the various business events that occur within the accomplishment of functions and then for each business event the various collections of business information systems that are executed in support

of that particular business event.

A *Business Information Systems* metadata system component enables the identification and interrelationship of various business information systems and their components to the application views that reference the databases upon which the business information systems act, and the business events that act as the triggers for the systems. Through these relationships the various business events along with their business cycles and calendars can be listed to then know of processing loads for each business information system.

A *Conceptual Data Models* metadata system component enables the exposition of the various conceptual data models that contribute conceptual subjects, entities and/or attributes to the development of one or more logical data models that are to be implemented within the enterprise. Because conceptual data models exist at a level of abstraction higher than logical data models they function as a coalescing mechanism for the use of the different data concepts employed within the logical data models. These models server as collection of data model templates available for use in the construction of logical databases. Because the conceptual data model is a level of abstraction lower than a 11179 Data Element metadata model each conceptual entity's attributes represents a deployment of the complete set of semantics of that 11179 data element. The conceptual data model metadata system component is supported by a full set of data modeling creation and re-engineering facilities including the importing and exporting of SQL DDL. It enables enterprise uses to view conceptual database models individually or across the enterprise.

A *Database Domains* metadata system component enables the full exposition of the data classes that exist within the context of a mission.

A *Database Object Classes* metadata system component enables the full specification of the data and processes that are contained within the DBMS layer of any modern database application environment. Included within the database object classes are its data structure that comprise the data segments of the database object class, the processes that create, modify, or delete rows of data of the database object tables, the states through which the database objects are transformed, and the database object information systems that transform the database objects from one valid state to the next.

A *Databases* metadata system component enables the visibility of the various databases of the various database architecture classes (i.e., original data capture, transaction data staging area, operational data store, data warehouses, and reference data) and their attendant schema based data model views, along with the associated business information systems that are supporting the functions of the enterprise through the various business calendars and events.

A *Functions* metadata system component enables the enumeration of the various functional hierarchies and commonly accepted variants of the business functions that represent the accomplishment of knowledge work by various organizations in their performance of the enterprise's mission.

An *Information Need and Characterizations* metadata system component enables the identification and characterization of the various information needs of the enterprise. Information needs are then interrelated with the various functions, organizations, and missions so that they can be viewed together.

An *ISO 11179 Data Element Metadata* system component enables the creation of the various business fact templates and their semantics that are then employed to regularize all the attributes of entities and columns of tables. Include are the various components of the 11179 standard including concepts, data element concepts, data elements, conceptual value domains, value domains, and

value domain values. Collectively, when interrelated with all the other data-based metadata enables data standardization and sharing across all the various database architecture classes and database applications that operate on these databases.

A *Logical Data Models* metadata system component defines the various databases that are to be implemented within the enterprise. Each such implemented data model has yet to be transformed into the design required by a particular SQL DBMS. Each logical data model consists of tables, columns, and relationships. Each column is related to its 11179 data element and to its appropriate conceptual data model entity attribute. Logical data models can be boot-strapped into existence through conceptual data model entity, entity-set, or attribute imports. Conversely, conceptual data models can be built through the promotion of a logical data model. Logical data model table column value domains may be restricted by valid value lists, ranges, and/or excluded value lists. Within an large functional area of an enterprise there may be several dozen original data capture databases, a large quantity of TDSA databases depending on their architecture, a dozen or so operational data store databases, a similar quantity of data warehouse databases, any number of data marts, a few reference data databases depending on decisions regarding distribution. The logical data model metadata system component is supported by a full set of data model creation and re-engineering facilities including the importing and exporting of SQL DDL. It enables enterprise users to view logical database models individually or across the enterprise.

A *Missions* metadata system component enables the identification and definition of the set of missions that are undertaken by the enterprise. Once identified these would be able to be interrelated with the appropriate database domains, functions, and organization, and through other relationships to know of the various databases and business information systems that operate on various business events and cycles.

An *Organizations* metadata system component enables the incorporation of the various organizations that exist within the enterprise and the interrelationship with enterprise missions functions, business events and calendars and their associated business information systems. These enable the full exposition of the activities of various organizations via their functions and business information systems.

A *Persons and Roles* metadata system component enables the capture of the various staff that exist within the enterprise and the roles they play within functions and organizations.

A *Physical Data Models* metadata system component enables the creation of the actual DBMS-based data models that are then compiled and are operating with the business information systems to collect, store, evolve and report enterprise data. These operational physical data models consist of its database reference, DBMS schema, DBMS tables, DBMS columns, and relationships. The operational physical data models can be boot-strapped into existence through logical data model table, table-set, or column imports. The physical databases are interrelated directly with the application view models and also with their logical data models. The physical data model metadata system component is supported by a full set of data model creation and re-engineering facilities including the importing and exporting of SQL DDL. It enables enterprise users to view physical database models individually or across the enterprise or within the context of logical data models.

A *Resources and Life Cycles* metadata system component enables the identification of the various resources within the enterprise that collectively represent either the infrastructure or external product set of the enterprise. Infrastructure resources include for example, staff, facilities, contracts, finance, and the like. External products include manufactured products, services to customers, and the like. Each resource is then defined in terms of its life cycle. Resource life cycle nodes from different life cycles are interrelated to show enterprise-based interdependencies. Databases and Business information systems, and information needs are then interrelated to each life cycle node. Collectively

the fully attributed resource life cycles enable the enterprise to view its complete operation in terms of its essential resources that define its very existence.

### 3.3 Metadata Management Technical Architecture

The technical architecture of any database application consists of an enumeration of the characteristics of its logical database, physical database, interrogation, system control, and computing infrastructure operating environment.

The characteristics of the logical database of a comprehensive metadata management system include:

- Each meta-entity is expressed as a separate ANSI standard SQL table.
- All relationships are expressed as traditional ANSI standard SQL relationships.
- All referential integrity and referential actions are schema based.
- Column and table constraints as well as assertions and triggers are SQL based.
- All SQL statements are not vendor proprietary.

The characteristics of the physical database include:

- All metadata is loadable through either vendor provided 4GL, or 3GLs, or through an ODBC access metadata application presentation layer.
- Supported by a process that can read SQL DDL for use in creating data models at the conceptual, logical, and physical levels.
- Able to export XML wrapped metadata through DBMS vendor and/or metadata application provided facilities.

The characteristics of interrogation include:

- Ability to report directly from the metadata database's explicit SQL schema through report writers provided by the SQL Vendor or through third party vendors such as Crystal Reports.
- Ability to publish metadata in HTML for the Internet.

The characteristics of system control include:

- Supported by the SQL DBMSs facilities for at least audit trails, transaction rollback, logical and physical database re-organization, and security and privacy.

The characteristics of the operating environment include:

- Operational on any MS/Windows operating system
- Able to place the meta data database on a server running under either a Windows O/S or Unix
- Operational under any commonly available SQL based DBMS through ODBC.

## 4.0 Metadata Management System Use Scenarios

A comprehensive metadata management system can either be a passive repository for knowledge work accomplished or can be integral component of accomplishing knowledge work. Clearly the later is preferred as the population and use of the facility cannot then be ignored. If the policy is made that a deliverable exists only after it is able to be retrieved from the metadata repository, and that corrections or revisions of deliverables are accomplished only when they are retrieved from the metadata repository then the repository will certainly take on a critical, central, and active role within any knowledge work project environment. With that as a given, the following are typical use scenarios for a comprehensive metadata repository and its attendant metadata system:

- Build, maintain, and employ business cycles, calendars and interrelate business information system execution cycles
- Build, maintain, and employ business information system specifications
- Build, maintain, and employ conceptual data models
- Build, maintain, and employ database application projects
- Build, maintain, and employ database domain models
- Build, maintain, and employ database object classes
- Build, maintain, and employ function models
- Build, maintain, and employ information needs analysis
- Build, maintain, and employ information systems plans
- Build, maintain, and employ ISO 11179 data elements and supporting metadata
- Build, maintain, and employ logical data models
- Build, maintain, and employ mission models.
- Build, maintain, and employ organization models
- Build, maintain, and employ physical data models
- Build, maintain, and employ resource life cycles

Each scenario is briefly described.

**Build, maintain, and employ business cycles, calendars and interrelate business information system execution cycles.** The actual workflow of a collection of business information systems exists within business cycle, business events, and business calendars. Each business cycle is defined so that the sequence for the accomplishment of business information systems is clear. Each business information system is then activated by the business event that is associated with the business cycle. Business calendars need to also be defined as they may contain specific days on which certain processes must be completed or cannot occur. Business calendars must be interrelated with business cycles.

**Build, maintain, and employ business information system specifications.** Each business information system specification is hierarchical and thus includes subsystems. Each subsystem is named and generally described as to its content and purpose. The levels of detail, for example pseudo code for business information system modules are purposely omitted because that is best left to database information system development environments. If that level of detail were in the metadata repository then there would be a 100% likelihood that it would be out of synch with the actual business information system. Business information systems are integrated with the database object classes that they invoke to transform database objects from one state to the next and are also integrated with the business events that in the name of the business function cause the execution of the business information system.

**Build, maintain, and employ conceptual data models.** Conceptual data models are collections of entities, attributes and relationships that can be used as data model templates for logical databases. Each entity within a conceptual model should be the data specification of a well defined policy within the enterprise. A collection of entities within a particular subject should conform to a larger and more complex policy. A logical database is bounded by schema and is intended to be implemented by a particular DBMS thus arising in an operational database that collects, stores, and maintains actual business data. In contrast, the conceptual database's entities are bounded only by the subject within which it is defined. In the construction of a logical data model, one more entities may contribute attributes to form the column of the logical data model's tables. Conceptual data models enable the creation of standard data structures that when employed in a logical data model ensure completeness, rigor, and the data standardization essential for data sharing. The semantics of attributes of a conceptual data model entity are derived from ISO 11179 data elements. In total, the ISO 11179 data elements, conceptual data models, logical data models, and physical data models all form a general hierarchy of business facts within the enterprise that enable a clear picture of where and how all business facts are defined and deployed. Conceptual data models can be created inductively through the promotion of a single logical data model to the conceptual data model level. Then, data modeling activities would occur to break apart the conceptual data model into individual subjects and collections of entities within those subjects. Entities can be interrelated across subject areas to represent conceptual data model factoring.

**Build, maintain, and employ database application projects.** Each database application project consists of a work plan, deliverables, assigned staff and a work environment. As projects are proposed they are set within the context of information systems plans. Each project's metadata is linked to the actual deliverable's metadata so they can be reviewed to better understand the work accomplished. As work is performed, work-accomplishment time-cards are entered so that earned value reporting is automatically produced. Since the projects would have been estimated via standard metrics, the actual accomplishments can be used to adjust the metrics. Finally, since all projects exist within the metadata repository they can be viewed and analyzed collectively or individually, or in groups of contained project tasks.

**Build, maintain, and employ database domain models.** Database domains are "noun-intensive" descriptions of the data that is inferred by the lowest level of a mission hierarchy. Each database domain is thus restricted in scope to that of the mission leaf. Additionally, each database domain is represented by a simple entity-relationship diagram (ERD). When all the relevant database domains are completed their ERDs are combined to ensure that the entities that are named the same are in fact the same and are represented at the same level of granularity.

**Build, maintain, and employ database object classes.** Database object classes are the encapsulated data structures, processes, and constraints necessary to transform a set of data from one value state to the next. Database object classes are essential to the integrity of databases. In modern SQL DBMSs, database object classes are largely able to be constructed through the use of

persistent views that map to a collection of columns across a set of tables. The value state integrity is governed by columns and table constraints. The value states are transformed through stored procedures within assertions and triggers. It is important to define database object classes within the domain of the DBMS to ensure that all external language agents such as 4GLs, query languages, and 3GLs are forced to proceed through these DBMS defined and encapsulated database object classes.

**Build, maintain, and employ function models.** As a database project commences, it is important to know just what role it will play within the manual functions that are accomplished by any organization within the scope of a mission. The hierarchical function models are created and interrelated with the various organizations that perform them. Because functions are human activities, there may be multiples sets of functions that are generally equivalent but differ in style of knowledge worker processes. The differences are not critical because the relationship between a business information system and a business function is through the intermediary, business event.

**Build, maintain, and employ information needs analysis.** As a database project is started, it is important to know just what are the information needs that are to be encompassed within the database design. The information needs are thus gathered and stored into the metadata repository along with their characteristics such as timeliness, granularity, production needs, and the like. The information needs are interrelated with both the functions that are being supported by the information needs, and the resource life cycle nodes for which the information needs essentially become the work product evidences of the resource life cycle node state.

**Build, maintain, and employ information systems plans.** Every project within an enterprise commonly requires the specification and implementation of multiple information systems. Within an enterprise as a whole there may be hundreds of information systems being planned. A comprehensive information system plan sets all the information systems within the context of the resource life cycle nodes, and then estimates their duration via standardized project methodologies and standard metrics. This enables the enterprise to view all its projects, and to know the effects of accelerating and/or delaying any particular project.

**Build, maintain, and employ ISO 11179 data elements and supporting metadata.** Attributes of entities and columns of tables should all draw their semantics from data elements. A data element is a context independent (i.e., entity and/or table independent) business fact semantic template. It is well accepted practice that the quantity of data elements are a small fraction of attributes and/or columns. Supporting data elements are multiple higher levels of data element metadata including concepts, conceptual value domains, value domains and sets of values. The values sets can be directly allocated to DBMS schema columns as constraints. More likely they would form the rows of data within the reference data database.

**Build, maintain, and employ logical data models.** A logical data model is a collection of tables, columns, and relationships bounded by a schema. Logical data models are built as a precursor to the design of the database object classes that operate to maintain data integrity and value transformations. It is common to build a logical database within the scope of a reasonably large mission hierarchy such as human resources, finance, facilities, customers, sales management, distribution, or inventory. Database object classes are accomplished through business information systems. Logical databases commonly conform to particular data architecture classes such as original data collection, transaction data staging area (TDSA), data warehouses, data marts, and reference data databases. Logical database table columns should all be derived from attributes from entities of one or more conceptual data models. Logical data models also act as the "parent" of one or more physical data models. In total, the ISO 11179 data elements, conceptual data models, logical data models, and physical data models all form a general hierarchy of business facts within the enterprise that enable a clear picture of where and how all business facts are defined and

deployed. Logical data models can be created inductively through physical data model imports that exist within a certain scope, and then through the promotion of a single physical data model to the logical data model level. Then, data modeling activities would occur to expand the scope of the logical data model to be that of the union of all the physical data models.

**Build, maintain, and employ mission models.** The mission models are the boundaries of the scope of the enterprise. It is within mission models that database domains that lead to database designs are created. Missions are also the scope boundaries for all enterprise organizations and functions.

**Build, maintain, and employ organization models.** As a database project commences An organization model is built to then allocated to the various missions and functions. This permits the easy identification of those components of the enterprise that are involved in any database project effort.

**Build, maintain, and employ physical data models.** The physical database is a logical database that may have been subsetting and/or transformed to server the particular needs of a DBMS, or performance requirement. Physical databases are mapped back to their "parent" logical models through a column (logical data model) to DBMS column (physical data model) mapping. Physical data models, are "hosts" to the various SQL views that in turn act as intermediaries to the business information systems that access the databases. There may be multiple transformations of a particular logical database, and each exists and is mapped back to its "parent" logical database. In total, the ISO 11179 data elements, conceptual data models, logical data models, and physical data models all form a general hierarchy of business facts within the enterprise that enable a clear picture of where and how all business facts are defined and deployed.

**Build, maintain, and employ resource life cycles.** Enterprises can be viewed as a collection of resources that are moved through well defined life cycles. In this context, resources, some concrete and some abstract would include staff, finance such as payables, receivables, and payroll, as well as facilities, contracts, customers, sales management, distribution, products, manufacturing lines, inventory, missions, functions, organizations, reputation and the like. Each life cycle node of a resource, for example the recognition of a receivable is commonly supported either by manual or automated systems and databases. A resource life cycle node from one resource life cycle can be related to a node on another life cycle as a way of facilitating the related-to node. For example, Issuance of a contract from within the contacts resource facilitates the recognition of the receivable within the receivables resource. This interdependence enables the enterprise, as a whole, to be seen as an network of interconnected resources that has to function effectively as a complete system for the enterprise to be successful. Assisting in the effectiveness of a given resource life cycle node of a resource are the databases and information systems that assist persons who are performing functions within their organizations in support of the enterprise's mission.

---

*A longer version of the Comprehensive Metadata Management paper will be available on July 2, 2003. To view the longer version, please click here ... <http://www.wiscorp.com/featuredpapers.html>*

**Copyright 2003 © Whitemarsh Information Systems Corporation**

---

*Michael M. Gorman, President of Whitemarsh Information Systems Corporation, has been involved in database and DBMS for almost **35** years. Mr. Gorman has been the Secretary of the ANSI Database Languages Committee, X3H2 for **25** years. X3H2 standardizes SQL. A full list of*

*Whitemarsh's clients and products can be found on the web site, [www.wiscorp.com](http://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.*

**[\[Return to the Article Archive\]](#)**

**The Data Administration Newsletter (TDAN.com)**

**Robert S. Seiner - Publisher - [rseiner@tdan.com](mailto:rseiner@tdan.com)**

**© Copyright 1997-2003 - The Data Administration Newsletter (TDAN.com) - [Disclaimer](#)**