Whitemarsh
Information Systems Corporation

Inductive Enterprise Architectures

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

1.0 Objective

The objective of this paper is to present the Whitemarsh approach to inductively discovering and then creating an enterprise architecture.

Why this type of effort be undertaken arises because it is common for an enterprise to have created over a long time a large collection of requirements, specifications, databases, data models, procedures and all manner of Information Technology work product artifacts without there first having been a “grand design.”

It is not that any one application information system or database, or use of a DBMS may have been wrong, or misguided, or mis-created, it’s just now that a very large collection of all these IT artifacts have been created and there likely is the desire to regularize, optimize, and even standardize them all into a “grand architecture. “

An initial question might arise as to whether there could have been a “grand design” from the very beginning? It is Whitemarsh’s position that the answer is simple: No. That said, however, there is an overall approach to the inductive development, mainly through reverse engineering of such a grand architecture.

From Whitemarsh’s perspective, enterprises are not recognizing that John Zachman’s most favorite saying may have come to pass. That is:

"Someday, you are going to wish you had
all those models,
Enterprise wide
horizontally and vertically integrated
at an excruciating level of detail"

A key facilitator in the inductive achievement of an inductive enterprise architecture is the use of the Whitemarsh software system, Metabase. It can be used to identify, capture, integrate and interrelate in an non-redundant manner all the required data management work products that comprise a comprehensive enterprise architecture.

The Metabase System’s metadata data models clearly show that the Metabase System is a database-centric software fulfillment of John Zachman’s always stated, and cited above, admonishment to the data management community, that is, “Some day.....” Through the work accomplished via the steps outlined in this short paper, John Zachman’s admonition can not only be replaced but also established such that there will be a continuing evolution and maintenance of an enterprise’s architecture.

Whitemarsh employs a framework that has been specially designed for the knowledge worker, which all IT and “white-collar” persons are. This ***Knowledge Worker Framework*** was modeled after John Zachman’s framework in terms of it’s 6x6 row and column design. The rows are the same. The Knowledge Worker Framework columns, however are somewhat different from John Zachman’s framework columns as they were arrived at through an inductive analysis of the Whitemarsh comprehensive data-centric business information system methodology that has been employed very successfully and has been continuously improved since the middle 1980s.

The six columns of the Knowledge Worker Framework are:

- Mission
- Database Objects
- Business Information Systems
- Business Event
- Business Function, and
- Business Organization

The cells of this framework for the knowledge worker are populated in a highly engineered fashion with the work products needed to “tell” the stories of collections of enterprise architectures. Table 1 sets out the rows, columns, and cell-base enterprise work products of the Knowledge Worker Framework. As an aside, this collection of work products has been evaluated against many $100 Million large scale IT failure reports published by the U.S. General Accountability office. The Knowledge Worker Framework is set out in Table 1.

Whitemarsh Knowledge Worker Framework						
Perspective	**Mission**	**Database Object**	**Business Information System**	**Business Event**	**Business Function**	**Organ-ization**
Scope	Business missions	Major business resources	Business information Systems	Interface events	Major business scenarios	Organizations
Business	Mission hierarchies	Database Domains, and Resource Life Cycles	Information sequencing and hierarchies	Event sequencing and hierarchies	Business scenario sequencing and hierarchies	Organization charts, jobs and descriptions
System	Policy hierarchies	Data Elements Specified data models and Identified Database Objects	Information system designs	Invocation protocols, input and output data, and messages	Best practices, quality measures and accomplishm ent assessments	Job roles, responsibilities, and activity schedules
Technology	Policy execution enforcement	Implemented data models and Detailed Database Objects	Information systems application designs	Presentation layer information system instigators	Activity sequences to accomplish business scenarios	Procedure manuals, task lists, quality measures and assessments
Deployment	Installed business policy and procedures	Operational data models	Implemented information systems	Client & server windows and/or batch execution mechanisms	Office policies and procedures to accomplish activities	Daily schedules, shift and personnel assignments
Operations	Operating business	View data models	Operating information systems	Start, stop, and messages	Detailed procedure based instructions	Daily activity executions, and assessments

Table 1. Knowledge Worker Framework.

An analysis of the critical errors cited GAO IT failure reports was undertaken and these errors were allocated to the cells within which the failure occur. Over 40% of all IT system failures occur from errors in the cells across all six columns of the first two rows. 50% of IT system failures occur from not properly adapting the enterprise to newly created IT systems. This 50% resides in the last two columns from rows three through six. The remaining errors, only10% of all IT system failures can be properly laid the feet of IT.

One of core collections are the artifacts reside squarely in the Database Object column and form an enterprise's Data Architecture Reference Model. This model contains:

- Data Elements
- Data Models of Concepts
- Logical Data Models of Databases
- Physical Data Models of Databases
- View Models that bridge Physical Databases and Application Information Systems

These five data architecture layers form the nucleus around which the large collections of supporting IT work products that are interrelated, interoperable, and non-redundant.

When the complete set of IT work products, including those contained in the Data Architecture Reference model are depicted within in a single diagram, Figure 1 occurs.

Now, if it is the intent of such an effort to merely collect all the work products associated with Figure 1 within various text, report, and diagram deliverables the result would fill file-drawers of several filing cabinets. Clearly, enterprise organizations already has enough of both.

While such collections can be seen as elegant and very informative, they are not able to integrated, interrelated, or be interoperable. How do you "query" Word documents? How do you do "joins?" Those three characteristics, that is, interrelated, interoperable, and non-redundant are the characteristics of a well organized database that is supported by a collection of highly engineered work-product functional modules of an overall application information system.

Needed then is an application information system which has at its core, a "metadata database." The data model of this "metadata database" is a data-centric rendition of the majority of the work products from the Knowledge Worker Framework including the core models form the Data Architecture Reference Model. This "metadata database" data model is highly engineered such that it truly supports the interrelatedness, interoperability, and non-redundancy.

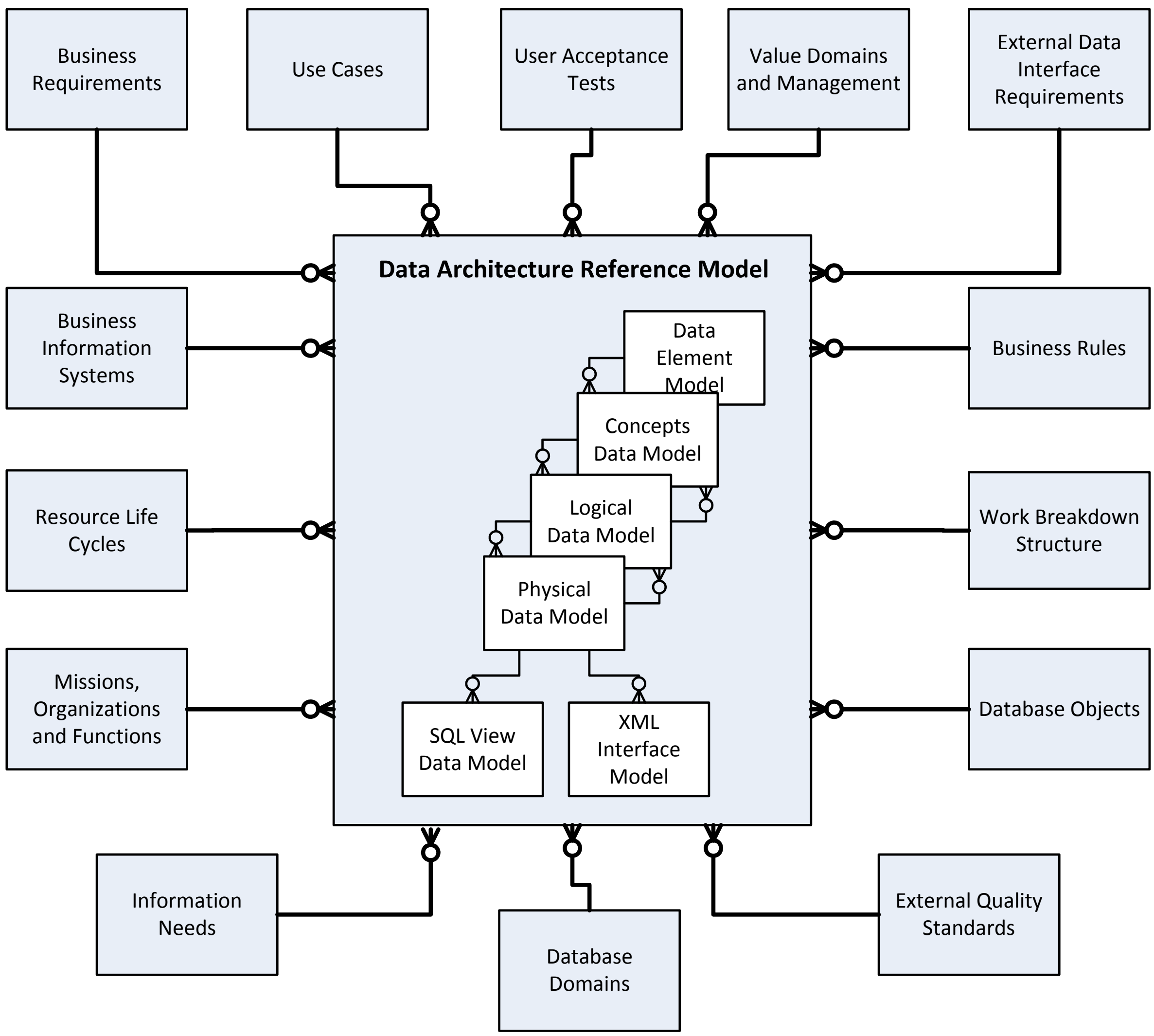

Figure 1. Enterprise Architecture Work Products set within the Data Architecture Reference Model.

The overall domain of the Metabase System is depicted in Figure 2.

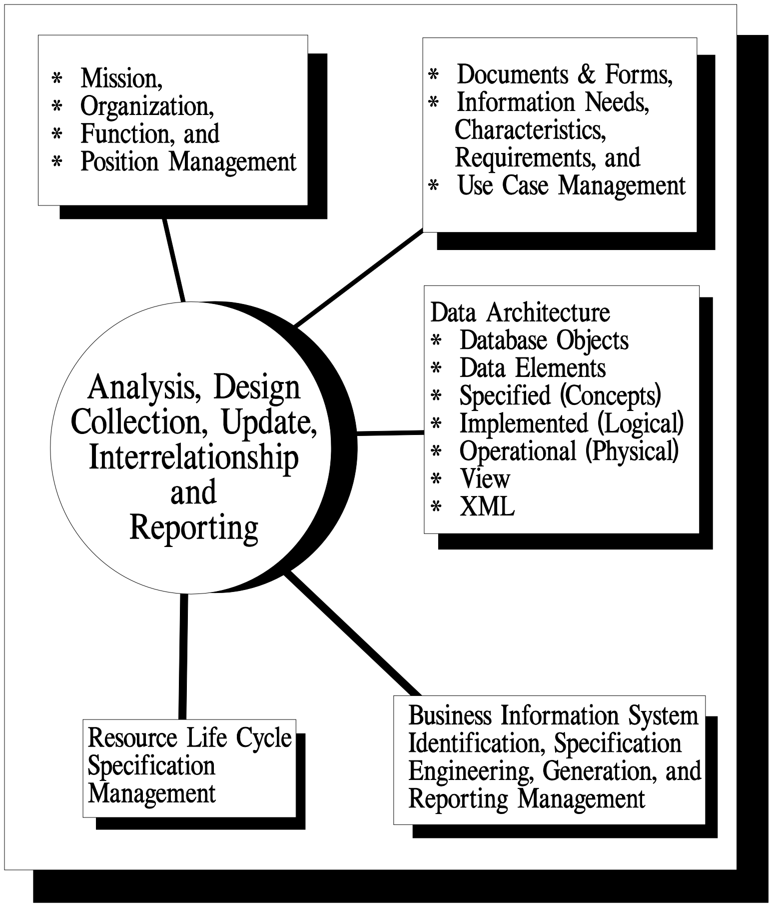

Figure 2. Metabase System Metadata Domain.

It is therefore within this context that the John Zachman saying, the Knowledege Worker Framework, the Data Architecture Reference Model, and the Metabase System that Whitemarsh's approach enables the analysis of existing IT work products with a view towards inductively arriving at an enterprise architecture..

2.0 Topics Covered

The topics is this paper include:

- The overall Whitemarsh metabase meta model

- Meta category value meta entities
- Data element model meta entities
- Specfied data model meta entities
- Implemented data model meta entities
- Operational data model meta entities
- Value domain meta entities
- Data type meta entities

3.0 Approach

Whitemarsh’s approach includes the following activities, listed in their general order of accomplishment:

- An inventory of COTS tools and systems employed to define, manage, and evolve these enterprise architecture components (see Section 3.1).

- An exposition of the customer's enterprise Missions, Organizations, and Functions (see Section 3.2).

- An exposition of the major Resources and their Life Cycles that enable the customer to carry out its functions through its organizations to accomplish its missions (see Section 3.3).

- The cross mapping of the Resources and their Life Cycles with the customer's Missions, Organizations, and Functions (see Section 3.4).

- An inventory of Databases, Business Information Systems, and Database Management Systems along with their descriptions (see Section 3.5).

- A cross mapping among those Databases, Business Information Systems, and Database Management Systems (see Section 3.6).

- The cross mapping of the Resources and their Life Cycles with Business Information Systems and Databases (see Section 3.7).

- An exposition of the major Business Event Cycles and Calendars over which business information systems and databases are utilized (see Section 3.8).

- The cross mapping of the Resources and their Life Cycles with Business Event Cycles and Calendars (see Section 3.9).

- An inventory of major past, current and future planned IT projects including major Deliverables (see Section 3.10).

- An inventory of staff, their areas of specialties and skills (see Section 3.11).

- The cross mapping of IT projects with Personnel and IT contractors.

Based on the inventory and storage of the enterprise architecture components above, the following analyses should be conducted on this “As-Is” enterprise architecture component inventory.

- Coverage
- Gaps
- Efficiencies
- Interoperability and redundancy
- Personnel and subcontractor deployment

Once these analyses are completed, executive level and actionable recommendations will be made regarding existing tools, the improvement of efficiencies, and any costs associated with data information management. These will point to a “To-Be” enterprise architecture component set.

The sections of this approach set out a more detailed explanation of the various Enterprise Architecture Components to be identified, characterized, stored in the Whitemarsh Metabase System and analyzed so as to produce a way-ahead.

Each section includes one or more diagrams that shows the collected information and how it is related to other sets of relevant information. These diagrams are at a higher level than the actual Metabase System data model diagrams.

Identified as well in each of the Section 2 subsections is the identification of the individual deliverables from Section 1.4 that are directly or indirectly affected by the subsection’s work.

Sections 3 though 7 of this proposal are outside the scope of both the deliverables and the intended work effort. These sections are provided so the next steps can be contemplated knowing that the work can just continue because the result of this effort is not just a series of reports, but the construction of a Metabase System database that contains the various enterprise architecture components identified in this Section 2.0.

If the generated reports are favorably received, both the Metabase System database used to store the collection enterprise architecture component information and the Metabase software system, can be procured and can remain to support the report's recommendations. Sections 3 through 7 address:

- The Metabase System
- The Business Information System Generator
- Independent Verification and Validation
- Information Systems Planning
- Follow-on Deliverables

The final section, Section 8, describes a number of relevant work efforts.

3.1 COTS Tools and Systems Inventory and Descriptions

An inventory of COTS tools and systems employed to define, manage, and evolve enterprise architecture components will be accomplished.

Included are data modeling tools, metadata repositories, DBMSs and their supporting DBMS management tools, ETL tool suites, metadata management systems, and business information system generators.

The descriptions of each of these tools and the mapping to the databases and business information systems that are managed, facilitated, and/or evolved through these tool sets will be developed.

Each of the tool types is identified and related to the collection of enterprise architecture components directly affected in terms of architecture, design, implementation, deployment, and maintenance.

The resulting "data" captured and/or generated by the tool is assessed in terms of redundancy, interoperability and integration. Addressed for each tool data-set is the ability for the tool's data set to be used across tools including the ease of data update and maintenance. The ability to making changes via one tool-set and having that changed data appear in an updated manner in other tool sets is assessed.

Examined as well is the deployment of DBMSs that act as database engines, agents of access for business information systems, and the like. Critically analyzed is the integration, interoperability and non-redundancy of all DBMS managed data.

3.2 Missions, Organizations, and Functions Discovery and Description

Figure 3 depicts a high level data model that supports the capture and interelationship of enterprise level missions, organizations and functions.

An exposition of the customer's enterprise Missions, Organizations, and Functions is undertaken. Collectively, the Missions, Organizations and Functions form the overarching architecture within which all the databases and business information systems are deployed, and through which all enterprise resources are transformed along their life cycles.

Missions define the very existence of the enterprise, both infrastructure and external, that are the ultimate goals and objectives that measure enterprise accomplishment from within different business functions and organizations.

Organizations accomplish the various aspects of missions with which databases, information systems and through which functions.

Functions are the human-based processes performed by organizations in their achievement the various missions of the enterprise. Human processes are common across and within organizations, business events and by indirection, business information systems and databases.

It is within the overall context of Missions, Organizations and Functions that the enterprise operates. Within these contexts the databases and business information systems exist. It is therefore important to the understanding of Enterprise Data Management that these three models be created and interrelate.

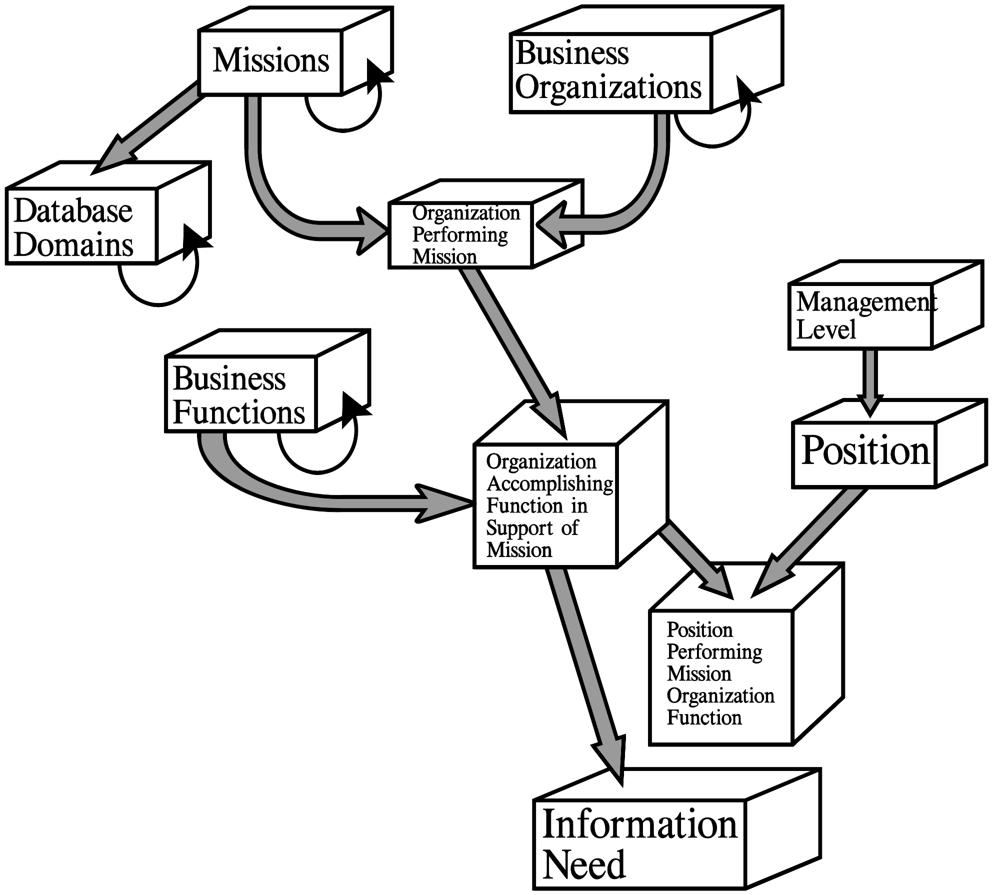

Figure 3. Mission Organization Function Position Metadata Model.

3.3 Resource Life Cycle Analysis and Description of the Enterprise

Figure 4 presents a high level data model for the capture of Resources and their Life Cycles. Collectively, these set out the major ways that the enterprise carry out its functions through its organizations in the accomplishment of its missions. Resources include, for example, facilities, materiel, staff, transportation, reputation, marketing, and finances.

As shown in Figure 3, Organizations accomplishing functions in support of missions have certain information needs. From Figure 4, the satisfaction of these information needs is accomplished by the database object and business information systems that produce the information made available from the various resource life cycle nodes of the essential resources of the enterprise.

Now, while Resources and their Resource Life Cycles are independent of missions, organizations and functions, Resources and their Life Cycles are accomplished through the deployment of databases and business information systems set within their respective mission,

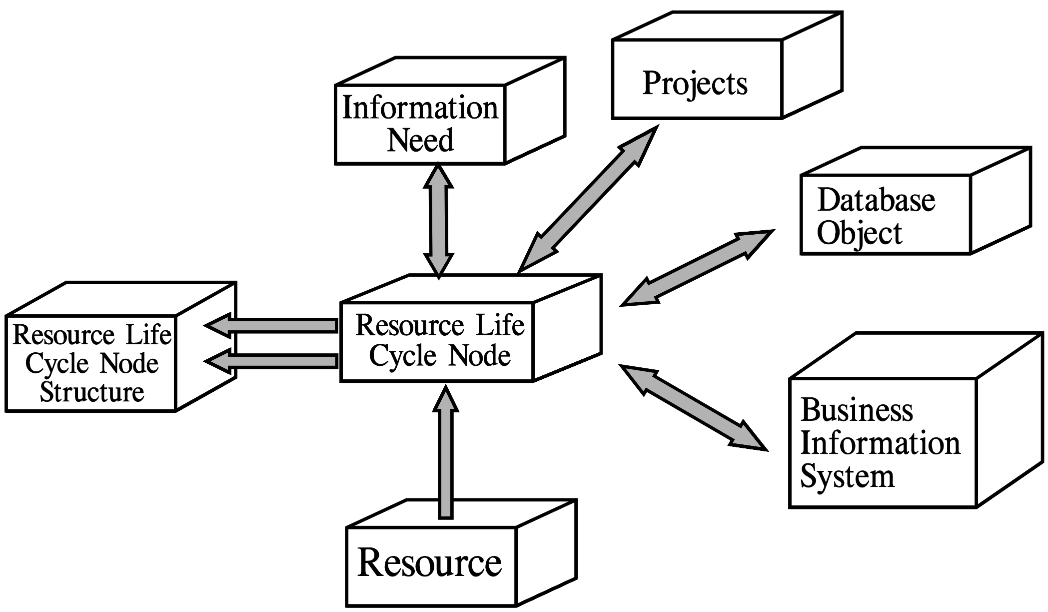

Figure 4. Resource Life Cycle Metadata Model.

organization and function contexts.

Resource Life Cycle nodes from different Resource Life Cycles are able to be intersected through precedence vectors that are represented in the Resource Life Cycle Node Structures.

Not only do these Resource Life Cycle Nodes and supporting Database Objects and Business Information Systems provide clear data gathering, manipulation and execution maps through the enterprise's Missions, Organizations and Functions, they also quickly enable the identification of gaps in both databases and also business information systems.

Finally, the Resource Life Cycle Node structures within and between enterprise resources form the critical backbone for the development of enterprise-wide business information system plans. These plans are outside the scope of this effort but are described in Section 6 of this proposal.

3.4 Resource Life Cycles with Missions, Organizations, and Functions Cross Mapping

Figure 5 shows how Figures 1 and 2 are interrelated to support the production of cross reference mappings between Missions-Organizations-and-Functions with Resources and their Life Cycles.

This cross-mapping enables the enterprise to understand how it accomplishes the creation, evolution and accomplishment of its identified resources within their life cycles. An essential by product is the surfacing of redundancies, conflicts and gaps which, in turn represent problems that have to be addressed.

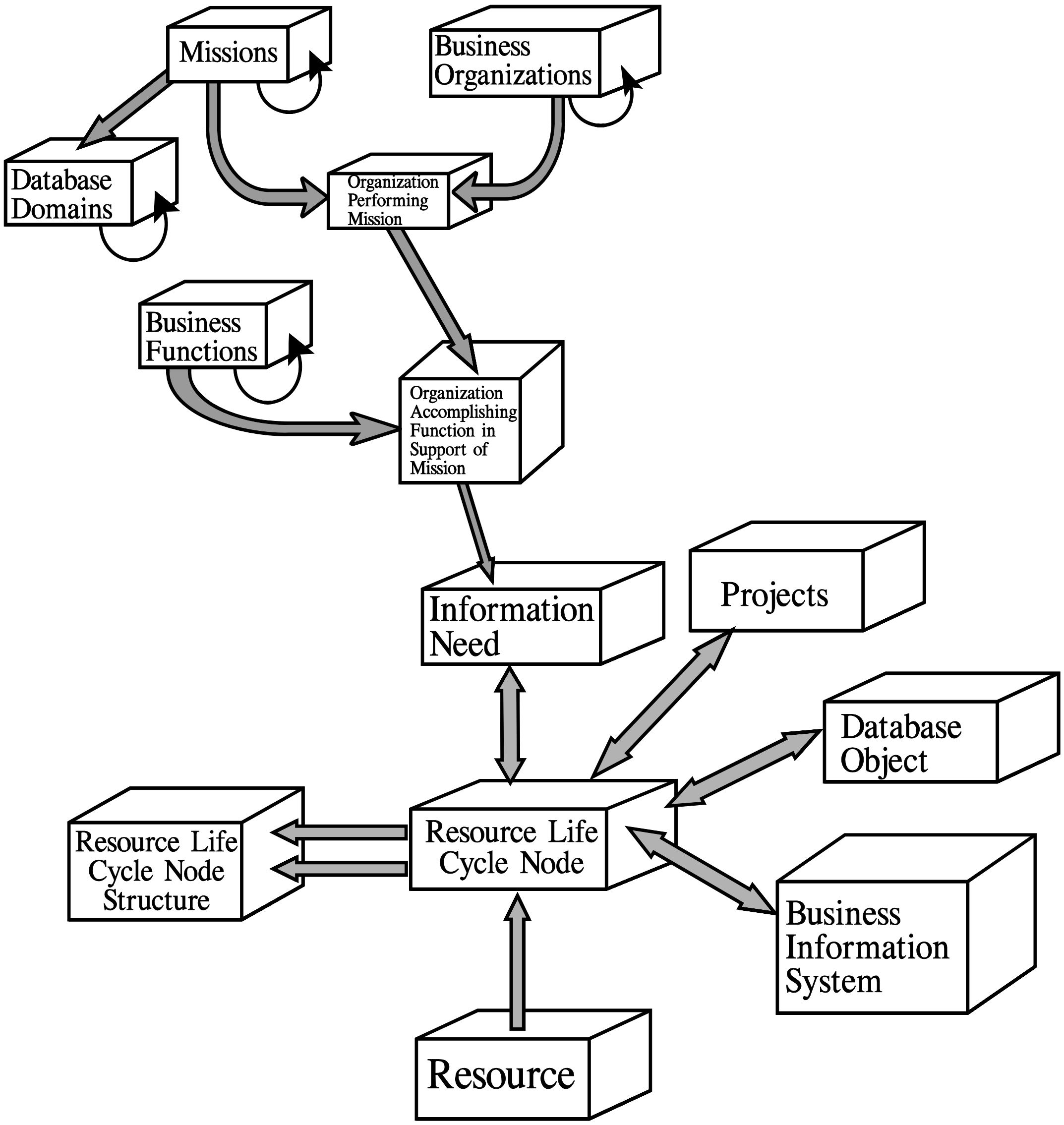

Figure 5. Mission-Organization-Function Cross Reference with Resource Life Cycles Metadata Model

3.5 Databases, Business Information Systems, and Database Management Systems (DBMS) Inventory

In this activity, an inventory of the Databases and their attendant data models, Business Information Systems, and Database Management Systems along with their descriptions is developed.

Databases include the database's schema (tables, columns, and relationships), the DBMS that acts as the database's engine, and the fundamental orientation of the database, that is, strategic, tactical, or operation, and the exposition of the database's architecture (original data capture, transactional data staging area, reference, operational data store, or data warehouse (wholesale and/or retail).

The business information systems are identified and described including, how they are related to mission, organization, function, and databases. Included as well are the impacts these business information systems have when policy (a.k.a., data) is required or changed.

3.5.1 Databases

Databases exist across the enterprise and are identified, described and their supporting DBMS engine noted. The data models from each database are store stored within the Metabase System databases as described in Section 2.5.2. In addition, which of their possible five data architecture classes is also recorded.

The classes are:

- Original Data Capture
- Transaction Data Staging Area
- Subject Area Databases (also called Operational Data Stores)
- Data Warehouses (wholesale and retail)
- Reference Data
-

Figure 6 presents the overall relationship among these database architectures classes along with their most significant characteristics at the bottom of this figure.

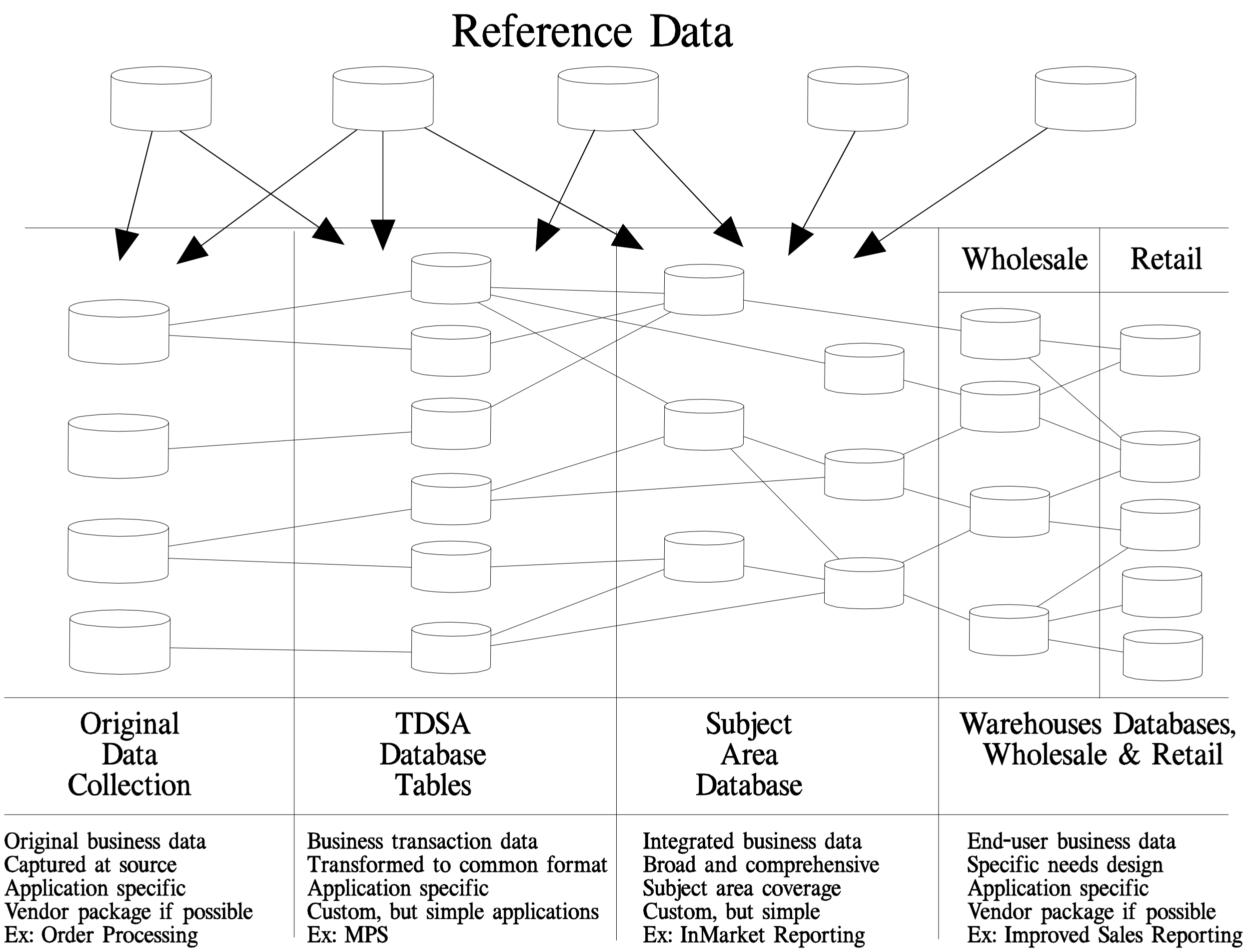

Figure 6. Database Architecture Classes and Most Critical Characteristics

The *Original Data Capture Database* architecture class represent the actual databases that reside within organizations within an enterprise and may provide data to other database classes. These database types are often called OLTP databases because they support on-line transaction processing.

The *Transaction Data Staging Area* database architecture class represents the data extracts from the original data capture databases that are then, if necessary, modified to the required semantics for any of the other database architecture classes. Any interface between any database architecture class may proceed through a transaction data staging area.

The *Subject Area Database* architecture class, also similar to Operational Data Stores, represents the subject-based integrated databases of data that, in turn, support some measure of analyses and reports, analysis results retention, and supports the generation of other classes of databases, that is, data warehouses.

The *Data Warehouse* (wholesale and retail) database architecture represents the transformed and likely redundant sets of data that serve special reports and analyses. The key set of differences between wholesale data warehouses and retail data warehouse is one of volume, duration and specialization. Data mart data warehouse designs are commonly created along the lines of "star schemas" or "snow-flake schemas," and when compared to wholesale data warehouse have smaller volumes, shorter durations, and are more specialized.

The Reference Data database architecture class represents data that form the critical characterization and discrimination characteristics of data from within the entire set of business facts. Included, for example are genders, city names, state names, all codes, and like. Ideally, all reference data would be exactly the same across all the other four data architecture classes. Realistically, however, different organizations have different reference data value versions for the same reference data and different data value versions across time.

Master Data is a form of reference data but not as specially designed tables within an enterprise reference data database but as specially designed databases that have a unique purpose and interfaces to the other database architecture class databases. Consequently, the data models of Master Data databases are treaded just as are the data models of the other databases.

3.5.2 Data Architecture Reference Model

It has long been an assertion that an enterprise data architecture and its attendant data models are a critical enterprise asset. Figure 1 from the Preface shows just how critical. The criticality is not only that data models form the basis for all policy exposition within the enterprise, they also are the ultimate “intersection” among all the IT work products.

Figure 7 illustrates and overall data architecture reference model that can be employed enterprise-wide. It can be used enterprise-wide because the individually contained data models within the architecture are independent of each other and address uniquely different dat abstractions and generalizations. This architectures facilitates integration, interoperability, and non redundancy across the models. Additionally, “higher” models are able to be used as templates for the manufacturing of “lower” models.

Once a data architecture reference model is accomplished, it serves as a framework for understanding the scope and interrelationships among clearly definable sets of component models that address data.

Accomplished, such a model enhances clear communication among deployed instances of the data architecture model's contained components.

After many years of evolution in understanding data from its natural conception through its deployment via databases, and business information systems, a data architecture reference model (see adjacent figure) consists of six distinct models. The models are:

- Data Element Model
- Concepts (Specified) Model
- Logical (Implemented) Database Model
- Physical (Operational) Database Model
- SQL View Model
- XML Schema Model

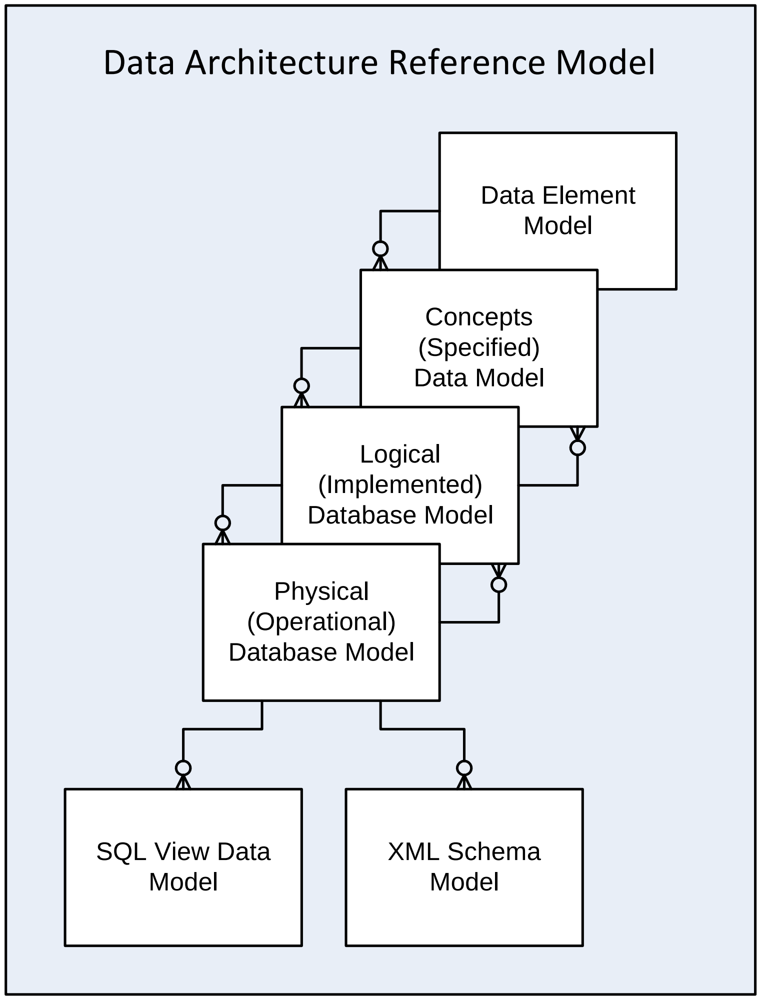

Figure 7. Data Architecture Reference Model Data Models.

Figure 8 illustrates how the Data Architecture Reference Model data models are integrated, one with the other and how their accomplishment can be distributed across the enterprise.

Please note that in this diagram, Specified Data Models are the data models of concepts, Implemented Data Models are equivalent to logical data models of databases, and Operational Data Models are equivalent to physical data models of databases.

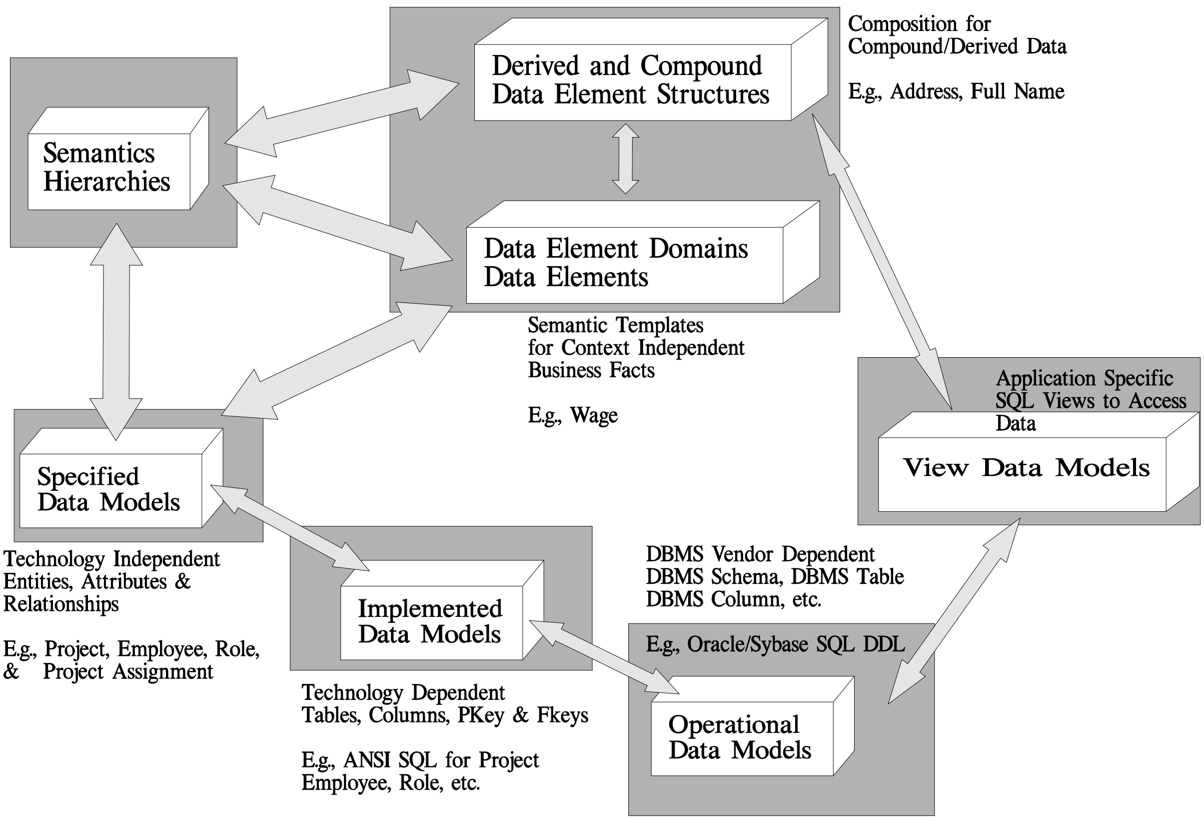

Figure 8. Interrelationship Among Data Architecture Reference Model Data Models

3.5.2.1 Data Element Model

Figure 9 illustrates a high level data element meta model. These data elements will be created primarily through the process of reverse engineering as described in Section 2.5.3.
Data elements are the enterprise facts that are employed as the semantic foundations for attributes of entities within data models of concepts (Specified Data Models), columns of tables within database models (Implemented Data Models) that support the requirements of business and are implemented through database management systems (Operational Data Models), that, in

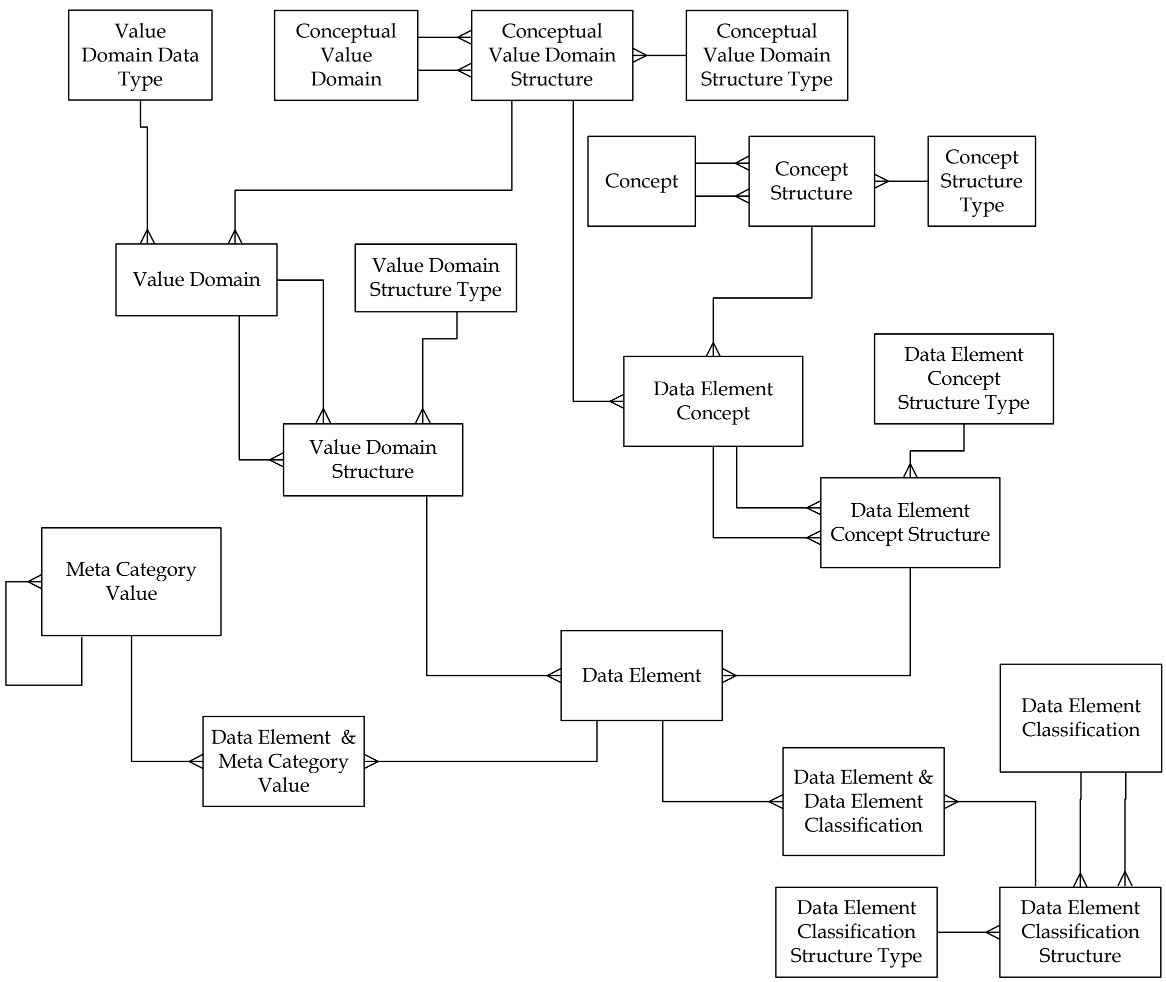

Figure 9. Data Element Metadata Model.

turn, are employed by business information systems (View Data Models) that materialize the database objects necessary for within the resources of the enterprise that support the fulfillment of enterprise missions.

Key components are Concepts, Conceptual Value Domain, Data Element Concepts, Data Elements and Value Domains. Semantic and data use modifiers can be assigned to every data element concept and data element.

3.5.2.2 Specified Data Model (Data Models of Concepts)

Figure 10 illustrates a high level Specified Data Model meta model. These specified data models will be created primarily through the process of reverse engineering as described in Section 2.5.3.

Specified Data Models are the data models of concepts. These models consist of subjects, entities, attributes, and inter-entity relationships. Relationships can interrelate entities within multiple subjects. Each data model should address only one concept such as a person's name, or an address, etc. These Specified Data Models can be templates for use in developing database models (Implemented or Operational). Every entity attribute should map to its parent data element. Semantic and data use modifiers can be assigned to every entity attribute. Key components are subjects, entities, attributes, and relationships.

A Specified Data Model is a data model of a specific concept, represented as a container such as student, school, organization, or address. These containers (e.g., student or school) must be specified before they can be implemented in one or more different database collections of tables that ultimately become operational through a DBMS such as Oracle.

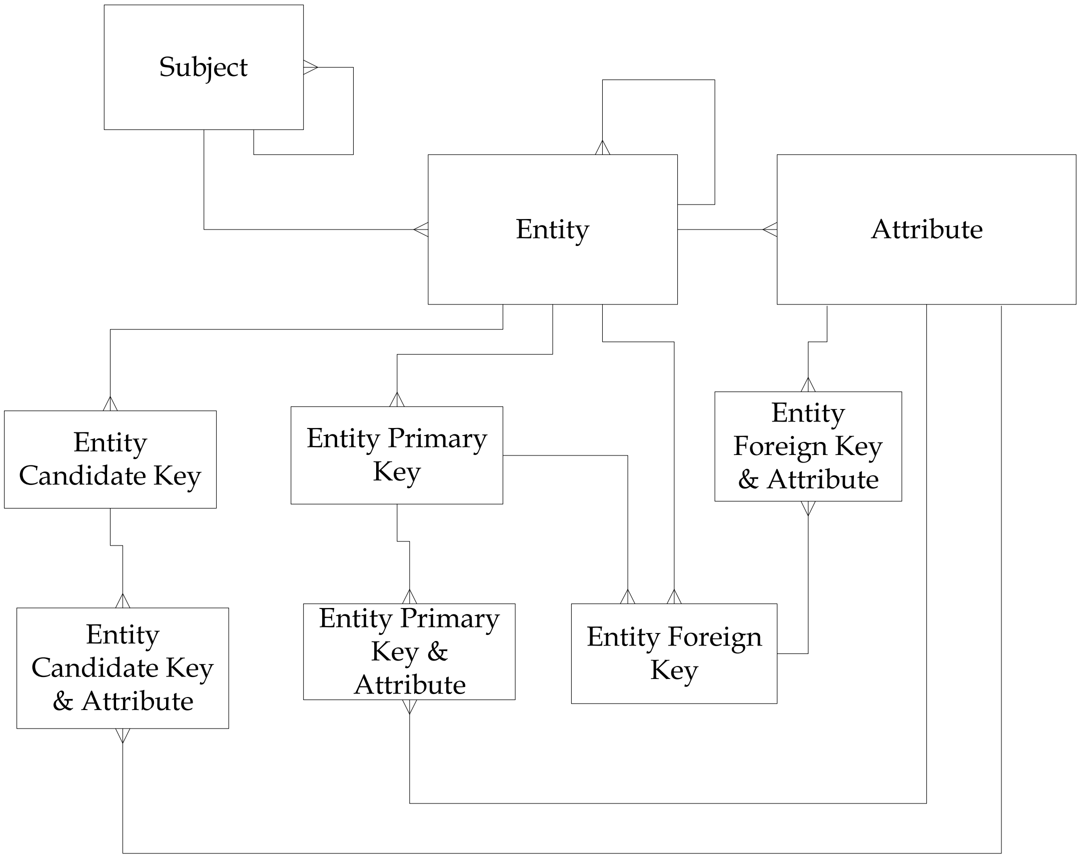

Figure 10. Specified Data Models (data models of concepts).

3.5.2.3 Implemented Data Model (Logical Database Models)

Figure11 illustrates a high level Implemented Data Model meta model. These Implemented Data Models will be created primarily through the process of reverse engineering as described in Section 2.5.3.

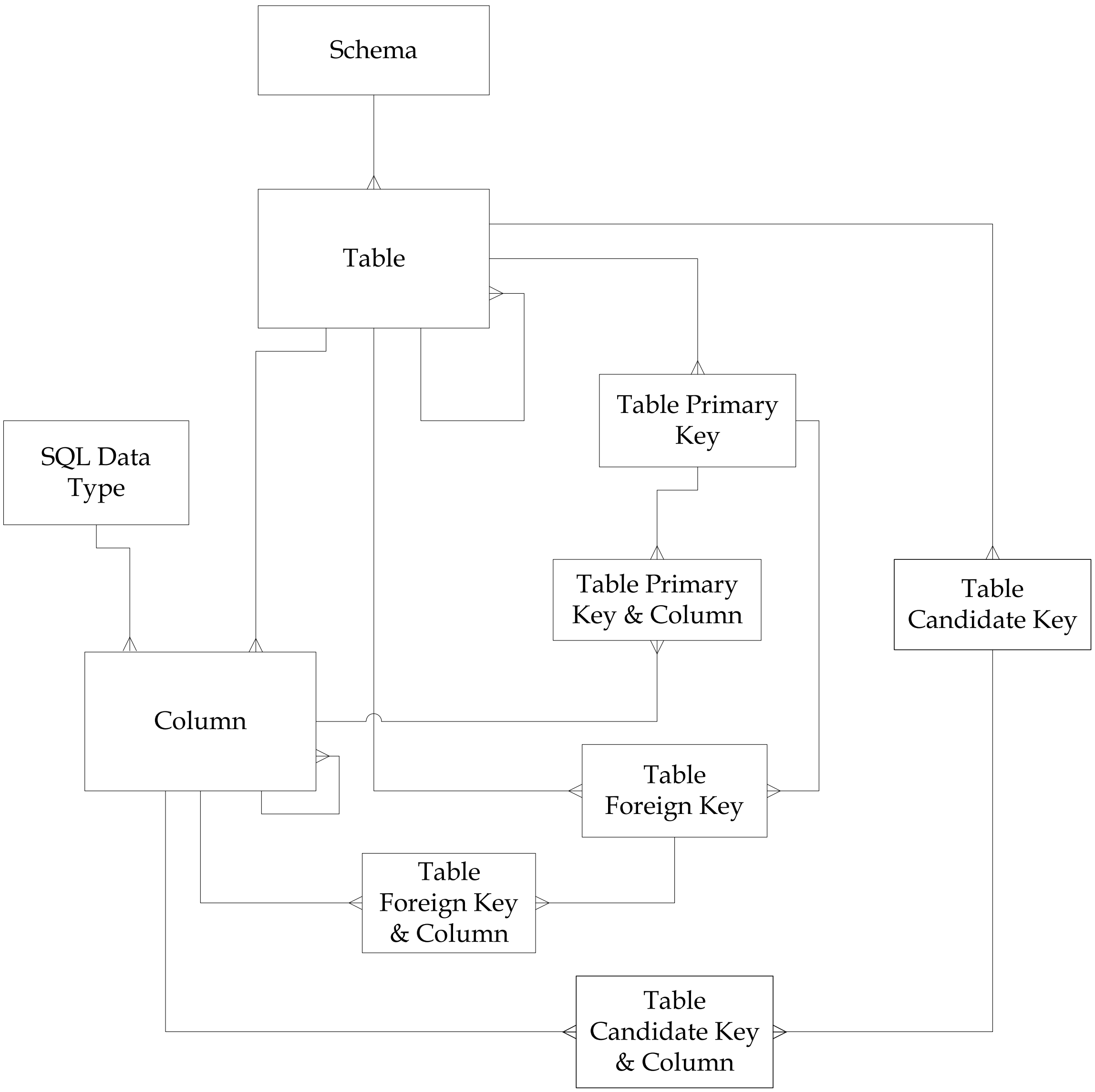

Figure 11. Implemented Data Model (Logical Database Model.)

Implemented Database Models, are the data models of databases that are independent of DBMSs. These models consist of the data structure components: schema, tables, columns, and inter-table relationships. Relationships are restricted to tables within a single schema. While each Implemented Database Model can address multiple Specified Data Models from the collection of Specified Data Models , each Implemented Database Models should address only one broad subject. Every table column should map to a parent Attribute. Semantic and data use modifiers can be assigned to every column. There is a many-to-many relationship between the Specified

Data Model and the Implemented Database Models. Key components are schemas, tables, columns, and relationships.

3.5.2.4 Operational Data Model (Physical Database Models)

Figure 12 illustrates a high level Operational Data Model meta model. These Implemented Data Models will be created primarily through the process of reverse engineering as described in Section 2.5.3.

Operational Database Models, are the data models of databases that have been bound to a specific DBMSs. These models consist of the data structure components: DBMS schema, DBMS tables, DBMS columns, and inter-table DBMS relationships. DBMS Relationships are restricted to DBMS tables within a single DBMS schema. Each Operational Database Models can address multiple Implemented Database Models. Every DBMS Column should map to a parent Column. There is a many-to-many relationship between the Implemented Database Models and the Operational Database Models. Key components are DBMS schemas, DBMS tables, DBMS columns, and DBMS Relationships.

In this state, that is, dependent upon a particular DBMS and upon the performance requirements of a particular software application, this data model is termed “physical.” These data models are the Operational Database Models that are bound to application business information systems through view data models. These data models are often not in third normal form as a way to meet needed performance requirements. DBMS Columns from the DBMS tables from within these Operational Database Models are deployments of a single column of a table from a Implemented Database Model.

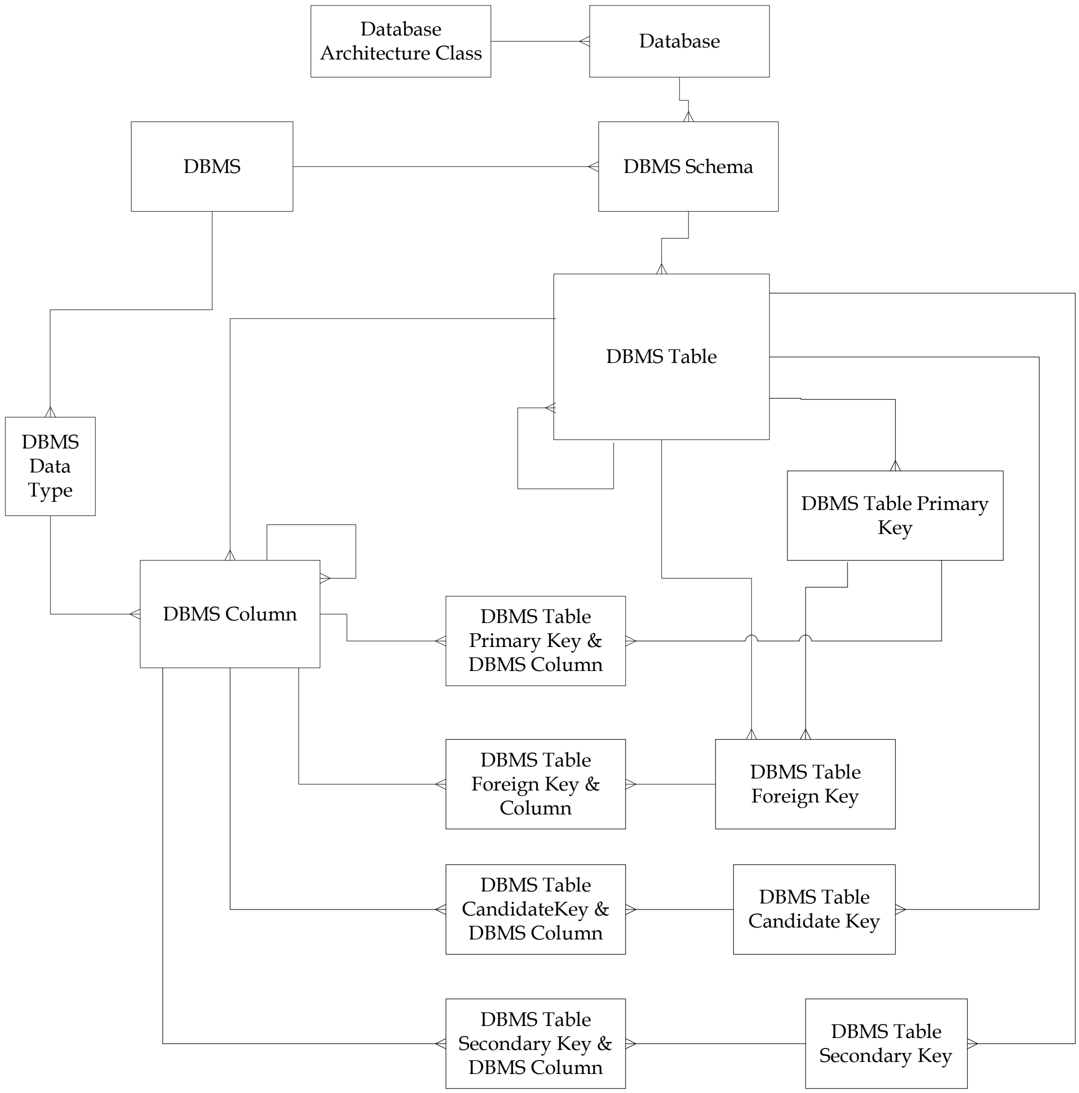

Figure 12. Operational Data Model (Physical Database Models).

3.5.2.5 View Data Model

Figure 13 illustrates a high level View Data Model meta model. These View Data Models will be created primarily through the process of reverse engineering as described in Section 2.5.3. The View data models represent the interfaces between Operational Database Models and business information systems. View and their view columns can be characterized as Input and/or Output. Additionally, these views can be mapped one to the other on a view column basis and

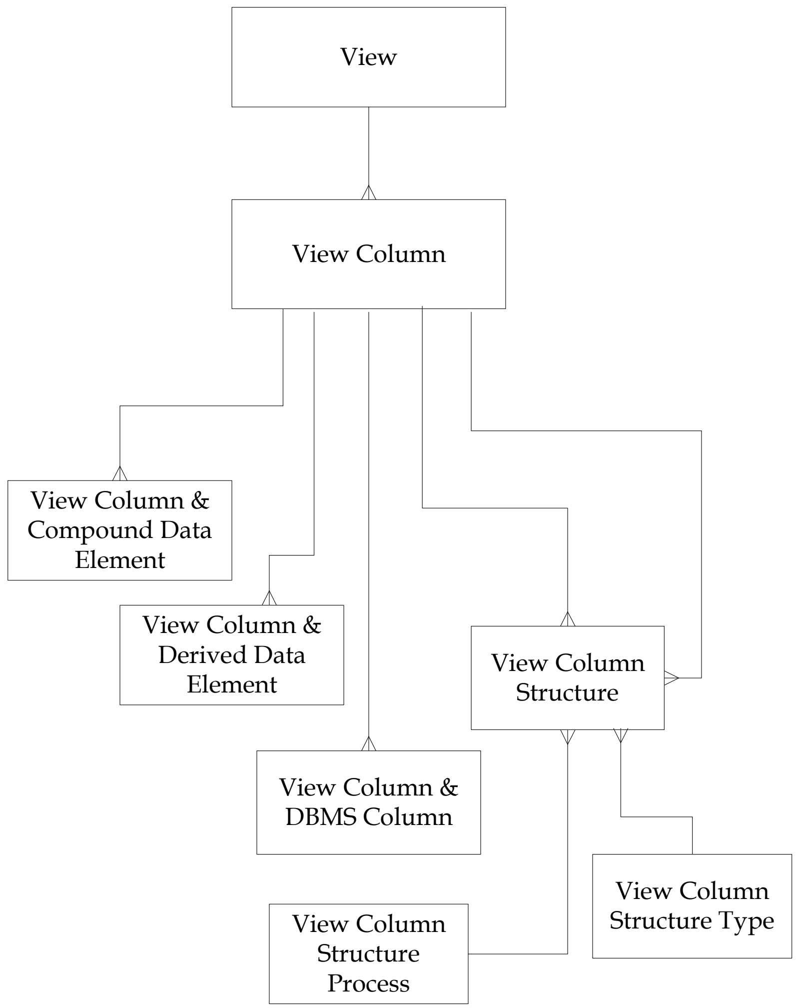

Figure 13. View Data Models.

processes can be specified to define any appropriate data value transformation. Key components are Views, View columns, and view-column interrelationships.

View data models are bound to the particular DBMS through which they are defined. View data models enable application systems to select, employ, and update databases according to their physical data models without having to include physical data model details within the application systems.

3.5.3 Reverse Engineering

Reverse Engineering, depicted in Figure 14, is essential for Enterprise Data Management assessments. That is simply because it is common due to the exigencies of business that will be many physical database schema adjustments over the years long after any logical data model has been created.

Reverse engineering exists within two different scenarios: “intersection” or “union” data models across a set of legacy systems data models.

Under the intersection scenario, each of the legacy data models is imported into the metabase. Then, one by one, their common data structures within tables are “promoted” to the Implemented Data Model layer. Once complete, the Implemented Data Model layer represents the intersection data across all those legacy models.

That intersection data model can then imported back down into the Operational Data Model layer to make a new Operational Data Model. Then, that model can be exported to SQL schema DDL to make the database design that represents the database application that is the “intersection” of the legacy databases. The legacy systems are then able to put and get data to the “intersection database” with the full confidence that the mapping semantics have been pre-engineered.

Under the union scenario, the resulting Implemented Data Model represents the design of a “Subject Area” database. It is broader than all the feeding databases. Similar to the “intersection” data model scenario, its design can be imported back into a different Operational Data Model

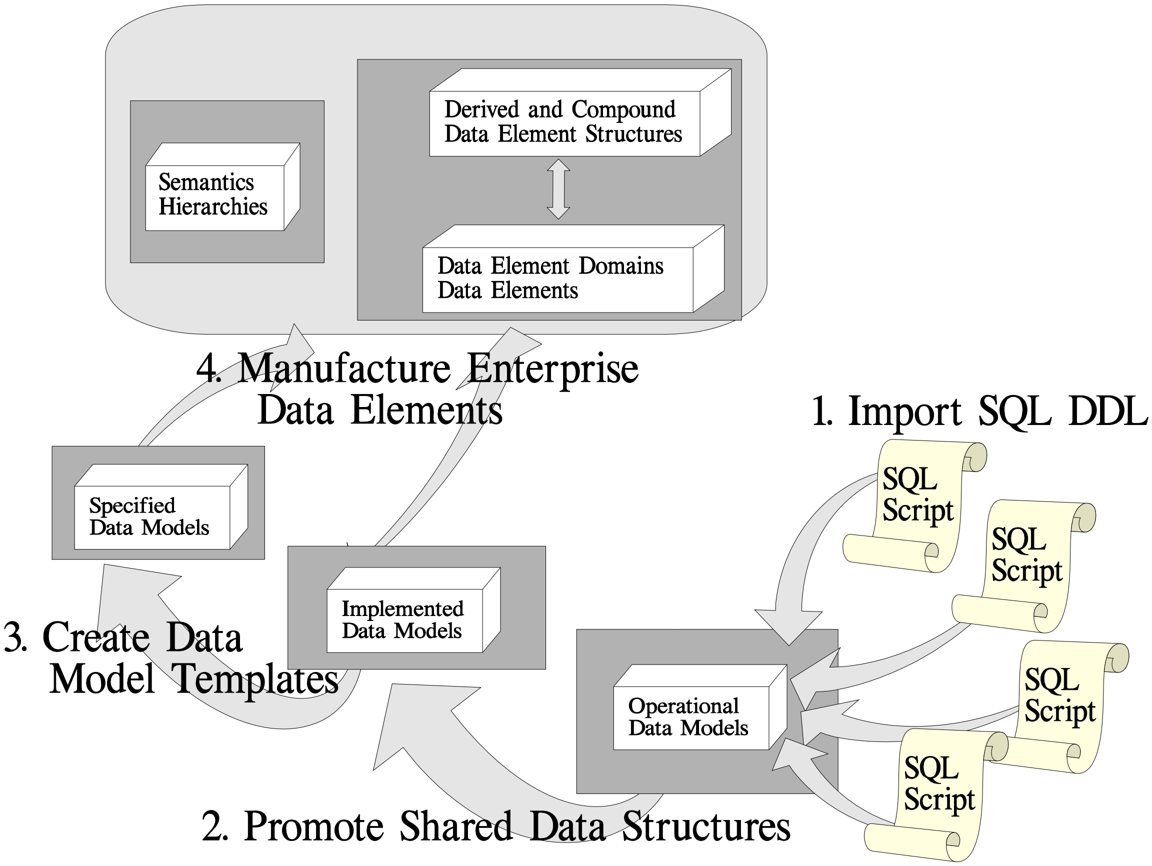

Figure 14. Reverse engineering process flow.

and then it's SQL schema DDL is used to make the database design of a "union" or subject area database application. The legacy systems are then able to put and get data to the "union database" with the full confidence that the mapping semantics have been pre-engineered.

There are three uses for the “union” scenario:

- Creating data semantics for enterprise-wide data management
- Creating “intersection data models” across a community of legacy systems that desire to exchange data
- Creating “union data models” that represent a subject area database across a community of systems that are contributing to a more expansive database

The overall process is started by importing the database design into the Operational Data Model layer by reading the database’s SQL DDL. This design is promoted to an Implemented Data Model layer, and then into the Specified Data Model layer. Within the Specified Data Model layer, the set of entities are re-engineered into different and newly created subjects. Then the ISO 11179 Data Element layer including all the upper levels of metadata are created. The reverse engineering ends here.

3.5.4 Business Information Systems

Figure 15 depict only a subset of the relationships that define the immediate content and context related to Business Information Systems. These systems are the necessary computer software systems triggered by enterprise business events instigated by functions.

Business information systems are directly related to mission, organization, function, and databases. Business information systems are interrelated to each other including their calendar and business event execution schedules. Collectively, business information systems are the mechanisms that carry out the automation aspects of enterprise policy.

A full understanding about a business information system includes the following other sets of metadata:

1. Data Integrity Rules (Business Rules)
2. Database Objects
3. Mission-Organization-Function-Position Assignments
4. Project Management
5. Reports
6. Requirements Management
7. Resource Life Cycle Nodes
8. Use Cases

Whenever any of the Section 2.0 subsections touches one of these items when the metadata of a business information system is added, links to these metadata contexts is accomplished. During any of the Section 2.0 activities, if, for example, a list of reports are generated by a business

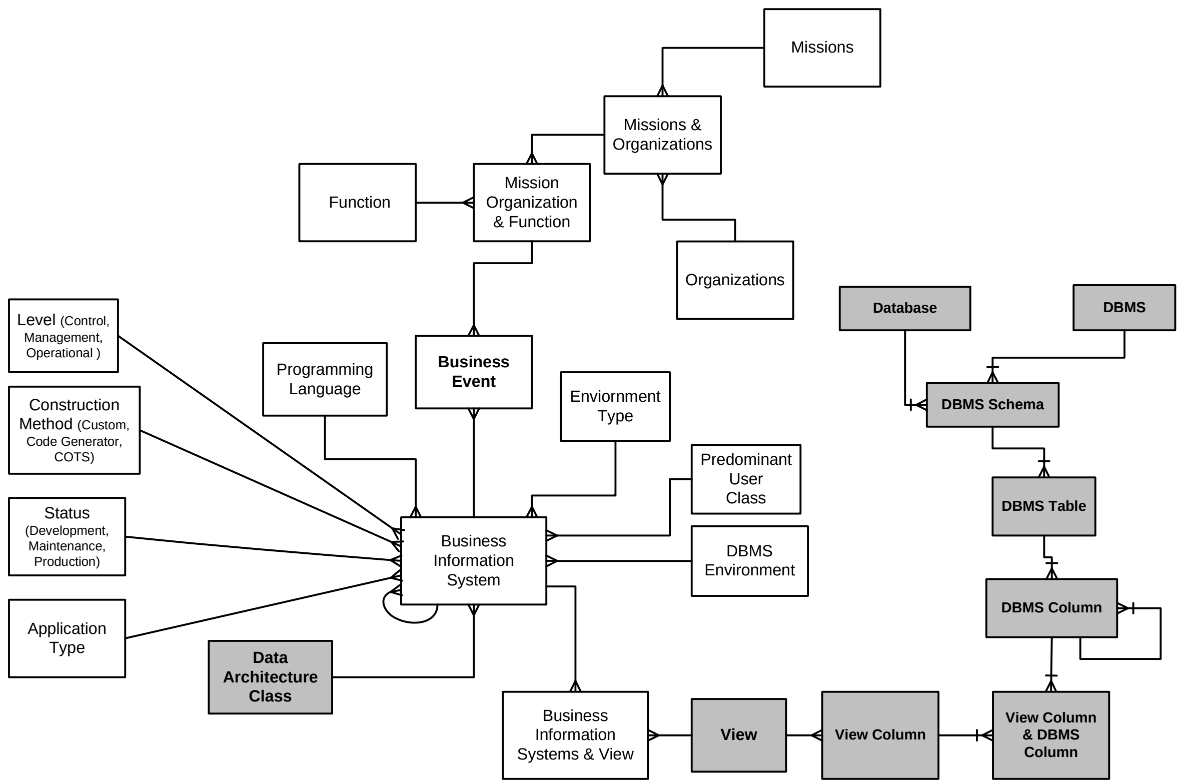

Figure 15. Business Information System Metadata Model

information system, the names and brief descriptions of the reports is entered and then related to the business information systems.

It is through this process that the overall Enterprise Data Management architecture will be filled out.

3.5.5 Database Management Systems (DBMS)

The database management systems to be examined are DB2, Oracle, and MS SQL. ISO/ANSI SQL Standards exists in editions such as 1992, 1999, 2003, 2008, and 2012. SQL:2016 is in the process of final standardization.

Each DBMS will be assessed as to which “Features” the DBMS has implemented. An example of a features is E051, Basic Query Specification, E0051-01 Select Distinct. These features have been allocated to the various SQL DBMS Standards (e.g., SQL:1998).

To determine whether an examined DBMS has implemented a given feature, the INCITS DM32.2 representative will be contacted and a request will be made for their feature implementation list.

A Metabase System DBMS subordinate table will store a cross reference between a feature, the SQL standardization year (e.g., SQL:1999) and a specific DBMS, e.g., DB2.

It is through this assessment and cross reference that compatibility across DBMSs can be established. To further validate this cross reference, specific SQL commands from within that DBMS can be entered into an Internet-based SQL validator to ensure that the specific SQL DBMS can execute the SQL command.

The result of the SQL DBMS cross reference analysis is that any given feature employed in one of the assessed SQL-based DBMS that is not implemented in another SQL-based DBMS cannot be compatibly included within or accessed by a given business information system that is intended to use multiple SQL DBMSs.

3.6 Database, Business Information System, and DBMS Cross Mapping

Figure 16 presents a high level data model that shows the intersection of DBMS and the Business Information Systems. The databases, business information systems and Database Management Systems inventoried in Section 2.5 are cross mapped.

The cross mapping among the Databases, Business Information Systems, and Database Management Systems identifies which databases support which business information systems, and also which DBMSs are employed as the engines for the databases.

For databases and business information systems that are related to each other but are managed by different DBMSs, or exist as stove-pipe implementations, the identification and description of required ETL processes to support the “appearance” of data interoperability is set out.

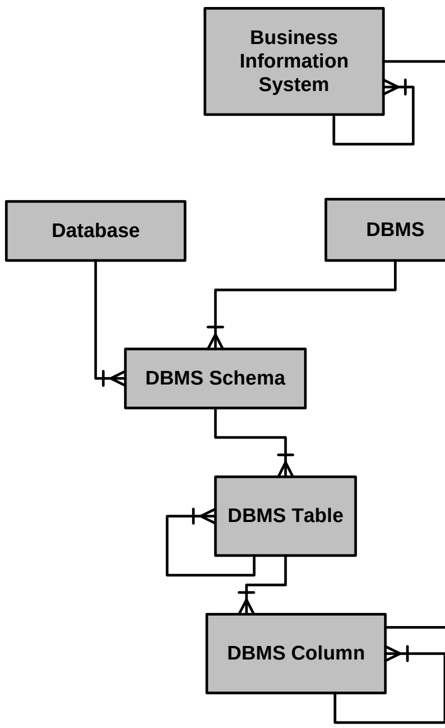

Figure 16. Database, Business Information System and DBMS Cross Reference Metadata Model

3.7 Resource Life Cycles with Business Information System and Databases Cross Mapping

Figure 17 presents the high level data model that shows the cross mapping of the Resources and their Life Cycles with Business Information Systems and Databases enable the enterprise to know specifically which databases and business information systems are needed to accomplish the Resource Life Cycle node states.

This cross mapping of Resources and their Life Cycles with Business Information Systems and Databases are a critical component in Information Systems Planning, that, in turn, form the schedules and dependancies that undergird the "just in time" development and maintenance of databases and business information systems.

An essential report from these cross references are the gaps across the resource life cycles, and by implication in the business information systems and databases.

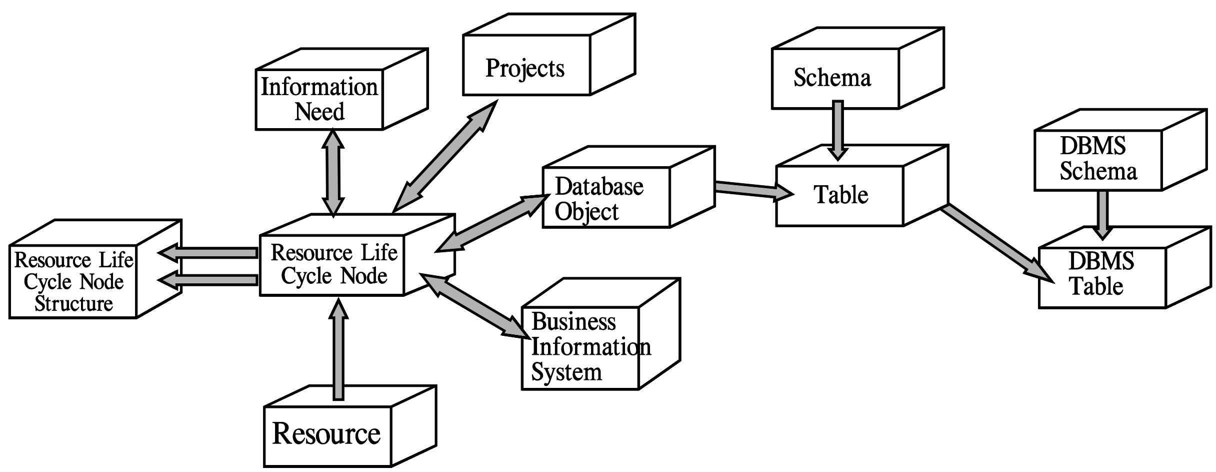

Figure 17. Resource Life Cycles, Business Information Systems, and Databases Cross Mapping.

3.8 Business Event Cycles and Calendars

Figure 18 sets out the high level data model The major Business Event Cycles and Calendars over which business information systems and databases are employed are identified and interrelated to Mission-Organization-Functions.

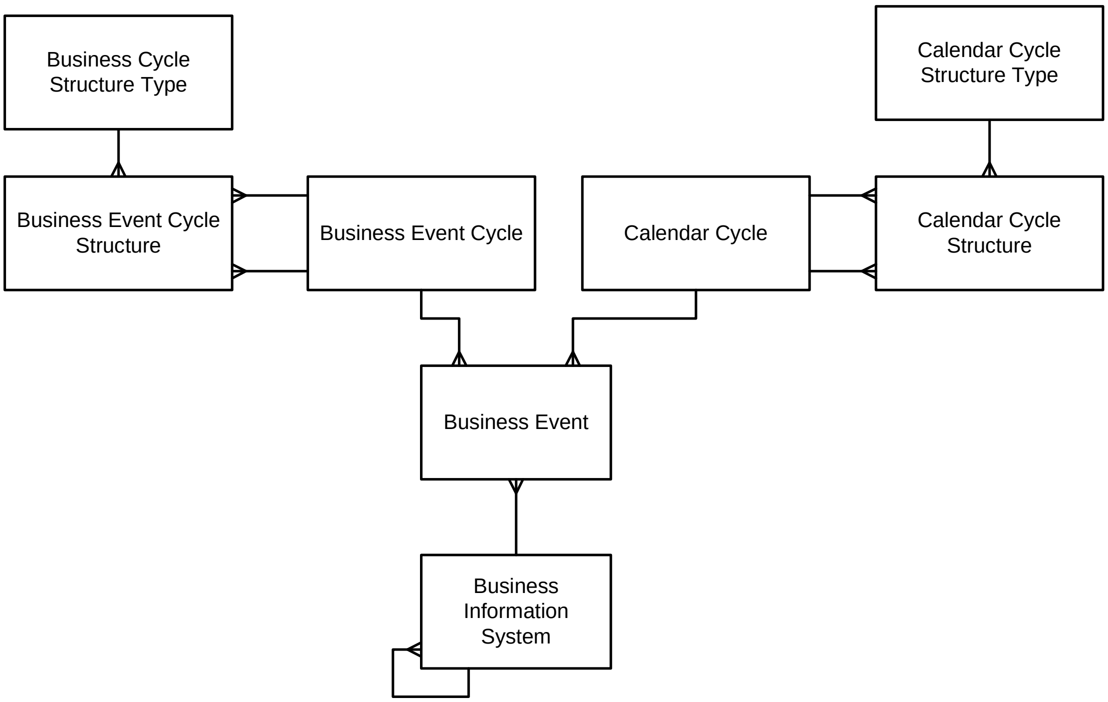

Figure 18. Business Event and Calendar Cycles.

A Business Event Cycle is a cycle during which business events, such as financial reports, holidays, business planning and the like occur.

A business event cycle may be simple or complex. If complex, the business event cycle actually consists of other business event cycles that are represented in the business event cycle structure.

Taken together, organizations accomplish their work through business functions, which, at the instigation of a business event within either a business or calendar cycle invoke a business information system that collects, updates or reports data all in support of one or more enterprise missions.

3.9 Business Event with Resource Life Cycles Cross Mapping

Figure 19 depicts the metadata model that supports the cross mapping of the Resources and their Life Cycles with Business Event Cycles and Calendars is established. These cross-mappings enable the enterprise to understand specifically when the various Resource Life Cycle nodes are to be completed. That is, within certain business events of a business event cycle, and/or on the occurrence of a particular business event during a business event calendar.

3.10 IT Projects Inventory and Descriptions

The metadata model in Figure 20 depicts the key metadata tables that supports the creation and management of project management metadata necessary for the effective management of Enterprise Data Management projects.

An inventory of major past, current and future planned IT projects including major Deliverables will be described.

The various projects address one or more collections of activities that create an IT work product, that, in turn, supports the enterprise. These are critical to meeting the schedules and expectations set out in the Resource Life Cycles.

Each project consists of deliverables, collections of tasks, assigned staff, and work environment factors that affect the accomplishment of project work.

Work environment factors are those tools and working conditions that either enhance or impede the accomplishment of work.

Understanding, interrelating, and performing meaningful comparisons of work accomplishment across the enterprise requires that projects, deliverables, and tasks be standardized as much as possible. To effect standardization, Project, Deliverable, and Task templates are employed as the basis for generating interoperable, integrated, and non-redundant projects.

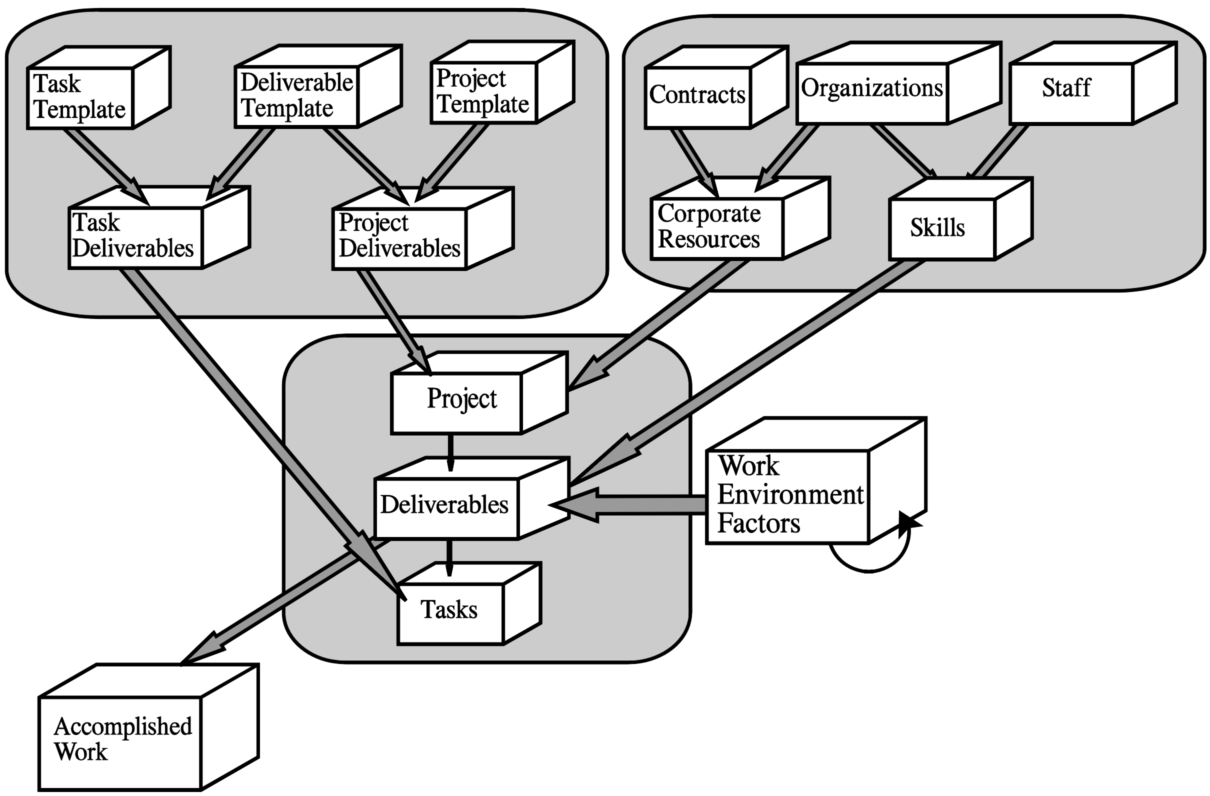

Figure 19. IT Projects Inventory and Descriptions with Resource Life Cycle Mappings.

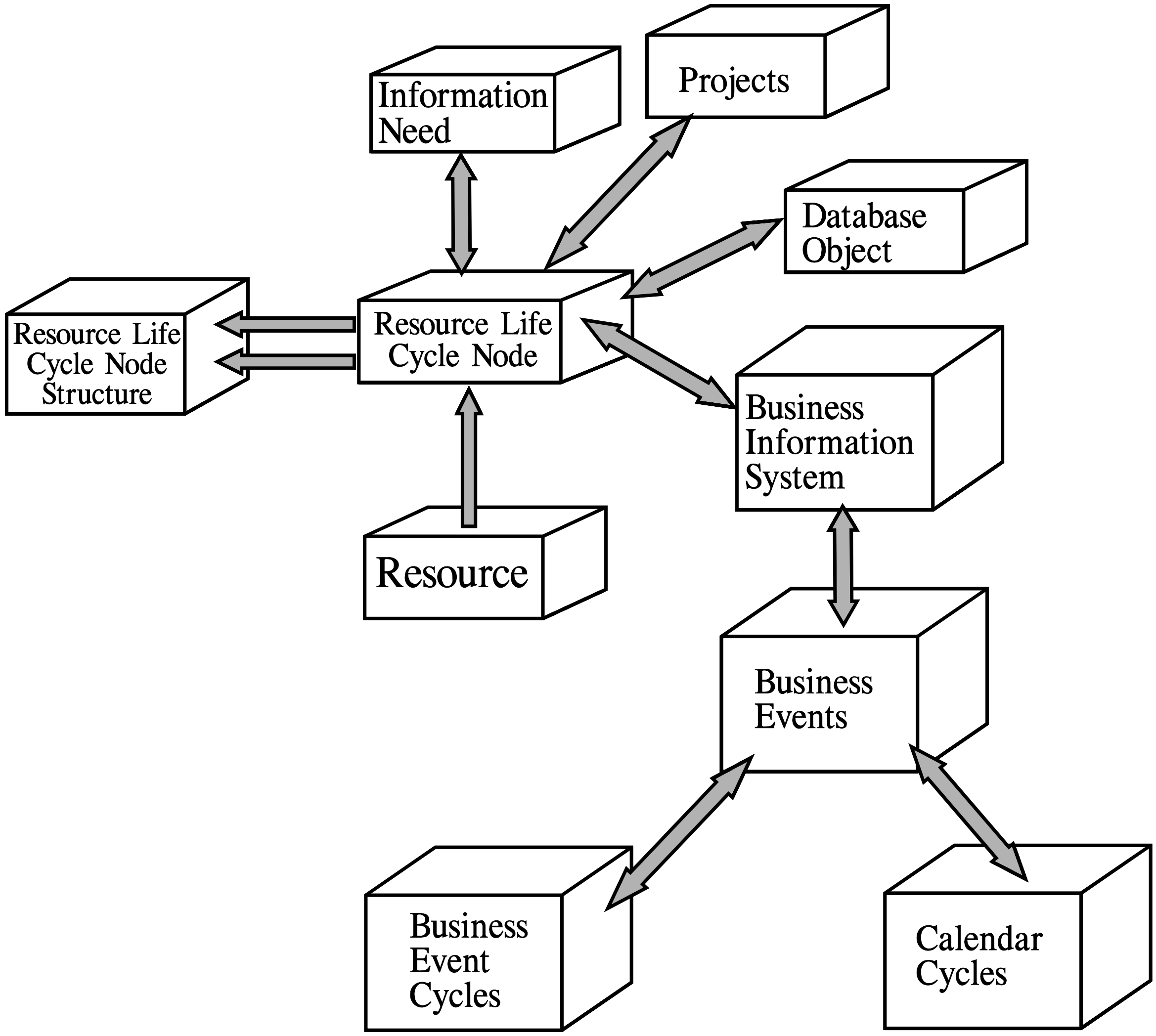

Figure 20. Business Event and Resource Life Cycle Cross Mappings.

3.11 Personnel and Subcontractors

Figure 21 presents the simple metadata data models that enables an inventory of Personnel and Subcontractors including their areas of specialties and skills will be developed. These provide a critical set of information to understand when a project can be expected to be accomplished, and with what level of completeness and quality.

These staffing identification and assessments enable a more refined development of information technology projects as their schedules will not be based primarily on having a good work break down structure but rather upon having the right persons with the right skills working on an IT project at the right time.

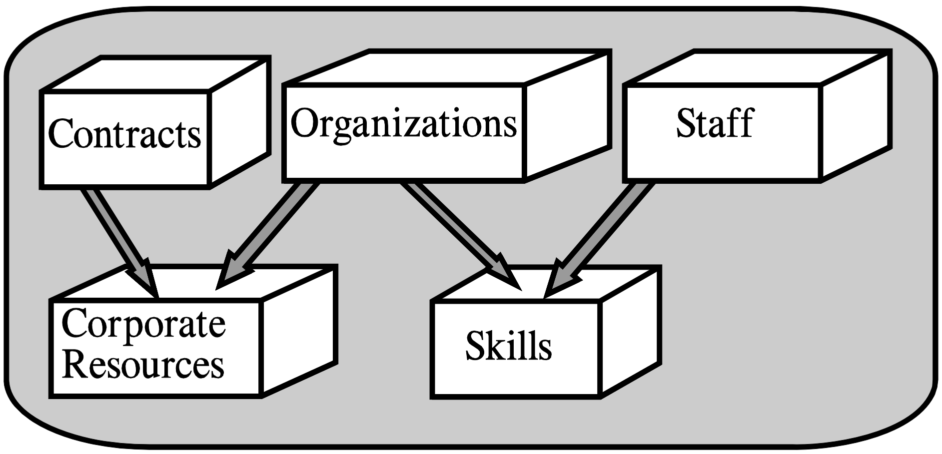

Figure 21. Personnel and Subcontractors Metadata Models

3.12 Cross mapping of IT projects with Personnel and IT contractors

Figure 20 also forms the basis for the cross mapping of IT projects with Personnel and IT contractors support two purposes. First, the generation of the same project's schedule with varying quantity and skill level quality and velocities. Second, the ability to explain why certain projects take longer or are accomplished in shorter time.

In addition to staffing project schedules, the mapping of various work environment factors (tools, reviews, availability of subject matter experts) also affect schedules. These too are to be mapped.

4.0 Conclusions

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

- An Enterprise Architecture can not only be constructed bottom-up but will ultimately be more valid.
- A bottom-up approach enables the development of both an As-Is that is founded on reality and a To-Be that is the ultimate objective.
- As-Is to To-Be Enterprise Architecture migration projects can be developed effectively and accurately because of the use of the Metabase System
- Whitemarsh's Project Management module supports automatic generation and estimation of projects that when executed interconnects Project Plans with actual Project Deliverables.
- Use of Resource Life Cycle Analysis enables the sequencing of Enterprise Architecture Migration projects that are based on just-in-time development.
- The use of this approach, the Metabase System, and the Whitemarsh Project Management module enables continuous Enterprise Architecture incremental improvement as the need arrises.

5.0 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
Enterprise Architectures	www.wiscorp.com/sp/sp08.pdf
Engineering and Managing Information Systems Plans	www.wiscorp.com/sp/sp11.pdf
Data Modeler Architecture and Concept Of Operations Reverse and Forward Engineering Guide Metabase Module User Guides	http://www.wiscorp.com/metabase_demo.html
Metabase Overview	http://www.wiscorp.com/metabase.zip
Challenge for Business Information System Success	www.wiscorp.com/sp/sp19.pdf

Data Models: The Center of the Business Information System Universe	www.wiscorp.com/sp/sp20.pdf
Metadata The Information Technology Intellectual Property Of The Enterprise	www.wiscorp.com/sp/sp21.pdf
Database Objects Introduction	www.wiscorp.com/sp/sp22.pdf