Whitemarsh
Information Systems Corporation

Integration and Interoperability of Enterprise Architecture, Knowledge Worker Framework & Metabase System to Produce ROIs

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

Figures

Tables

1 Objective

The objective of this paper is to set out the interrelationships among the following components of efficient and effective enterprises with respect to Information Technology:

- Returns on Investment
- Enterprise's Architectures
- Knowledge Worker Framework
- Data Architecture Reference Model
- Resource Life Cycle Analysis
- Database Object Classes
- Business Information Systems
- Metabase System Modules

In each section, the overall component is described along with its key subordinate components. For example, for each specific ROI, it is described, and the relevant interrelationships between the component and the other components (e.g., ROIs to Knowledge Worker Framework) are described.

The ultimate outcome of this paper is that once the work products, as set out in the 36 cells of the Knowledge Worker Framework, are completed, the ROIs are achieved.

A number of the sections have data model diagrams. Please note that when reviewing these data models, the data for the tables (rectangles) in white is created by the Metabase System module that has its name very close to the Figure's caption. For example, Figure 7 is not only the data model for Data Element specification but Data Elements is also the name of the Metabase System module for creating that data.

When there is a grey shaded table (rectangle), it means that the represented data will have already been created through another Metabase System Model. For example, Figure 5 has seventeen shaded tables. These all relate to several other functional data models which means that the data would have been created through other Metabase System modules. In this case, the other modules are:

- Mission-Organization-Function-Position
- View

The data model diagrams show the following three data model patterns:

- Single row table
- Recursive row tables
- Network tables

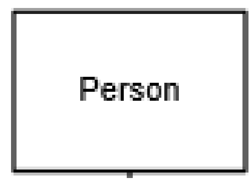
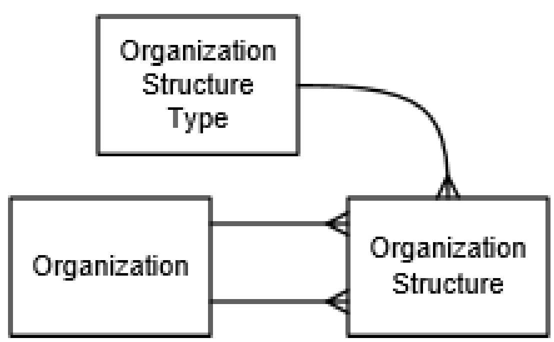
For reviewers who are savvy in data modeling notation, a simplified form of the Information Engineering "IE" notation is used. Each line with the "crows-feet" means zero, one, or more. The other end of the line means "one."

In Figure 2, the circular relationship means that a Database Object can have a hierarchically collection of organizations.

In the case of Figure 3, it means that for each Organization Structure Type rows there can be zero, one or more Organization Structure rows. Again, in the case of Figure 3 where there are two one-to-many lines, the top relationship is termed "Active," and the bottom relationship is termed "Passive."

For the "Active" relationship, an Organization may be decomposed into multiple other organizations, and for the Passive relationship, an Organization may have one "parent" organization.

Descriptions and diagram illustrations are set out in Table 1.

Data Structure	Description	Diagram
Single Row Table	A single row table is shown through a straight forward name such as Person.	 **Figure 1**. Single Row Table
Recursive Row Table	A recursive row table is one that has a looped relationship associated with that table. For example, Database Object.	 **Figure 2**. Recursive Row Table
Network Row Table	A network of tables consists of three tables with the naming convention of <name>, <name> Structure, and <name> Structure Type. An example is Organization, Organization Structure, and Organization Structure Type. This enables true network structures to be represented.	 **Figure 3**. Network Data Structure

Data Structure	Description	Diagram

Table 1. Data Structures employed in Metabase System Data Model diagrams.

All the Metabase System data models are completely described in the Metabase System's User Guides (http://www.wiscorp.com/metabase/MetabaseUserGuides.zip).

2 ROIs

The seven ROIs and links to their detailed descriptions are listed in Table 2.

ROI	Link
Enterprise-wide Project Management (16 to 1 ROI)	http://www.wiscorp.com/roi_enterprisewideprojectmanagement.html
Information Systems Planning (5.9 to 1 ROI)	http://www.wiscorp.com/roi_informationsystemsplanning.html
Data-Centered Development and Management (28 to 1 ROI)	http://www.wiscorp.com/roi_datacentereddevelopmentandmanagement.html
Data Model Manufacturing (8.6 to 1 ROI)	http://www.wiscorp.com/roi_datamodelmanufacturing.html
Business Information Systems Environments (7.7 to 1 ROI)	http://www.wiscorp.com/roi_businessinformationsystemenvironments.html
Business Information System Manufacturing (2.8 to 1 ROI)	http://www.wiscorp.com/roi_rusinessinformationsystemsmanufacturing.html
Enterprise Architecture Development (14.6 to 1 ROI)	http://www.wiscorp.com/roi_enterprisearchitecturedevelopment.html

Table 2. ROIs and Links to ROI Descriptions

Associated with this paper are a collection of "messages" that have been sent via email to members of a Whitemarsh email list. The link to the messages is:

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

2.1 Enterprise-wide Project Management (16 to 1 ROI)

Project Management "data" is just another form of metadata. When project management data is stored in the Metabase System's database, individual project plans, set within the context of the enterprise, are able to be manufactured.

The Whitemarsh approach is based on project, deliverable and task templates that enable the automatic generation of project plans. Once staff, by skill and work performance, are assigned, and once work environment factors are allocated, project plan resources are automatically generated.

Project accomplishment status can be recorded as work is accomplished. In addition, the actual created or modified work products can be directly accessed through the project management's entered data such as Deliverables or Tasks.

Whitemarsh's project management creates and manages its data in an integrated database in common with all the other IT work products. The Whitemarsh project management approach is dramatically easier, faster, and more effective than traditional approaches and products. Its ROI over the traditional Project Management approach is 16:1.

2.1.1 Relationship Between Enterprise-wide Project Management and Knowledge Worker Framework

All the cells from the Knowledge Worker Framework are related to the project management ROI as the cells are merely an enumeration of the work products determined to be appropriate for that cell.

2.1.2 Relationship Between Enterprise-wide Project Management and Enterprise's Architecture

The Enterprise's Architecture consists of two types: Over-Arching, and Operational. The projects that address these are:

- Activity Sequences and Schedules
- Activity Schedules
- Best Practices
- Business Functions
- Business Resources
- Business Organization
- Business Events
- Business Information Systems
- Daily Activity Executions and Assessments
- Daily Schedules

- Detailed Procedures
- Job Roles and Responsibilities
- Management and Positions
- Missions
- Office Policies
- Personnel Assignments

2.1.3 Relationship Between Enterprise-wide Project Management and Data Architecture Reference Model

The projects that import, construct, and/or modify the data represented by the various data models contained in the data architecture reference model are constructed through project management. The projects include:

- Database Domains
- Data Element Models
- Implemented Data Models
- Information Needs Analyses
- Missions, Organizations, Functions, and Positions
- Operational Data Models
- Requirements Management
- Specified Data Models
- Views

2.1.4 Relationship Between Enterprise-wide Project Management and Resource Life Cycle Analysis

The various projects that import, construct, and/or modify the various resource life cycle models are constructed through project management. The projects include:

- Missions, Organizations, Functions, and Positions
- Resource Life Cycle Analysis

2.1.5 Relationships Between Enterprise-wide Project Management and Database Object Classes

The various projects that import, construct, and/or modify the various database object classes are constructed through project management. The projects include:

- Database Objects
- Database Domain
- Data Element Models
- Implemented Data Models
- Information Needs Analyses
- Missions, Organizations, Functions, and Positions
- Specified Data Models

2.1.6 Relationships Between Enterprise-wide Project Management and Business Information Systems

The various projects that import, construct, and/or modify the various business informaiton system specifications are constructed through project management. The projects include:

- Business Information Systems
- Database Objects
- Database Domain
- Data Element Models
- Documents and Forms
- Implemented Data Models
- Information Needs Analyses
- Missions, Organizations, Functions, and Positions
- Operational Data Models
- Reports Management
- Requirements Management
- Specified Data Models
- Use Cases
- Views

2.1.7 Relationships Between Enterprise-wide Project Management and Metabase System Modules

The Metabase System's database, which as of February 2016 has 279 tables, serves as the database for storing the work products identified and managed by the already defined deliverable templates of Whitemarsh Project Management. The Metabase System Modules involved are:

- Business Information Systems
- Data Elements
- Database Objects
- Documents and Forms

- Implemented Data Model
- Information Needs Analysis
- Mission Organization Function Position
- Operational Data Model
- Project Management
- Reports Management
- Requirements Management
- Resource Life Cycle Analysis
- Specified Data Model
- Use Cases
- View Data Model

2.2 Information Systems Planning (5.9 to 1 ROI)

Information System Plans (ISP) are essential to a well-ordered multi-year IT strategy for the enterprise. That said, the traditional strategies for developing ISPs are a profound disappointment.

Traditional techniques and strategy developing and evolving enterprise-wide Information Systems Plans take 16 to 25 staff years. Additionally, the traditionally developed ISPs do not exhibit the critical characteristics of timeliness, usability, maintainability, quality, and reproducibility.

Through Whitemarsh strategies and techniques, the development of Information Systems Plans can be efficiently and effectively created that not only exhibit the characteristics cited above, but also can be created almost six times faster that through the traditional approach. The Whitemarsh ROI for Information Systems Plans is 5.9.

2.2.1 Relationship Between Information Systems Planning and Enterprise-wide Project Management

The projects involved in enterprise wide project management for this ROI include:

- Business Information Systems
- Database Objects
- Implemented Data Models
- Information Needs Analyses
- Missions, Organizations, Functions, and Positions
- Operational Data Models
- Reports Management
- Requirements Management
- Use Cases
- Views

2.2.2 Relationship Between Information Systems Planning and Knowledge Worker Framework

Information systems planning is constructed through the work products mainly contained in the Business Information System columns, and the cells containing Resource Life Cycle Analysis operational data models, view data models and the database object classes.

2.2.3 Relationships Between Information Systems Planning and Data Architecture Reference Model

Information systems planning is best completed in concert with Operational Data Models, and View Data Models and the Database Object Classes.

2.2.4 Relationships Between Information Systems Planning and Resource Life Cycle Analysis

Business Information Systems are the main source for achieving the states of the various resource life cycle nodes.

2.2.5 Relationships Between Information Systems Planning and Database Object Classes

Business information systems should be directly related to the requirements of accomplishing the database object states through the execution of database object information systems.

2.2.6 Relationships Between Information Systems Planning and Metabase System Modules

The Metabase System Modules related to the specification of Business Information System are:

- Business Information Systems
- Documents and Forms
- Mission Organization Function Position
- Operational Data Model
- Project Management

- Reports Management
- Requirements Management
- Resource Life Cycle Analysis
- Use Cases
- View Data Model

2.3 Data Centered Development and Management (28 to 1 ROI)

Traditionally developed business information systems developed through process-driven techniques cost significantly more than those created through data-driven techniques. The conservative estimate is that process-driven business information systems take 4.6 times longer to develop because the quantity of work products is 4.6 times larger.

Increased costs are not the sole problem. Business Information System bloat greatly increases maintenance time and cost, reduces efficiency, and because of the 4.6 times greater work products factor, the ability to easily and effectively integrate, interoperate, and have consistent semantics is compromised.

The Whitemarsh data-driven techniques not only greatly reduce Business Information System bloat and maintenance, but also increase integration, interoperability and consistency. In addition to all these benefits, the Whitemarsh approach has an ROI of 28:1. This is a 3x silver bullet.

2.3.1 Relationship Between Data Centered Development and Management and Enterprise-wide Project Management

The projects involved in enterprise wide project management for this ROI include:

- Business Information Systems
- Database Objects
- Database Domains
- Data Element Models
- Implemented Data Models
- Information Needs Analyses
- Missions, Organizations, Functions, and Positions
- Operational Data Models
- Requirements Management
- Specified Data Models
- Views

2.3.2 Relationship Between Data Centered Development and Management and Knowledge Worker Framework

Data Centered Development and Management is mainly accomplished through the development of the work products contained in the Database Object and the Business Information System columns of the Knowledge Worker Framework.

2.3.3 Relationship Between Data Centered Development and Management and Data Architecture Reference Model

The data models involved include:

- Database Objects
- Database Domains
- Data Element Models
- Implemented Data Models
- Specified Data Models
- Views

2.3.4 Relationship Between Data Centered Development and Management and Database Object Classes

Database objects represent the specification of the data-centered actions for transforming a database object from one state to the next as needed within any given business information system.

2.3.5 Relationship Between Data Centered Development and Management and Metabase System Modules

The Metabase System Modules related to the specification of Business Information System are:

- Business Information Systems
- Data Elements
- Database Objects
- Implemented Data Model
- Information Needs Analysis
- Mission Organization Function Position
- Operational Data Model
- Project Management

- Requirements Management
- Specified Data Model
- Use Cases
- View Data Model

2.4 Data Model Manufacturing (8.6 to 1 ROI)

Enterprise data models are not only at the very center of all Enterprise Business Information System environments but also enable the interrelationship among all the IT work products. Data Models are key in enabling the scope of business information systems to be restricted to just what is necessary.

Despite all the worth of Enterprise Data Models, their cost through traditional techniques has not only been excessive, they have been cited as a reason for over budget and late business information systems.

Whitemarsh's Enterprise Data Model development techniques have resolved these common complaints and also have dramatically increased the velocity through which data models are created. These techniques have produced an ROI of 8.6.

2.4.1 Relationship Between Data Model Manufacturing and Enterprise-wide Project Management

The projects involved in enterprise wide project management for this ROI include:

- Database Objects
- Database Domains
- Data Element Models
- Implemented Data Models
- Information Needs Analyses
- Missions, Organizations, Functions, and Positions
- Operational Data Models
- Requirements Management
- Specified Data Models
- Views

2.4.2 Relationship Between Data Model Manufacturing and Knowledge Worker Framework

Data Model Manufacturing is mainly accomplished through the development of the work products contained in the Database Object column of the Knowledge Worker Framework.

2.4.3 Relationships Between Data Model Manufacturing and Data Architecture Reference Model

The data models involved include:

- Data Element Models
- Implemented Data Models
- Operational Data Models
- Specified Data Models
- Views

2.4.4 Relationships Between Data Model Manufacturing and Metabase System Modules

The Metabase System Modules related to the specification of Business Information System are:

- Data Elements
- Implemented Data Model
- Information Needs Analysis
- Mission Organization Function Position
- Operational Data Model
- Requirements Management
- Specified Data Model
- View Data Model

2.5 Business Information System Environments (7.7 to 1 ROI)

While data models are essential for business information system development, business information systems are where "the rubber meets the road." It is through business information systems that enterprise-policies are executed and the data resulting from those policy executions is captured, stored, analyzed and reported.

Like enterprise data models, the traditional processes for business information system creation are both expensive and fraught with error. Whitemarsh, over the years has participated in IT projects where business information system development techniques have been dramatically improved and made cheaper. The ROI derived from all these improvements is 7.7.

2.5.1 Relationship Between Business Systems Environments and Enterprise-wide Project Management

The projects involved in enterprise wide project management for this ROI include:

- Business Information Systems
- Database Objects
- Information Needs Analyses
- Missions, Organizations, Functions, and Positions
- Operational Data Models
- Reports Management
- Requirements Management
- Resource Life Cycle Analysis
- Use Cases
- Views

2.5.2 Relationship Between Business Systems Environments and Knowledge Worker Framework

Business Information System Environments are mainly accomplished through the development of the work products contained in the Business Information System column of the Knowledge Worker Framework.

2.5.3 Relationships Between Business Systems Environments and Data Architecture Reference Model

The data models involved include:

- Implemented Data Models
- Operational Data Models
- Views

2.5.4 Relationships Between Business Systems Environments and Metabase System Modules

The Metabase System Modules related to the specification of Business Information System are:

- Business Information Systems
- Database Objects

- Documents and Forms
- Implemented Data Model
- Information Needs Analysis
- Mission Organization Function Position
- Operational Data Model
- Project Management
- Reports Management
- Requirements Management
- Resource Life Cycle Analysis
- Specified Data Model
- Use Cases
- View Data Model

2.6 Business Information System Manufacturing (2.9 to 1 ROI)

From the very beginning of IT, there has been a dramatic year-over-year decrease in the cost of computer hardware even in the face of significant increases in computer hardware capacity and performance. Sadly, the same cannot be said about business information systems. Over the years, even in the face of increased computer software sophistication, the cost of business information system software has continuously increased.

Starting however, in the middle 1980s, there has been a steady growth in the ability to actually manufacture business information systems. Whitemarsh has been a continuous supporter and user of business information system generators. For example, the Whitemarsh Metabase System has 279 database tables. Under traditional business information system development techniques, the Metabase System should have cost $4.5 million. In fact, it has cost greater than four times less than that amount.

In another example, an association membership management system for an international standards organization had a database of about 90 database tables. The traditional cost for this system should have been $1.5 million. The actual invoices were for less than $300K.

The key strategy for a very dramatic cost savings is to have an environment within which business information systems are manufactured rather than created through highly-skilled, hand-crafting artisans.

Under this approach, well in excess of 90% of every business information system function can be generated through business information system generators.

In addition to generating correct-the-first time business information systems, this changed approach produces an ROI of 2.8.

2.6.1 Relationship Between Information Systems Manufacturing and Enterprise-wide Project Management

The projects involved in enterprise wide project management for this ROI include:

- Business Information Systems
- Documents and Forms
- Information Needs Analyses
- Operational Data Models
- Reports Management
- Requirements Management
- Resource Life Cycle Analysis
- Use Cases
- Views

2.6.2 Relationship Between Information Systems Manufacturing and Knowledge Worker Framework

Business Information System Manufacturing is done through the development of the work products contained in the Business Information System column of the Knowledge Worker Framework.

2.6.3 Relationships Between Information Systems Manufacturing and Data Architecture Reference Model

The data models involved include:

- Implemented Data Models
- Operational Data Models
- Views

2.6.4 Relationships Between Information Systems Manufacturing and Metabase System Modules

The Metabase System Modules related to the specification of Business Information System are:

- Business Information Systems
- Database Objects
- Documents and Forms
- Implemented Data Model
- Information Needs Analysis
- Mission Organization Function Position
- Operational Data Model
- Project Management

- Reports Management
- Requirements Management
- Resource Life Cycle Analysis
- Specified Data Model
- Use Cases
- View Data Model

2.7 Enterprise Architecture Development (14.6 to 1 ROI)

It is very important for an enterprise to have an overarching enterprise architecture. Creating such an architecture through traditional techniques has been very expensive, time consuming, and almost always not exhibiting the following information systems plan characteristics of: timeliness, usability, maintainability, quality, and reproducibility.

To achieve these characteristics, an enterprise architecture cannot just be a large document containing a myriad of diagrams. Rather, an enterprise architecture must be an overarching layer to the overall collection of IT work products that form the implementing mechanisms for the enterprise architecture.

Because of this construction strategy, that is, bottom up and through complete integration with other IT work products, the ROI for this approach is the average of the four directly related ROI's. That is, 14.6

2.7.1 Relationship Between Enterprise Architecture Development and Enterprise-wide Project Management

The projects involved in enterprise wide project management for this ROI include:

- Business Information Systems
- Database Objects
- Database Domains
- Data Element Models
- Documents and Forms
- Implemented Data Models
- Information Needs Analyses
- Missions, Organizations, Functions, and Positions
- Operational Data Models
- Reports Management
- Requirements Management
- Resource Life Cycle Analysis
- Specified Data Models
- Use Cases
- Views

2.7.2 Relationship Between Enterprise Architecture Development and Knowledge Worker Framework

Enterprise Architecture development is accomplished through the development of the work products contained all the columns of the Knowledge Worker Framework.

2.7.3 Relationships Between Enterprise Architecture Development and Enterprise's Architecture

The Enterprise's Architecture is essential to the overall Enterprise Architecture Development as it is one of the architectures that is developed.

2.7.4 Relationships Between Enterprise Architecture Development and Data Architecture Reference Model

The data models involved include:

- Data Element Models
- Implemented Data Models
- Operational Data Models
- Specified Data Models
- Views

2.7.5 Relationships Between Enterprise Architecture Development and Resource Life Cycle Analysis

Resource Life Cycle Analysis is essential to Enterprise Architecture Development as it is one of the architectures that is developed.

2.7.6 Relationships Between Enterprise Architecture Development and Database Object Classes

Database Object Classes are essential to Enterprise Architecture Development as it is one of the architectures that is developed.

2.7.7 Relationships Between Enterprise Architecture Development and Business Information Systems

Business Information Systems are essential to Enterprise Architecture Development as it is one of the architectures that is developed.

2.7.8 Relationships Between Enterprise Architecture Development and Metabase System Modules

The Metabase System Modules related to the specification of Enterprise Architectures are:

- Business Information Systems
- Data Elements
- Database Objects
- Documents and Forms
- Implemented Data Model
- Information Needs Analysis
- Mission Organization Function Position
- Operational Data Model
- Project Management
- Reports Management
- Requirements Management
- Resource Life Cycle Analysis
- Specified Data Model
- Use Cases
- View Data Model

3 Enterprise's Architecture (www.wiscorp.com/sp/sp08.pdf)

The Enterprise's Architecture is the specification and operational architecture for the enterprise itself, the engineering and structure of the enterprise's mission, organizations, functions and database domains are provided such that they can be extended and/or integrated with other more technical architectures such as hardware, information systems, and business events.

3.1 Enterprise's Architecture Components and Descriptions

The Enterprise's Architecture consists of two types: Overarching, and Operational. The work products that address the Overarching Enterprise's architecture are set out in Table 3. The work products for the Operational Enterprise's Architecture are set out in Table 4.

Overarching Enterprise's Architecture	
Knowledge Worker Column (Scope and Business rows)	**Work Product**
Mission	Business Missions and Mission hierarchies
Database Object	Major business resources, Database domains and resource life cycles
Business Information System	Business information Systems, Information sequencing, hierarchies and use cases
Business Event	Interface events and event sequencing and hierarchies
Business Function	Major business Scenarios and business scenario sequencing and hierarchies
Business Organization	Organizations and organization charts, jobs and descriptions

Table 3. Overarching Enterprise's Architecture Work Products

Operational Enterprise's Architecture		
Knowledge Worker Framework		
Column	Row	**Work Product**
Business Function	System	Best practices, quality measures, and accomplishment assessments
Business Function	Technology	Major business resources, Database domains and resource life cycles
	Deployment	Office policies and procedures to accomplish activities
	Operations	Detailed procedure-based instructions
Business Organization	System	Job roles, responsibilities, and activity schedules
	Technology	Procedure manuals, task lists, quality measurers, and assessments
	Deployment	Daily schedules, shift and personnel assignments
	Operations	Daily activity executions and assessments

Table 4. Operational Enterprise's Architecture Work Products

The data models for the Overarching work products are depicted in Figure 4. Its data is collected and interrelated in the Metabase System Mission-Organization-Function-Position (MOFPA) module.

The data models for the Operational Architecture work products are depicted in Figure 5. Its data is collected and interrelated in the Metabase System Documents and Forms module.

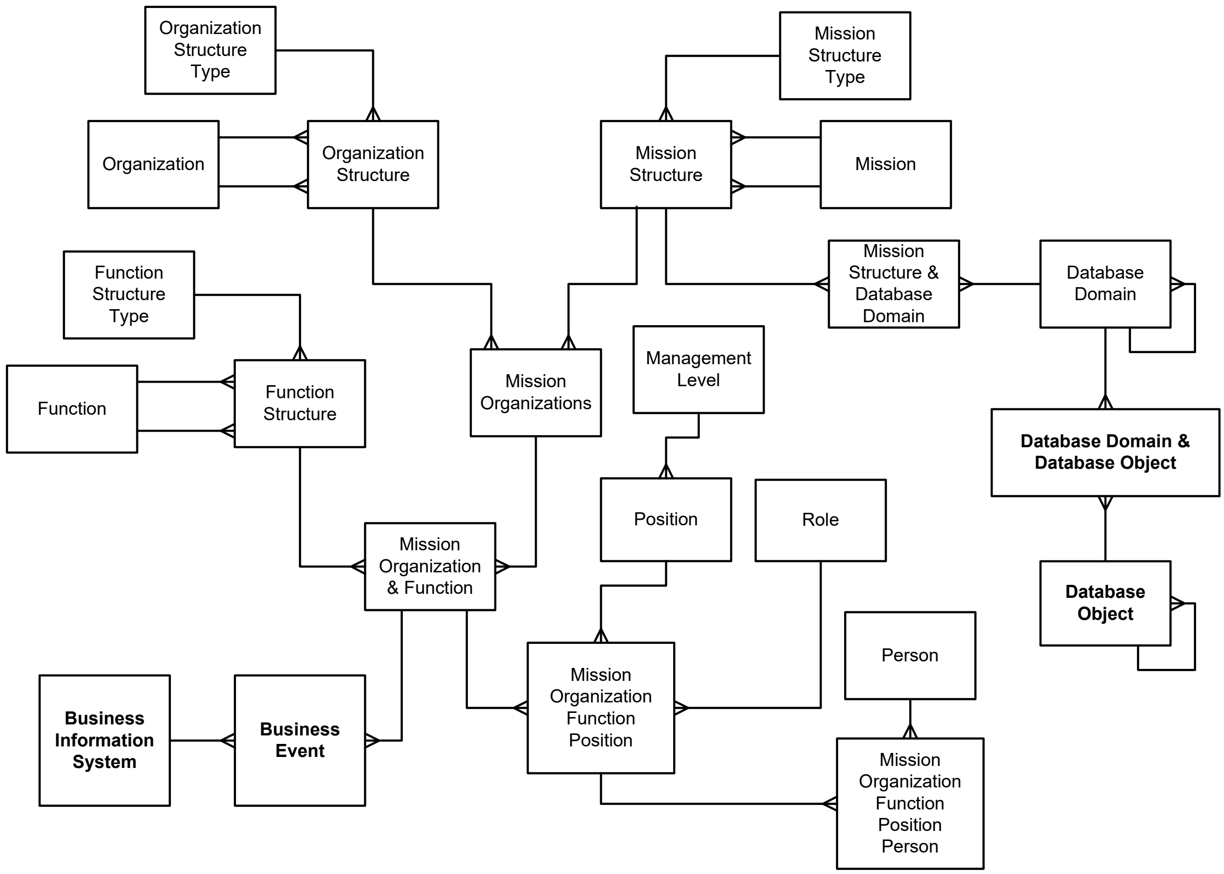


Figure 4. Mission-Organization-Function-Position Data Model.

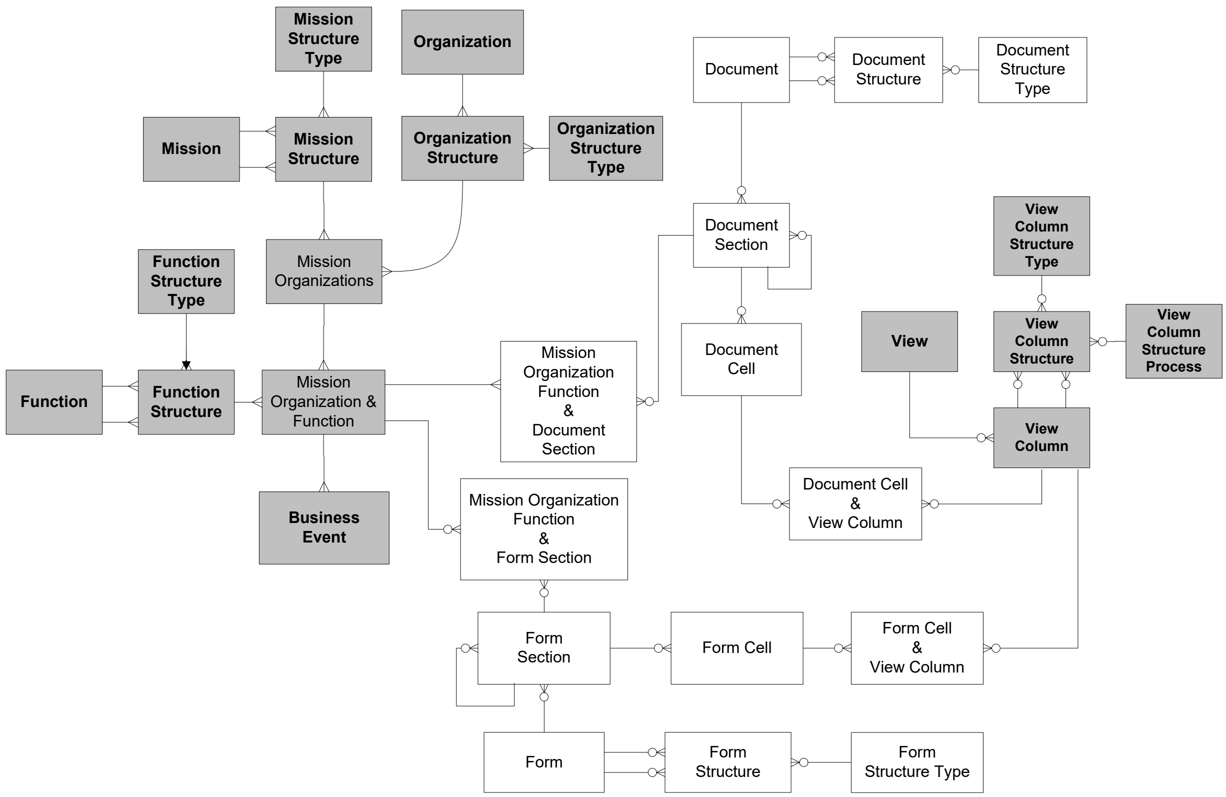


Figure 5. Documents and Forms Data Model.

Relationship Between Enterprise's Architecture and ROIs

The relationships between the various columns of the Enterprise's Architecture and the ROIs is contained in Section 2 and its subsections. Specifically:

- Section 2.1, Enterprise-wide Project Management (16 to 1 ROI)
- Section 2.7, Enterprise Architecture Development (14.6 to 1 ROI)

3.2 Relationship Between Enterprise's Architecture and Knowledge Worker Framework

The relationships between the Enterprise's Architecture and the Knowledge Worker Framework is contained in Section 3.1 and Tables 3 and 4.

3.3 Relationships Between Enterprise's Architecture and Data Architecture Reference Model

The work product developed during the creation of the Enterprise's architecture that lead to the direct creation of the various data architecture models is: Database Domains, which are based on the Mission Models.

3.4 Relationships Between Enterprise's Architecture and Resource Life Cycle Analysis

The work product developed during the creation of the Enterprise's architecture that lead to the direct creation of resource Life Cycle Analysis is: Major Business Resources.

3.5 Relationships Between Enterprise's Architecture and Metabase System Modules

The Metabase System modules involved development of the Enterprise's Architecture are:

- Business Information Systems
- Documents and Forms
- Mission Organization Function Position
- Project Management
- Requirements Management
- Resource Life Cycle Analysis

4 Knowledge Worker Framework

(www.wiscorp.com/KnowledgeWorkerFramework.pdf)

A knowledge worker is someone who primarily works with information and abstract concepts. Another type of worker is the real product worker. White collar workers such as clinicians and clinical support personnel are knowledge workers because they develop care plans, provide treatments, and record results. Administrative staff is also a class of knowledge workers that includes executives, administrators, data processing/information systems personnel, and most other office workers. Alternatively, workers on a manufacturing line are not knowledge workers because they are primarily focused on the creation and/or assembly of real products.

Both knowledge workers and real product workers share common characteristics including plans, schedules, estimates and result assessments. Notwithstanding, the fundamental work methods and environment that underlies the knowledge worker and the real product worker are different at the core. Thus, trying to make one a clone of the other is both frustrating and invalid.

Due to the abstract nature of their work, information required by knowledge workers can best be stored, assimilated and used as objects, which are encapsulations of data, processes and business rules. To most effectively support knowledge workers, the enterprise should strive to create object oriented environments.

These two concepts, knowledge worker and object oriented environments are brought together into technology architectures since both uniquely characterize the ideal working environment.

The knowledge worker's environment involves both automated and non-automated activities. Some non-automated activities involve the use of automation, for example, once a patient receives a treatment from a clinician (non-automated activity), the characteristics of the treatment, and the clinicians observations about the patient's reaction to the treatment are typically recorded in some automated system. A knowledge worker's framework must therefore address manual and automated activities.

Knowledge workers perform groups of functions to accomplish their designated job or to accomplish some aspect of the enterprise's mission. Knowledge workers may perform these function groups in different combinations depending on the enterprise's organization. For example, if an organization is highly distributed into multi-functional units, there may be staff that perform diverse groups of functions. Conversely, a highly centralized organization may have certain staff devoted to specific and highly specialized functions. The knowledge worker is therefore a complex multi-faceted person who performs diverse functions of different constructions for one or more organizations.

Enterprises commonly create computing supports for knowledge workers under the assumption that the functions they perform and the organizations through which they act are fixed and seldom change. Not only are these assumptions wrong, but when the functions and organizations do change, computing environment changes seldom keep pace because they are time consuming to specify, difficult to implement, and slow to accomplish. Slow-to-react computing environment changes, therefore, become the very reason why information technology support to business functions and organizations cannot keep pace with the demands of change. What is needed are computing environments that are object oriented, sensitive to knowledge worker functions and organizations, and that can react to the demands of change in a timely fashion.

The Knowledge Worker Framework is a rows and columns collection of work products suitable for accomplishment by persons primarily focused on knowledge-based work. The columns represent a homogeneous collection of work products that while different, cascade through a series of rows that each represent a different viewpoint for that overall Knowledge Work column heading.

The Knowledge Worker columns, which occur left-to-right in a definite sequence are:

- Mission
- Database Object

- Business Information System
- Business Event
- Business Function
- Business Organization

The Missions column represents the essence of the enterprise. Its descriptions are in an idealized textual hierarchical or network form that devoid of who, what, when, where and how. Those are implementation mechanisms. Thus, missions are the "why" basis of the enterprise.

The Database Objects column presents the policy-based proofs that an aspect of an enterprise mission has been accomplished. Database objects too are in an idealized form and, as well are devoid of "who," "when," or "where." The "what" of database objects are the data structures, and attendant database object based processes that ensure correct, accurate and complete data collection are the "how" with respect to Database Object Classes.

The Business Information Systems column represents the business information systems necessary to capture, update, and report the data required by the data structures of the database objects columns. They are devoid of "who" or "when."

The Business Events column is the triggering column that interlinks the Business Functions to the "right" with the Business Information Systems to the "left". Business Events are the "when" of the Business Information Systems execution triggers.

The Business Functions column represents the manual functions performed by knowledge workers as they proceed to accomplish their jobs.

All seven of the columns are interlinked through many-to-many relationships so that reality can be accurately reflected.

Collectively, the Knowledge Worker Framework story is as follows: When an organization proceeds through its Business Function work processes, it sometimes needs IT support. In those cases, the appropriate Business Event is triggered. It, in turn, invokes the execution of the Business Information System, that, in turn, causes the collection, reporting and updating of data represented by the Database Object. All of this services to fulfill the needs of a mission from the Missions column.

4.1 Knowledge Worker Framework Columns and Rows

Table 5 sets out the columns and rows from the Knowledge Worker Framework..

Knowledge Worker Framework						
Perspective	**Mission**	**Database Object Classes**	**Business Information System**	**Business Event**	**Business Function**	**Organization**
Scope	Business missions	Major business resources	Business information Systems	Interface events	Major business scenarios	Organizations

Knowledge Worker Framework						
Perspective	**Mission**	**Database Object Classes**	**Business Information System**	**Business Event**	**Business Function**	**Organization**
Business	Mission hierarchies	Database Domains, and Resource Life Cycles	Information sequencing, hierarchies, and use cases	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 Object Classes	Information system designs	Invocation protocols, input and output data, and messages	Best practices, quality measures and accomplishment assessments	Job roles, responsibilities, and activity schedules
Technology	Policy execution enforcement	Implemented data models and Detailed Database Object Classes	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 model	Operating information systems	Start, stop, and messages	Detailed procedure based instructions	Daily activity executions, and assessments

Table 5. Knowledge Worker Framework Columns and Rows.

4.2 Relationships Between Knowledge Worker Framework and ROIs

The relationships between the various columns of the Knowledge Worker Framework and the ROIs is contained in Section 2 and its subsections. Specifically:

- Section 2.1, Enterprise-wide Project Management (16 to 1 ROI)
- Section 2.2, Information Systems Planning (5.9 to 1 ROI)
- Section 2.3, Data Centered Development and Management (28 to 1 ROI)
- Section 2.4, Data Model Manufacturing (8.6 to 1 ROI)
- Section 2.5, Business Information System Environments (7.7 to 1 ROI)
- Section 2.6, Business Information System Manufacturing (2.9 to 1 ROI)
- Section 2.7, Enterprise Architecture Development (14.6 to 1 ROI)

4.3 Relationships Between Knowledge Worker Framework and Data Architecture Reference Model

The Database Objects column contains the complete set of data models contained in the Data Architecture Reference Model. These models include:

- Data Elements
- Specified Data Model
- Implemented Data Model
- Operational Data Model
- View Data Model

These five models are described in Section 4, Data Architecture Reference Model.

The Data Architecture Reference Model is accomplished through the work products and Knowledge Worker Framework cells itemized in Table 6.

Work Product	Knowledge Worker Framework	
	Column	Row
Database Domains	Database Object	Business
Data Elements		System
Specified Data Models		System
Implemented Data Models		Technology
Operational Data Models		Deployment
View Data Models		Operations

Table 6. Knowledge Worker Framework Work Products for Data Architecture Reference Model

4.4 Relationships Between Knowledge Worker Framework and Resource Life Cycle Analysis

Resource Life Cycle Analysis is accomplished through the work products and Knowledge Worker Framework cells itemized in Table 7.

Work Product	Knowledge Worker Framework	
	Column	Row
Major Business Resources	Database Object	Scope
Resource Life Cycles		Business

Table 7. Knowledge Worker Framework Work Products for Resource Life Cycle Analysis

4.5 Relationships Between Knowledge Worker Framework and Database Object Classes

Database Object Classes are accomplished through the work products and Knowledge Worker Framework cells listed in Table 8.

Work Product	Knowledge Worker Framework	
	Column	Row
Identified Database Object Classes	Database Object	System
Detailed Database Object Classes		Technology

Table 8. Knowledge Worker Framework Work Products for Database Object Classes

4.6 Relationships Between Knowledge Worker Framework and Business Information Systems

Business Information Systems are accomplished through the work products and Knowledge Worker Framework cells listed in Table 9.

Work Product	Knowledge Worker Framework	
	Column	Row
Business Information Systems	Business Information System	Scope

Work Product	Knowledge Worker Framework	
	Column	Row
Information Sequencing, hierarchies, and use cases		Business
Information Systems Designs		System
Information System Application Designs		Technology
Implemented Information System		Deployment
Operating Information Systems		Operations

Table 9. Knowledge Worker Framework Work Products for Business Information Systems

4.7 Relationships Between Knowledge Worker Framework and Metabase System Modules

The Metabase System modules are involved in all rows and columns of the Knowledge Worker Framework. The Metabase System Modules identified with the Knowledge Worker Framework cells set out in Table 10.

Knowledge Worker Framework Identified Metabase System Modules						
Perspective	**Mission**	**Database Object Classes**	**Business Information System**	**Business Event**	**Business Function**	**Organization**
Scope	Missions Organizations Functions Position Assignment	Major business resources	Business information Systems	Business Event Management	Major business scenarios	Organizations

Knowledge Worker Framework Identified Metabase System Modules						
Perspective	**Mission**	**Database Object Classes**	**Business Information System**	**Business Event**	**Business Function**	**Organization**
Business	Missions Organizations Functions Position Assignment	Database Domains Resource Life Cycles Information Needs Analysis Business Rules	Business Information Systems Use Cases Data Integrity Rules Information Needs Analysis Business Rules	Business Event Management	Business scenario sequencing and hierarchies	Organization charts, jobs and descriptions
System	Missions Organizations Functions Position Assignment	Data Elements Specified Data Models Database Object Classes	Business Information Systems Wire Frames User Acceptance Tests	Business Event Management	Documents and Forms	Documents and Forms
Technology	Documents and Forms	Implemented Data Models Database Object Classes	Business Information Systems Wire Frames User Acceptance Tests	Business Event Management	Documents and Forms	Documents and Forms
Deployment	Documents and Forms	Operational Data Models	Business Information Systems Wire Frames User Acceptance Tests	Business Event Management	Documents and Forms	Documents and Forms

Knowledge Worker Framework Identified Metabase System Modules						
Perspective	**Mission**	**Database Object Classes**	**Business Information System**	**Business Event**	**Business Function**	**Organization**
Operations	Documents and Forms	View Data Model	Business Information Systems Wire Frames User Acceptance Tests	Business Event Management	Documents and Forms	Documents and Forms

Table 10. Knowledge Worker Framework With Identified Metabase System Modules

5 Data Architecture Reference Model

The data architecture reference model is in the area of data engineering and serves as a framework for understanding the scope and interrelationships among clearly definable sets of its contained data models.

Accomplished, the individual contained data models enhance clear communication among deployed instances of the other contained data architecture models. Figure 6 clearly shows a hierarchical collection of data models. These can be constructed top-down or bottom-up.

The final result enables semantic homogeneity of data across the enterprise. Additionally, this collection of data models enable the integrity and interoperability of all the key work products needed for business information systems. The Data Architecture Reference Model enables quick, efficient and effective traceability.

5.1 Data Architecture Reference Models and Descriptions

The data architecture consists of four distinct technology models and then two technology-dependent interface models. The four main models depicted in Figure 6 are:

- Data Element Model
- Concepts (Specified) Model
- Logical (Implemented) Database Model
- Physical (Operational) Database Model

The two technology dependent interface models are:

- SQL View Model
- XML Schema Model

The data models for each of the models are depicted as follows:

- Data Elements, Figures 7
- Specified Data Model, Figure 8
- Implemented Data Model, Figure 9
- Operational Data Model, Figure 10
- View Data Model, Figure 11
- XML Data Model, <not shown>

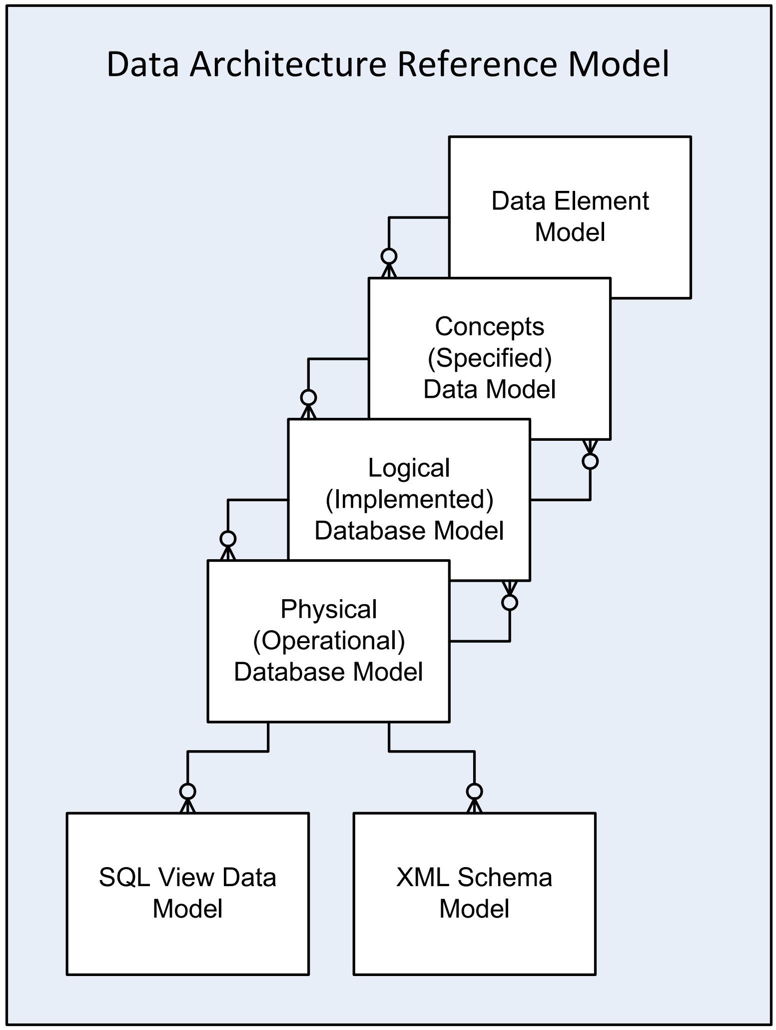


Figure 6. Data Architecture Reference Model.

All the Data Architecture Reference Models are detailed within the Data Architecture Reference Model paper and in their respective Metabase System module user guides.

Within this environment of six distinct data models, the Data Element model captures the once-only identification, specification, and definition of data elements that may be represented as database table columns in many different database tables.

Similarly, the Specified Data Models enable the specification of data structures for data-based concepts that, in turn, form the standardized building blocks of all database models (Implemented and Operational) and for View models. Collectively, these enable integration, interoperability across all the enterprise's business information systems.

These Specified Data Models can be deployed in one or more Implemented Database Models, which, in turn are operationally deployed in one or more DBMS specific Operational Database Models.

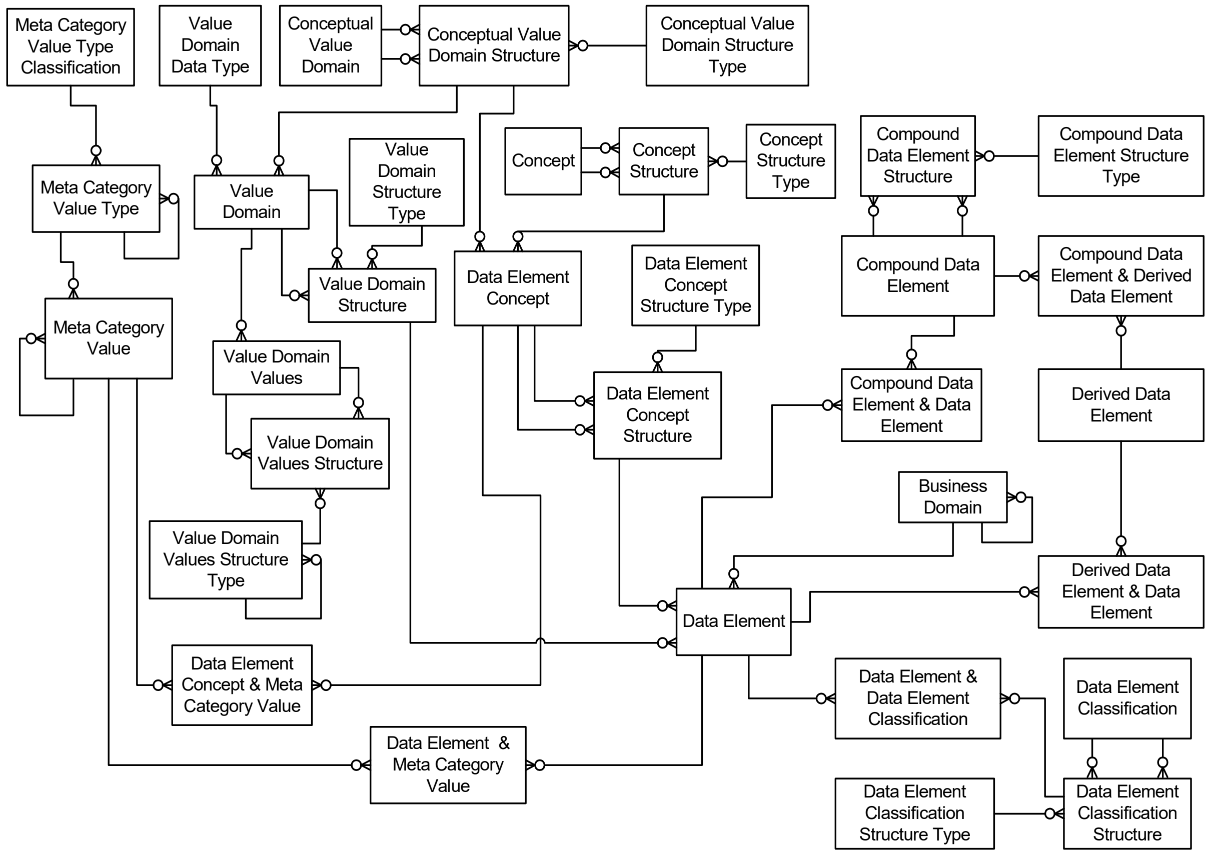


Figure 7. Data Element Model.

Each of these six data model classes serve a special purpose and each is interrelated with the others in some integrity-enhancing and work-saving manner.

Figure 6 depicts a left-side set of one-to-many relationships going "down." This relationship supports two meanings. The first is the mapping of an individual component of a model, and the second is the mapping to a whole collection from within the “parent” model to the “child” model.

In the Data Element model there can be individual data elements such as Person First Name, and there can be collections of data elements within a specific data element concept collection, for example, Person Related Information such as Person Identifier, Person Birth Date, Person First Name, Person Middle Name, and Person Last Name.

In the first type of left-side one-to-many relationship, the individual data element, Persons First Name would be semantically mapped to zero, one, or more attributes within different entities. For example, to Employee First Name, to Customer Contact First Name, or to Causality Insurance Contract First Name.

In the second type of "left-side" relationship, a whole collection of data elements can be mapped to a whole collection of attributes across one or more entities. For example, all the Data Elements within a Data Element Concept collection called Biographic Data Elements might be

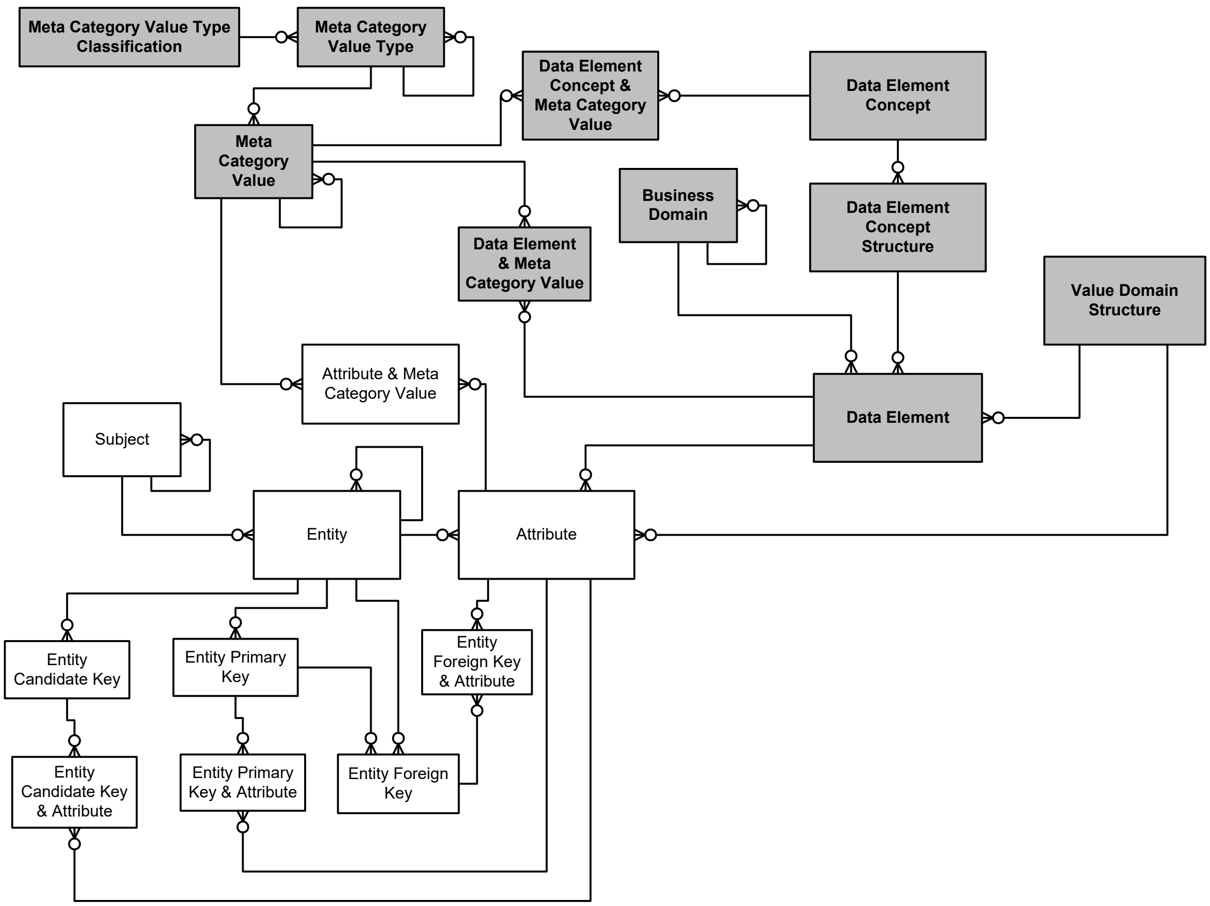


Figure 8. Specified Data Model

mapped to the entity, Person Information, or to the entity, Customer Contact Information. In this case, the mapping of the data elements, Person Identifier, Person First Name, etc., is mapped to a corresponding set of attributes within one or more entities.

On the right-side of Figure 6, there is also a set of one-to-many relationships. These go "up." This set, like the left-side one-to-many relationships has two meanings: individual component, and whole collections. The meanings of the right-side one-to-many relationship are different from the left-side one-to-many. The first type of right-side relationship, the mapping of an individual component is not one-to-many, but one-to-one. Thus, an individual DBMS Column, for example, EmpFrstNam can be inherited from only one higher level component, for example, the single column, EmployeeFirstName.

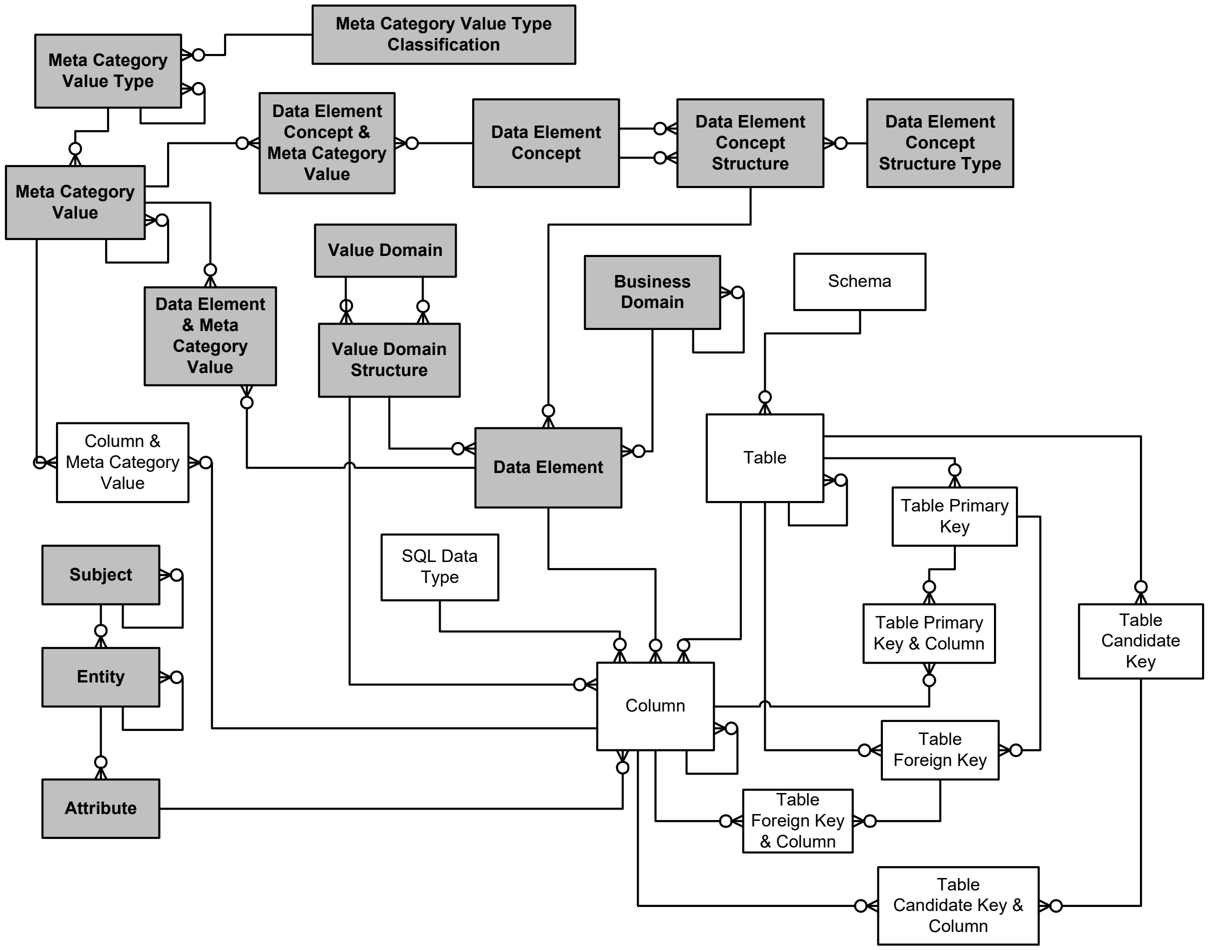


Figure 9. Implemented Data Model

The second type of right-side relationship, the mapping of collections can be one-to-many. That is, one collection can map to one set of columns within one table of a single Implemented Database Model while another collection from the same Implemented Database Model can be mapped to a different collection within a different Implemented Database Model. Hence, the collections can be seen as "from" one Implemented Database Model to zero, one, or more different Implemented Database Models.

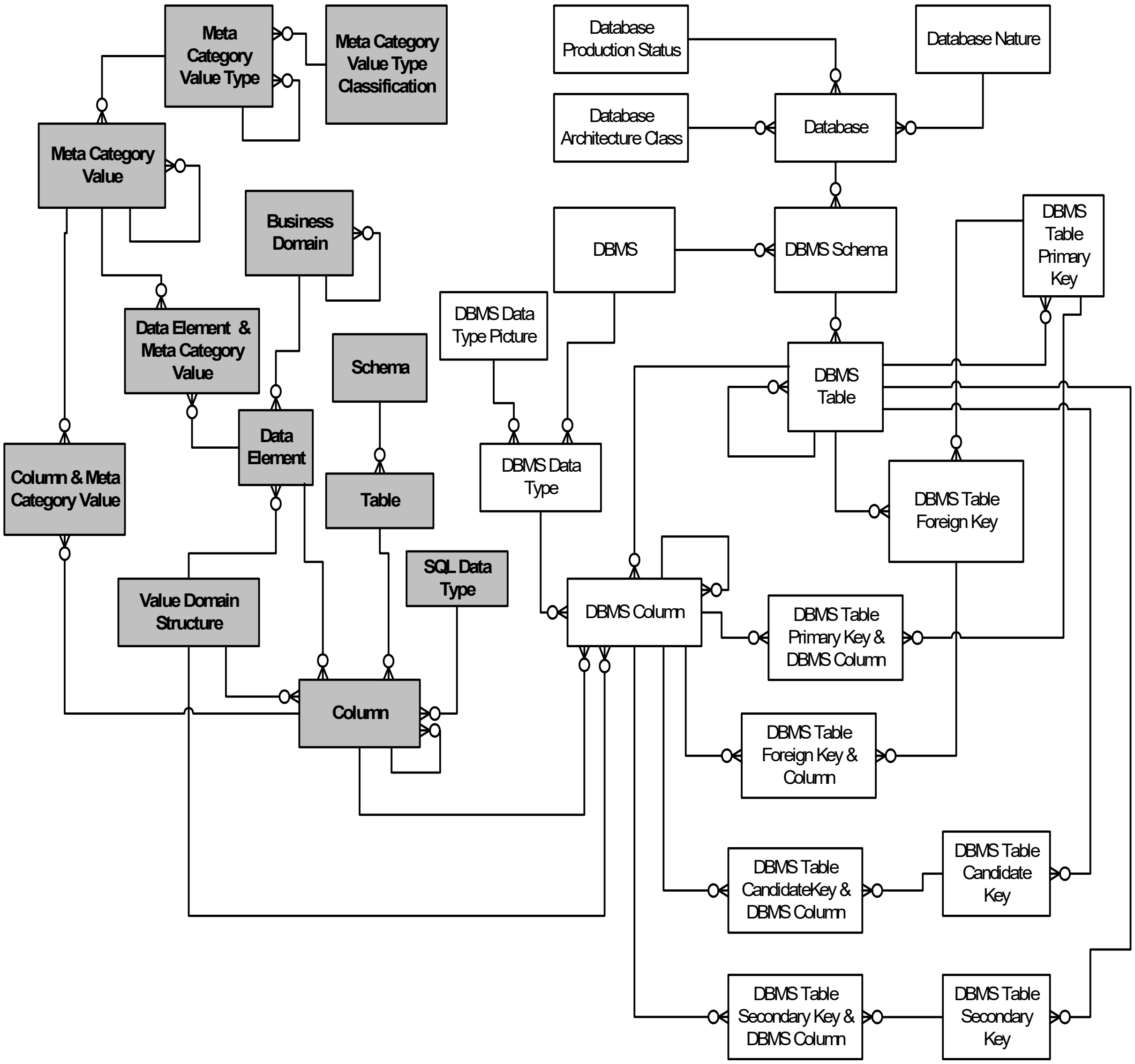


Figure 10. Operational Data Model

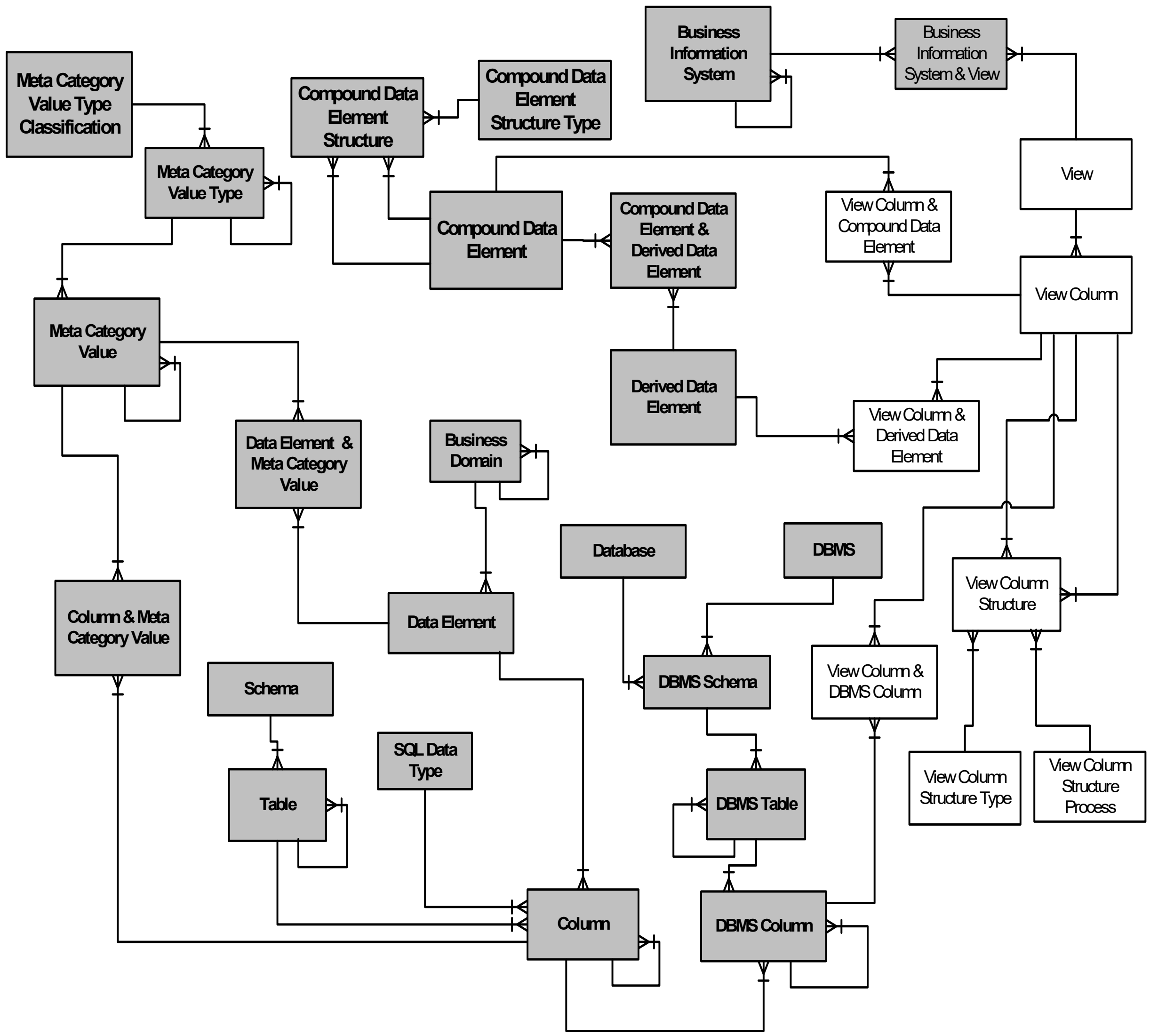


Figure 11. View Data Model

The description of the models is set out in Table 11. Table 12 provides a cross-reference between work products and data architecture reference model component.

Data Model Class	Description
Data Element	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 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.
Specified Data Model	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.
mplemented 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
Operational Database Models	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

Data Model Class	Description
	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.
View Data Model	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 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.
XML Interface Data Models	A XML interface model represents a specialized construction of importable or exportable data with respect to the business information system. Each XML data stream is defined through a XML Schema. Both XML schemas and XML data streams are independent of the software applications that create and/or use them. XML Schemas and XML data streams are DBMS represented in plain ASCII text. Key Components are XSDs, XML elements, and XML attributes. The structure of XML data is expressed through an XML schema that is employed to then understand the contents of XML data records. XML schemas are created through special software applications. XML data streams are created by source application business information systems and are subsequently read and processed by target application business information systems.

Table 11. Data Model Layers within the Data Architecture Reference Model.

Work Products	Data Architecture Reference Model Component					
	Data Element	Concepts Data Model	Logical Data Model	Physical Data Model	View Data Model	XML Data Model
Business Information Systems				✔	✔	✔
Business Requirements		✔	✔	✔		✔
Business Rules	✔		✔	✔	✔	✔
Database Domains			✔			
Database Objects			✔			
External Data Interface Requirements				✔	✔	✔
Eternal Quality Standards	✔	✔	✔	✔	✔	✔
Information Needs	✔		✔			
Mission Organization Functions			✔			
Resource Life Cycles			✔			
Use Cases			✔	✔	✔	✔
User Acceptance Tests				✔	✔	✔
Value Domains and Management	✔	✔	✔	✔	✔	✔
Work Breakdown Structure (WBS)	✔	✔	✔	✔	✔	✔

Table 12. Cross reference between work products and data architecture reference model component.

5.2 Relationships Between Data Architecture Model and ROIs

The relationships between the Data Architecture Reference Model and the ROIs is contained in Section 2 and its subsections. Specifically:

- Section 2.1, Enterprise-wide Project Management (16 to 1 ROI)
- Section 2.2, Information Systems Planning (5.9 to 1 ROI)
- Section 2.3, Data Centered Development and Management (28 to 1 ROI)
- Section 2.4, Data Model Manufacturing (8.6 to 1 ROI)
- Section 2.5, Business Information System Environments (7.7 to 1 ROI)
- Section 2.6, Business Information System Manufacturing (2.9 to 1 ROI)

- Section 2.7, Enterprise Architecture Development (14.6 to 1 ROI)

5.3 Relationship Between Data Architecture Model and Knowledge Worker Framework

The data models within the Data Architecture Reference model are all created within the Knowledge Worker Framework column, Database Objects, rows Business through Operation as presented in Table 5.

5.4 Relationships Between Data Architecture Model and Resource Life Cycle Analysis

The relationships between the data models within the Data Architecture Reference model and the Resource Life Cycle Analysis are all indirect. That is, a given Database Object Class is mapped to a given Resource Life Cycle Node. The Database Object, in turn, is mapped to tables from the Implemented Data Model. This indirect relationship is depicted in Figure 12.

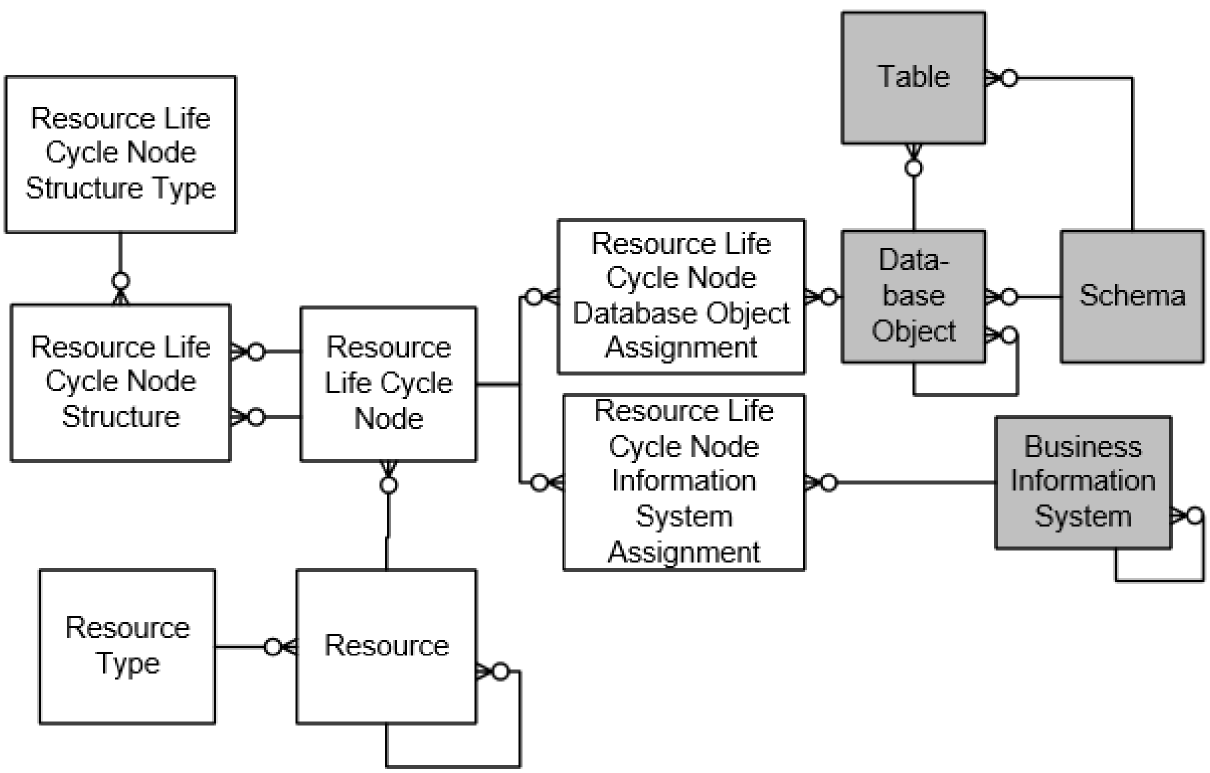


Figure 12. Relationship between Resource Life Cycle Analysis and Data Architecture Reference Model: Implemented Data Model through Schema, Table, and Database Object Class.

5.5 Relationships Between Data Architecture Model and Database Object Classes

The relationships between the data models within the Data Architecture Reference model and the Resource Life Cycle Analysis are all indirect. That is, a given Database Object Class is mapped to a given Resource Life Cycle Node. This is also depicted in Figure 12.

5.6 Relationships Between Data Architecture Model and Business Information Systems

5.7 Relationships Between Data Architecture Model and Database Object Classes

The relationships between the data models within the Data Architecture Reference model and the Resource Life Cycle Analysis are all indirect. That is, a given Database Object Class is mapped to a given Resource Life Cycle Node. This is also depicted in Figure 12.

5.8 Relationships Between Data Architecture Model and Business Information Systems

During the specification, development, and evolution of the work products that ultimately comprise a business information system, there can be an endless array of names for some of these work products. These work products generally fall into the categories listed in column one of Table 13 A depiction of the interrelationships of the data models and of the work products is presented in Figure 13. Table 13 also shows the cross reference between the Data Architecture reference model data models and the typical work products of a business information system effort.

Business Information System Work Products, for example, Business Requirements, shown in Figure 13 can have a one-to-many relationship between one or more different business information system work products and zero, one or more of the data models.

When two or more business information system work products are interrelated with one or more of the data models, there may exist relationship between those two business information system work products through artifacts contained in the data models. For example, there is a business requirement to capture student addresses.

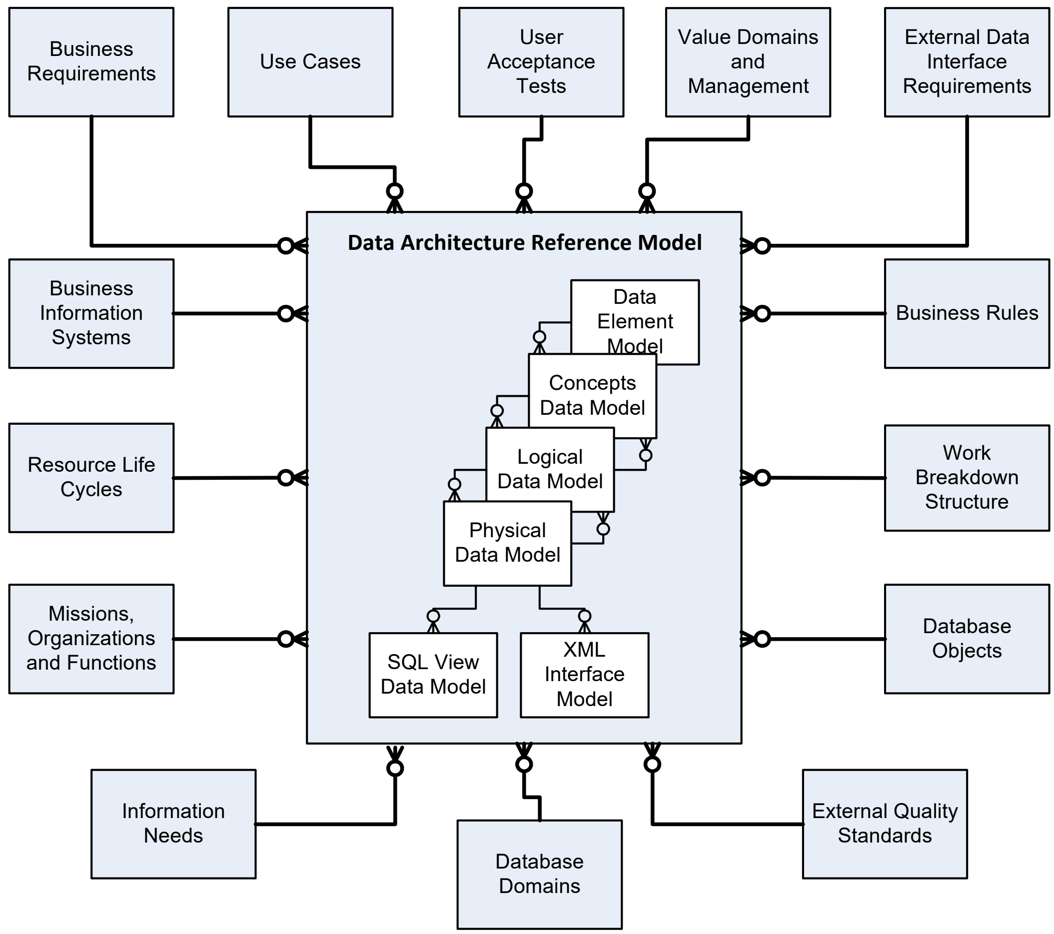


Figure 13. Cross Reference between Data Architecture Models and Business Information System Work Products.

Business Information System Life Cycle Work Products		
Work Product	**Description**	**Data Model Components**
Business Requirements	Business Requirements are the identification, specification, and definition of the components that must exist in the ultimate solution delivered by the contractor. Business Requirements form the foundation upon which all components of are engineered, implemented, and maintained. Business requirements will evolve over time. Thus, it is important to be able to track the initial and evolved requirements. It is unrealistic to initially have all requirements because new and/or revised requirements are discovered all during the project's architecture, engineering, and implementation.	Subjects, Entities Attributes Data Elements Database Domains Database Objects

Business Information System Life Cycle Work Products		
Work Product	**Description**	**Data Model Components**
Business Rules	Business Rules are assertions of truth-states in the database. Each business rule includes the identification, name, description, and exact specification of data-based rules that must either be true or that, after the execution of an information system process, results in a state of truth. There are two classes of Business Rules: data and process. Almost invariably, business rules depend on existing data, reference or control data, or data that is determined as a consequence of a process's execution. Almost all business rules are mappable to data, whether persistent or temporary. Business rules are almost always discovered during design sessions, and use-case walkthroughs. As business rules are discovered, the various data models need to be concurrently examined to determine whether the database can support the rules. Since every business rule has a process component, a key component of each rule is the specification of the process and a determination of where that rule is bound. That is, bound into the data model component (e.g., Data Element, or DBMS column), a low-level application component, a mid-level application process, or in the user presentation layer. Because of this multiplicity of possible bindings, business rules need to be centrally defined and managed, but bound only into the data model or business information system work product within which it is accomplished.	Schema Tables Columns DBMS Schema DBMS Tables DBMS Columns Value Domains
Work Breakdown Structure (WBS)	Work Breakdown Structures are hierarchical representations of two classes of effort: What, and How. The "what" type of WBS contains an action phrase and a noun phrase that together describe what is to be done, and the name of the work product creation or evolution. The second class of WBS, the "how WBSs" are tuned to deliverables but to the actual techniques employed to accomplish the data model or business information system work product. Both the "What" WBS and the "How" WBSs need to be interlinked. Well developed metadata management systems include complete project management so that work plans and progress against work plans can be directly tied to and integrated with accomplishments.	Subjects, Entities Attributes Data Elements Database Domains Database Objects Schema Tables Columns DBMS Schema DBMS Tables DBMS Columns
External data interface Data Requirements	External data interface Requirements are the specifications of an interface between an internal database data structure and some external data source. Each interface essentially consists of a fully defined data model that defines every field in the interface to the extent that a software module can be created to read the records represented by the	DBMS Schemas DBMS Tables DBMS Columns DBMS Relationships

Business Information System Life Cycle Work Products		
Work Product	**Description**	**Data Model Components**
	interface and take appropriate action against the database. Such actions can be to insert, delete, or modify database records.	Value Domains Views View Columns Relationships XML Schemas XML Elements XML Attributes
External Quality Standards	External Quality Standards are de jure and de facto standards through which data management products and/or processes can be judged. The ISO Standard 11179 enables assessment of the adequacy and completeness of the metadata associated with data elements. Included in this class of assessment are Concepts, Conceptual Value Domains, Data Element Concepts, Value Domains including mapping among equivalent values, Data Element Classifications, and Administrative Information (for Stewardship). The ISO/ANSI SQL standard enables the assessment of SQL data structures and languages employed in database designs and application program access. WC3 XML standards enable an analysis of the names and engineering of XML schemas and XML data streams so as to ensure maximum interoperability conformity to existing sets of XML schema models.	Schemas Tables Columns Relationships Views View Columns Relationships XML Schemas XML Elements XML Attributes
Business Information Business Information Systems	Business Information Systems are the broad characterizations of the application systems that capture, update, and report data. Business Information Systems are additionally detailed into their specific components, and those that deal with database data are mapped to the appropriate data model component.	Views View Columns Relationships XML Schemas XML Elements XML Attributes
Use Cases	Use Cases are highly engineered pseudo-process models that clearly define the behavior of the users and business information systems, and also the responses provided from the databases as they take in, modify, or provide data to users. Use-cases are detailed to the level such that a programmer can interpret the process intent and write an application system module without semantic and/or process misunderstanding. Use Cases present behavior-based scenarios of the use of the database to accomplish the requirements. Because use-cases directly identify database data, mappings can be created between one use-case and another to identify redundancy and possible conflict. Mappings can also be made between the detailed data and process aspects of use-cases and	Schemas Tables Columns Relationships Value Domains DBMS Schemas DBMS Tables DBMS Columns DBMS Relationships Views View Columns

Business Information System Life Cycle Work Products		
Work Product	**Description**	**Data Model Components**
	business requirements, deployments of use-cases in software and hardware, value domains, business rules, and WBS.	Relationships XML Schemas XML Elements XML Attributes
User Acceptance Tests	User Acceptance Tests are stylized user-application system interaction scripts that can be exercised to the extent that fully informed users can determine that all the requirements have been met and all use-cases are satisfactorily performed. User Acceptance Tests (UAT) are the ultimate mechanisms through which organizations determine that it has received the business information system that was specified. Because of all the different data models, and work products, the User Acceptance tests can be very comprehensive and very telling.	DBMS Schemas DBMS Tables DBMS Columns DBMS Relationships Value Domains Views View Columns Relationships XML Schemas XML Elements XML Attributes
Value Domains	Value domains relate to the allowed, disallowed, or other defined collections of values and interconnection of values that represent discrete choices (Gender = Male, Female, unknown), or sequenced states such as Applied, Reviewed, Accepted, Rejected, or Appealed. Included as well are the mappings across time of evolved and/or changed value domains. Value domains commonly stand alone and are allocated to data elements, or to attributes, columns, or DBMS columns. In all cases of value domain allocation, the allocations must be such that semantics conflicts are prohibited.	DBMS Schemas DBMS Tables DBMS Columns DBMS Relationships Value Domains Views View Columns Relationships XML Schemas XML Elements XML Attributes
Resource Life Cycle Analyses	Resource Life Cycle of Analysis identifies, defines, and sets out the resources of the enterprises, the life cycles that represent their accomplishments, and the interrelationships among the different enterprise resource life cycles. Resource life cycle nodes represent the end-state data resulting from the execution of business information systems. The end-state data is represented through database object classes.	Schemas Tables Columns Relationships
Missions Organizations and Functions	Missions, organizations, functions, and position assignments represent the identification, definition, and interrelationships among the persons who, through their positions, perform functions within their organizations that accomplish enterprise missions. Missions define the very existence of the enterprise, and that are the ultimate goals and objectives that measure enterprise accomplishment from within different	Schemas Tables Columns Relationships

Business Information System Life Cycle Work Products		
Work Product	**Description**	**Data Model Components**
	business functions and organizations. Functions represent the procedures performed by enterprise organization groups as they achieve the various missions of the enterprise from within different enterprise organizations. Organizations represent the bureaucratic units created to perform through their functions the mission of the enterprise.	
Information Needs	Information needs represent the identification, definition, and interrelationship of the information needed by various organizations in their functional accomplishment of missions and what databases and information systems provide this information?	Schemas Tables Columns Relationships
Database Domains	Database domains are the data-intensive bridge between mission descriptions and database object classes. While database object classes are non redundant, they may be referenced by several database domains.	Schemas Tables Columns Relationships
Database Object Classes	Database object classes represent the identification of 1) the critical data structures, 2) the processes ensure high quality and integrity data within these data structures, 3) the value-based states represented by these data structures, and 4) the database object centric information systems that value and transform database objects from one state to the next. Database Objects are transformed from one valid state to another in support of fulfilling the information needs of business information systems as they operation within the business functions of organizations.	Schemas Tables Columns Relationships

Table 13. Data Management Components and affected Data Models.

5.9 Relationships Between Data Architecture Model and Metabase System Modules

There is a one to one relationship between the Data Architecture Reference Model and the Metabase System modules. This is depicted in Table 14.

Business Information System Models	Metabase System Module
Data Elements	Data Elements
Specified Data Model	Specified Data Model
Implemented Data Model	Implemented Data Model
Operational Data Model	Operational Data Model
View Data Model	View Data Model

Table 14. Relationship between Data Architecture Reference Models and Metabase System Modules

6 Resource Life Cycle Analysis

The goal of RLC analysis is to build a bridge between the operational level needs of information management organizations and the strategic level organization business process needs required by upper management. The main goal of the strategic level is to identify and describe the major resources that are essential to the enterprise's survival, and the main goal of an information systems organization is to plan, develop, deliver, and maintain the various information systems projects that are required to implement the enterprise resources in the most effective manner possible. RLC analysis achieves this bridge goal by determining:

- The resource life cycle networks
- The database object projects and business information system projects and establishing their proper sequence for analysis, design, and implementation.
- A strategic view of the ongoing information systems development and major maintenance work

RLC analysis determines three components of the resource life cycle networks, that is, the resource, life cycle and the precedence vectors between resource life cycles. A resource is an enduring asset of value to the enterprise. The life cycle is a linear identification of the major states that must exist within life of the resource. The life cycle of a resource represents the resource's "cradle to grave" set of state changes. The precedence is a vector that may occur

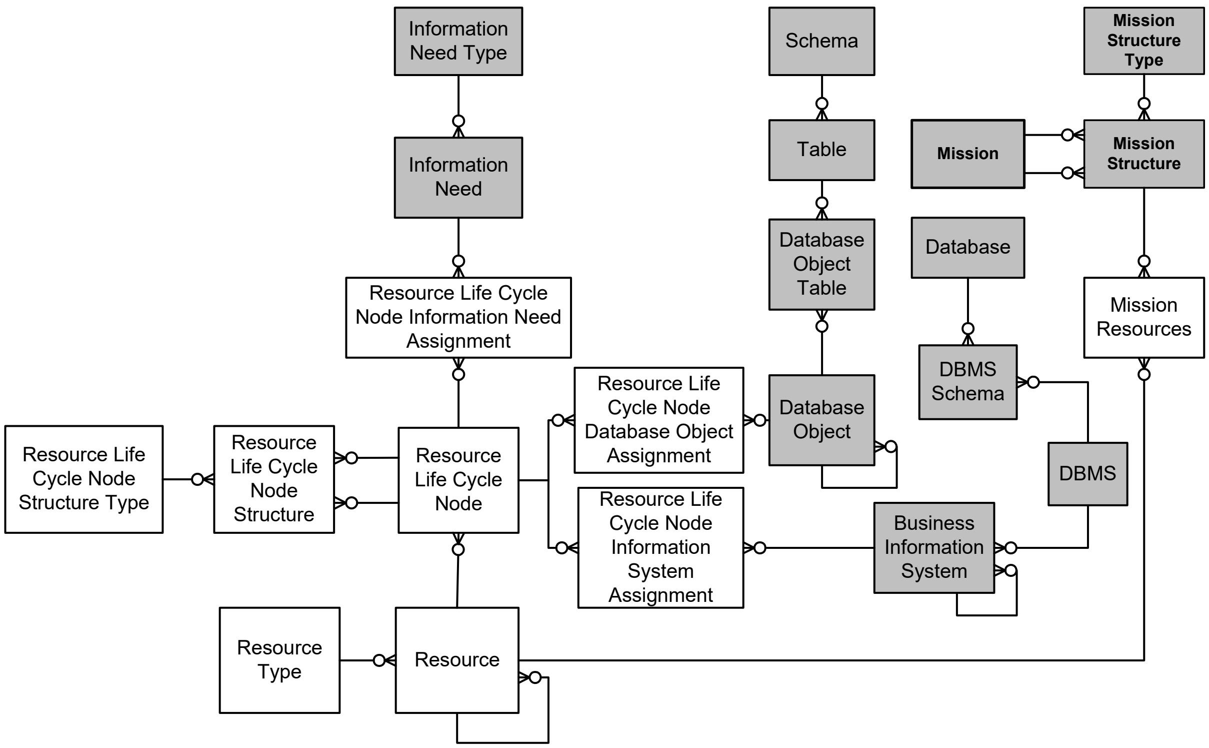


Figure 14. Resource Life Cycle Data Model.

between nodes on different resource life cycles, and thus indicates which resource life cycle node enables another resource life cycle node.

6.1 Resource Life Cycle Analysis Tables and Descriptions

Figure 14 depicts the data model for Resource Life Cycle Analysis. The main tables and their descriptions are presented in Table 15.

Resource Life Cycle Analysis	
Table	**Descriptions**
Resource Life Cycle Node Business Information Systems Assignment	A Resource Life Cycle Node Business Information Systems Assignment is an association between a resource life cycle node and a business information system. A business information system may be assigned to one or more resource life cycle nodes and a resource life cycle node may be related to one or more business information systems.
Resource Life Cycle Node Business Information System Assignment	The Resource Life Cycle Node Business Information System Assignment meta entity represents the association of the business information system with one or more resource life cycle nodes.
Resource Life Cycle Node Database Object Assignment	A Resource Life Cycle Node Database Assignment is an association between a resource life cycle node and a database object. A database object may be assigned to one or more resource life cycle nodes and a resource life cycle node may be related to one or more database objects.
Resource Life Cycle Node Information Need Assignment.	A Resource Life Cycle Node Information Need Assignment is the association of one resource life cycle node and an information need. A resource life cycle node may be related to one or more information needs and an information need may be related to one or more resource life cycle nodes.
Resource Life Cycle Node Structure.	The Resource Life Cycle Node Structure is the association of one resource life cycle node an another. The association represents a relationship between the two resources for some purpose.
Resource Life Cycle Node.	A Resource Life Cycle Node is a life cycle state within the resource. If the resource is employee then the life cycle node may be employee requisition, employee candidate, employee new hire, assigned employee, reviewed employee, and separated employee.
Resource Life Cycle Node Structure Type.	A Resource Life Cycle Node Structure Type is a classification of a set of resource life cycle node structures that explain the collection. An example might be enablement and the associated resource life cycle node structures might related a recognized receivable resource life cycle node from the receivables resource “enables” a paid invoice resource life cycle node from the invoice resource.
Resource Type	A Resource Type represents a collection of resources. For example, finance resource resources, or property resources.
Resource	A Resource is an enduring asset of value to the enterprise. Included for example are facilities, assets, staffs, money, even abstract concepts like reputation. If a resource is missing then the enterprise is incomplete.

Table 15. Resource Life Cycle Analysis Components and Descriptions

6.2 Relationships Between Resource Life Cycle Analysis and ROIs

The relationships between Resource Life Cycle Analysis and the ROIs is contained in Section 2 and its subsections. Specifically:

- Section 2.1, Enterprise-wide Project Management (16 to 1 ROI)
- Section 2.2, Information Systems Planning (5.9 to 1 ROI)
- Section 2.7, Enterprise Architecture Development (14.6 to 1 ROI)

6.3 Relationship Between Resource Life Cycle Analysis Knowledge Worker Framework

The Resource Life Cycle Analysis work products are accomplished entirely within the Database Objects Column. Specifically rows, 1 (Scope), and 2 (Business).

6.4 Relationships Between Resource Life Cycle Analysis Data Architecture Reference Model

The relationship between the Resource Life Cycle Analysis model and the Data Architecture is indirect. That is, the Recourse Life Cycle Analysis table, Resource Life Cycle Node & Database Object associates one or more Database Objects to a given Resource Life Cycle Node. In turn, the Database Object is associated with one or more Implemented Data Model tables that exist within one schema. This relationship is shown in Figure 14.

6.5 Relationships Between Resource Life Cycle Analysis Database Object Classes

Figure 14 illustrates the relationship between Resource Life Cycle Analysis and Database Objects. The Resource Life Cycle Node is related to the Resource Life Cycle Node & Database Object, which, in turn, is related to one or more Database Objects.

6.6 Relationships Between Resource Life Cycle Analysis Business Information Systems

Figure 14 also illustrates the relationship between a Resource Life Cycle Node and a Business Information System.

6.7 Relationships Between Resource Life Cycle Analysis Metabase System Modules

Resource Life Cycle Analysis is accomplished and the results stored through the Metabase System module, Resource Life Cycle Analysis.

7 Database Object Classes

A database object class is a formally defined IT artifact that contains a set of related tables, primitive processes for transforming rows of data within each contained table, a set of states that model the overall value-based transformations of the objects within the object class, and a defined information system to carry out the state-based transformations. Database Object Classes are used to model complex object such as insurance policies, invoices, companies, persons, bills of lading, and the like.

7.1 Database Object Class Components and Descriptions

The main tables and their descriptions are presented in Table 16. Figure 15 depicts the data model for Database Objects.

Database Object Classes	
Table	**Descriptions**
Database Object Class	A Database Object Class is a type of data structure that proceeds through predefined states according to embedded process transformations. The database object class, the specification of database objects, is defined in four parts: database object data structure, database object process, database object information system and database object state. An instance of a database object class is a database object.

Database Object Classes	
Table	**Descriptions**
Database Object Data Structure	A database object class data structure is the set of data structures that map onto all the different value sets for real world database objects such as an auto accident, vehicle and emergency medicine incident. A database object class data structure's set of columns may be drawn from one or more tables, all of which must be semantically related through primary and foreign key relationships.
Database Object Information System	A Database Object Information System is a set of specifications that controls, sequences, and iterates the execution of various database processes that cause changes in database objects to achieve specific value-based states in conformance to the requirements of business policies. For example, the reception and database posting of data from business information systems activities (screens, data edits, storage, interim reports, etc.) that accomplish entry of the auto accident information.
Database Object Assigned Property Class	A Database Object Assigned Property Class represents one or more property classes assigned to a database object.
Database Object Information Systems Assigned Database Object Table Process	A Database Object Information Systems Assigned Database Object Table Process represents one or more Database Object Table Processes that are assigned to a Business Information System. These are set into a sequence for proper accomplishment.
Database Object Resource Life Cycle Node Assignment	A Database Object Resource Life Cycle Node Assignment represents the association between a Resource Life Cycle node and a database object class. A database object may be assigned to one or more Resource Life Cycle nodes and a Resource Life Cycle node may be related to one or more database objects.
Database Object State	Database Object States are the value states of one or more database objects that represent the after-state of the successful accomplishment of one or more recognizable business events. Examples of business events are auto accident initiation, involved vehicle entry, involved person entry, and auto accident DUI involvement. Database state changes are initiated through named business events that serve business functions. The business function, auto accident investigation includes the business event, auto-accident-incident initiation, which, in turn, causes the incident initiation database object information system to execute, which, in turn, causes several database processes to cause the auto accident incident to be materialized in the database.
Database Object State and Database Object Information System	A Database Object State and Database Object Information System is the association of one or more Database Object States and Database Object Information Systems. The association exists within a sequence so that the state is properly achieved.
Database Object Table	A Database Object Table is an association of an Implemented Data Model table with a database object. Membership rationale classifies the reason why a table belongs to the database object.

Database Object Classes	
Table	**Descriptions**
Database Object Table Process	Database Object Table Process is the set of database processes that enforce the integrity of data structure columns, referential integrity between database objects and actions among contained data structure segments, the proper computer-based rules governing data structure segment insertion, modification, and deletion. An example is the proper and complete storage of an auto accident.
Database Object Table Process Column	A Database Object Table Process Column is an association of a specific database object table process and a specific column of a table.

Table 16. Database Object Class Tables

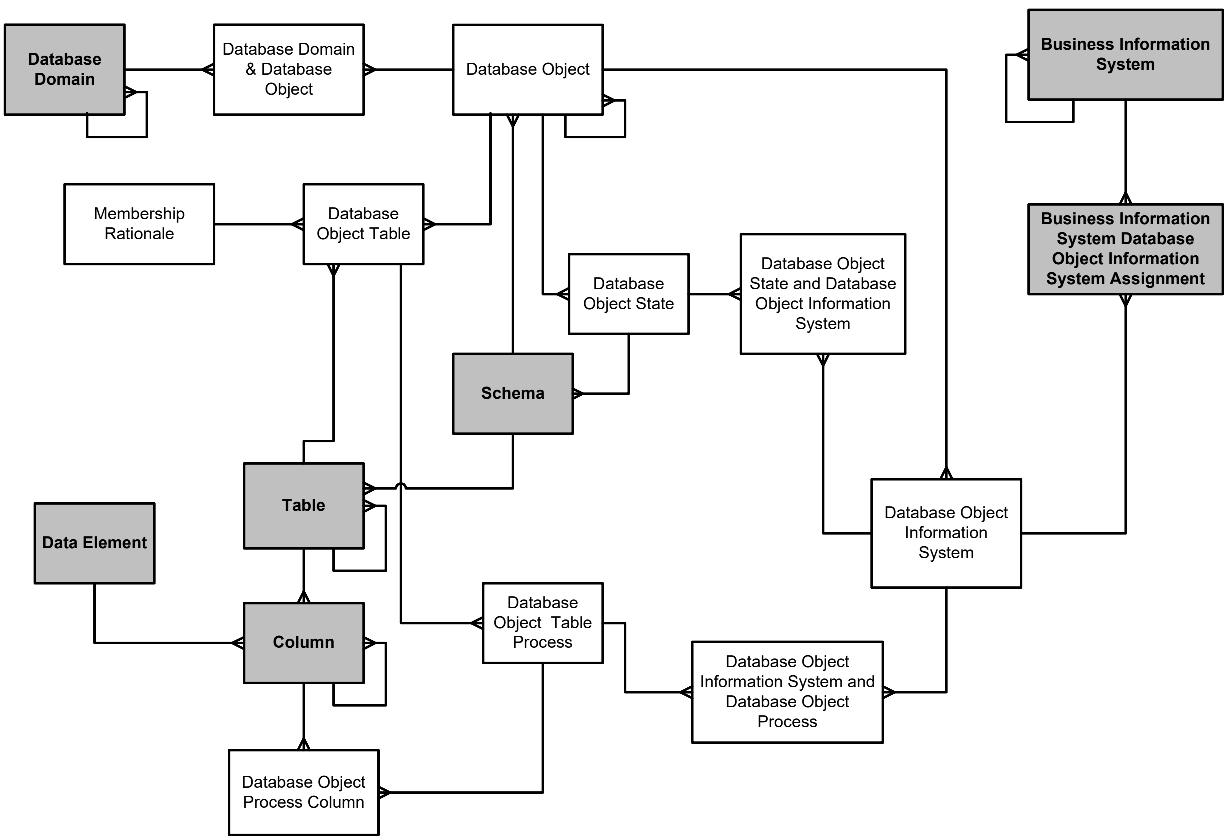


Figure 15. Database Object Class Data Model.

7.2 Relationship Between Database Objects and ROIs

The relationships between Database Object Class and the ROIs is contained in Section 2 and its subsections. Specifically:

- Section 2.1, Enterprise-wide Project Management (16 to 1 ROI)
- Section 2.2, Information Systems Planning (5.9 to 1 ROI)
- Section 2.3, Data Centered Development and Management (28 to 1 ROI)
- Section 2.7, Enterprise Architecture Development (14.6 to 1 ROI)

7.3 Relationship Between Database Objects and Knowledge Worker Framework

Database Object Classes are accomplished in the two rows, System and Technology of the Database Objects Column.

7.4 Relationships Between Database Objects and Data Architecture Reference Model

Database Object Classes are related only to the Implemented Data Model through the relationship between Table and Database Object via the relationship table, Database Object Table. The relationship is illustrated in Figure 15

7.5 Relationships Between Database Objects and Resource Life Cycle Analysis

The relationship between a Database Object and a Resource Life Cycle Node is also indirect. That is, a Database Object Class is mapped to a Resource Life Cycle Node Database Object Assignment, which, in turn is mapped to a Resource Life Cycle Node. This enables a Database Object Class to be mapped to multiple Resource Life Cycle Nodes.

7.6 Relationships Between Database Objects and Business Information Systems

Database Object Classes are not directly related to Business Information Systems. Rather, a Business Information System is directly mapped to the Operational Data Model through a view as shown in Figure 15.

7.7 Relationships Between Database Objects and Metabase System Modules

Database Objects are entirely implemented through the Metabase System module, Database Objects.

8 Business Information Systems

As stated in Section 2.5, business information system development, business information systems are where "the rubber meets the road." It is through business information systems that enterprise-policies are executed and the data resulting from those policy executions is captured, stored, analyzed and reported.

8.1 Business Information System Components and Descriptions

The data model for business information system is presented in Figure 16. The tables that enable the complete specification of a Business Information System are set out in Table 17.

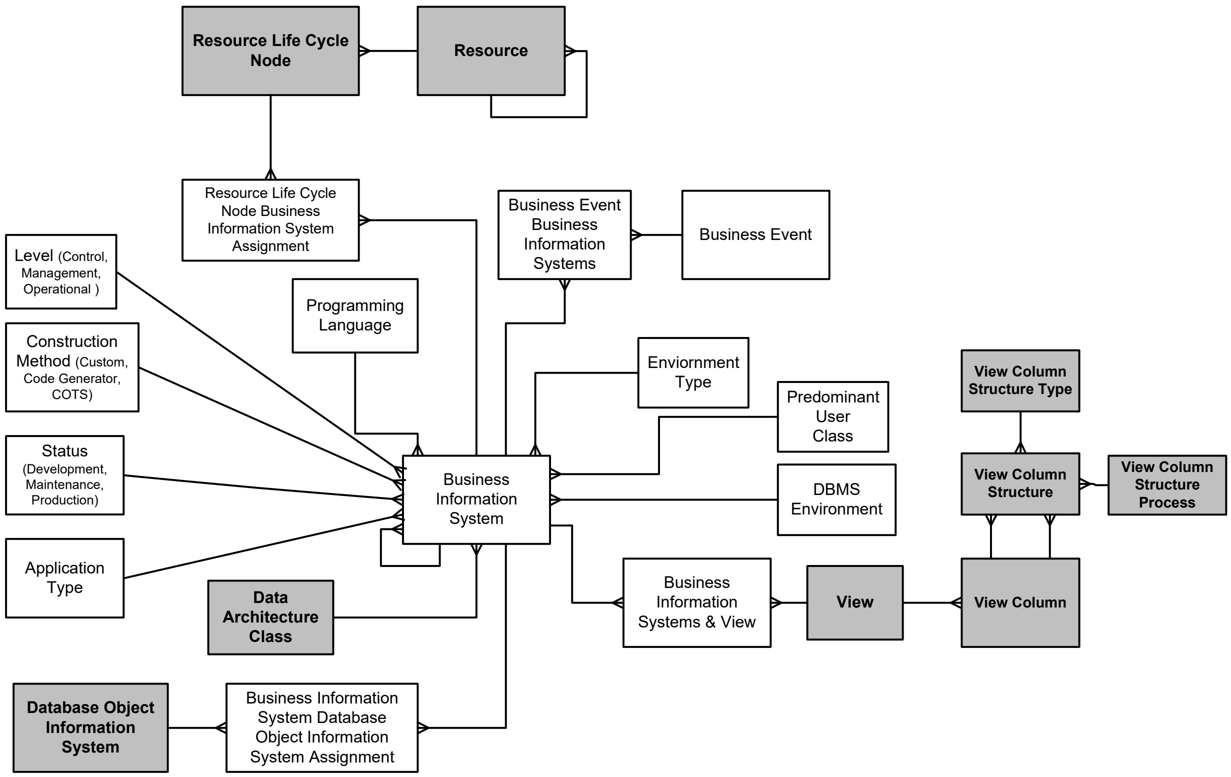


Figure 16. Business Information System Data Model.

Business Information Systems	
Table	**Description**
Application Type	An is a classification of the application such as distribution, finance, human resources and the like.
Business Event Cycle	A Business Event Cycle is a cycle during which business events occur such as financial reports, holidays, business planning and the like. A business event cycle may be simple or complex. If complex then the business event cycle actually consists of other business event cycles as represented in the business event cycle structure.
Business Event Cycle Structure	A Business Event Cycle Structure is a collection of business event cycles, for example, a Summer cycle may also consist of a End of School cycle, Back to School Cycle, Vacation Cycle, and a Holiday Cycle.
Business Event Cycle Structure Type	A Business Event Cycle Structure Type classifies a collection of Business Event Structure instances.

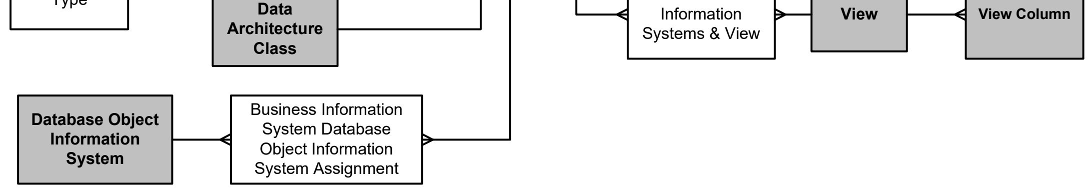


Figure 16: Business Information System Data Model.

Business Information Systems	
Table	**Description**
Business Information System	A Business Information System is a computer based information system that is being managed through the metabase. It is know by its characteristics, its operation cycles (business event and calendar), subordinate business information systems, employed databases, views, and associated resource life cycle nodes.
Business Information System and View	A Business Information System and View is an association between a view and a business information system. This then enables knowledge of the DBMS columns and DBMS tables that are accessed by the business information system.
Business Information System Database Object Information System Assignment	A Business Information System Database Object Information System Assignment records the database object information systems that are invoked by the business information system.
Calendar Cycle	A Calendar Cycle is a set of recurring calendar based dates that are of interest to the enterprise. For example, quarterly, bi-weekly, monthly, daily, and the like. Calender cycles are linked to Business Events so that the timing of business event triggering can be know.
Calendar Cycle Structure	A Calendar Cycle Structure is a collection of calendar cycles, for example, a Financial Report cycle may consist of a Second Week of the Month, the Last Friday of the month, and the first day of the quarter.
Calendar Cycle Structure Type	A Calendar Cycle Structure Type is a classification of a set of Calendar Cycle Structures.
Data Architecture Class	The Data Architecture Class is a classification of a style of database design supported by the application. The most common data architecture classes are original data capture, transaction data staging area, reference data, wholesale data warehouses and retail data warehouses (also called data marts).
DBMS Environment	The DBMS environment meta entity is intended to carry information that would indicate that the business information system is serviced by one or more than one DBMS such as Oracle or Sybase.
Environment type	The Environment type meta entity is intended to distinguish whether the business information system is executing on a desktop, server, or mainframe.
Predominant User Class	The Predominant User Class meta entity is intended to distinguish among the types of users of the business information system. Examples are executive, middle management, and line managers
Programming Language	The Programming Language meta entity distinguished the development language of the business information system such as Cobol, Fortran, etc.

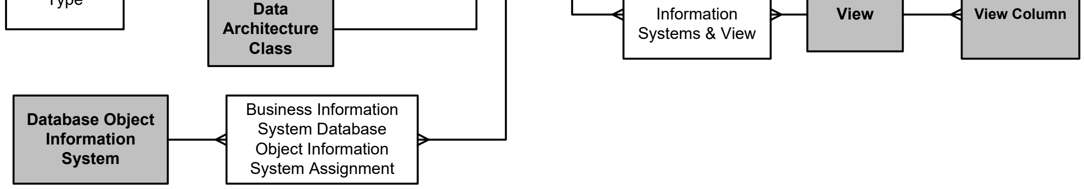


Figure 16. Business Information System Data Model.

Business Information Systems	
Table	**Description**
Construction Method	The Construction Method meta entity type of the application includes custom, COTS, and the like.
Level	The Level meta entity type of the application is its fundamental orientation such as operational, executive and control.
Production Status	The Production Status meta entity type of the application indicates whether the application is in development, test, or production.
Resource Life Cycle Node Business Information System Assignment	The Resource Life Cycle Node Business Information System Assignment meta entity represents the association of the business information system with one or more resource life cycle nodes.

8.2 Relationships to ROIs

The relationships between Business Information Systems and the ROIs is contained in Section 2 and its subsections. Specifically:

- Section 2.1, Enterprise-wide Project Management (16 to 1 ROI)
- Section 2.5, Business Information System Environments (7.7 to 1 ROI)
- Section 2.6, Business Information System Manufacturing (2.9 to 1 ROI)
- Section 2.7, Enterprise Architecture Development (14.6 to 1 ROI)

8.3 Relationship Between Business Information Systems and Knowledge Worker Framework

The entire Knowledge Worker Framework column, Business Information Systems addresses the identification through requirements, specification, implementation and operation of Business Information Systems.

8.4 Relationships Between Business Information Systems and Data Architecture Reference Model

Business Information Systems are related only to the two data models: Operational Data Model and the View. The relationship is illustrated in Figure 16

8.5 Relationships Between Business Information Systems and Resource Life Cycle Analysis

Figure 16 also illustrates the relationship between a Resource Life Cycle Node and a Business Information System.

8.6 Relationships Between Business Information Systems and Database Object Classes

Database Object Classes are not directly related to Business Information Systems. Rather, a Business Information System is directly mapped to the Operational Data Model through a view as shown in Figure 15.

8.7 Relationships Between Business Information Systems and Metabase System Modules

The Metabase System module, Business Information System supports the complete specification of Business Information Systems.

9 Metabase System

The Metabase System is a metadata management system. It is not generalized. Rather, it is specialized to meet the majority of the metadata needs of the Knowledge Worker Framework. The Metabase System is SQL-engine based with an explicit SQL schema in support of ODBC-based reporting through report writers like Crystal Reports. The functional of the Metabase System are:

- Business Event Management
- Business Information Systems
- Data Elements
- Data Integrity Rules
- Database Objects
- Documents and Forms
- Governance
- Implemented Data Models

- Information Needs Analysis
- Mission-Organization-Function-Position-Assignment
- Operational Data Models
- Project Management
- Reports Management
- Requirements Management
- Resource Life Cycle Analysis
- Specified Data Models
- Use Cases
- User Acceptance Tests
- View Data Models

Because the Metabase System operates through ODBC it can be hosted on an SQL DBMS. The Metabase System is fundamentally a client-server system. Hence its screens can range from very simple menus, lists, and forms, to very complex multiple lists per screen supported by look-ups, updates, and sophisticated error checking.

The Metabase System is inherently multi-user, and operates either on an Intranet or through the Internet.

9.1 Metabase System Modules and Descriptions

The Metabase System is a metadata management system. It is not generalized. Rather, it is specialized to meet the majority of the metadata needs of the Knowledge Worker Framework. The Metabase System is SQL-engine based with an explicit SQL schema in support of ODBC-based reporting through report writers like Crystal Reports. The functional of the Metabase System are:

- Business Event Management
- Business Information Systems
- Data Elements
- Data Integrity Rules
- Database Objects
- Documents and Forms
- Governance
- Implemented Data Models
- Information Needs Analysis
- Mission-Organization-Function-Position-Assignment
- Operational Data Models
- Project Management

- Reports Management
- Requirements Management
- Resource Life Cycle Analysis
- Screens
- Specified Data Models
- Use Cases
- User Acceptance Tests
- View Data Models

Because the Metabase System operates through ODBC it can be hosted on an SQL DBMS. The Metabase System is fundamentally a client-server system. Hence its screens can range from very simple menus, lists, and forms, to very complex multiple lists per screen supported by look-ups, updates, and sophisticated error checking.

The Metabase System is inherently multi-user, and operates either on an Intranet or through the Internet.

9.1.1 Business Event Management

Business Event Management represents the events that are invoked by a Business Function that, in turn, triggers the execution of a Business Information System. Business events exist independently or within two types of cycles: Business Event Cycle, or Calendar Cycles.

9.1.2 Business Information Systems

Business Information 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.

9.1.3 Data Elements

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 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.

Data elements are derived through the specification of concepts, conceptual value domains, data element concepts, and value domains. Additionally, the data element model supports the definition of a full complement of semantic and data use modifiers that can be allocated to data element concepts, data elements, attributes, and columns. Because of these allocations, naming and definitions can be fully automatic.

9.1.4 Database Objects

Database Object Classes represent the identification of 1) the critical data structures, 2) the processes that ensure high quality and integrity data within these data structures, 3) the value-based states represented by these data structures, and 4) the database object centric information systems that value and transform database objects from one state to the next.

Database Objects are transformed from one valid state to another in support of fulfilling the information needs of business information systems as they operation within the business functions of organizations. Database objects are encapsulated within database management systems (DBMS) so that they can be independent of any end-user environment or programming language.

9.1.5 Data Integrity Rule Specification and Binding (In Development)

Data Integrity Rule Specification and binding is in two parts: Specification and Binding. The rules are specified generically and can exist in network and/or hierarchical structured collections. The individual components of each data integrity rule consists of four classes of rules: Input, Select, Validate, and Action. Each rule is then mapped to the Implemented Data Model metadata object of column.

The Binding of each specified rule is to the highest level in the overall metadata component within which the data integrity rule is to be executed. Included are: data elements, entities, attributes, tables, columns, DBMS tables, DBMS columns, views, view columns, database object table process columns, database object table processes, compound data elements, and derived data elements.

9.1.6 Documents and Forms

Documents and Forms enable the structuring of both documents and forms into documents/forms, their sub-sections, and cells. Documents and Forms can be interrelated with network data structures. Both Document and Form section cells can be mapped to mission-organization-functions, and also to view columns.

9.1.7 Governance (In Development)

The Governance Metadata model contains no new metadata. Rather, it is collection of meta entities and relationships that support the understanding of the governance that is set into place. There are mappings between the Mission-Organization-Function-Position-Person meta entity and one or more of the following meta-entities: Data Element, Subject, Schema, Data Integrity Rule, and Business Information System through associative meta entities. These associative meta entities are each able to be mapped to a project task.

9.1.8 Implemented Data Model

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

9.1.9 Information Needs Analysis

Information Needs Analysis represents the identification, definition, and interrelationship of the information needed by various organizations in their functional accomplishment of missions and what databases and information systems provide this information.

9.1.10 Missions-Organizations-Functions-Position Assignments

The Mission, Organization, Function, and Position Assignment module represents the identification, definition, and interrelationships among the persons who, through their positions, perform functions within their organizations that accomplish enterprise missions.

9.1.10.1 Missions

Missions 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.

9.1.10.2 Organizations

Organizations represent the bureaucratic units created to perform through their functions the mission of the enterprise. Missions are intersected with Missions through the Missions-Organization table

9.1.10.3 Functions

Functions represent the procedures performed by enterprise organization groups as they achieve the various missions of the enterprise from within different enterprise organizations. Functions are intersected with the Mission-Organization tables through the Mission-Organization-Function table

9.1.10.4 Position and Person Management

Position Assignments represent the identification of both persons and position titles that are both interrelated and allocated to the various functions performed by organizations that achieve enterprise missions.

9.1.11 Operational Data Models

Operational Data Models, are the data models of databases that have been bound to a specific DBMSs. These models, which are comprised of 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.

While each operational database data model can address multiple implemented data models. Every DBMS Column should map to a parent Column. There is a many-to-many relationship between the Implemented Data Model and the Operational Data Model.

9.1.12 Project Management.

The project management model is founded on the principal that project management data is just another form of metadata, and that all the other metadata model metadata can be appropriately integrated with the project management metadata. The Project Management module is founded on project templates. These exist for: project, deliverable, and task.

Once a project template is picked, the tasks are generated and set up for full specification. That is, the adding of work environment factors, skill levels of assigned staff, and the identification of the particular Resource Life Cycle node that is the target of the project.

As the project progresses, work-accomplishments records are stored so that earned-value reporting can be generated.

9.1.13 Reports Management

The Reports Management module enables the specification of reports to be produced by the business information system. The reports consist of a report that can be contained in a network of report blocks. Each report can be hierarchically organized into report sections. Each report section can then be mapped to serve the needs of one or more Mission-Organization-Function combinations. Report sections then contain report cells which are then mapped to view columns.

9.1.14 Requirements Management

The purpose of this Metabase System module, Requirements Management, is to provide:

- Identification and description of requirements.
- Interrelationship among different requirements.
- Relationship between requirements and other metadata artifacts.

- Mapping to Business Events
- Mapping to Business Information Systems
- Mapping to DBMS Columns
- Mapping to User Acceptance Test Steps
- Mapping to Database Objects
- Mapping to User Cases
- Mapping to Data Integrity Rules
- Mapping to Resource Life Cycle Nodes
- Mapping to Mission Organization Functions

The Requirements Management module permits recording of the characteristics of the requirements associated with the enterprise. Each requirement can be described and interrelated with other requirements. Once requirements are identified and described, they can be allocated to any of the "mapping" items above. This permits enterprises to know the requirements needed by whom within the different organizations in the performance enterprise missions.

9.1.15 Resource Life Cycle Analysis

Resource Life Cycle of Analysis identifies, defines, and sets out the resources of the enterprises, the life cycles that represent their accomplishments, and the interrelationships among the different enterprise resource life cycles. Resource life cycle nodes represent the end-state data resulting from the execution of business information systems. The end-state data is represented through database object classes.

9.1.16 Screens (In Development)

Screens, once specified, are mapped to business information systems. Screens can be decomposed into hierarchical collections of screen sections. Screen sections can be mapped to one or more Use Case Events. Screen sections contain screen controls. Screen controls that are specified as columns are mapped to view columns. Screen controls are mapped to process specifications associated with each control.

9.1.17 Specified Model

Specified Data Models are the data models of concepts. These models, which are comprised of data structure components: subjects, entities, attributes, and inter-entity relationships. Relationships can span interrelate entities within multiple subjects.

Each data model should address only one concept such as a person's name, or an address, etc. These concept data models can then be templates for use in developing database models (Implemented or Operational). Every entity attribute should map to its parent Data Element. Every attribute can be allocated both semantic and data use modifiers.

9.1.18 Use Cases

The use case module enables the creation, updating, and interrelationships of detailed function model specifications. The use cases contain use case networks structures, use case events, pre conditions, post conditions, special conditions, use case facts, and use case actors.

These use case components are able to be interrelated with business information systems, database table columns, and mission organization functions, and persons functioning with positions.

9.1.19 User Acceptance Tests (In Development)

The User's Acceptance Tests are based on network structured collections of processes that are mapped to view columns. User acceptance test steps are related to one or more requirements.

9.1.20 View Model

The View data model represents the interfacing between operational data 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 processes can be specified to define any appropriate data value transformation.

9.2 Relationships Between Metabase System and ROIs

The ROIs that are directly affected by the use of the Metabase System are:

- Section 2.1, Enterprise-wide Project Management (16 to 1 ROI)
- Section 2.2, Information Systems Planning (5.9 to 1 ROI)
- Section 2.3, Data Centered Development and Management (28 to 1 ROI)
- Section 2.4, Data Model Manufacturing (8.6 to 1 ROI)
- Section 2.5, Business Information System Environments (7.7 to 1 ROI)
- Section 2.7, Enterprise Architecture Development (14.6 to 1 ROI)

9.3 Relationship Between Metabase System and Knowledge Worker Framework

The Metabase System is the computer-based software system for collecting all the work products identified in the Knowledge Worker Framework.

9.4 Relationships Between Metabase System and Data Architecture Reference Model

The Metabase's database schema is an Operational Data Model.

9.5 Relationships Between Metabase System and Business Information Systems

The Metabase Modules cited in Section 9.1 are the Business Information Systems of the Metabase System. In addition to those functional modules there is an Administrative System that manages the overall operation and security of the Metabase System's users and use.

10 Way Forward Projects

The Way Forward is accomplished through the employment of the Whitemarsh Project Management System. Figure 17 depicts a high-level architecture of the project management database tables. In the upper left, the table collection addresses the development of project templates. The result of selection of Project Templates for a project causes the complete generation of a precise set of deliverables and task collections.

For the Way Ahead sections identified below, the overall project type would be for example, Data Interoperability, and the specific work-accomplishment projects would be the individual bullet points such as "Develop Mission Model."

Within each bullet point, the work products that are, in turn, listed in the Knowledge Worker Framework table, Table5, are proper terms. For example, Mission Model.

Each of the way-ahead sections is divided into two subsections: Problem Characteristics and Solution Approach. The first identifies the likely characteristics that indicate the likelihood of the problem. The second, Solution Approach sets out the high-level action-bullets that enable a solution.

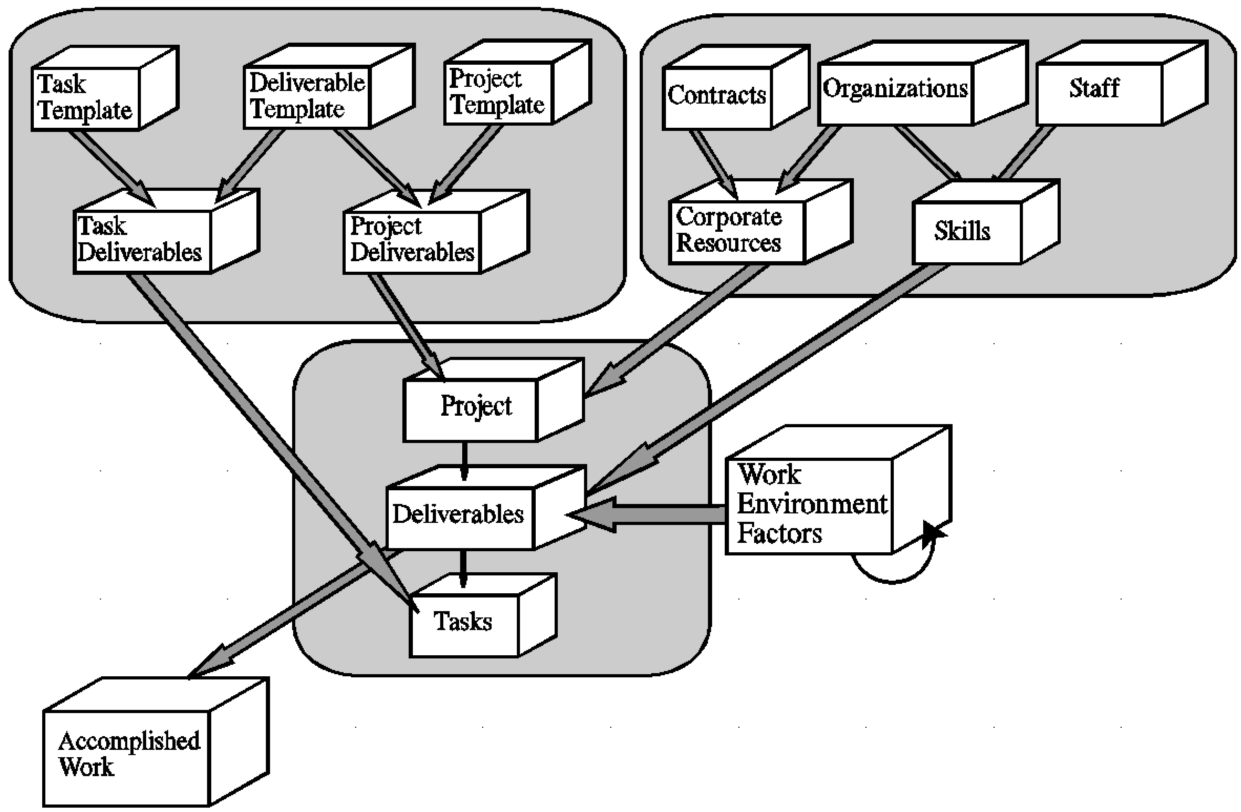


Figure 17. High Level Depiction of Tables within Whitemarsh Project Management.

10.1 Data Interoperability

10.1.1 Problem Characteristics

- A Department or Enterprise is looking for a strategy to accomplish data interoperability across a number of systems.
- There is a large quantity of un-integrated systems. Or, there is data everywhere but information no-where, or conflicts as to answers to the same question exist.
- A Department or Enterprise wants to "kick the tires" before making a commitment to a particular approach.
- A Department or Enterprise wants either to engage in discussions about data interoperability problems across multiple organizations or from multiple staff members within one or more Departments and/or projects before deciding on an approach.

10.1.2 Solution Approach

- Develop Mission Model.
- Develop Database Domains.
- Develop Database Object Class Data Structures.
- Identify related DBMS Schemas based on Database Object Class Data Structures.
- Import related Operational Database Models from related DBMS Schemas.
- Inductively build Implemented Data Models and create "local definitions."
- Inductively build Data Elements and create "local definitions."
- Construct Meta Category Value Class Types, Meta Category Value Classes, and Meta Category Values including "local definitions."
- Construct Value Domains and Value Domain Values and create "local definitions."
- Generate complete definitions for Implemented Data Model columns.
- Generate where-mapped reports from Implemented Data Models and Operational Data Models.
- Perform semantic conflicts analysis and adjust all models as appropriate.
- Posit a quantity of data semantics harmonization projects for short to long range solution to the data interoperability problems.

10.2 Interoperable Business Information Systems

10.2.1 Problem Characteristics

- An unacceptable quantity of discrete information systems and databases have been individually created over a long period of time.
- There is the exception that there is data everywhere but information is no-where.
- There are conflicting answers to the same question across the organization, such as How many Customers, What is the Gross Sales, Net Sales, count of employees, what's on back-order, etc.
- There is an unacceptable effort devoted to creating programs that do Extract-Transform-and-Load.
- There is a need to integrate databases across functional areas of the organization to support inter-departmental and inter-functional area reports and analyses.
- There is a need to integrate data from multiple procured packages such an SAP, or Oracle Applications, or those resulting from Mergers and Acquisitions.

10.2.2 Solution Approach

- Accomplish the Solution Approach as stated in Section 10.1.2.
- Identify collections of business information system that are intended to be interrelated.
- Develop Views from these business information systems and relate the DBMS Columns from the Operational Data Models created in Section 10.1.2.
- Based on Implemented Data Models mapping through to Operational Data Models and View create where-mapped to Business Information Systems reports.
- Perform semantic conflicts analysis and adjust all models as appropriate.
- Posit a quantity of business information systems data interoperability projects for short to long range solution to the interoperable business information systems problem.

10.3 Project-Level Integrated, and Non-redundant Artifacts Across the System Development Life Cycle.

10.3.1 Problem Characteristics

- Projects are needed to build data warehouses and/or a traditional business information system. These need to be planned, resourced, and started.
- Projects that have been started but are now in trouble.
- Many different project members who are building the same component (data model, process model, etc.) differently.
- No agreement on what the project deliverables should be or how they should be constructed, integrated, and made non-redundant.
- No overall methodology for each project, and across projects.
- Project members are not able to access and employ each other's work.
- There is no easy way of integrating the work products of one part of a project with another project.

10.3.2 Solution Approach

- Identify and/or construct subset of an overall Mission Model.
- Identify and/or construct subset of an overall Database Domains.
- Identify and/or construct subset of an overall Business Function Model.
- Identify and/or construct subset of an overall Business Organization Model.
- Identify and/or construct subset of an overall Business Event Model.
- Identify and/or construct subset of an overall Database Object Class Data Structures.
- Review and revise Missions through Database Object Class Data Structures.

- Identify and or construct a subset of a Specified Data Model appropriate for the Database Domains relevant for the project.
- Identify and or construct an Implemented Database Data Model appropriate for the Database Domains based on the data structures from the Specified Data Model.
- Identify and or construct Data Elements and create “local definitions” as appropriate.
- Identify and or construct Meta Category Value Class Types, Meta Category Value Classes, and Meta Category Values including “local definitions” as appropriate.
- Identify and or construct Value Domains and Value Domain Values and create “local definitions.”
- Generate complete definitions for Implemented Data Model columns.
- Review and revise Specified through Implemented Data Models.
- Generate Data Model DDL and import into Business Information System Generator.
- Tune resulting business information system to meet behavior model implied by Business Function Model.
- Review and Revise with Functional Users until requirements are quiescent.

10.4 Department-wide Integrated, and Non-redundant Artifacts Across All Departmental Functions and Their Interrelationships to Project Level Sets of Artifacts.

10.4.1 Problem Characteristics

- There exists collections of projects within one or more overall functional areas of the Department that have been unintegrated, conflicting, and redundant specifications.
- There are data warehouse projects that have been built from multiple and individual operational databases and business information system projects that result in un-integrated semantics, granularities, precision, and temporal characteristics.
- There are projects that represent the intersection of separately captured and updated data because of a consequence of multiple vendor-specific application packages that contain conflicting, redundant, and unintegrated data.
- The projects that exist solely to extract, transform, and load data from one application system database into another database are becoming too many, too large, and that consume too many resources.
- There are a number of traditional business information systems that are about to be planned, or started, or that are under development that are in trouble because of bad planning, under specified deliverables, and best-guess estimates.
- There are project staff from different projects who build the same component (data model, process model, etc.) differently because there is no standardization on format or content.

- There are disagreements on what the project deliverables should be or how they should be constructed.
- There are multiple contractors who are building databases and/or systems based on deliverables that are not standardized across all the contractors.
- There are contractors that cannot use each other's deliverables to enhance integration, to reduce conflicts or to eliminate redundancy.
- There is no comprehensive and detailed methodology for the projects that supports deliverable integration and non-redundancy. Every project is done differently. There is no overall multi-project reporting.
- There are project members from the different projects who not able to access another project staff's work, and who are not able to use other's work as a way to save time or to cause data and process integration.
- There is no easy way of integrating the work products of one part of a project with another project.
- There is a lack of Department-wide deliverables that can be used by multiple projects within and across departments.

10.4.2 Solution Approach

- Identify and/or construct enterprise-wide Mission Model.
- Identify and/or construct enterprise-wide Business Organization Model
- For the specific Department, select the relevant Missions
- Identify and/or construct subset of an overall Database Domains based on Department Missions.
- Identify and/or construct subset of an overall Business Function Model based on Department Missions.
- Develop Database Object Class Data Structures based on Department Database Domains.
- Identify related DBMS Schemas based on constructed Database Object Class Data Structures.
- Import related Operational Database Models from related DBMS Schemas.
- Inductively build Implemented Data Models and create “local definitions.”
- Inductively build Data Elements and create “local definitions.”
- Construct Meta Category Value Class Types, Meta Category Value Classes, and Meta Category Values including “local definitions.”
- Construct Value Domains and Value Domain Values and create “local definitions.”
- Generate complete definitions for Implemented Data Model columns.
- Generate where-mapped reports from Implemented Data Models and Operational Data Models.
- Perform semantic conflicts analysis and adjust all models as appropriate.

- Identify collections of operative business information system within the Department.
- Develop Views from these business information systems and relate the DBMS Columns from the Operational Data Models.
- Based on Implemented Data Models mapping through to Operational Data Models and View create where-mapped to Business Information Systems reports.
- Perform Department-based data element, specified, implemented and operational data semantic conflict analyses and construct appropriate reports.
- Perform Department-based Business Information System process conflict analysis, redundancy analysis, and lack of interoperability analyses and construct appropriate reports.

10.5 Enterprise-wide Integrated, and Non-redundant Artifacts Across All Enterprise Functions and Their Interrelationships to Department Level and, in Turn, Relationships to Project Related Artifacts

10.5.1 Problem Characteristics

- There is a need to integrate databases and business information systems across multiple departments functions in support of multi-department data warehouses to eliminate data redundancy and conflicts.
- There is a need to develop standard Enterprise-wide data models that contain constructs to be used by different Departments and projects within those Departments to eliminate conflicting specifications unnecessary development, and reduce maintenance.
- There is a need for Enterprise "data elements," semantics, and value domains such as Gender, and Project Codes to accelerate database design, business information system development, to reduce redundant development, and to speed change impact analysis and maintenance.
- There is a need to standardize Master data across business-essential strategic data structures to commonly employ this data across the enterprise to maximize common data and data structures reuse within databases and business information systems.
- There is a need for a metadata management system to capture and store Master Data specifications that can be used within and across Departments and projects to make standardized data and process structures
- There is a need to "see across" metadata collected and employed from multiple projects from different Departments and in turn, projects across the enterprise to support the re-use of specifications that, in turn, makes the development of databases and business information systems faster, and more transparent to end-users.
- When multiple contractors are involved, there is a need to have the databases and/or systems based on deliverables that are standardized, non-conflicting, integrated, and

non-redundant so that the most expensive component of IT, people, can be reduced and optimized.
- There is a need to reduce and/or eliminate the quantity of resources focused on the development and maintenance of data extract, transformation and loading projects that exist solely as a consequence of integrating separately captured and updated data from different databases. The quantity of these projects becomes reduced whenever there is an increase in the common use of Enterprise-wide data and process semantics.
- There is a need to harmonize standardized data structures and/or Enterprise Data Elements across Departments and projects so that development and maintenance is reduced and so that the quality of end-user use is increased.
- The is a need for project staff from different projects to cooperate and without conflict on the building of the same component (data model, process model, etc.) to eliminate conflicting and redundant development, and to reduced the quantity and duration of maintenance.
- There is a need to eliminate disagreement on project deliverables, their specifications and construction so that work can proceed more quickly with shorter reviews and easier maintenance.
- There is a need for a detailed and comprehensive methodology for the projects to eliminate unnecessary differences, to enable multi-project reporting, to increase re-use of commonly developed components, and to be able to re-deploy staff resources without significant retraining.
- There is a need for project members from the different projects to access each other's work to reduce redundantly developed artifacts, to cause data and process integration, to increase overall quality and to reduce individual project life cycle time.

10.5.2 Solution Approach

- Identify and/or construct enterprise-wide Mission Model.
- Identify and/or construct enterprise-wide Business Organization Model
- Identify and/or construct subset of an overall Database Domains
- Identify and/or construct subset of an overall Business Function Model
- Develop Database Object Class Data Structures based on Database Domains.
- Identify related DBMS Schemas based on constructed Database Object Class Data Structures.
- Import related Operational Database Models from related DBMS Schemas.
- Inductively build Implemented Data Models and create “local definitions.”
- Inductively build Data Elements and create “local definitions.”
- Construct Meta Category Value Class Types, Meta Category Value Classes, and Meta Category Values including “local definitions.”

- Construct Value Domains and Value Domain Values and create “local definitions.”
- Generate complete definitions for Implemented Data Model columns.
- Generate where-mapped reports from Implemented Data Models and Operational Data Models.
- Perform semantic conflicts analysis and adjust all models as appropriate.
- Identify collections of operative business information systems.
- Develop Views from these business information systems and relate the DBMS Columns from the Operational Data Models.
- Based on Implemented Data Models mapping through to Operational Data Models and View create where-mapped to Business Information Systems reports.
- Perform data element, specified, implemented and operational data semantic conflict analyses and construct appropriate reports.
- Perform Business Information System process conflict analysis, redundancy analysis, and lack of interoperability analyses and construct appropriate reports.

10.6 Community of Interest Data Interoperability Specification Artifacts

10.6.1 Problem Characteristics

- There is a need for inter-departmental projects and supporting organizations to engineer databases and business information systems /or database data exchanges across functional communities of interest across Departments in support of reducing redundancy and increasing both data and process integration without an increase of data or process components.
- There is a need to collect and store the Community of Interest metadata in the same rigorous fashion as it does for project and/or Department metadata so that these Community of Interest artifacts can be easily integrated and incorporated with other data and process specifications.
- There is a need to make the Community of Interest metadata permanent and resident at the Enterprise level because it spans multiple Departments or because it spans multiple projects within a particular Department and it needs to be conserved as opposed to being redundantly specified, developed, and maintained.
- There is a need to support queries, extracts and reports across all the organizations involved in Communities of Interest and their use of shared metadata.

10.6.2 Solution Approach

- Identify and/or construct enterprise-wide Mission Model.
- Identify and/or construct enterprise-wide Business Organization Model
- Identify and describe the community of interest for which the solution is to be accomplished.
- Identify the relevant Missions associated with the Community of Interest.
- Identify the specific Departments involved with the Community of Interest.
- Identify and/or construct subset of an overall Database Domains based on the Community of Interest Department's Missions.
- Identify and/or construct subset of an overall Business Function Model based on the Community of Interest Department's Missions.
- Develop Database Object Class Data Structures based on the Community of Interest Department's Database Domains.
- Identify related DBMS Schemas based on constructed Database Object Class Data Structures.
- Import related Operational Database Models from related DBMS Schemas.
- Inductively build Implemented Data Models and create "local definitions."
- Inductively build Data Elements and create "local definitions."
- Construct Meta Category Value Class Types, Meta Category Value Classes, and Meta Category Values including "local definitions."
- Construct Value Domains and Value Domain Values and create "local definitions."
- Generate complete definitions for Implemented Data Model columns.
- Generate where-mapped reports from Implemented Data Models and Operational Data Models.
- Perform semantic conflicts analysis and adjust all models as appropriate.
- Identify collections of operative business information system within the Community of Interest's Departments.
- Develop Views from these business information systems and relate the DBMS Columns from the Operational Data Models.
- Based on Implemented Data Models mapping through to Operational Data Models and View create where-mapped to Business Information Systems reports.
- Perform Community of Interest's Department-based data element, specified, implemented and operational data semantic conflict analyses and construct appropriate reports.
- Perform Community of Interest's Department-based Business Information System process conflict analysis, redundancy analysis, and lack of interoperability analyses and construct appropriate reports.

10.7 Enterprise and/or Department Information Systems Plan

10.7.1 Problem Characteristics

- There is a need to have an information systems plan that identifies what to build and/or maintain, when to build and/or maintain, and the overall costs to build and/or maintain databases and business information systems.
- There is a need for the Information Systems Plan to be founded upon the same metadata that is created, developed and maintained within the metadata management system.
- There is a need to have the Information Systems Plan easily reflect changes in both what is built and/or maintained, and in the sequence of what is built and/or maintained as these changes take place during the evolution of the metadata management system data.
- There is a need to have an overall information systems plan defensible based on business needs with respect to enterprise missions, organizations, functions, and business events.
- There is a need to have an Information Systems Plan call for the building of the right components at the right time and in the right sequence to maximize data and process interoperability, minimize or eliminate redundancy, and to maximize re-use of already created information technology components.

10.7.2 Solution Approach

- Develop Mission Model.
- Develop Database Domains.
- Develop Database Object Classes.
- Develop Database Object Class Data Structures.
- Develop Enterprise-wide set of Resources
- Develop Resource Hierarchies..
- Develop Resource Life Cycles and their nodes.
- Allocate precedence vectors among Resource Life Cycle Nodes..
- Identify existing Business Information Systems.
- Allocate Business Information Systems to the Resource Life Cycle Nodes..
- Allocate Database Object Classes to Resource Life Cycle Nodes.
- Identify Resource Life Cycle Nodes for which IT projects need to be developed.
- Identify Resource Life Cycle Nodes that contain mappings from single Business Information Systems.
- Identify Resource Life Cycle Nodes that contain mappings from single Database Object Classes.
- Refactor Resource Life Cycle Nodes so that Business Information Systems are employed only within Resource Life Cycle Nodes.

- Refactor Resource Life Cycle Nodes so that Database Objects are created only through only one Resource Life Cycle Node..
- Create IT Project specifications for the accomplishment of Resource Life Cycle Nodes.
- Generate Deliverables and Tasks for Resource Life Cycle Node based projects.
- Resource-load IT Projects through allocation of Staff and Work Environment Factors.
- Create the overall Information Systems Plan.
- Present, review and revise the Information Systems Plan.
- Execute and adjust the Information Systems Plan over time.

10.8 Metadata Management System Requirements

10.8.1 Problem Characteristics

- There is a need of an Enterprise and/or large Department to have a metadata management system that collects, stores, evolves, and reports the complete set of metadata artifacts required for the complete understanding of the missions, organizations, functions, databases, business information systems, and other critical metadata artifacts necessary to understand, plan, execute, and evolve the enterprise.
- There is a need for a formal needs analysis and requirements determination prior to issuing a Request for Proposals (RFP) for, or for a custom-engineered Enterprise-wide repository so that good proposals and/or development are received and are able to be evaluated.
- There is a need to prototype a repository within the scope of an individual project, across multiple projects within a Department, and possibly across the Enterprise before proceeding so that realistic efforts can be determined and set against determined benefits.
- The need to accomplish the implementation of a metadata management system with the same rigor and precision as is done for quality database projects so that there can be a high probability of success given a solid set of requirements and project plans as evidenced in a rigorously defined set of deliverables, a highly engineered methodology, time line, and resources.

10.8.2 Solution Approach

- Acquire Metabase System
- Establish Metabase System infrastructure software environment
- Establish Metabase System infrastructure software environment
- Install Metabase System
- Install Metabase System supporting DBMS

- Establish connections to metabase system databases
- Identify and establish metabase system databases
- Bootstrap load all default data into newly established metabase system databases
- Identify, name, and establish metabase system database users and permissions
- Determine that all metabase system modules are operational
- Create cross reference between and among all organizational work product deliverables and metabase system work products
- Validate that all metabase system use procedures are operational
- Conduct all metabase system training
- Commence metabase system operations

10.9 Enterprise Architectures

10.9.1 Problem Characteristics

- There is a need to better understand the entirety of the enterprise.
- There is a recognition that there are duplications, conflicts and even competition of efforts across multiple organizations within the enterprise.
- There is a recognition that there are duplicate staffs across the enterprise that, if consolidated could result in savings of scarce enterprise resources.
- There is a recognition that there are duplications, conflicts and even competition of product lines across multiple organizations within the enterprise.
- There is a recognition that there are duplications, conflicts and even competition of infrastructure organizations, staff, logistics and other functional areas across multiple organizations within the enterprise.
- There is a recognition that there are duplications, conflicts and even competition of Information Technology efforts to develop and manage databases and business information systems across multiple organizations within the enterprise.
- There is a recognition that there are duplications, conflicts among existing databases in terms of data structures, and DBMS Column semantics.
- There is a recognition that there are duplications, conflicts and even competition of business information systems across multiple organizations within the enterprise.
- There is a recognition that there is no overall information systems plan for the development of an integrated and interoperable databases and business information systems across the enterprise.

10.9.2 Solution Approach

- Develop Enterprise's Architecture
  - Identify and/or construct an enterprise-wide Mission Model.
  - Develop Database Domains
  - Identify and/or construct an enterprise-wide Business Organization Model
  - Identify and/or construct an enterprise-wide Business Function Model.
  - Identify and/or construct an enterprise-wide Position and Staff Model

- Develop Database Object Classes
  - ♦ Develop Database Object Class
  - ♦ Develop Database Object Class Data Structure.

- Develop Resource Life Cycles
  - ♦ Develop Enterprise-wide set of Resources
  - ♦ Develop Resource Hierarchies.
  - ♦ Develop Resource Life Cycles and their nodes.
  - ♦ Allocate precedence vectors among Resource Life Cycle Nodes.
  - ♦ Identify existing Business Information Systems.
  - ♦ Allocate Business Information Systems to the Resource Life Cycle Nodes..
  - ♦ Allocate Database Object Classes to Resource Life Cycle Nodes.

- Develop Data Models
  - ♦ Identify related DBMS Schemas based on constructed Database Object Class Data Structures.
  - ♦ Import related Operational Database Models from related DBMS Schemas.
  - ♦ Inductively build Implemented Data Models and create "local definitions."
  - ♦ Inductively build Data Elements and create "local definitions."
  - ♦ Construct Meta Category Value Class Types, Meta Category Value Classes, and Meta Category Values including "local definitions."
  - ♦ Construct Value Domains and Value Domain Values and create "local definitions."
  - ♦ Generate complete definitions for Implemented Data Model columns.
  - ♦ Generate where-mapped reports from Implemented Data Models and Operational Data Models.

- Develop Information Systems Plans
  - ♦ Identify existing Business Information Systems.
  - ♦ Allocate Business Information Systems to the Resource Life Cycle Nodes..
  - ♦ Allocate Database Object Classes to Resource Life Cycle Nodes.

- Identify Resource Life Cycle Nodes for which IT projects need to be developed.
- Identify Resource Life Cycle Nodes that contain mappings from single Business Information Systems.
- Identify Resource Life Cycle Nodes that contain mappings from single Database Object Classes.
- Refactor Resource Life Cycle Nodes so that Business Information Systems are employed only within Resource Life Cycle Nodes.
- Refactor Resource Life Cycle Nodes so that Database Objects are created only through only one Resource Life Cycle Node..
- Create IT Project specifications for the accomplishment of Resource Life Cycle Nodes.
- Generate Deliverables and Tasks for Resource Life Cycle Node based projects.
- Resource-load IT Projects through allocation of Staff and Work Environment Factors.
- Create the overall Information Systems Plan.

10.10 Enterprise's Architecture

10.10.1 Problem Characteristics

- There is a need to understand the entirety of the enterprise.
- There are duplications, conflicts and even competition of efforts across multiple organizations within the enterprise.
- There are duplicate staffs across the enterprise that, if consolidated could result in savings of scarce enterprise resources.
- There are duplications, conflicts and even competition of product lines across multiple organizations within the enterprise.
- There are duplications, conflicts and even competition of infrastructure organizations, staff, logistics and other functional areas across multiple organizations within the enterprise.
- There are duplications, conflicts and even competition of Information Technology efforts to develop and manage databases and business information systems across multiple organizations within the enterprise.

10.10.2 Solution Approach

- Identify and/or construct an enterprise-wide Mission Model.
- Identify and/or construct an enterprise-wide Business Organization Model

- Identify and/or construct an enterprise-wide Business Function Model.
- Identify and/or construct an enterprise-wide Position and Staff Model
- Develop Database Domains.
- Develop Database Object Classes
- Develop Database Object Classes.
- Develop Database Object Class Data Structures.

10.11 Enterprise-Wide Semantics

10.11.1 Problem Characteristics

- There is a recognition that there are duplications, conflicts among existing databases in terms of data structures across operational data models.
- There is no overall data element model across the enterprise
- There is no overall specified data model across the enterprise
- There are not taxonomies in support of Meta Category Value Class Types, Meta Category Value Classes, and Meta Category Values including "local definitions" as appropriate.
- There are not taxonomies in support of Value Domains and Value Domain Values and create "local definitions."
- There are conflicting names and definitions for DBMS columns that mean the same but have differently named and defined columns.
- There are same named DBMS columns but with different definitions.

10.11.2 Solution Approach

- Identify DBMS Schemas associated with enterprise-wide business information systems.
- Import related Operational Database Models from identified DBMS Schemas.
- Inductively build Implemented Data Models and create "local definitions."
- Inductively build Data Elements and create "local definitions."
- Construct Meta Category Value Class Types, Meta Category Value Classes, and Meta Category Values including "local definitions."
- Construct Value Domains and Value Domain Values and create "local definitions."
- Generate complete definitions for Implemented Data Model columns.
- Generate where-mapped reports from Implemented Data Models and Operational Data Models.
- Perform semantic conflicts analysis and adjust all models as appropriate.

11 References

Topic or Title	URL
Business Information Systems Life Cycle Costs	http://wiscorp.com/sp/sp26.pdf
Challenge for Business Information Systems Success	http://wiscorp.com/sp/sp19.pdf
Comparative Analysis of Business Information Systems Development Methodologies	http://wiscorp.com/sp/sp27.pdf
Data Architecture Reference Model	http://wiscorp.com/sp/sp28.pdf
Data is Executed Policy	http://wiscorp.com/sp/sp02.pdf
Data Management Independent Verification and Validation	http://wiscorp.com/sp/sp24.pdf
Data Modeler Architecture and Concept of Operations	http://www.wiscorp.com/metabase/MetabaseDataModelerArchitectureandConceptofOperations.zip
Data Models, the Center of the Business Information Systems Universe	http://wiscorp.com/sp/sp20.pdf
Database Objects Overview	http://wiscorp.com/sp/sp23.pdf
Engineering and Managing Information Systems Plans	http://wiscorp.com/sp/sp11.pdf
Enterprise Architectures	http://wiscorp.com/sp/sp08.pdf
Inductive Enterprise Architecture Development	http://wiscorp.com/sp/sp25.pdf
Managing Data Names, Value Domains, Abbreviations and Definitions	http://www.wiscorp.com/sp/sp03.pdf
Managing the Meaning of Data	http://wiscorp.com/sp/sp01.pdf
Messages that Convey the Value	http://www.wiscorp.com/messagesthatconveythevalue.html
Metabase Reverse and Forward Engineering Users Guide	http://www.wiscorp.com/metabase/MetabaseReverseAndForwardEngineeringUsersGuide.zip
Metabase System and the Knowledge Worker Framework	http://www.wiscorp.com/metabase/MetabaseRationaleKnowledgeWorkerFrameworkandMetaModels.pdf
Metabase System Overview	http://www.wiscorp.com/metabase/MetabaseOverview.zip

Topic or Title	URL
Metabase System User Guides	http://www.wiscorp.com/metabase/MetabaseUserGuides.zip
Metadata, The Information Technology Intellectual Property of the Enterprise	http://wiscorp.com/sp/sp21.pdf
Modeling Data and Designing Databases	http://wiscorp.com/sp/sp06.pdf
Resource Life Cycle Management: The Enterprise Architecture Component Integrator	http://wiscorp.com/sp/sp10.pdf
RFP Development	http://wiscorp.com/sp/sp18.pdf
ROI: Business Information Systems Environment	http://www.wiscorp.com/roi_businessinformationsystemenvironments.html
ROI: Business Information Systems Manufacturing	http://www.wiscorp.com/roi_rusinessinformationsystemsmanufacturing.html
ROI Data Centered Development and Management	http://www.wiscorp.com/roi_datacentereddevelopmentandmanagement.html
ROI: Data Model Manufacturing	http://www.wiscorp.com/roi_datamodelmanufacturing.html
ROI: Enterprise Architecture Development	http://www.wiscorp.com/roi_enterprisearchitecturedevelopment.html
ROI: Enterprise Wide Project Management	http://www.wiscorp.com/roi_enterprisewideprojectmanagement.html
ROI: Information Systems Planning	http://www.wiscorp.com/roi_informationsystemsplanning.html
Whitemarsh Project Management	http://wiscorp.com/WhitemarshProjectManagementArchitectureAndConceptOfOperations.pdf