Whitemarsh
Information Systems Corporation

Data Models: The Center of the Business Information Systems Universe

Whitemarsh Information Systems Corporation
2008 Althea Lane
Bowie, Maryland 20716
Tele: 301-249-1142
Email: Whitemarsh@wiscorp.com
Web: www.wiscorp.com

Table of Contents

1. Objective

The objective of this paper is to present a detailing of an aspect of the June TDAN paper, *Challenge for Business Information System Success*. That paper is located at:

http://www.tdan.com/view-articles/15265

The TDAN paper's Figure 4 contains a collection of interrelated deliverables that are created during a normal business information system's development or evolution life cycle. For the purposes of this paper, these deliverables are limited to just data models and a collection of work products more precisely tied to specific business information systems. Excluded therefore are for example, hardware, network, and system software. Table 8 from the TDAN paper presents a cross between the data models and the other business information system work products.

Figure 4 and Table 8 clearly imply that data models are a key component of any business information system development effort. Properly created and deployed, these deliverables can be the cause for increased productivity, increased quality, decreased cost and decreased risk. Explaining this is the objective of this short paper. Figure 4 and Table 8 are repeated here as Figure 2 and Table 3. Diagrams of all the work product deliverables identified in Table 2 are rendered as third normal form data models and are located at:

http://www.wiscorp.com/pub/MetadataModels.pdf

2. Topics Covered

The topics in this paper include:

- The Problem at Hand
- Data Architecture Reference Model
- Business Information System Work Products
- Conclusions

3. The Problem at Hand

What prompted this paper was an email to the data management discussion group which explained, in a response to the question, “Why data modeling?” that data modeling lies at the center of three universes.

- First, as a way to validate requirements through prototyping prior to a First-Implementation-Is-Correct effort.

- Second, as a way to integrate, make non redundant, and make interoperable the full set of work products on the System Development Life Cycle (SDLC).

- Third, as the specification of persistent enterprise policy.

The first two universes are the subject of this paper. The way that the question is answered is through a demonstration of how a data model is manufactured using pre developed parts, exporting the data model into a business information system generator (e.g., Clarion from SoftVelocity (www.SoftVelocity.com)), and several reviews and iterations that finally result in a comprehensive specification that can be successfully implemented into a production system that is correct the first time it is implemented.

During subsequent discussion group iterations, it was stated that to achieve the second universe, deliverable integration, interoperability and non redundancy requires an enterprise-level and comprehensive metadata management system and database that can accomplish any of the following (and many more) creation, reporting, and evolution scenarios:

- Going from business data elements to entity collections of concepts to defined database model to SQL schema to code generation.

- Selecting one business data element and show all the SQL Views that employ it.

- Selecting one business data element and show all the use-case business facts from pre-, post-, and special-conditions.

- Show all the different mappings among equally valid value domains that are both current and across time.

- Show wherever database table columns are deployed within user-graphical interfaces, business information systems, views, and the like.

- Show end to end traceability from requirements across and within related design and implementation artifacts to actual deployments within databases, and/or business information systems.

- Show how artifacts in use-cases are employed in multiple logical and physical database models, and in which business information systems cause values to be created, updated, or reported.

The response to the discussion concluded with the assertion that unless and until we can actually demonstrate accomplishment, for example, the scenarios above, organizations are unlikely to put data modeling work on their critical paths. To be welcomed to the critical path, we have to enable the project to be accomplished faster, cheaper, higher quality, and lower risk.

To accomplish this, we need a comprehensive metadata management system that collects, stores, interrelates, and integrates all IT work products. Data Modelers are uniquely situated to do this as data models are at the "intersection" of almost all business information system deliverables. If we, however, just focus on data models, we're missing out on 99.44% of all our productive and contributory use.

4. Data Architecture Reference Model

Figure 1 presents an overall data architecture reference model. An overview of each data model is provided in Table 1. A thorough presentation of the Data Architecture Reference model is contained in the book, *Data Modeler Architecture and Concept of Operations*, that can be downloaded from: http://www.wiscorp.com/metabase_demo.html

Figure 1 shows that the core of the data management environment is the collection of data model layers. Within this environment there are six distinct data models. For example, 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, there may be concept data models, for example, for students, schools, organizations, or addresses. These concept data models can be deployed in one or more implemented or logical database models, which, in turn are operationally deployed in one or more DBMS specific operational or physical data models.

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

Figure 1 shows a left-side set of one-to-many relationships going "down." This relationship supports two meanings. The first is the mapping of an individual

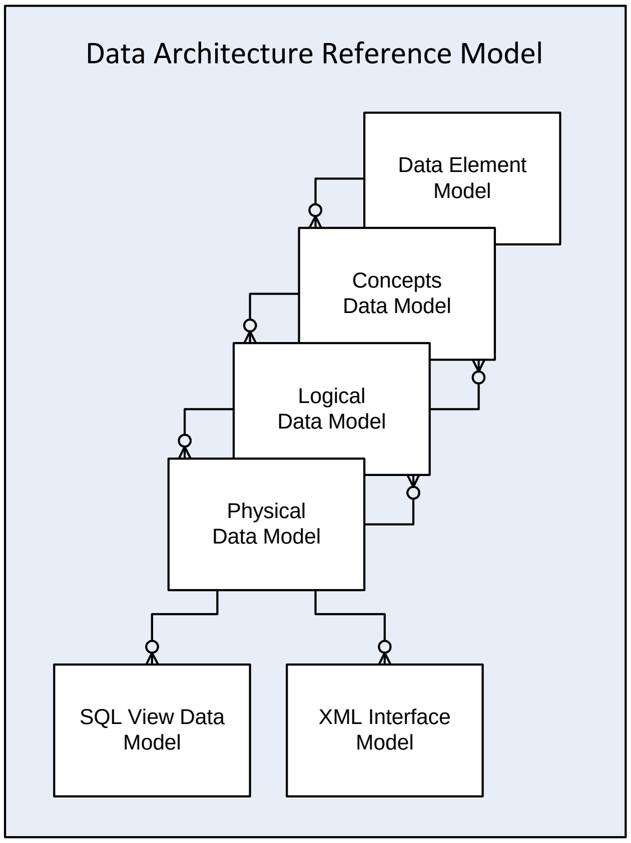

Figure 1. Data Architecture Reference Model

component of a model, and the second is the mapping of a whole collection from within a data 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 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 1, there is also a set of one-to-many relationships. This set, like the left-side one-to-many relationships have 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.

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 Logical Data Model while another collection from the same Physical Data Model can be mapped to a different collection within a different Logical Data Model. Hence, the collections can be seen as "from" one Physical Data Model to zero, one, or more Logical Data Models.

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 Data Elements and Value Domains. Semantic and data use modifiers can be assigned to every data element concept and data element.
Specified or Concepts 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 concept data models can be templates for use in

Data Model Class	Description
	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
Implemented or Logical Data Model	Implemented Data 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 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. 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 Data Model. Key components are schemas, tables, columns, and relationships
Operational or Physical Data Models	Operational, or Physical Data 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 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. Key components are DBMS schemas, DBMS tables, DBMS columns, and DBMS Relationships.
View Data Model	The View data models represent 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. Key components are Views, View columns, and view-column interrelationships.
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.

Table 1. Data Model Layers within the Data Architecture Reference Model.

5. Business Information System Work Products

We are all familiar with the collection of work products that are identified, engineered, created, evolved and maintained during the life cycle of business information systems. While there can be an endless array of names for some of these work products, they generally fall into the work product categories that are listed in column one of Table 2. A depiction of the interrelationships of the data models and of the work products is presented in Figure 2.

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				✔	✔	✔
External Quality Standards	✔	✔	✔	✔	✔	✔
Information Needs	✔		✔			
Mission Organization Functions			✔			
Resource Life Cycles			✔			
Use Cases			✔	✔	✔	✔
User Acceptance Tests				✔	✔	✔
Value Domains and Management	✔	✔	✔	✔	✔	✔
Work Breakdown Structure (WBS)	✔	✔	✔	✔	✔	✔

Table 2. Cross reference between work products and data architecture reference model component.

An explanation of these work products and how they involved one or more of the data model layers is presented in Table 3. It presents a name and brief description of each work product and also the data model components that are created, evolved, and possibly affected as a consequence of developing and/or evolving business information system deliverables.

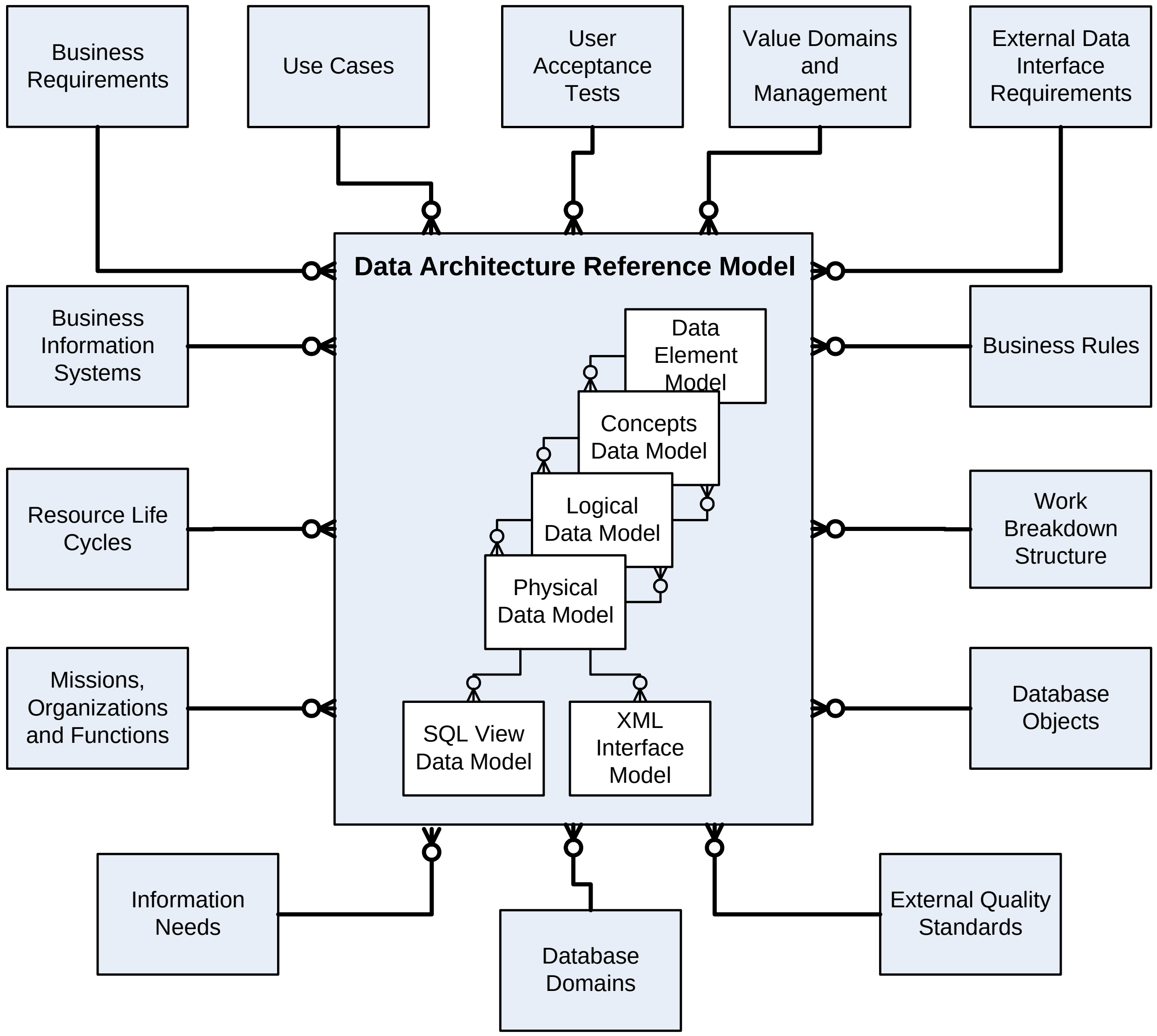

Figure 2. A high-level depiction of the interrelationship between the relationship between data models and business information system work products.

Business Information System Life Cycle Work Products		
Component	**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 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	Subjects, Entities Attributes Data Elements Database Domains Database Objects Schema

Business Information System Life Cycle Work Products		
Component	**Description**	**Data Model Components**
	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.	Tables Columns DBMS Schema DBMS Tables DBMS Columns
External Interface Data Requirements	External 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 interface and take appropriate action against the database. Such actions can be to insert, delete, or modify database records.	DBMS Schemas DBMS Tables DBMS Columns DBMS Relationships 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 Software 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

Business Information System Life Cycle Work Products		
Component	**Description**	**Data Model Components**
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 business requirements, deployments of use-cases in software and hardware, external interfaces, value domains, business rules, and WBS.	Schemas Tables Columns Relationships Value Domains DBMS Schemas DBMS Tables DBMS Columns DBMS Relationships Views View Columns 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 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	Resource Life Cycle of Analysis identifies, defines, and sets out the	Schemas

Business Information System Life Cycle Work Products		
Component	**Description**	**Data Model Components**
Cycle Analyses	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.	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 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.	Schemas Tables Columns Relationships
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 3. Data Management Components and affected Data Models.

6. Conclusions

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

- Comprehensive Business Information System Development Life Cycles have work items such as data models and other work products identified, described, and through process-oriented work breakdown structures created and evolved during the business information system life cycle. These all need to be integrated, non redundant, and interoperable across the entire time span of the creation, implementation, and evolution of the business information system.

- The time and effort expended on the development of work items commonly include the creation of data model and work product specifications, and some level of interrelationships. Typically, however, this effort does not result in work item integration, non redundancy and interoperability. Without integration, non redundancy, and interoperability, significant time and resources are wasted accomplished unnecessary research, conflict resolution, and ad-hoc repairs to all these deliverables. And when done these repairs often are inconsistent leading to more and more problems later on.

- A comprehensive, enterprise-engineered metadata management system enables work items to be captured once, made non redundant, and integrated and also which makes all these deliverables reportable within and across projects, functions, and the enterprise. This results in higher productivity, higher quality, lowered cost and lowered risk.

- Data model work items include data elements, concepts data models, logical and physical database models, along with view and XML data models enable the interrelationship of the key business information system life cycle work products.

- The cost and effort required to create the various work item models in a comprehensive metadata management system that is supported by a quality methodology, seminars, workshops, and is available to all appropriate users of business information system projects is negative. That’s because the resources to create, make non redundant, integrated, and interoperable these data models and business information system work products are always fewer that the opposite approach, which sadly is all too common.

7. References

The following references to Whitemarsh materials provide a more detailed exposition practical application of the significant content of this paper.

The following documents are available free from the Whitemarsh website:

Paper	URL

The following documents are available for Whitemarsh Website Members. The URLs that follow provide descriptions of the pages. Members should log in and proceed to the appropriate page, e.g., Enterprise Database, find the book, paper, or course and perform the download.

Paper	URL