



Whitemarsh
Information Systems Corporation

Enterprise Architectures

Michael M. Gorman

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

Designations used by companies to distinguish their products are often claimed as trademarks. In all instances where Whitemarsh Press is aware of a claim, the product names appear in initial capital or all capital letters. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

©2007 by Whitemarsh Business Information Systems Corporation
All rights reserved.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If legal advice or other expert assistance is required, the services of a competent professional person should be sought. FROM A DECLARATION OF PRINCIPLES JOINTLY ADOPTED BY A COMMITTEE OF THE AMERICAN BAR ASSOCIATION AND A COMMITTEE OF PUBLISHERS.

Reproduction or translation of any part of this work beyond that permitted by section 107 or 108 of the 1976 United States Copyright Act without the permission of the copyright holder is unlawful. Requests for permission or further information should be addressed to Whitemarsh Press.

ISBN 978-0-9789968-2-6

Printed in the United States of America

Table of Contents

Preface	xiii
1. Enterprise Architectures	1
1.1 What are Enterprise Architectures?	2
1.2 Five Contained Architectures	3
1.3 Remainder of this Book	7
1.4 Questions and Exercises	9
2. Components of an Architecture	11
2.1 Engineering an Architecture Process	12
2.2 Architecture Process Integration	14
2.3 One Architecture for all Reasons	17
2.4 Integrated Architectures	18
2.5 Architecture Components Summary	20
2.6 Questions and Exercises	21
3. Knowledge Worker Framework	23
3.1 What is a Knowledge Worker?	23
3.2 The Killer Nature of Stove-pipes	25
3.3 Knowledge Worker Framework	27
3.4 Knowledge Worker Framework Rows	30
3.4.1 Scope Row	34
3.4.2 Business Row	35
3.4.3 System Row	36
3.4.4 Technology Row	38
3.4.5 Deployment Row	39
3.4.6 Operations Row	40
3.5 Knowledge Worker Framework Columns	40
3.5.1 Mission Column	41
3.5.2 Database Object Class	42
3.5.3 Business Information System Column	45
3.5.4 Business Event Column	48
3.5.5 Business Function Column	49
3.5.6 Organization Column	52
3.6 Knowledge Worker Framework Summary	54
3.7 Questions and Exercises	56

4. Enterprise's Architecture	59
4.1 The Enterprise's Architecture Scope	59
4.2 The Enterprise's Architecture Components	60
4.3 Enterprise's Architecture Process Flow	60
4.4 Enterprise's Architecture Summary	65
4.5 Questions and Exercises	66
5. Database Object Class Architecture	67
5.1 Contrasting Database Object Class with Other Object Classes	68
5.2 The Business Case for Database Object Classes	69
5.3 Database Object Class Examples	72
5.4 Origin of Database Object Classes	74
5.5 DBMS Technology Need for Database Object Classes	75
5.6 DBMS Encapsulation Requirement	77
5.7 Four Components of a Database Object Class	78
5.8 Database Object Class Components	82
5.9 SQL DBMS Alternative Implementations of Database Object Classes	86
5.10 Database Object Class Summary	87
5.11 Questions and Exercises	88
6. Data Architecture	89
6.1 Database Architecture Classes	90
6.2 Data Model Generalization Levels	92
6.2.1 Data Element Models	95
6.2.2 Specified Data Models.	98
6.2.3 Implemented Data Models.	102
6.2.4 Operational Data Models.	105
6.2.5 View Data Models.	109
6.3 Data Model Integration and Reuse	110
6.4 Top-down versus Bottom-up Modeling	113
6.5 Division of Effort	116
6.6 Data Architecture Summary	119
6.7 Questions and Exercises	122
7. Resource Life Cycle Analysis	125
7.1 Resource Life Cycle Analysis Objectives	127
7.2 Building Resource Life Cycles	128

Table of Contents

7.2.1	Resource Identification	128
7.2.2	Resource Life Cycles	129
7.2.3	Resource Life Cycle Networks	130
7.3	Resource Life Cycles as an Architecture	131
7.4	Employing Resource Life Cycle Networks	133
7.5	Real Benefit from Resource Life Cycle Analysis	136
7.6	Resource Life Cycle Analysis Components	137
7.7	Resource Life Cycle Analysis Summary	139
7.8	Questions and Exercises	140
8.	Business Information Systems Plans Architecture	143
8.1	Are Business Information Systems Plans Really Needed?	143
8.2	Characteristics of a Quality Business Information Systems Plan ...	145
8.3	Business Information Systems Plans within the Context of the Metadata Environment	147
8.4	The Business Information Systems Plan Steps	147
8.5	Executing and Adjusting the Business Information Systems Plan Through Time	151
8.6	Business Information Systems Plan Summary	154
8.7	Questions and Exercises	155
9.	Role of the Metabase	157
9.1	Structural Basis for the Metabase	160
9.2	Business Rationale for the Metabase	160
9.3	Metabase Model Relationships	163
9.3.1	Knowledge Worker Framework Relationships	164
9.3.2	Enterprise's Architecture Relationships	165
9.3.3	Database Object Relationships	166
9.3.4	Data Model Generalization Level Relationships	166
9.3.5	Resource Life Cycle Analysis Relationships	167
9.3.6	Generated Metadata Relationships	167
9.4	Metabase Component Modules	176
9.5	Functional Requirements for a Metabase System	180
9.6	Metabase, The Architecture Integrator	183
9.7	Metabase Benefits	198
9.8	Role of Metabase Summary	199
9.9	Questions and Exercises	201

Enterprise Architectures

10. Summary and Conclusion	203
10.1 Enterprise Architectures	203
10.2 Components of an Architecture	204
10.3 Knowledge Worker Framework	204
10.4 Enterprise's Architecture	206
10.5 Database Object Class Architecture	206
10.6 Data Architecture	207
10.7 Resource Life Cycle Analysis	209
10.8 Business Information Systems Plans	210
10.9 Role of the Metabase	212
10.10 Enterprise Architecture Way Ahead	213
10.11 Questions and Exercises	215
Attachment 1. Application for Employment	219
Attachment 2. Database Architecture Class Descriptions	223
Attachment 3. U.S. Department of Defense Architecture Framework	229
Attachment 4. Cross Reference Between Methodology, Architectures, and the Knowledge Worker Framework	234
Index	241

Figures

Figure 1. IDEF-0 process cell construction.	12
Figure 2. Measures of a tool's infrastructure contents.	13
Figure 3. Relationships between cells and adjacent or related cells.	14
Figure 4. Complex networks of process cells.	15
Figure 5. Mortgage calculator screen.	16
Figure 6. Lifting the one-world architecture.	17
Figure 7. Intersecting three architectures.	18
Figure 8. M1 A2 Abrams Battle Tank.	18
Figure 9. Overall relationship across Knowledge Worker Framework columns.	55
Figure 10. Enterprise architecture products and interrelationships.	61
Figure 11. Methods and language alternatives to database update.	77
Figure 12. Graphic displaying the composition of a database object.	80
Figure 13. Database object transformation from one state to the next.	81
Figure 14. High level data model for database object classes.	83
Figure 15. Database architecture classes.	91
Figure 16. Relationships among Data Element through View Data Models.	95
Figure 17. Key components of the Data Element model.	96
Figure 18. Key components of the Specified Data Model.	99
Figure 19. Key components of the Implemented Data Model.	103
Figure 20. Key components of the Operational Data Model.	106
Figure 21. Key components of the View Data Model.	109
Figure 22. Integration and reuse of Data Element to Specified Data Model.	111
Figure 23. Integration and reuse of Specified Data Model to Implemented Data Model metadata.	112
Figure 24. Integration and reuse of Implemented Data Model and Operational Data Model metadata.	114
Figure 25. Reverse engineering approach to data architecture management.	115
Figure 26. Functional distribution of data architecture efforts.	117
Figure 27. Court Person.	129
Figure 28. Case.	129
Figure 29. Document	129
Figure 30. Precedence vectors among court persons, case, and document.	132

Figure 31. Interrelationships Between Key Components of Enterprise Database and Resource Life Cycles. 134

Figure 32. Allocated Information Systems to RLC, Case. 136

Figure 33. Key components of the resource life cycle analysis model. 138

Figure 34. Knowledge worker framework and metadata domain. 148

Figure 35. Refined view of metadata involved in information systems plan. 148

Figure 36. Continuous flow process. 152

Figure 37. Interrelationship between Metabase and Clarion. 158

Figure 38. Relationship between framework cell and a metadata database. 164

Figure 39. Metadata integration across multiple enterprise architecture models. 168

Figure 40. Illusion of metadata integration and non redundancy. 169

Figure 41. Three separate projects with one integrated set of metadata. 172

Figure 42. Relative quantities of metadata across data model generalization levels. 174

Figure 43. Integrated methodology phases because of a metabase. 175

Figure 44. Unified operating knowledge worker environment with frameworks, metabase (metadata repository), and methodology. 215

Tables

Table 1. Examples of Enterprise Architectures in a Knowledge Worker environment.	4
Table 2. Knowledge Worker Framework contained architectures.	27
Table 3. Knowledge Worker Framework.	29
Table 4. Percent Allocation of the GAO reasons for information technology failure to the cells of the Knowledge Worker Framework.	32
Table 5. Summary of U.S. General Accountability Office Reasons for Information Technology System Failure	33
Table 6. Cross reference between Resource Life Cycle Nodes and Business Information Systems.	46
Table 7. Resource Life Cycle cross relationship with Business Information Systems and Business Events.	48
Table 8. Enterprise’s architecture components.	64
Table 9. Database Object Class components.	86
Table 10. Data model generalization levels.	94
Table 11. Data element model components.	98
Table 12. Specified Data Model components.	101
Table 13. Implemented Data Model components.	105
Table 14. Operational Data Model components.	108
Table 15. View data model components	110
Table 16. Resource Life Cycle Analysis model components.	139
Table 17. Characteristics of a quality of a Business Information Systems Plan.	146
Table 18. Steps for Business Information Systems Plan.	151
Table 19. Metabase processing modules.	180
Table 20. Metabase component reuse across Enterprise Architectures.	198
Table 21. Summary of U.S. General Accountability Office Reasons for Information Technology System Failure	205
Table 22. Methodology Phases with Work Products versus Enterprise Architectures.	236
Table 23. Methodology Phases with Work Products versus Knowledge Worker Framework..	239

Preface

A lack of demonstrable integration across various architecture classes is what prompted this book. It is common for different teams to be assembled across the enterprise to “do their thing.” One group does the enterprise’s¹ architecture while another group does data models, and some third group does Business Information Systems² plans.

The result is often three large piles of paper, and in the case of data models, really neat looking diagrams. Sometimes, Picasso comes to mind as the artist. Then, everybody sits back and waits for the magic caterpillars to butterfly transformations to occur. Nothing seems to happen. A “Winter of discontent” sets in. Everybody searches for the Spring, but it is nowhere to be found.

This book shows that not only can Spring happen, but also that three architecture-based product sets, that is, the enterprise’s architecture, the data models, and the Business Information System plans are actually part of a larger collection of architectures that are tightly coupled, interdependent, integrated, and non-redundant. In the process of creating the architecture product sets over the past 40+ years, its become painfully obvious that two architectures are missing, and that one architecture, data models, is badly engineered.

The two architectures that are missing are Database Object Classes and Resource Life Cycle Analysis. Database Object Classes are needed to

¹. As defined within this book, an enterprise is merely a term to relate to a collection of organization units that have common collections of data, processes, activities within a business or a company and sometimes beyond corporate affiliations as in the case of data interchanges. An enterprise is therefore not just a synonym for business or a company. Rather, it is intended to convey a common data, process, and activity view across the organizational units sharing that view.

². In this book, a business information system is a set of application-specific software that created, manipulates, evolves or deletes data—most commonly—from a database through a database management system in support of some mission area of the enterprise. A business information system is class of information system, and it is distinguished from other classes such as a computer’s operating system, or “systems software” information system such as telecommunications management, database management systems, and end-user security management.

overcome the real business perception problems that were caused by the popularity of the relational data model.

The Resource Life Cycle Analysis is needed because of two reasons. First, Resource Life Cycle Analysis enables the creation of an integration lattice work between the enterprise's architecture products and the enterprise's databases and business information systems. The second reason is that not only does Resource Life Cycle Analysis makes Business Information Systems Planning possible in realistic time frames and with reasonable resources, it enables these plans to be repeatable, reliable, and malleable.

The architecture that is badly engineered is data models. After extensive research into what data architects and data modelers actually do, a layered data architecture approach is clearly more appropriate. This approach is not fundamentally new. Rather it sets down into actual practice what is actually done. This approach is more practical, is accomplished all the time by data architects and data modelers but without persistence and repeatability. The result is a collection of data architecture products that support maximal reuse, complete integration, and very controlled redundancy.

In addition, the data architecture approach completely integrates the only two ISO/ANSI standards in this area: ISO 11179 for data element metadata, and ISO/ANSI SQL for comprehensive, portable, and interoperable data model development.

This book sets out five distinct architectures. While that's good, what integrates the work and the work products from all these architectures? It is the Knowledge Worker Framework. This framework, first explicitly posited in 1992, was squarely founded on a database project methodology that has been successfully evolving over the past 30 years. Further, the organization, sequencing, and work products in the Knowledge Worker Framework are those that satisfy 100% of the U.S. General Accountability Office's reasons for information technology system failure. These reasons were discovered from analysis of eight multi-hundred million dollar information technology failures.

It is not enough however, to just assert integration. It must be proven. Thus, there is an underlying metadata repository system that supports the creation, manipulation, evolution, and integration of all the work products. It is upon this highly engineered, experience-fired foundation that this book is based.

1

Enterprise Architectures

The objective of this book is to present an approach to the creation of architectures that should exist across the enterprise to ensure integration of data, semantics, and policy. To that end, this book addresses material about five architectures:

- The Enterprise's architecture.
- The Database Object Class architecture.
- The Data architectures.
- The Resource Life Cycle Analysis architecture.
- The Business Information Systems Plans architecture.

The book starts with explaining just what enterprise architectures are, and the components that should exist in a quality architecture. The book then sets all five architectures within the Knowledge Worker Framework. The Knowledge Worker Framework both has unique products and also products from the five architectures. The Knowledge Worker Framework serves as an overarching framework. Finally, the book describes the all-important role of the metadata repository as the mechanism for integration and non-redundant representation of all the work products.

This book shows that the five architectures are important to comprehensively manage the work products of knowledge workers. This book further shows that all these materials reinforce each other, are integrated, and enable enterprises to understand their past, operate the present, and plan for the future.

This chapter is all about enterprise architectures, that is, what are they, and why the architectures are necessary.

1.1 What are Enterprise Architectures?

The term, Enterprise Architecture contains both Enterprise and Architecture. In Wikipedia, enterprise contains no special definition. Rather, the Wikipedia states that “almost any business or organization can be called an enterprise.” From the Whitmarsh book, Data Interoperability Community of Interest Handbook, an enterprise is:

... a term to relate to a collection of organization units that have common collections of data, processes, activities within a business or a company and sometimes beyond corporate affiliations as in the case of data interchanges. Ideally, there is also a common governance of these items. An enterprise is therefore not just a synonym for business or a company. Rather, it is intended to convey a common data, process, and activity view across the organizational units sharing that view.

As to architecture, the Wikipedia states that an architecture is:

... the design of the total built environment, from the macrolevel of town planning, urban design, and landscape architecture to the microlevel of creating furniture. Architectural design usually must address both feasibility and cost for the builder, as well as function and aesthetics for the user.

Planned architecture often manipulates space, volume, texture, light, shadow, or abstract elements in order to achieve pleasing aesthetics. This distinguishes it from applied science or engineering, which usually concentrate more on the functional and feasibility aspects of the design of constructions or structures

The architectures contained in this book are in the class: Science and Engineering. Specifically, the Wikipedia provides a definition for Enterprise Architecture as follows:

Enterprise architecture is the practice of applying a comprehensive and rigorous method for describing a current and/or future structure and behavior for an organization's processes, business information systems, personnel and organizational sub-units, so that they align

with the organization's core goals and strategic direction. Although often associated strictly with information technology, it relates more broadly to the practice of business optimization in that it addresses business architecture, performance management, organizational structure and process architecture as well.

1.2 Five Contained Architectures

The definition set out in the previous section sets out the context for the five enterprise-related architectures identified in Table 1.

Enterprise Architectures	
Architecture Class	Brief Description
Enterprise	In the 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, business information systems, and business events.
Database Object Classes	A Database Object Class architecture is a self-contained set of data structures, processes, states and state-transformation systems that advance a collection of highly engineered data within the domain of a database management system (DBMS).
Data	A data architecture is the set of data-specific structures that enable enterprise-wide data and semantics integration critical to data interoperability.
Resource Life Cycle Analysis	A Resource Life Cycle Analysis is a set of enterprise resources, each set within a life cycle of major state transformations of those resources. These Resource Life Cycles can be interconnected, and used as a lattice work to attach the enterprise's inventory of database and business information systems, which in turn, greatly assists in the formation of Business Information Systems Plans.

Enterprise Architectures	
Architecture Class	Brief Description
Business Information Systems Plans	A Business Information Systems Plan is a highly engineered sequence of the development of databases and business information systems throughout the enterprise so as to cause maximum reuse and to accomplish the building and evolution of databases and business information systems in a business-rationale-based and highly ordered fashion.

Table 1. Examples of Enterprise Architectures in a Knowledge Worker environment.

The five architectures presented are all contained within the framework for the knowledge worker. The first architecture, the enterprise's architecture provides the overall context within which the knowledge worker operates. The enterprise's architecture identifies and describes the enterprise's mission, organization and functions. It also defines the data which ultimately provides the "proof of execution" of the operating enterprise.

The second architecture, Database Object Classes represent the objects that are valued, updated, employed, and deleted by the knowledge worker as the enterprise's business is conducted. Proof of mission accomplishment exists in the form of data. The data must be consistent, highly engineered, integrated, and non-redundant. Database Object Classes provide such constructions.

The third architecture, data, enables a high degree of engineering across all the enterprise's databases. That's not enough, however, because many enterprises have thousands of databases that are incoherent, in semantic conflict, not integrated, and very redundant. Key to enterprise-wide data engineering are the three levels of generalization above and one level of detailed interfaces below the level of the operating databases.

The highest level of generalization above the operating databases, Data Elements, enables the singular definition of the maximally used business facts and the setting of these business facts within business concepts and controlled value domains. These business facts, once defined are reused many times across the data model generalization levels so as to provide semantic consistency, value domain, and data type management.

The second highest level of generalization above the operating databases, Specified Data Models, enables the specification of standardized functionally-based data models of concepts that ultimately, as whole or partial collections, get used over and over during the construction of implementable database models. These Specified Data Models are used many times to effect standard data structures, semantics, and meaning across enterprise database models.

The first level of generalization above the operating databases, that is, the Implemented Data Models, represents data models of databases that may ultimately become one or more variations of the operating databases. Additionally, whole or partial data structure collections from these Implemented Data Models may be employed in one or more of the operating databases.

The point to these “above” levels of generalizations is that these enable order from data specification chaos, integration and reuse of semantics from the current discord, and control or even elimination of large scale redundancy of data across the enterprise. Ultimately, database should promote harmony, civilized discourse, and not be the cause of the bureaucratic food-fights.

The one level of detailed interfaces below the level of the operating databases is the level of data exchange between database and business information systems. The exchange level has two forms, technology bound, and technology independent. The technology bound level, SQL views, support the direction interaction between the database management system (DBMS) databases and the business information systems. The second form, technology independent, is commonly seen in the form of XML, which has two parts: definition, and data stream. These two parts are either composed by a business information system, or shredded into a database’s required structure by the DBMS. This enables data to be put and/or retrieved by DBMS-disconnected business information systems. XML is not however a know-nothing, do-everything data exchange environment. The four layers above the XML layer (that is, Data Elements through Operational Data Models) must already exist for XML to be effective.

The fourth architecture, Resources Life Cycles, are an enterprise resource’s organizing architecture. The key premise is that every enterprise consists of a finite set of well-known resources that are created, manipulated, and ultimately dissolved by the enterprise. Examples include locations, facilities, products, monies, staff, real and abstracts, and the like. Enterprise operations are thus seen through the operating life cycles of resources.

Resource Life Cycles are able to be interconnected, one life cycle with another. Collectively, the interconnected Resource Life Cycles form a lattice work through which missions, organizations and functions are operationally viewed, and upon which database objects and business information systems are integrated and coalesced.

The fifth architecture, Business Information Systems Planning, is an architecture that reflects the build and evolution sequence of all information technology projects. This Business Information Systems Planning architecture is built directly from the foundation of the Resource Life Cycle Analysis architecture, which in turn was built from the database objects, data, and enterprise architectures.

All together, these individual architectures enable the enterprise to engineer, evolve, and know answers to the following questions:

- What is the enterprise's mission, organization, functions and domains of data?
- What is the key policy-based engineered data that has to be coherent, integrated, and non-redundant?
- What are the enterprise resources and how do they progress individually and in concert to achieve enterprise missions in the most efficient and effective manner possible?
- What is the overall Business Information Systems Plan, and in what sequence must the database and business information system projects must be accomplished to achieve just-in-time readiness?

Key to all these architectures is the organizing framework for the knowledge worker, that is, the Knowledge Worker Framework. This framework organizes and interrelates these five interconnected and contained architectures and enables all the architecture work products to be integrated, and non-redundantly stored in one metadata repository. This metadata database repository, hereinafter called the Metabase³, supports the creation,

³ Metadata is a generic term that identifies all classes of information technology specifications across the enterprise. Hence all data and process
(continued...)

manipulation, and reporting of only what the enterprise needs to exist and thrive.

The material about the Knowledge Worker Framework contains, at varying levels of detail, descriptions about all the products, including the products contained in the architectures encompassed by the Knowledge Worker Framework. The chapters on the individual five architectures, for example, Database Object Classes, provide additional levels of detail about Database Object Class components, reasons for the need for Database Object Classes, the relationship among the components of Database Object Classes, and the relationship between components Database Object Class and components from other architectures.

1.3 Remainder of this Book

The chapters in this book set out the engineering and the content of the Knowledge Worker Framework and its five contained architectures. This chapter started with the enumeration of the architectures and stated why each needs to exist and how all the architectures are integrated into an overall whole.

Chapter 2 describes architectures in general and identifies the critical components and measures of successful architectures. Bad architectures are just above not worth having, and may even be counterproductive if they cause the creation of work products that are not able to be integrated, or are supported by methodologies that have not been detailed or proven, end-to-end. Good architectures are fully detailed. Every work product is expertly engineered, is supported by a high-quality methodology, and is able to be integrated with all the other work products within a given architecture and across architectures. Notwithstanding the high degree of engineering, good

³(...continued)

specifications are metadata. All requirements could also be considered metadata. A metadata repository is a database within which all metadata is stored. A metabase database is a metabase. Whitmarsh has employed the term, Metabase, in this context since 1982. A metadata repository system is a software system that captures, stores, reports, and manages all metadata. The system that manages a metabase is the Metabase system. Sophisticated metadata repositories are multi-user and support the capture and reporting of metadata in a non-redundant, integrated manner across the enterprise.

architectures are just highly proceduralized common sense. Common sense is the acid test, and the inverse of common sense is non-sense.

Chapter 3 fully describes the Knowledge Worker Framework, row by row, and column by column. Every cell is detailed including a high level description of every work product that results for the cell contained work processes. All cells are, of course, completely integrated.

Chapter 4, the Enterprise's Architecture, describes the contents of the enterprise's architecture and describes at a high level how the enterprise's architecture is created. This chapter also describes the work products, how they are interrelated, and how these work products fit with the work products of the other architectures.

Chapter 5, Database Object Classes, defines what a Database Object Class is, how it contrasts with other types of objects, and what unique role Database Object Classes play in a database environment. This chapter also presents how Database Object Classes can be implemented in SQL DBMSs that are not object-oriented and how these classes can be implemented in SQL:1999 and above compliant DBMS that are object-oriented. This chapter also describes the work products, how they are interrelated, and how these work products fit with the work products of the other architectures.

Chapter 6, Data Architecture, starts with a presentation of the various classes of the database architectures that most commonly exist within enterprises, be they small or large. The chapter then proceeds to fully define each of the five data model generalization levels, that is, Data Elements, Specified Data Models, Implemented Data Models, Operational Data Models, and View Data Models. The chapter sets out the work products that must be created for each model, and then how all these work products are thoroughly integrated one with the other.

Chapter 7, Resource Life Cycle Analysis, defines just what Resources and Resource Life Cycles are, what business objectives are fulfilled by their employment, where these fit within all the other architectures. While valuable in their own right, Resource Life Cycles also serve as a key linking mechanism between the Enterprise's Architecture, Database Object Classes, and Data Architectures on the one hand and Business Information Systems Plans on the other hand.

Chapter 8, Business Information Systems Plans, explains their necessary characteristics. These characteristics have been glaringly unfilled by other Business Information Systems Plan development methodologies. This explained methodology is the next logical step from the prior architectures

and employs all their work products. The Business Information Systems Planning methodology then leverages the large quantity of project methodology and metrics' templates to produce Business Information Systems Plans that are reliable and repeatable, and are malleable as new technologies unfold.

Chapter 9, The Role of the Metabase, shows how the Metabase should be employed to store all the work products from these architectures. Metadata repositories that store just operational or technical metadata miss about 99% of their true value to the enterprise. Clearly, if we focus solely on the optimization of the past we will never be able to plan, see, and accomplish the future. It is through the exposition, optimization, and recasting of our missions, organizations, functions, database objects, databases and business information systems that the future will be unfolding in ever better forms.

Chapter 10, Summary and Conclusions, brings together the goals and summary of every chapter and sets out the way ahead: a work plan for success.

It needs to be said, and remembered that this book is not a theory book. If it had been, it would have been written 25 or more years ago. Rather, this book is the consequence of the evolution and development of enterprise architecture and data management practice that has been conducted in industry and government since the early 1970s.

Every product in every chapter of this book is fully detailed in other Whitemarsh books, and is fully supported by detailed methodology, courses, workshops, and of course, the Metabase, Metabase. In short, this book is a professional practice book and is another integral component of the Whitemarsh Data Management Series.

1.4 Questions and Exercises

1. Are you aware of stand-alone architecture projects such as the one-big data model, or a Business Information Systems Plan? What was its value? Did it integrate with existing metadata? Did it become expensive shelf-ware? What was management's attitude towards it over time?
2. What are the effects of a bad architecture? How do you know if one is bad? Do you follow it nonetheless or try to have it fixed?

3. How many different kinds of architectures are there in Information Technology? Can there be just “one for all reasons” that is specific, unambiguous, useful, and able to be accomplished?
4. Do the five architectures in this book address the needs of the Knowledge Worker? What are other architectures? Are there common components across architectures? How are common components defined, valued, and kept in synch?
5. Does this book provide a practical definition for the enterprise? Architecture? Enterprise Architecture? How would you define these terms? Compare and contrast with the book’s definitions.
6. Why should there be five data model generalization levels? What purpose does each serve? How are the levels interrelated? Does the interrelationship make sense? What would be the benefit to all these levels? Costs?
7. Should architectures be designed such that they tell stories? How do you update your architectures? How often? Are the updates in one architecture automatically carried over to another when there are common components? If yes, what’s the benefit. If not, what’s the cost?