Whitemarsh

Information Systems Corporation

Knowledge Worker Framework

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

1

Knowledge Worker Framework

1.1 The Knowledge Worker

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 and for example, food service personnel 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 complexities 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.

1.2 The Zachman Framework

Figure 1.1 depicts the Zachman Framework[1]. John Zachman and John Sowa [2] both extended and then tried to represent the contents of all 36 cells of the Zachman framework within an integrated repository meta model. A review of the Zachman and Sowa materials along with the extensive article in Database Programming and Design (March 1997, Vol. 10 No. 3) clearly demonstrate that the focus of the framework is the information system and its immediate surroundings rather than the enterprise or the knowledge worker environment.

Over the years a number of organizations[3] have attempted to rigorously implement the Zachman framework, but few have had more than “spiritual” success. Nonetheless, the Zachman framework has an intrinsic simplicity, and an almost instantaneous acceptance by all those who see it.

---

1 *A Framework for Information Systems Architecture*. John A Zachman. IBM Systems Journal, Vol 26, No 3, 1987. IBM Publication G321-5298. The Zachman Framework is a six by six matrix. The columns are data (what), function (how), network (where), organization (who), time (when), and rationale (why). The rows are scope, business model, systems model, technology model, implementation, and executing system. The contents of the 36 cells are intended to be the products that result form accomplishing the row-column effort. That is, accomplishing the scope of data, or the systems model of data, etc.

2 *Extending and Formalizing the Framework for Information Systems Architecture*. John F Sowa and John A. Zachman. IBM Systems Journal, vol 31, no. 3, 1992. IBM Publication G321-5488.

3 MITRE, for example, in 1994 attempted to deploy the framework at EPA. After a number of attempts, it was concluded that it was impossible to both validly and faithfully represent the framework through an integrated repository.

Zachman Enterprise Architecture Framework						
	Data	**Function**	**Network**	**People**	**Time**	**Motivation**
Scope	List of Things Important to Business	List of Processes the Business Performs	List of Locations in Which the Business Operates	List of Organizations Important to the Business	List of Events Significant to the Business	List of Business Goals/strategies
Enterprise Model	Semantic Model	Business Process Model	Logistics Network	Work Flow Model	Master Schedule	Business Plan
System Model	Logical Data Model	Application Architecture	Distributed System Architecture	Human Interface Architecture	Processing Structure	Business Rule Model
Technology Model	Physical Model	System Design	System Architecture	Presentation Architecture	Control Structure	Rule Design
Detailed Representations	Data Definition	Program	Network Architecture	Security Architecture	Timing Definition	Rule Specification
Functioning Enterprise	Data	Function	Network	Organization	Schedule	Strategy

Figure 1.1, Zachman Information Systems Architecture/Enterprise Architecture Framework

In the Summer of 1996, Barbara von Halle presented the Zachman Framework to a meeting of the New Jersey Data Administration Management Association. At that meeting, Ms. von Halle, a principle for Spectrum Technology Group described how Spectrum employs the Zachman framework as a mechanism for identifying the overall work effort of consulting assignments. According to von Halle, the framework is quite suitable for these purposes.

Ms. von Halle also presented a repository design sketch that Spectrum employed with one of their clients. The repository design is very similar to the essential components of the Whitemarsh repository, which was based on an earlier repository design from the Yourdon, Inc. course, Database Administrator's Workshop (Gorman, July 1981).

While the Zachman framework is a great scoping and framework document it is neither detailed nor integrated. Past attempts to faithfully detail the framework and build a repository meta-model from its basis for other clients have been unsuccessful. Success exists when risks are lowered, quality is increased, estimates are accurate and projects are delivered on-time or early. Ultimately, the Zachman framework is best seen solely as a framework of high-level guidance and a way to judge the overall comprehensiveness of a planned effort.

1.3 The Knowledge Worker Framework

The Whitemarsh Knowledge Worker Framework, depicted in Figure 1.2, is similar to the Zachman framework but was inductively built from the Whitemarsh methodology. Whitemarsh Information Systems Corporation, starting in 1981, has been employing the methodology for large scale database projects and installing repositories for both industry and government[4]. Databases, treaties that govern the behavior of its users, are broad in scope, inter-organizational and multi-functional in nature. Thus, database projects by their very nature deal with most aspects of the Knowledge Worker Framework.

[4] Repositories have been implemented at Hartford Insurance (1982), U.S. Army (1984), Hershey Chocolate(1985), Social Security Administration (1986), MITRE Corporation (State of Ohio, MITRE, State of Delaware, and State of California)(1991-1995), the Mars Corporation (1995 - 1996).

Whitemarsh's Knowledge Worker Framework								
Viewpoint			Man-Machine Interface					Primary Responsi-bility
			Machine		Interface	Man		
Project	Deliverables	Mission	Database Object	Business Information System	Business Event	Business Function	Organization	
Specification	Scope	List of business missions	List of major business resources	List of business information Systems	List of interface events	List of major business scenarios	List of organizations	Architect
Specification	Business	Mission hierarchies	Resource life cycles	Information sequencing and hierarchies	Event sequencing and hierarchies	Business scenario sequencing and hierarchies	Organization charts, jobs and descriptions	Architect
Specification and Implemen-tation	System	Policy hierarchies	Specified data models and Database object models	Information system designs	Invocation protocols, input and output data, and messages	Best practices, quality measures and accomplishmen t assessments	Job roles, responsibilities, and activity schedules	Architect and Engineer
Implemen-tation	Technology	Policy execution enforcement	Implemented data models	Information systems application designs	Presentation layer information system instigators	Activity sequences to accomplish business scenarios	Procedure manuals, task lists, quality measures and assessments	Engineer
Implemen-tation	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	Engineer
Operation	Operations	Operating business	Application Interface data models	Operating information systems	Start, stop, and messages	Detailed procedure based instructions	Daily activity executions, and assessments	Engineer

Figure 1.2, Whitemarsh Knowledge Worker Framework

At the heart of the Whitemarsh methodology is a single taxonomy and terminology that forms the basis for understanding database, database management system technology, database projects, and repository. Whitemarsh's methodology and its use of repository is described in several books.[5]

The Knowledge Worker Framework modifies the Zachman framework's columns and row headers. The contents of the 36 cells were also reworked to produce already known successful database deliverables. While many of the words and sequences have been changed, the overall framework parallels Zachman's intent: a broad framework across multiple disciplines. The Whitemarsh Knowledge Worker Framework is more comprehensive than the Zachman framework because the Whitemarsh Knowledge Worker Framework:

- Moves the Why column, the last column in the Zachman framework from its role as justification or reason to be the first column in the Whitemarsh knowledge worker frame work and labels it mission so that it can "drive" all the remaining columns and their unfolding rows;

- Incorporates a database object column that is a melding of the Zachman data and function column, and the rules portion (rows 3, 4, and 5) of the motivation column. Database objects are an essential component for heterogeneous, world wide database efforts. In contrast, the Zachman framework represents objects in row 3, columns 1, 3, and 6[6].

- Drops the network column because it really is a technology mechanism of implementation. This column, like other technology implementation mechanisms such as operating systems and database management systems form a technology dimension that is invisible within the Knowledge Worker Framework--as it should be.

- Creates a function column to address human functions and/or human processes that may or may not involve business information systems

5 Managing Database, Four Critical Factors (Gorman, QED, 1984), Database Management Systems, Understanding and Applying Database Technology (Gorman, QED, 1991), Enterprise Database in a Client Server Environment (Gorman, Wiley & QED, 1994), and Database Objects, The Foundation Stones of Enterprise Database (Whitemarsh, 1996), Implementing A Corporate Repository (Tannenbaum, Wiley, 1994)

6 *Data Architect* by Barbara von Halle in Database Programming and Design, Vol. 9, No 11, page 11.

- Creates a business information systems column that interfaces with the function column and the database objects column

- Partitions five of the columns into two for “machine”, one for “interface”, and two for “man” to enable more flexibility, reuse,

- Redefines the first column of row headers into two subcolumns that classifies deliverables and indicates the overall major project phase in which the deliverables are created, and finally,

- Adds a last column of row headers to indicate “who” is responsible for overseeing the rows.

This last modification is discussed by both Zachman[7] and von Halle’s[8]. These changes have transformed the Zachman framework away from focusing on just information systems and their supporting technologies towards a more complete framework that addresses the complete knowledge worker environment.

The Knowledge Worker Framework is important because it provides the opportunity to view knowledge work on a single continuum from mission through organization (the columns) and at multiple levels, that is, from scope through day to day operations (the rows). If all the analysis, specification, and implementation metadata resulting from this holistic approach is stored in a repository, then complete job descriptions, job duties, procedures, guidelines, work effort measures can be stored, interrelated, examined for conflicts and, over time, evolved and maintained to remain current with the evolution of an enterprise’s mission.

Both the Zachman and the Knowledge Worker Frameworks have been reviewed–side by side–and there is general agreement that the Knowledge Worker Framework is more comprehensive in the areas critical to the success of the knowledge worker.

[7] *Enterprise Architecture: The Issue of the Century.* Database Programming and Design, Vol 10, No. 3, March 1977.

[8] *Data Architect*. Database Programming and Design, Vol. 9, No 11, November 1996.

1.4 Analysis of Troubled Environments

During reviews of troubled computing environments[9], a common finding is the lack of effective, stored, traceable systems requirements, valid functional adaptations, applicability to multiple organizations, design, implementation, operation, and maintenance artifacts. These must exist both from the machine and the human perspective. Additionally there must be an effective interface between "man" and "machine." Simply, organizations and staff seem not to know where they've come from, where they're going, how they'll know when they get there, and why. Anecdotal reviews and surveys reveal that when projects are in trouble, critical elements of institutional knowledge have either been ignored or not integrated into a cohesive whole so that the organization's information needs are unclear. Without that clarity, the entire systems development and database design effort cannot be undertaken with confidence.

In a thorough review of a $500 million Federal project, it became obvious that critical aspects of the Knowledge Worker Framework were unaddressed. Every time a problem was identified within the project, the knowledge worker cell that most closely related to the problem was checked. After the project's review was finished the problems were classified as fatal, near-fatal, or fixable. The diagram and its check marks were reviewed. Every check mark that was fatal or near-fatal was in a specific domain of cells, and the fixable problems were in an entirely different domain.

Figure 1.3 presents a version of the Knowledge Worker Framework in which twelve cells are "missing" and 24 other cells are shaded. The 24 cells are referred to as the knowledge worker environment. The 12 cells that are missing are directly related to traditional perceptions of information technology support to the knowledge worker. Figure 1.4 presents the framework with the 24 squares missing and the 12 inner cells shaded.

The studies of failed or in-trouble information technology projects almost always result in the following conclusions:

- One or more of the cells within the knowledge worker environment set (the "24" squares) were not properly addressed or have been identified as the reasons for the problems and/or failures. It didn't really matter whether the 12 information technology cells were well done or not.

---

9 In contrast to troubled environments, many quality computing environments have database at its center or focus. Database is a technology based exposition of coherent business organization and policy. The data in a database is the business' executed policy. Well engineered, a database system accurately mirrors enterprise policy and procedures in some well bounded business area and is employed to analyzed the past, facilitate/control the present and to assist in planning for the future. The quality of database is directly proportional to the quality of the enterprise.

Whitemarsh's Knowledge Worker Framework									
Viewpoint			Man-Machine Interface						Primary Responsi-bility
			Machine		Interface	Man			
Project	Deliverables	Mission	Database Object	Business Information System	Business Event	Business Function	Organization		
Specification	Scope	List of business missions	List of major business resources	List of business information Systems	List of interface events	List of major business scenarios	List of organizations		Architect
Specification	Business	Mission hierarchies	Resource life cycles	Information sequencing and hierarchies	Event sequencing and hierarchies	Business scenario sequencing and hierarchies	Organization charts, jobs and descriptions		Architect
Specification and Implemen-tation	System	Policy hierarchies				Best practices, quality measures and accomplishmen t assessments	Job roles, responsibilities, and activity schedules		Architect and Engineer
Implemen-tation	Technology	Policy execution enforcement				Activity sequences to accomplish business scenarios	Procedure manuals, task lists, quality measures and assessments		Engineer
Implemen-tation	Deployment	Installed business policy and procedures				Office policies and procedures to accomplish activities	Daily schedules, shift and personnel assignments		Engineer
Operation	Operations	Operating business				Detailed procedure based instructions	Daily activity executions, and assessments		Engineer

Figure 1.3, Knowledge Worker Environment Components of the Whitemarsh Knowledge Worker Framework

Whitemarsh's Knowledge Worker Framework								
Viewpoint			Man-Machine Interface					Primary Responsi-bility
			Machine		Interface	Man		
Project	Deliverables	Mission	Database Object	Business Information System	Business Event	Business Function	Organization	
Specification	Scope							Architect
	Business							
Specification and Implemen-tation	System		Specified data models and Database object models	Information system designs	Invocation protocols, input and output data, and messages			Architect and Engineer
Implemen-tation	Technology		Implemented data models	Information systems application designs	Presentation layer information system instigators			Engineer
	Deployment		Operational data models	Implemented information systems	Client & server windows and/or batch execution mechanisms			
Operation	Operations		Application interface data models	Operating information systems	Start, stop, and messages			

Figure 1.4, Information Technology Components of the Whitemarsh Knowledge Worker Framework

- When most or all of the “24” cells are properly handled the project is almost always considered a success. It didn’t really matter whether the 12 information technology cells were well done or not.

In a number of studies performed by the United States General Accounting Office, an arm of the United States Congress, it becomes very clear that virtually all information systems failures occur for reasons almost totally divorced from information technology. The primary reasons why information technology projects fail has virtually nothing to do with information technology. The reasons almost always point to one or more of the “24” cells. The following documents were reviewed and support this conclusion:

- HUD Information Resources: Strategic Focus and Improved Controls Needed, GAO/AIMD-94-34 (4/14/94)
- Executive Guide: Improving Mission Performance Through Strategic Information Management and Technology, GAO/AIMD-94-115 (05/01/94)
- Defense Management: Impediments Jeopardize Logistics Corporate Information Management, GAO/NSIAD-95-28 (10/01/94)
- Information Technology: Best Practices can Improve Performance and Produce Results, GAO/T-AIMD-96-46 (2/26/96)
- Defense IRM: Critical Risks Facing New Materiel Management Strategy, GAO/AIMD-96-109 (09/01/96)
- Defense Financial Management, GAO/HR-9703 (02/01/97)
- Information Management and Technology, GAO/HR-97-9 (02/01/97)
- USDA Information Management, GAO/T-AIMD-97-90 (5/14/97)

All these documents are available from the Government Accounting Office’s web site. The GAO is charged with determining how well the Executive branch is spending taxpayers monies. A complete listing of the key points made by these studies is provided in Appendix 1. Readers are strongly encouraged to see if they “read” their company or organization within this appendix. If so, then the prescriptions contained within this book may help alleviate problems.

The message is thus short and simple: pay significantly more and proper attention to the "24" cells than to the "12." While not paying attention to the "24" will almost always guarantee failure, paying attention to the "12" will almost never guarantee success.

1.5 GAO Versus Zachman and Knowledge Worker Frameworks

The GAO studies of in-trouble or failed information technology projects provides an excellent opportunity to compare the Zachman and Knowledge Worker frameworks. The question simply stated is: which framework, if implemented and followed completely will address more of the critical issues cited in the GAO studies. The relevance of the answer depends on whether the issues cited in the GAO study are important to a specific organization. If an analysis of the GAO issues with respect to your organization ranges from "painful" through "excruciating" then the comparison is similarly significant.

Across all the studies there were about 120 reasons for project failures. Many of these reasons occur multiple times. Hence, the list is not an unduplicated one. When these 120 reasons were "allocated" to the Zachman and Knowledge Worker frameworks, one or more cells were "checked" whenever the "checked" cell represented an activity that if properly accomplished would have caused the failure reason to either be resolved or not to occur at all. Figure 1.5 then, identifies the allocation of the GAO failure reasons to the Zachman framework. This figure shows that less than 10% of the GAO failure reasons would have been addressed by the Zachman framework. In contrast, Figure 1.6 shows the same allocation to the Knowledge Worker framework. 100% of the GAO failure reasons are allocated therein. The conclusion is straight forward: the Knowledge Worker Framework addresses more of the cells than does the Zachman framework.

During the presentation of the Knowledge Worker Framework at the New Jersey DAMA meeting in July 1997, it was asked what would be an appropriate strategy for getting the message of the Knowledge Worker Framework to "management." It was suggested that the information technology organizations must convince "management" that they should stop being victims of information technology projects. Rather, they should own, control, and take responsibility for significant portions of information technology efforts. Clearly, "management" should own, control, and take responsibility for the outer-24 cells. And, if at all possible, projects should never be started without these 24 cells well under control. When these cells are controlled, risk is reduced, costs are lowered, and projects generally come-in on time and within budget.

Zachman Enterprise Architecture Framework						
	Data	**Function**	**Network**	**People**	**Time**	**Motivation**
Scope	2	1	None	None	None	5
Enterprise Model	1	1	None	None	None	None
System Model	None	None	None	None	None	None
Technology Model	None	None	None	None	None	None
Detailed Representa-tions	None	None	None	None	None	None
Functioning Enterprise	None	None	None	None	None	None

Figure 1.5. Allocation of the GAO Information Technology Critical Success Factor Issues

Knowledge Worker Framework									
Viewpoint			**Man-Machine Interface**						**Primary Responsi-bility**
			Machine		**Interface**	**Man**			
Project	**Deliverables**	**Mission**	**Database Object**	**Business Information System**	**Business Event**	**Business Function**	**Organization**		
Specification	**Scope**	13	5	6	1	8	10	**Architect**
	Business	12	6	6	1	15	14	
Specification and Implemen-tation	**System**	8	6	5	0	28	18	**Architect and Engineer**
Implemen-tation	**Technology**	3	0	0	0	18	14	**Engineer**
	Deployment	1	0	0	0	12	11	
Operation	**Operations**	1	0	0	0	8	8	

Figure 1.6. Allocation of the GAO Information Technology Critical Success Factor Issues

1.6 Object Oriented Environments

All work environments should be object oriented. Knowledge worker environments operate in whole activities rather than specialized subsets of activities. In contrast, most real product workers operate in highly specialized environments in order to produce individualized aspects of real products such as aircraft, automobiles, and other manufactured products. While both environments are obect-oriented, knowledge worker environments are more “data” intensive and real product worker environments are more “process” intensive. In addition, real product worker environments are required to be exact and correct for every iteration of the product or sub-product that is manufactured, while knowledge worker products are not. Knowledge worker environments involve whole collections of data and processes in support of products that are abstract and are highly variable either by design or necessity.

For example, a real-product worker may spend entire days, weeks, and even years putting bolts into a component of a larger machine without ever knowing whether the final assemblies fly through the air, run on the ground, or move through water. A medical clinician, however, a classical example of a knowledge worker, who performs a history and physical must consider the entire patient, prior medical treatments, home and work environment, current physical and mental state, and the possibilities of treatment; not just whether there needs be one more bolt.

The three heralded characteristics of objects are encapsulation, inheritance, and polymorphism. The most critical aspect of an object-oriented environment is encapsulation. Encapsulation means that the object (that is, its contained data and attendant processes) are accessible only through the well defined and bounded rules. An EKG machine is an example of complete encapsulation. The machine contains all the materials and processes necessary to capture the EKG signals, record them on paper, and to react to unacceptable signal changes. A knowledge worker’s history and physical is also encapsulated.

Encapsulation has been a characteristic of well defined, designed, implemented, and maintained data processing systems and environments since the early 1950s. Encapsulation existed well before data processing. A business form that contains both the blank fields for data entry and also all the sets of instructions, look-up tables, etc., is encapsulated. If the data entry person had to rely on external tables and/or external instructions and if the wrong set was employed then there would be two consequences:

- broken encapsulation
- bad data entry

From a formal viewpoint, encapsulation means that the object is shielded from the “influences” of its outside environment. Standard money arithmetic processes defined within insurance policy object regarding premium computation can be configured independent of the currency of the money represented by the premium given the arguments of dollars exchange rate and other standard inputs.

Inheritance means that contained objects presume on the properties of containing objects. For example, a particular assessed patient assumes predominate of all men, all humans, and if a native of a community, the traits of the populations of the community in which the patient was raised.

Polymorphism means that a contained object may perform acceptably despite different, but acceptable outside factors. For example, invoking a command to compute REMAINING BALANCE may in fact invoke different processes depending on the “invoking environment” such as whether the required remaining balance is for a loan, and invoice, or a real-estate mortgage. In a clinical setting, a patient might exhibit a number of different behaviors. Within a diagnostic manual, different combinations of behaviors would lead to specific diagnoses. The contained object is the mechanism of diagnosis, the methods are the processes by which the diagnosis is determined, and the outside data are the behaviors of the patient. Because the object can produce different diagnoses for different collections of behaviors, the object exhibits the characteristic of polymorphism.

All successful knowledge worker environments must strive to be object-oriented because this lessens errors, increases the quality of designs, and lowers the costs of development and maintenance. These characteristics result because complex objects, e.g., the patient, the computer system, or the clinician is broken down into definable components that are nonetheless whole, reactive, and simpler to construct, test, and maintain. Containing objects are then build-up by linking collections of already known-to-be-correct contained objects. While objects are identified top-down and are built bottom-up, they can also can contain references or make use of “outside” object-oriented components. Those references must however be unambiguous so that the references will always produce dependable results.

The Knowledge Worker Framework itself is object oriented because it embraces all enterprise missions, organizations, functions, business information systems, databases, interfaces between functions and business information systems. Each component within the framework should also be object-oriented.

Given that the automation component of an enterprise’s knowledge worker environment is heavily dependent on commercial off-the-shelf (COTS) packages, these COTS packages must also be as object-oriented as possible. Given also that there are large, broad, and integrated databases that are used by the COTS packages, then the business information systems that support the databases too must be object-oriented to ensure that each business information system receives consistent data. When multiple business information systems store data, that data must be stored through singly defined, and consistently applied encapsulated business rules.

If, during the evolution and maintenance of the computing environment, an object-oriented component is changed then the change/evolution must be available to and employed by all referencing object-oriented components at the same instant. Otherwise, data and process corruption will occur.

The key benefit from the Knowledge Worker Framework is that it is broad and all encompassing, starting with list and ending with day-to-day activities that contribute to the accomplishment of the enterprise's mission. The framework belongs to the entire enterprise, not just one department. The key benefit from an object orientation is that once the objects are selected and installed they can operate within the environment with minimum disruption. Together, the knowledge worker environment and the object orientation provide for a framework of progress within which the technology architecture can operate in a beneficial manner.

1.7 Remainder of Book

Chapter 2 of this book describes the various columns and rows of the Knowledge Worker framework. Chapter 3 identifies the high level Whitemarsh work breakdown structure elements needed to produce the cell's contents, and Chapter 4 identifies the meta-entity types from the Whitemarsh Repository required to support the framework.. Chapter 5 provides a subset example of key framework cells.

1.8 Whitemarsh *on* Database WebSite

Whitemarsh publishes a series of books, courses, and software on database. Each item represents the *lessons learned* from almost thirty years of database projects. It's often said that wisdom is 99% scar tissue. Given that truth, using Whitemarsh *on* Database books is a wise move. Information for Whitemarsh *on* Database materials is available on the web site: www.wiscorp.com.

All Whitemarsh on Database books, courses, and software are live. That is, they are published electronically in an Adobe PDF format for the books, HTML for the courses, and self-extracting zip files for the software. Whitemarsh materials are updated on an as-needed basis. Whitemarsh *on* Database subscribers are notified about updates through email. Updates can then be downloaded through the Whitemarsh web site.