Standard Practice for Information Modeling[1]

This standard is issued under the fixed designation E2145; the number immediately following the designation indicates the year of original adoption or, in the case of revision, the year of last revision. A number in parentheses indicates the year of last reapproval. A superscript epsilon (ε) indicates an editorial change since the last revision or reapproval.

1. Scope*

1.1 Information models are increasingly important in the analysis, design, and sharing a common understanding in information engineering, in business process improvement, in building information systems, in developing informatics standards, and in many other uses.[2]

1.2 The purpose of this practice is to identify best practices for the creation, use, and assessment of various types of information models.

1.3 Included in this practice are recommended organizational policies and procedures, where modeling is best used and recommended modeling methods, best practices and evaluation criteria.

1.4 Excluded from this practice are detailed specifications of modeling techniques that are specified or described in other sources.

2. Referenced Documents

2.1 *ANSI Standards:*[3]

ANSI X3.172-1990 Dictionary for Information Systems

IEEE 1320.1-1998 Standard for Functional Modeling Language—Syntax and Semantics for IDEF0

IEEE 1320.2-1998 Standard for Functional Modeling Language—Syntax and Semantics for IDEF1X (Object 97)

2.2 *ISO Standards:*[3]

ISO 8601-88 Data Elements and Interchange Formats—Representation of Dates and Times

ISO/IEC 1087 Terminology Work—Vocabulary-Part 1: Theory and Application

ISO/IEC 11179 Information Technology—Specification and Standardization of Data Elements, Parts 1-6

ISO/IEC 19501:2005 Information technology—Open Distributed Processing—Unified Modeling Language (UML) Version 1.4.2

ISO/IEC 2382 Information Processing Systems—Vocabulary

2.3 *Other Documents and Standards:*[4]

Unified Modeling Language Specification version 1.5, March 2003

3. Terminology

3.1 The following sections present terms, definitions, and acronyms found in this practice and in various modeling activities. These terms and definitions are referenced or derived from ANSI X3.172 or ISO/IEC 2382 unless otherwise cited.

3.2 *Definitions:*

3.2.1 *activity, n*—a group of logically related tasks performed for a purpose.

3.2.2 *alternate key attribute, n*—in a logical data model, any candidate key of an entity other than the primary key.

3.2.3 *application model, n*—a representation or description of the application software, programs, or components needed to support a business function.

3.2.4 *attribute, n*—a characteristic of an object or entity (see ISO/IEC 11179).

3.2.5 *attribute value, n*—a representation of an instance of an attribute (see ISO/IEC 11179).

3.2.6 *behavior, n*—in the object-oriented methodology, behavior constitutes the observable effects of an operation or event, including any results generated or obtained; and represented by operations, methods, and state machines (see UML 1.5).

3.2.7 *business model, n*—a representation of the strategy, situation, environment, objectives, direction, and similar characteristics of a business enterprise or business area.

3.2.8 *business process model, n*—a representation of business processes, often where these processes are successively decomposed to describe component activities, to identify the events to which the business shall respond, and to identify the results produced.

[1] This practice is under the jurisdiction of ASTM Committee E31 on Healthcare Informatics and is the direct responsibility of Subcommittee E31.25 on Healthcare Data Management, Security, Confidentiality, and Privacy.

Current edition approved March 1, 2013. Published March 2013. Originally approved in 2001. Last previous edition approved in 2007 as E2145 – 07. DOI: 10.1520/E2145-07R13.

[2] An information model in the context of this standard is any representation of process, data, etc., used in any aspect of information technology or information management.

[3] Available from American National Standards Institute (ANSI), 25 W. 43rd St., 4th Floor, New York, NY 10036, http://www.ansi.org.

[4] Available from Object Management Group. OMG Headquarters, 250 First Avenue, Needham, MA 02494. http://www.omg.org.

***A Summary of Changes section appears at the end of this standard**

Copyright © ASTM International, 100 Barr Harbor Drive, PO Box C700, West Conshohocken, PA 19428-2959. United States

3.2.9 *column, n*—a physical data model and relational database structure that is analogous to the attribute of a logical data model.

3.2.10 *concept, n*—a unit of thought constituted through abstraction on the basis of characteristics common to a set of objects (see ISO/IEC 1087).

3.2.11 *conceptual model, n*—an abstract representation of any type of model used to identify the principal components and relationships of the subject of the model, and avoiding unnecessary or confounding detail.

3.2.12 *context, n*—a designation or description of the application environment or discipline in which a name is applied or from which it originates (see ISO/IEC 11179).

3.2.13 *data, n*—a representation of facts, concepts, or instructions in a formalized manner, suitable for communication, interpretation, or processing by humans or automatic means (see ISO/IEC 11179).

3.2.14 *data dictionary, n*—a database used for data that refers to the use and structure of other data; that is, a database for the storage of metadata (see ANSI X3.172-1990).

3.2.15 *data element, n*—a unit of data for which the definition, identification, representation and permissible values are specified by means of a set of attributes (see ISO/IEC 11179).

3.2.16 *data model, n*—a description of the organization of data in a manner that reflects an information structure (see ISO/IEC 11179).

3.2.17 *data steward, n*—a person or organization delegated the responsibility for managing a specific set of data resources, (see ISO/IEC 11179).

3.2.18 *dependent entity, n*—in a logical data model, an entity that inherits one or more identifying attributes from another entity.

3.2.19 *domain, n*—the set of possible data values of an attribute (see ISO/IEC 2382); also the identification, description or scope of a segment of industry or commerce, or of a program, project, or problem.

3.2.20 *encapsulation, v*—the packaging of an object's data with its corresponding methods with access via a controlled messaging interface (see UML 1.4).

3.2.21 *entity, n*—any concrete or abstract thing of interest, including associations among things (see ISO/IEC 2382).

3.2.22 *external schema, n*—an external schema is the representation of data at the system architecture presentation layer as viewed by the user when interacting or communicating with the information system.[5]

3.2.23 *foreign key attribute, n*—in a logical data model, an attribute or combination of attributes of a child or category entity whose values match those in a primary key of a related or generic entity instance.

3.2.24 *function, n*—an activity, process or transformation identified by a verb that describes that which would be accomplished.

3.2.25 *independent entity, n*—in a logical data model, an entity that does not inherit identifying attributes from another entity.

3.2.26 *inheritance, v*—the transmission of characteristics from a parent process, entity, or object to its children (see UML 1.4).

3.2.27 *internal schema, n*—a schema of the ANSI Three Schema architecture in which views of the information are representations of data structure at the system architecture data layer.

3.2.28 *key attribute, n*—in a logical data model, an attribute used to identify an instance of an entity.

3.2.29 *key inheritance, v*—the transmission of a key attribute from a parent or independent entity to a child or dependent entity.

3.2.30 *lexical, n*—pertaining to words or the vocabulary of a language as distinguished from its grammar and construction (see ISO/IEC 11179).

3.2.31 *location model, n*—a representation or description of the components and interrelationships of a real or virtual location.

3.2.32 *logical model, n*—an expansion of a conceptual model, or a derivation from a physical model, that provides a level of detail sufficient to design or effect a business solution.

3.2.33 *metadata, n*—data that describes other data (see ISO/IEC 11179).

3.2.34 *model, n*—a representation in whatever form of a real or envisioned object, action, or process.

3.2.35 *modeling, v*—the process of creating, revising, documenting and presenting a model.

3.2.36 *model view, n*—a collection of models pertaining to a particular domain and from the perspective of a particular individual's role or viewpoint.

3.2.37 *non-key attribute, n*—in a logical data model, an attribute that is not the primary or part of a composite primary key of an entity.

3.2.38 *object, n*—any part of a conceivable or perceivable world (see ISO 1087).

3.2.39 *object class, n*—a set of objects, ideas, abstractions, or things in the real world that can be identified with explicit boundaries and meaning and whose properties and behavior follow the same rules (see ISO/IEC 11179).

3.2.40 *object-oriented methodology, n*—any of the formal modeling methods or techniques that employ objects to describe data and the methods or processes acting on those data as a single unit for analysis, design, or development (see UML 1.5, ISO/IEC 11179).

3.2.41 *organizational model, n*—a representation or description of the components and interrelationships of an organization or organizational component.

[5] ANSI Standards Planning and Requirements Committee—a layered model of database architecture comprising a physical schema, a conceptual schema, and user views.

3.2.42 *physical model, n*—a derivation from a logical model, or a physical implementation, providing a highly detailed definition of the solution needed to carry out the activities in business area.

3.2.43 *primary key attribute, n*—in a logical data model, the candidate key selected as the unique identifier of an entity.

3.2.44 *property, n*—a peculiarity common to all members of an object class (see ISO/IEC 11179).

3.2.45 *relational data methodology, n*—any of several modeling and information management methods and techniques that employ structured relationships among entities or tables.

3.2.46 *row, n*—a physical data model and relational database structure that contains a single instance of data in a table.

3.2.47 *structured analysis and design, n*— any formalized methodology for the analysis, design, development or lifecycle management of information systems where processes and data are initially treated individually and subsequently integrated in a system architecture.

3.2.48 *subject area, n*—a subject area is a portion of an entire data model which is created to facilitate understanding of a specific functional area or component task.

3.2.49 *table, n*—a physical data model and relational database structure that is analogous to the entity of a data model.

3.2.50 *technology model, n*—a representation or description of the hardware, system software, and network components needed to support the business area.

3.2.51 *view, n*—a collection SQL queries stored in a relational database under assigned names and represented as a temporary or virtual table that does not permanently store the data it presents.

3.3 *Acronyms:*

3.3.1 *ANSI*—American National Standards Institute

3.3.2 *ASTM*—American Society for Testing and Materials

3.3.3 *CORBA*—Common Object Request Broker Architecture

3.3.4 *CRC*—Class, Responsibility and Collaboration Approach

3.3.5 *DFD*—Data Flow Diagram

3.3.6 *ICOM*—Input, Control, Output and Mechanisms as used in IDEF activity modeling

3.3.7 *IDEF*—Integrated Definition Language

3.3.8 *IDEF0*—The IDEF activity modeling language

3.3.9 *IDEF1X*—The IDEF data modeling language

3.3.10 *IDO*—Integrated Delivery Organization

3.3.11 *IE*—Information Engineering diagramming methodologies

3.3.12 *IEC*—International Electrotechnical Commission

3.3.13 *ISA*—Information Systems Architecture

3.3.14 *ISO*—International Organization for Standardization

3.3.15 *NIST*—National Institute of Standards and Technology

3.3.16 *OMG*—Object Management Group

3.3.17 *RDB*—Relational Database

3.3.18 *RDBMS*—Relational Database Management System

3.3.19 *SADT*—Structured Analysis and Design Technique

3.3.20 *SML*—Standardized Modeling Language

3.3.21 *SQL*—Structured Query Language

3.3.22 *SSADM*—Structured Systems Analysis and Design Methodology

3.3.23 *UML*—Unified Modeling Language

4. Summary of Practice

4.1 This practice describes policies and procedures, best practices for the creation, use, and assessment of information models.

4.2 The foundation for these information modeling best practices is derived from Donnabedian's structure, process and outcome quality constructs (**1**).[6]

4.2.1 *Structure*—Structure constructs identify or describe the organization component where modeling activities occur, the composition of the modeling method or tool, its language and syntax, and the resources allocated to perform modeling processes, for example human resources, technology, materiel, etc.

4.2.2 *Process*—Process constructs identify or describe the appropriate use of modeling, the effective use of modeling methods, and the correct employment of the modeling technique.

4.2.3 *Outcome*—A quality information model that provides a meaningful and usable representation of past, present, or future reality, and can be assessed by five dimensions (**2**) of model quality:

4.2.3.1 *Conceptual Correctness*—The model accurately reflects functional or business concepts where all models are representations of real or envisioned objects or concepts. The model shall accurately describe the intended structures or processes or both.

4.2.3.2 *Conceptual Completeness*—The model contains sufficient objects to describe the full scope of the functional or business domain under consideration. The model describes the full scope or an element of the structures or processes of a concept or both.

4.2.3.3 *Syntactic Correctness*—The objects in the model do not violate syntactic rules of the modeling language. Each modeling method has a predefined language typically consisting of visual structures and symbols where these structures and symbols are assembled and interpreted according to a clearly defined and unambiguous set of rules. These rules shall be widely recognized in the industry and preferably be standardized (**4**). The modeling methodology and processes shall adhere to these rules. The model shall effectively communicate its representations and descriptions to anyone who understands the modeling method and syntax.

4.2.3.4 *Syntactic Completeness*—All essential functional or business concepts are captured at appropriate points in the

[6] The boldface numbers in parentheses refer to the list of references at the end of this standard.

modeling process and represented in the model products. A modeling method may use different objects, symbols or structures at different times in the modeling process. The model shall display the appropriate types and numbers of structures and symbols at the appropriate place in the model and at the appropriate time in the modeling process.

4.2.3.5 *Enterprise Awareness*—The model depicts the entire enterprise or can be seamlessly integrated with other models across the entire organization. Enterprise awareness is the ability of a model to describe the entire organization, domain, or a comprehensive segment or component of industry, business, or commerce, either by itself or when combined with other comparable models. Multiple models shall seamlessly integrate across the entire organization with representations and descriptions easily flowing from one model to another. The Zachman Information Systems Architecture (**3**) is one of several frameworks suitable for identifying models that represent the informatics structure and processes of the organization. The Zachman Framework consists of six columns and five rows where the columns describe data, function, network, people, time, and motivation, and answer the fundamental questions *what*, *how*, *where*, *who*, *when*, and *why*, and the rows describe different perspectives as contextual scope, conceptual models, logical models, physical design, and detailed representations (physical implementation). The foot of the framework represents the actual business enterprise.

In the framework, an upper row essentially presents requirements for a lower row and each cell can contain directional, conceptual, logical, or physical models or combinations thereof.[7]

4.3 *Additional Technical Considerations:*

4.3.1 In addition to the above quality dimensions, several qualitative and technical considerations impact the quality of an information model (**4**).

4.3.2 *Nonredundancy*—A fact shall be represented at only one point in the model.

4.3.3 *Business Rules Enforcement*—The model accurately reflects real or envisioned functional or business processes.

4.3.4 *Reusability*—The content of the model is accessible to other than the originally intended users.

4.3.5 *Stability*—The model accepts changing functional or business requirements without significant alteration in the structure of the model.

4.3.6 *Flexibility*—The model is capable of supporting change in domain concepts or business processes.

4.3.7 *Simplicity*—The model is easily understandable and clearly presented.

4.3.8 *Implementable*—The model may be readily and economically used in developing strategy, and in the analysis, design, development, or operation of business processes or information systems.

4.3.9 *Communicability*—The model readily conveys meaning to its users and readers.

4.4 *Assessing the Modeling Process and Model Quality:*

4.4.1 The dimensions of model quality that are derived from structure, process, and outcome quality constructs lead to best practices in the following areas:

4.4.1.1 *Structure*—The organization desiring to employ modeling creates and maintains an environment conducive to the creation, evaluation, communication and use of quality information models.

4.4.1.2 *Process*—Modeling methods are appropriately selected and applied through effective use of modeling language and adherence to syntax to provide a meaningful, comprehensive, and balanced representation of real world functional or business concepts.

4.4.1.3 *Outcome*—The resulting model(s) exhibit a high degree of excellence and wholeness, are identified by author, participants, technique and approach, perspective, view and subject areas, and may include test data for evaluation purposes.

4.4.2 Annex A1 of this practice provides a checklist for assessing the quality of models and modeling activities in terms of structure, process, and outcome.

5. Significance and Use

5.1 Modeling is increasingly used in business, industry, and commerce to develop a common understanding of processes, functions, activities, and supporting data. Typical users of such models are systems developers, operations researchers and business analysts, educators, and executives.

5.1.1 Information models are regarded widely as beneficial by saving cost through realignment of processes, risk reduction and elimination of redundancy. Information models convey ideas and facilitate the analysis and understanding of complex processes and structures. These models form the basis for software engineering practices that build systems and databases, redefine organizational structures, improve business processes, and develop standards.

5.1.2 This practice provides a practical means for developers and users of information models to employ appropriate modeling methods and to objectively determine model quality.

5.2 *Background:*

5.2.1 Models are representations of past, existing, or contemplated reality. Models may assist in the explanation or analysis of complex structures and processes that may exceed human capacity for direct visualization or understanding. Models enable a focus on the key elements of a process or structure while ignoring confounding or irrelevant elements. As such, models make an explicit statement of the meaning of the reality being modeled.

5.2.2 Integrated information engineering models provide a coherent view of the processes and data of an organization or enterprise (**5**). Activity models identify the fundamental tasks performed in a function. Process models accurately describe the detailed collection of these activities within an organization. Data models are derived from and support the functions described in activity models. Object models also characterize the processes and data required to understand business operations. Both structured and object-oriented models may be used to construct information systems that support those business operations. Application models describe in varying levels of

[7] Additional detail about the Zachman Framework is available at http://www.zifa.com/.

detail, the overall architecture and components of the software needed to support the envisioned business functions. Organizational models reflect the current and envisioned future state of the organization, especially as this impacts business processes enabled or supported by information systems. Location models identify and describe the business and geographic position and relationships of structural components of an organizational entity. Technology models describe in varying levels of detail those hardware, system software, and network components needed to operate the information systems supporting a business area.

Models may be textual, graphic, or mixed graphic and text forms, including, tables and structured lists, flowcharts, process flow diagrams, state diagrams, data flow diagrams, entity-relationship diagrams, and related techniques develop an understanding of business processes and the transformation of data through these processes (**6**). Modeling products may also relate two or more types of models, such as for an Application-Data Matrix, or Technology-Location Plan.

5.3 *Framework for Relating Models to Systems:*

5.3.1 Since proposed by Zachman in 1987 (**3**), the Information System Architecture (ISA) has become one of several means to view and to blueprint information systems within an entire organization, to design business processes and change enterprise strategies, and to describe and interrelate the components of an information system to the entire organization. The cells within the framework represent primitive architectural constructs where the sets of columns collectively represent a consistent integrated model for each perspective (row). These rows describe different types of artifacts, each having a different purpose:

5.3.1.1 Row 1—artifacts define the scope as boundaries of the enterprise.

5.3.1.2 Row 2—artifacts define the conceptual design as envisioned by the enterprise "owners" or stakeholders.

5.3.1.3 Row 3—artifacts present a logical design, how the enterprise concepts will be realized independent of technology.

5.3.1.4 Row 4—artifacts describe the physical design, how the enterprise implementation is generally constrained by technology.

5.3.1.5 Row 5—artifacts specify the physical implementation as a specific application of technology.

5.4 *Principles and Approaches to Modeling:*

5.4.1 Modeling activities have been prevalent throughout the history of the engineering disciplines. This rich history indicates four basic principles (**7**):

5.4.1.1 The choice of what models to create has a profound influence on how a problem is approached and how a solution is shaped.

5.4.1.2 Every model may be expressed at different levels of precision.

5.4.1.3 The best models are connected to past, current, or envisioned future reality.

5.4.1.4 No single model is sufficient; every nontrivial system is best approached through a small set of nearly independent models.

5.4.2 The three approaches to information modeling are characterized as top-down, bottom-up, and inside-out (**8**). Of these, the top-down and bottom-up approaches are most often used.

5.4.2.1 The "top-down" approach follows a progression of activities that first analyzes and understands functionality of the domain-of-interest and only then progresses to model preparation. The top-down paradigm typically consists of preparing a high level or conceptual model followed successively by logical and physical models in increasing levels of detail. The strength of this approach is it provides a broadly-based and comprehensive set of rationally derived models that are generally appropriate across the spectrum of functional or business activities. This paradigm saves time, conserves resources, and reduces risk over the long-term course of system development.

The "bottom-up" approach reflects the experience of subject matter experts in one or more specialized subdomains. These panels of experts prepare small models that are subsequently integrated. The strength of this approach is that it addresses detail that can be easily converted to a specific application, and may be tuned or optimized for a small and contained segment of the business environment. Often, it is used when the entire domain is not well understood or in the absence of a business model foundation. The bottom-up paradigm solves small problems quickly. It produces narrowly focused model components requiring extensive effort for subsequent integration into systems with a broader scope or enterprise solutions.

5.5 *Use of Modeling Techniques for Content*—The complexity and extensiveness of business processes, the information used in these processes, and the information systems supporting these processes inhibit or prevent a comprehensive understanding to the depth of detail frequently required. Formal methods are needed to facilitate the development and sharing of a common understanding about information, the processing of that information, and the systems supporting those processes.

5.6 *Use of Models for System Engineering:*

5.6.1 System engineering is considered in all leading system lifecycle methodologies as a sequential, cyclic, or iterative sequence of four essential activities from a visionary or inception activity to deployment or implementation (**9**) as illustrated in Fig. 1. Performance and outcome of these activities can be substantially improved through modeling. For example, system development models may consist of elementary structures like an organizational chart to detailed structural representation of a software package. Models typically are employed in the planning and analysis phase of software development, in system and software design, and in the coding process.

5.6.1.1 Booch, et al (**7**) , state that the single fundamental reason for modeling a system is to better understand the system being developed, achieving four aims:

(1) To visualize an existing or desired system;

(2) To specify the structure and behavior of a system;

(3) To provide a guiding template for system construction; and

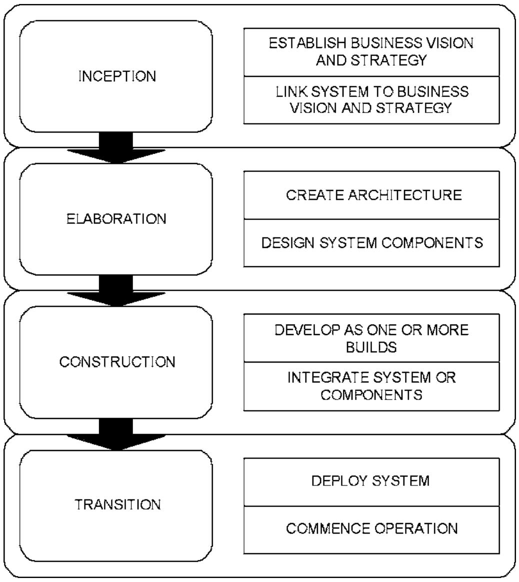

FIG. 1 Typical System Life Cycle Phases

(4) To document the decisions made in system development.

5.6.1.2 Pressman notes three additional reasons for modeling (**10**) in system development:

(1) The business problem and solution can be approached incrementally, thus reducing the probability that a major feature will be misinterpreted or overlooked;

(2) The customer or user can more easily review the analysis and design, to help preclude or eliminate areas of ambiguity and misunderstanding; and

(3) The development proceeds in a logical, efficient and more readily manageable manner through the migration of models from analysis through design and development.

Most importantly, models greatly enhance quality, reduce risk, and multiply capability in system development. Models enable the developer to proactively build-in quality during design rather than reactively assess quality during development. As models are transformed from analysis through physical software, sets of quality criteria can be created for each modeling step. As the models are reviewed against these criteria incrementally risk of failure decreases. Errors in requirements definition, design, and development can be uncovered easily and corrected at the earliest possible time with less effort and delay than if corrected during module or integration testing. Use of modeling tools, as in model-driven design and development, greatly magnifies the productivity of individual designers and developers.

5.6.2 *Modeling in Systems Inception*—Analytical models are essential to the systems inception and planning. In the inception phase, models help develop the business rationale and communicate this rationale in the system business case. Semantic and diagrammatic models help explain the thought processes leading to a system proposal. Activity models are useful to design future business activities and business process improvement. Financial models and simulations are important components of the business case and risk prediction (**11**) , and for software engineering project management (**12, 13**), and estimation (**14**).

5.6.3 *Modeling in Systems Elaboration*—The elaboration phase of the systems development paradigm includes analysis and high level design. Models are essential to system elaboration by analyzing, recording, and detailing business process and system functionality. Typical examples of analytical and high level design models include business process flow diagrams, conceptual entity models, conceptual technology models, etc.

5.6.4 *Migration of Models in System Construction*—A software engineering methodology provides development paradigms as a procedural framework for the design, construction, testing and implementation of systems. Within a development methodology, modeling methods prescribe the symbology and heuristics used to create models of varying levels of detail. Automation, via computer-assisted software engineering (CASE) tools, facilitates the modeling process and provides the vehicle that achieves transformability. The transformation from analysis to a functioning system is termed forward engineering while transformation from a functioning system to analytical or design models is termed reverse engineering.

The benefit of using standardized modeling methods is the software models that are created are easily understandable and readily transformable to a functioning system (**10**). As illustrated in Fig. 2, models follow a logical progression from Conceptual to Logical to Physical (**8, 15**) . Fig. 2 also illustrates the relationship between system lifecycle phases and Zachman Framework, along with representative types of business process and data models. This cascade of models facilitates the understanding of concepts and the implementation of information applications, databases or systems.

The systems development process originates with either an existing system in need of improvement or a vision of a new or reengineered system. The chain of models begins with conceptual modeling where activities and data are modeled at a high level of detail.

5.6.5 *Organizational Issues*—The structure and process elements of the model quality paradigm involve the creation and maintenance of an environment and procedures that facilitates creation, evaluation, communication, and utilization of information models. For the system development organization there should be a component responsible for the creation and maintenance of models. The organization should establish policies for modeling and standardize the model development process, methodologies, and tool sets. These policies constitute internal, or organizational standards or conventions, consisting of the following:

5.6.5.1 *Syntactic Conventions*—The library of model symbols to be employed;

5.6.5.2 *Positional Conventions*—The manner in which symbols are displayed or presented in the model; and

5.6.5.3 *Semantic Conventions*—The grouping of model components according to meaning.

5.6.5.4 The organization should provide a battery of standardized modeling methods from which the software developers can select as appropriate to the needs of a specific development project.

5.7 *Using Models for Standards Development:*

Zachman CONSTRUCT	LIFECYCLE PHASE	TYPICAL BUSINESS PROCESS MODELS	TYPICAL DATA MODELS
Scope / Context	Inception	Business Strategy	Data Direction Model
Conceptual Models	Elaboration	Conceptual Process Models	Conceptual Data Model
Logical Models		Logical Process Model	Logical Data Model
Physical Models	Construction	Physical Process Model	Physical Data Model
Component Models		Business Processes Instructions	DDL Script
	Transition		

FIG. 2 Migration of Models in System Development

5.7.1 Organizations may develop their own internal standards, such as standard operating procedures, or they may develop standards for use by the public, an industry or professional community, or other group of consumers. Modeling facilitates the development, communication and understanding of both internal and industry-wide standards. Developers of internal and external standards may use models to prepare and present a common reference to their work. Furthermore, consumers may find models facilitate the implementation and use of standards by their organizations by providing standards users a blueprint from which they can tailor and optimize model components for implementation. Here adherence to the structure, process, and outcome quality requirements shall be especially rigorous since the models produced provide the foundation for standards that may have substantial industry impact.

6. Best Practices for Information Modeling (Normative)

6.1 *Foundation Practices*—A quality model shall have a set of descriptive foundation features to aid in the preparation, communication and understanding of the model and to provide traceability of the decisions made in the evolution of the model.

6.1.1 *Model Identification*—Every model shall be identified by the following parameters:

6.1.1.1 Name or title of the model,

6.1.1.2 Version number of the model,

6.1.1.3 Name of the person who created or last edited the model, and

6.1.1.4 Date on which the model was created or last edited.

6.1.2 The above parameters should be presented as required by the selected modeling technique or notation system (**16**); otherwise present the model parameters as shown in Table 1.

TABLE 1 Example Model Identification Table

Identification Parameter	Example Identification Entry
Model name	Reimbursement Logical Data Model
Version	2.3
Author/editor	Anna Phalaxis
Date created/edited	20051130

6.1.3 Additional optional information, such as sponsoring organization, may be included. The format for the date entry should be ISO 8601 compliant (**17**) . Some modeling methods and CASE tools, such as for the IDEF1X data modeling method, present a standardized "kit" that includes these identification parameters.

6.1.4 *Modeling Technique*—Model builders shall state the modeling technique or notation system used to create the model.

6.1.5 *Approach*—Model builders should include the approach (as top-down or bottom-up) for reference.

6.1.6 *Views and Perspectives*—Model builders shall include the viewpoint for reference. For example, a data model constructed from the viewpoint of a system to be used by a computer operator will appear differently than a data model of the same system when used by a business manager.

6.1.7 *Clarity of Presentation*—Models shall be understandable and convenient to use. Objects shall be sized and arranged on the screen or page to be easily read at normal reading distance. Where the model is too large and or too complex for clear presentation, it should be decomposed into multiple submodels, subject areas, or other logical groupings of diagram objects.

6.1.8 *Including Sample Data for Quality Assessment*—Models, such as business process and data models, may be illustrated, exercised or tested through a process called instantiation. Instantiation involves insertion of real world values into the model elements for explanation and testing purposes. For example, test data may be applied to both logical and physical data models to facilitate testing and validation. Likewise, test financial data may be included in an activity model to validate activity-based costing features.

6.1.9 *Adherence to Standards:*

6.1.9.1 At present, only UML and the IDEF methods and notations are standardized, and only the IDEF notation is standardized by an ANSI-accredited voluntary consensus standards organization. Other traditional model notations and methods, such as described by Martin (**6**) , are not addressed in ANSI, ISO, or OMG standards and are not governed by formally constituted standards development organizations.

Where standards apply, such as for IDEF or UML modeling, both the modeling process and model products shall fully comply with the provisions of the relevant standards, and the specific standards shall be referenced in the model documentation.

6.2 *Best Practices by Type of Model*—Table 2 presents a summary of best practices as quality dimensions by type of model.

6.2.1 *Organizational Models*—Organizational models are frequently used to represent, analyze and communicate features of an organization, to represent the relationship of stakeholders to a project, to summarize the organizational change principles, etc. Organizational models may include hierarchical organization charts, role-participation diagrams, team networking diagrams, etc. The collection of organizational models provides a core understanding of the organization's management and control architecture. Quality dimensions for organizational models include the following:

6.2.1.1 Conceptual completeness requires representation of all essential organizational features, or components, such as departments or individual positions, along with the direct (solid line) and indirect (dotted line) relationships among these components.

6.2.1.2 Conceptual correctness requires accurate identification and description of these organizational elements.

6.2.1.3 Syntactic completeness requires all organizational components, their interconnections, and their relationships to other elements under study or analysis, be uniformly and completely identified.

6.2.1.4 Syntactic correctness requires all organizational features, components and their interconnections to be presented clearly and understandably.

6.2.1.5 Enterprise awareness requires a holistic view of the organization as a single enterprise, or a complete component within the organization.

6.2.2 *Location Models*—Location models provide a conceptual, logical, and physical representation of a business or business function. Location models may summarize or list business ground rules, such as where a customer will interact with a particular business function. Location models identify and concisely present assumptions and constraints as these apply to business and business processes. Location models may identify and describe facility inventories, illustrate the distribution of business processes by geography, and provide increasing levels of detailed description of geolocations and facilities. Quality dimensions for Location Models include the following:

6.2.2.1 Conceptual completeness requires representation of all essential location components, such as geographic and landscape features, buildings, components within buildings, etc.

6.2.2.2 Conceptual correctness requires accurate identification and description of these components and their connection to other components.

6.2.2.3 Syntactic completeness requires all appropriate location structural and performance specifications be included.

6.2.2.4 Syntactic correctness requires all physical components and connectivity be identified and documented consistent with standards and industry best practices.

6.2.2.5 Enterprise awareness requires the location model uniformly describe the enterprise, or discrete business components within the enterprise as a whole, contributing toward an integrated view of the entire business entity.

6.2.3 *Business Process Models*—Business process models provide a function-driven approach to describe the activities performed by an organization (**18**). Business process models are also called functional models. Business process models identify the events to which the business shall respond and the results produced. Business process models present both the current, or "as-is," process along with the future, or "to-be," process. Business process models reflect the activities and tasks performed at all levels of an organization. These models may represent the activities alone, activities combined with data or activities with measures, such as cost breakouts (**19**). The most basic business process model is the semantic model, a narrative representation of activities or tasks performed in an organization. The semantic model is used frequently to provide an ordered list of system functional requirements or business rules. Graphical business process models (**6**) may include conventional flowcharts, process models, use cases, data flow diagrams, process flow diagrams, etc. These modeling methods are well described in the literature (**6, 20**). Quality dimensions for Business Process Models include the following:

6.2.3.1 Conceptual completeness requires representation of all essential technology components, such as servers, routers, etc.

6.2.3.2 Conceptual correctness requires accurate identification and description of these technology components and their interconnections.

6.2.3.3 Syntactic completeness requires all appropriate technical and performance specifications be included.

6.2.3.4 Syntactic correctness requires all technology components and connectivity be presented consistent with industry best practices.

6.2.3.5 Enterprise awareness requires an enterprise view of the organization's business processes or a complete view of those processes performed in an organizational component, best represented as a top-down decomposition.

6.2.4 *Data Models*—The data-driven approach to modeling also is known as the Information Engineering Approach (**5**). Data schema diagrams were presented by Bachman in the

TABLE 2 Quality Dimensions by Model Type

Quality Dimension	Characteristics	Examples of Quality Dimensions by Model Type		
		Organizational Models	Location Models	Business Process Models
Enterprise awareness	The model depicts the entire enterprise, or a component that can be seamlessly integrated with other models across the entire enterprise.	An organization chart depicting the executives and their reporting relationships for the entire company.	An organization's shipping map identifies warehouses, processing and distribution routes in a geographic region.	The model describes business processes covering the entire organization.
Conceptual completeness	The model contains sufficient objects to describe the full scope of the target domain.	All executive management positions in the organization are represented.	The distribution illustrates all facilities and shipping routes.	The model presents a hierarchical breakdown of all essential components of the business processes.
Conceptual correctness	The model accurately reflects the concepts represented.	All individual names, positions, contact information, and positional interrelationships are correct and up to date.	All facility names, addresses, points of contact, etc., are current and correct.	The process model is an accurate representation of the current or envisioned processes, component activities, and their interrelationships.
Syntactic completeness	All important concepts are captured at appropriate points in the modeling process and represented in the model product.	The organization chart identifies all executive roles and relationships, the incumbents, and contact information.	The distribution map identifies all essential features like warehouse locations and distribution hierarchies.	The process model is decomposed to a level of sufficient detail to facilitate understanding at the level required by the intended readers.
Syntactic correctness	The objects displayed in the model conform to or are consistent with the syntactic rules of the modeling language.	The organizational chart is presented as a hierarchical wiring diagram consistent with industry best practices.	The distribution information is presented consistent with industry mapping best practices and USPS standards.	The model conforms to the IDEF0 standard for business process modeling.
Quality Dimension	**Characteristics**	**Examples of Quality Dimensions by Model Type**		
		Data Models	**Application Models**	**Technology Models**
Enterprise awareness	The model depicts the entire enterprise, or a component that can be seamlessly integrated with other models across the entire enterprise.	A conceptual entity-model diagram includes entities that support the business processes of the entire organization.	The application model illustrates the software architecture of an organization's commercial web portal.	A server and network diagram illustrates all major servers and their interconnections across the entire organization.
Conceptual completeness	The model contains sufficient objects to describe the full scope of the target domain.	A fully attributed logical data model presents all entities and attributes required for the enterprise materials management data system.	The model presents all objects required to comprehensively describe the application code enabling the business processes supported.	The technology model identifies all servers, network components, firewalls and external access points, etc. their interconnections, identification and physical and performance characteristics.
Conceptual correctness	The model accurately reflects the concepts represented.	A conceptual entity model diagram contains entities, and illustrates relationships among these entities to accurately describe relationships of business objects.	The application model accurately identifies the key components of the organization's web portal along with the relationships among these components.	The technology model employs graphic devices to accurately represent servers, network components, etc. according to industry best practices.
Syntactic completeness	All important concepts are captured at appropriate points in the modeling process and represented in the model product.	The logical data model is the product of a series of Joint Application Development sessions that were held to iteratively prepare a comprehensive conceptual entity model, a key-based logical data model and a fully attributed logical data model.	The application model identifies and describes all essential components along with their interconnections and process flows, at appropriate levels of detail.	The technology model consists of a nested hierarchy of increasingly detailed components that identify and locate all devices at appropriate levels of detail.
Syntactic correctness	The objects displayed in the model conform to or are consistent with the syntactic rules of the modeling language.	The logical data model conforms to IDEF1X data model standards	The components presented in the application model are consistent with industry best practices.	Equipment and connectivity are depicted consistent with industry best practices.

1960s, and subsequently improved, as rectangles for record types and connecting arrows for relationships (**21, 22**). Following this work, the Entity-Relationship approach was described by Chen as rectangles for entities and diamonds overlaying connecting arrows to detail the type of relationship (**23**) . Data models in particular have become widely recognized as essential to design quality databases (**24, 25**). Some authors have compiled sets of models in other industries that they present as guidelines or references for reusable data structures (**26**). Data Modeling techniques using the IE, Chen, and IDEF1X notations have been well described in the literature (**6, 20**) . In all of the types of data models noted below, enterprise awareness focuses on depicting the entire enterprise or object, or a component that can be seamlessly integrated with other models across the entire organization or component. Quality dimensions for conceptual, logical, and physical data models include the following:

6.2.4.1 *Conceptual Data Model:*

(1) Conceptual completeness requires all essential entities be represented and all relationships among these entities be

correct representations of the functional interrelationships of the business objects represented in the model.

(2) Conceptual correctness requires all essential entities be derived from and support a prerequisite conceptual activity model.

(3) Syntactic completeness requires all entities and attributes are defined or described for clarity of understanding.

(4) Syntactic correctness requires that there is strict adherence to the modeling methodology requirements.

(5) Enterprise awareness requires a complete and overarching representation of the key and essential data required to meet the business needs of the organization or organizational component.

6.2.4.2 *Logical Data Model*—A logical data model flushes out the detail of a conceptual data model without consideration of implementation technology. Logical data models may be represented as key-based models that follow the migration of keys through a data structure, and fully attributed data models that provide a comprehensive view of highly structured data required to support and enable business process.

6.2.4.3 *Key-based Logical Data Model:*

(1) Conceptual completeness requires all entities reflect or expand upon those in the precursor conceptual data model.

(2) Conceptual correctness requires candidate entity keys are appropriate, descriptive, and unique.

(3) Syntactic completeness requires that all many-to-many relationships are converted through interposition of associative entities, and all entities, key attributes, and relationships are defined or described.

(4) Syntactic correctness requires all entities and relationships are normalized through the Third Normal Form; referential integrity is maintained through proper migration of all key attributes throughout the model; and the model presentation strictly adheres to the modeling methodology employed.

(5) Enterprise awareness requires a complete view of all data entities and primary keys required to support all targeted business functions in the enterprise.

6.2.4.4 *Fully Attributed Logical Data Model:*

(1) Conceptual completeness requires that the fully-attributed model provides detail that expands upon the precursor key-based logical data model; all reference entities contributing identifying and nonidentifying foreign keys are displayed; and all required descriptive detail is contained in non-key attributes.

(2) Conceptual correctness requires that all non-key attributes contribute detail to the appropriate entities, and that entity subtypes appropriately reflect choices in populating data structures.

(3) Syntactic completeness requires that all reference entities are named and defined; all reference entity foreign key and non-key attributes are named and defined; and all subtype delimiters, entities, and attributes are named and defined.

(4) Syntactic correctness requires normalization through the Third Normal Form is preserved; referential integrity is maintained through proper migration of all identifying and nonidentifying key attributes throughout the model; all entities, key attributes, and relationships are defined or described; and the model presentation strictly adheres to the modeling methodology employed.

(5) Enterprise awareness requires all entities, attributes and relationships are represented sufficient to enable and support all intended enterprise business processes.

6.2.4.5 *Physical Data Model*—Physical data models expand upon the fully attributed logical data model to provide detail sufficient to implement a relational database or construct objects.

(1) Conceptual completeness requires preservation of the data structures developed in the precursor logical data model, and addition of metadata appropriate for the target database management system.

(2) Conceptual correctness requires metadata such as field length be appropriately specified for the intended function of the attribute.

(3) Syntactic completeness requires all metadata be included in the model.

(4) Syntactic correctness requires specification of metadata consistent with the target database management system and modeling methodology.

(5) Enterprise awareness requires all databases, tablespaces, tables, columns, and constraints exist to fully and efficiently integrate data across the total enterprise or within a defined component of the enterprise.

6.2.5 *Application Models*—Application models describe, in varying levels of detail, the application software needed to support business functions. Various types, stages and representations of application models exist. The Application Diagnostic Model summarizes and evaluates existing systems, and defines characteristics of envisioned systems. A Conceptual Application Model describes an overall application architecture, defining individual applications, and assigning business processes to applications. A Logical Application Model defines in increasing detail the subsystems within applications, the interfaces among applications and subsystems as well as interfaces to external systems. The User System Interface Model describes the physical appearance of the screens, reports, and menus that make up the user system interface. The Physical Application Model defines the physical software modules and interface files that make up the application. Quality dimensions for Application Models include the following:

(1) Conceptual completeness requires representation of all essential application components, such as modules, functions etc.

(2) Conceptual correctness requires accurate identification and description of these processing components and their interfaces to other components.

(3) Syntactic completeness requires all appropriate program structural and performance specifications be included.

(4) Syntactic correctness requires all components and connectivity be coded and documented consistent with industry best practices.

(5) Enterprise awareness requires that applications and processing components be integrated across the enterprise or seamlessly within a component of the enterprise.

6.2.6 *Technology Models*—Technology models provide a readily understandable view of the current and envisioned technical infrastructure of an organization, facility, or other entity. Technology models identify general technical capabilities and relate these to other domains such as application, business process, or data. Technology models exist at a conceptual, logical, and physical level. At a conceptual level, these models establish a high-level technical architecture and direction to achieve business goals. At the logical level, technology models define processing nodes and interconnections in terms of capacity requirements, reliability requirements, service capabilities, and standards. The physical technology model presents detailed physical characteristics of the technical infrastructure in sufficient detail to build and integrate its components. For example, a performance engineering technical model documents the volumes, service levels, resource usage, and performance estimates that drive the technical configuration and sizing of the technical infrastructure. Quality dimensions for Technology Models include the following:

(1) Conceptual completeness requires representation of all essential technology components, such as servers, routers, etc.

(2) Conceptual correctness requires accurate identification and description of these technology components and their interconnections.

(3) Syntactic completeness requires all appropriate technical and performance specifications be included.

(4) Syntactic correctness requires all technology components and connectivity be presented consistent with industry best practices.

(5) Enterprise awareness requires all technology components be represented to enable and support business process, communication, and data integration across the enterprise or uniformly within a defined component of the enterprise.

6.3 *Model Classification and Alignment:*

6.3.1 *Model Classification*—The above models may be classified as *Business Models*, consisting of Business Process Models, Organization Models, and Location Models, or *Technical Models*, consisting of Application Models, Data Models, and Technology Models.

6.3.2 Alignment of the above types of models ensures a close fit between business and technology as cited by Strassman and many others (**11**). In business process improvement and reengineering, for example, this alignment of models creates the business and technical blueprint for the creation of information systems to best support future business need, and the roadmap for the reorientation of business to best exploit the capabilities and benefits of information technology.

6.4 *Interfaces Among Types of Models*—The interfaces between types of models contains products that further clarify and describe the subject of the modeling activity. Table 3 illustrates the types of products, as matrices, that exist at the interfaces between model types. Examples of work products produced at these interfaces are:

6.4.1 *Organization-Location Matrix*—Organizational components that exist at a specific location.

6.4.2 *Organization-Business Process Matrix*—Business processes performed by organizational components.

6.4.3 *Organization-Data Matrix*—The data structures used by various organizations and their components.

6.4.4 *Organization-Application Matrix*—The programs and software applications employed by organizations and their components.

6.4.5 *Organization-Technology Matrix*—The physical technology, both items and configuration, used by organizational components.

6.4.6 *Location-Business Process Matrix*—The business processes performed at specific locations.

6.4.7 *Location Data Matrix*—The data structures and content created, manipulated, stored, or used at a specific location.

6.4.8 *Location Application Matrix*—The programs or software applications used at a location.

6.4.9 *Location-Technology Matrix*—The identification and configuration of equipment items by location.

6.4.10 *Business Process-Data Matrix*—The data structures and content supporting specific business processes.

6.4.11 *Business Process-Application Matrix*—The programs or software applications that enable business processes.

6.4.12 *Business Process-Technology Matrix*—The physical equipment and its configuration to support business processes.

6.4.13 *Application-Data Matrix*—The data components used by programs or application software.

TABLE 3 Interfaces Among Types of Models

Type of Model	Technology	Application	Data	Business Process	Location
Organization	Organization-Technology Matrix	Organization-Application Matrix	Organization-Data Matrix	Organization-Business Process Matrix	Organization-Location Matrix
Location	Technology-Location Matrix	Application-Location Matrix	Data-Location Matrix	Business Process-Location Matrix	
Business Process	Business Process-Technology Matrix	Application-Business Process Matrix	Business Process-Data Matrix		
Data	Technology-Data Matrix	Application-Data Matrix			
Application	Application-Technology Matrix				

6.4.14 *Data-Technology Matrix*—The data components and data characteristics stored and manipulated by technology components.

6.4.15 *Application-Technology Matrix*—The nature of the physical platforms and their configurations required to operate programs and software applications.

7. Practical Considerations in Modeling

7.1 The use of modeling to solve real-world problems should be approached as the strict use of methodology tempered by practical considerations.

7.2 *Understanding Modeling Anomalies:*

7.2.1 *Feature Interaction*—The phenomenon of feature interaction refers to semantic inconsistencies that are unrecognized at the time of assembly of components into a system. Feature interaction manifests as aberrant behavior during testing or operation. Modeling, especially from differing perspectives, may reveal these semantic inconsistencies and indicate appropriate changes in the component models to produce consistent behavior.

7.2.2 *Model Perspectives*—Models are only representations of reality, and the viewpoints and perspectives of that reality differ. For example, a business process model of clinic operation will appear differently depending upon whether it was constructed from a clinician's, administrator's, or patient's viewpoint, yet if prepared correctly all three models are accurate representations of the same reality. Likewise all three models would be valuable contributions to the design of an information system intended to support that clinic operation. Models should be expected to have differing appearance and meaning depending upon the perspective from which these are derived.

7.3 *Modeling to Understand Infrastructure Requirements:*

7.3.1 An understanding of the information technical architecture typically is developed early in the systems life cycle. Modeling facilitates this understanding by enabling the representation of large and complex concepts in a form more easily understood. Models have an essential role in accepted best practices in software engineering and formal system life cycle methodologies.

7.3.2 A variety of methods, techniques, and tools may be employed to produce clear, comprehensive, and explicit understanding of the business and technology environment. The types of models previously identified accommodate the evolution of the enterprise's operations and facilitate integration of the information architecture with the changing business environment. Such models are applicable to and usable by all constituencies within the organization.

7.4 *Modeling in Reengineering and Business Process Improvement*—Implicit in the term reengineering, is the intention to actively reconstruct the enterprise business processes in a new form. Examination and analysis of the business means representing both the operational and the management dimensions of the business information domain. This representation shall be performed in an integrated fashion using consistent notations that depict both concepts and relationships so that both dimensions can be clearly understood. The considerations provided elsewhere in this document involve the selection of methods, tools, and techniques that enable an efficient and effective representation of the enterprise. This representation is the starting point for the transition from conceptual understanding to implementing an information architecture designed to serve the future enterprise business functions.

7.5 *Modeling in Data Representation*—In an enterprise, models present a relatively static view of the structures and relationships among those concepts comprising the enterprise. Process, activity, and data models typically are employed and present trade-offs in the way that concepts are handled. The data model depicts the structural representation, for example, vocabulary, images, graphics, etc., as well as the relationships among those represented concepts. The data model, particularly in large and complex enterprises or functions, follows the understanding built by modeling the enterprise or function processes and activities.

7.6 *Modeling in Managing Security, Privacy, and Confidentiality of Personal Information*—Modeling in the realm of privacy/confidentiality and security involves employing various techniques to identify those enterprise scenarios, activities, and data where compromises of confidentiality and security may occur and negatively impact the enterprise, its customers, or the population at large. Modeling contributes to effective risk management and the reduction or elimination of those potentially compromising situations. The model developer should build functionality and utility into the model and only then constrain the application for security in implementation.

7.7 *Modeling in Application Development:*

7.7.1 Modeling is a necessary activity in the development of all but the most trivial of applications. Many business domains are so large and complex that modeling is essential to the effective and efficient development of applications, modification and adaptation of software, and in systems integration. The framework for the application, as discussed in 5.5, needs careful definition with respect to the stated enterprise boundaries. This definition will inventory the range of concepts that will be needed in the application area and the common conventions that will need to be considered and modeled.

7.7.2 A very helpful way to organize this information is given in IEEE 1362-98 Concept of Operations, which should precede other life cycle documents that attend the various processes that contribute to the application focal area. While modeling is a major contributor to the requirements engineering aspects in both development and maintenance/evolution of an application, it has utility in the design and construction activities of the life cycle. Because an application may deal with external or ancillary business functions, it shall interoperate with those applications serving such functions, with an overall, enterprise perspective, to avoid the feature interaction effect noted in Summary of Changes *(6)*.

7.8 *Combining Process and Data:*

7.8.1 An object model may be useful to represent information used by an organization or group of organizations sharing nearly identical business processes. The strength of the object model derives from its description of data in the context of processes specific to the organization or organizational component being modeled. The object model facilitates description or design of those system components where process and data are closely intertwined, such as at the presentation tier of system architecture.

7.8.2 The object-oriented approach is ideal for organizations with narrowly focused domains where the object model will highlight opportunities for interoperability and component reuse. The commonality that underpins the object-oriented approach rapidly fades with divergence of these shared business processes. For example, where healthcare business and care delivery processes are geographically bounded and tailored by organizations to meet specific local needs, object models serve best as local representations of that shared reality.

ANNEX

(Mandatory Information)

A1. BEST PRACTICES SCORECARD FOR MODELING

A1.1 The checklist shown in Table A1.1 determines a numerical score for the maturity of modeling processes within an organization.[8] Each item is scored using a zero to five scale, where zero means the item is not present or no part of the capability exists, and five means the item is fully present or maximally used. For example, in the quality requirement 'Organizational Commitment' a score of zero would mean the organization has no policy or intention to use modeling, while a score of five means the organization has demonstrated programs and policies in place to effectively use modeling in support of business goals.

The Final Score gives a decimal value (percentage) representing the maturity of the organization's use of modeling, where 1.0 (100 %) is optimum or ideal modeling maturity.

[8] After Donnabedian's structure, process, and outcome quality constructs.

REFERENCES

(1) Donabedian, A., *Explorations in Quality Assessment and Monitoring*, Vol I, "The definition of quality and approaches to its assessment," Ann Arbor: Health Administration Press, 1980.
(2) Reingruber, M. C. and Gregory, W. W., "The Data Modeling Handbook," New York: Wiley, 1994.
(3) Zachman, J. A., Framework for Information Systems Architecture, *IBM Systems Journal 26/3* , 1987.
(4) Simsion, Graeme, "Data Modeling Essentials," New York:Thomson, 1994.
(5) Martin, J., "Strategic Information Planning Methodologies," 2nd edition, Englewood Cliffs: Prentice Hall, 1989.
(6) Martin, J., "Diagramming Techniques for Analysts and Programmers," Englewood Cliffs: Prentice Hall, 1985.
(7) Booch, G., Rumbaugh, J., and Hacobson, I., "The Unified Modeling Language User Guide," Reading, MA: Addison-Wesley, 1999.
(8) Bruce, T. A., "Designing Quality Databases with IDEF1X Information Models," New York: Dorset House, 1992.
(9) Jacobson, I., Booch, G., and Rumbaugh, J., "The Unified Software Development Process," Reading, MA:Addison-Wesley, 1999.
(10) Pressman, R. S., "A Manager's Guide to Software Engineering," New York: McGraw-Hill, 1993.
(11) Strassmann, P. A., "The Business Value of Computers: An Executive's Guide," New Canaan, CT: Information Economics Press, 1990.
(12) Bruce, P., and Pederson, S. M., "The Software Development Project," New York:Wiley, 1982.
(13) Metzger, Philip W. with John Boddie (contributor), "Managing a Programming Project: People and Processes," Englewood Cliffs: Prentice-Hall ECS Professional, 1985.
(14) Jones, T. C., "Estimating Software Costs," New York:McGraw-Hill, 1998.
(15) Bourke, M. K., "Strategy and Architecture of Health Care Information Systems," New York:Springer, 1994.
(16) Integration Definitions for Functional Modeling (IDEF0) Federal Information Processing Standard 183, Gaithersburg, MD: National Institute of Standards and Technology, 1993.
(17) International Standard ISO 8601 Data elements and interchange formats—Information interchange—Representation of dates and times. First edition 1998-06-15. Ref No ISO 8601: 1988(E).
(18) DeMarco, T., "Structured Analysis and Systems Specification," Englewood Cliffs: Yourdon, Inc., 1978.
(19) Hoffer, J., George, J., and Valacich, J., "Modern Systems Analysis and Design," 2nd ed. Reading: Addison-Wesley, 1998.
(20) Bourke, M. K., "Strategy and Architecture of Health Care Information Systems," New York:Springer, 1994.
(21) Bachman, C., "Data Structure Diagrams," Database 1/2: pp. 4-10, 1969.
(22) Bachman, C., "The Evolution of Storage Structures," Communications of the ACM 15/7: pp. 628-634, July 1972.
(23) Chen, P., "The Entity-Relationship Model—Toward a Unified View of Data," ACM Transactions, Database Systems 1/1: pp. 9-36, Mar 1976 .
(24) Fleming, C., and VanHalle, B., "Handbook of Relational Database Design.," NY: Addison-Wesley, 1989.
(25) Silvershatz, A., Forth, H., and Sudarshan, S., "Database System Concepts," 3rd ed. New York: WCB McGraw-Hill, 1996.
(26) Silverston, L., Inmon, W., and Graziano, K., "The Data Model Resource Book," New York:Wiley, 1997.

TABLE A1.1 Best Practices Scorecard for Modeling

Quality Requirement	Explanation	Score (0-5)
Structure Characteristics		
Executive Buy-In	Organization executives and senior managers understand the need for and benefits of modeling.	
Organizational Commitment	Organization policy is to use modeling for problem solving, system development, future planning, and other activities.	
Stakeholder	A senior member of the organization sponsors the modeling effort.	
Model Master	One individual is accountable for and has the authority to impact modeling activities within the organization.	
Expertise	The organization maintains an appropriate number of modelers at appropriate levels of competence.	
Process Characteristics		
Environment	The organization maintains facility and materiel resources sufficient to accomplish modeling activities.	
Toolsets	The organization has selected, acquired, and maintains current tools appropriate for intended use and independent of unjustifiable personal preference for a specific tool or methodology.	
Internal Standards	The organization has adopted internal standards, policies and procedures for building, maintaining, and using models.	
Version Control	The organization has a formal process for managing model edits and versions.	
Outcome Characteristics		
Complete	The modeling efforts accurately depict and describe the necessary features of the objects being represented.	
Focus	The model efforts omit unnecessary or confounding detail.	
Understandable	The models are readily understandable by readers having a minimum of expertise in the modeling method or technique.	
Compliance	The models conform to standards and best practices for the particular modeling method or technique.	
Documentation	The models are accompanied by sufficient documentation, prepared in a clear and concise manner to facilitate understanding.	
Usefulness	The models are used and contribute to improved business practices, products, or services.	
	Total Score:	
	Final Score (Divide Total by 75):	

SUMMARY OF CHANGES

Committee E31 has identified the location of selected changes to this practice since the last issue, E2145 – 01, that may impact the use of this practice. (Approved June, 1, 2007)

(*1*) Expanded scope of applicability.
(*2*) Enlarged the scope of the document beyond health informatics to address information modeling generally in the information technology domain.
(*3*) Deleted old Section 6, Determining the Need for Modeling, because it was health domain specific. This topic is now addressed more generally in Section 7 and throughout the context of the whole document.
(*4*) Generalized the Zachman ISA description.
(*5*) Revised the types of models in Best Practices.
(*6*) Decomposed examples of Structural Models in the Best Practices section to bring parity among model types typically found in information technology modeling. Types of models addressed are: organizational models, business process models, data models, technology models, application models, location models.
(*7*) Removed old Tables 3, 4, and 5, moved quality dimension content to the text of each model type, and added the enterprise awareness dimension uniformly to each model type.
(*8*) Required compliance with modeling standards.
(*9*) Revised Fig. 1 to generalize the lifecycle beyond one methodology.
(*10*) Moved Object-Oriented modeling to Section 7 and expanded its content.

ASTM International takes no position respecting the validity of any patent rights asserted in connection with any item mentioned in this standard. Users of this standard are expressly advised that determination of the validity of any such patent rights, and the risk of infringement of such rights, are entirely their own responsibility.

This standard is subject to revision at any time by the responsible technical committee and must be reviewed every five years and if not revised, either reapproved or withdrawn. Your comments are invited either for revision of this standard or for additional standards and should be addressed to ASTM International Headquarters. Your comments will receive careful consideration at a meeting of the responsible technical committee, which you may attend. If you feel that your comments have not received a fair hearing you should make your views known to the ASTM Committee on Standards, at the address shown below.

This standard is copyrighted by ASTM International, 100 Barr Harbor Drive, PO Box C700, West Conshohocken, PA 19428-2959, United States. Individual reprints (single or multiple copies) of this standard may be obtained by contacting ASTM at the above address or at 610-832-9585 (phone), 610-832-9555 (fax), or service@astm.org (e-mail); or through the ASTM website (www.astm.org). Permission rights to photocopy the standard may also be secured from the ASTM website (www.astm.org/COPYRIGHT/).