Lexical Conventions for Enterprise Data Modeling

By Ed Johnson
Quality Information Solutions, Inc.
www.QISincorp.com

Table of Contents

Purpose

This document offers lexical conventions for authoring an ***Enterprise Data Model***, also called *Conceptual Data Model*[1]. The overarching aim is to serve as an aspect of governance over the model authoring process.

The lexical conventions for enterprise data modeling herein also aim to promote integrated enterprise definition through

1) Well defined things of significance to the enterprise,
2) Fundamental business facts and rules by which things of significance are related,
3) Enterprisewide data modeling semantics, and
4) Enterprisewide data semantics.

The Enterprise Data Model that reflects these quality characteristics will have high integrity and minimal ambiguity, redundancy, and false representations of the enterprise. It will be the basis for information resource management that ranges from strategic planning and operations management to application development and integration of disparate databases. It will minimize waste and costs from capturing, carrying, and using poor quality data that thwart enterprise flexibility, adaptability, responsiveness, and such other organic qualities for competitive advantage.

Moreover, the lexical conventions herein are independent of information technology, hence are unaffected by frequent changes in information technology. Thus, the conventions are presented to facilitate enterprise definition rather than as standards for database design and application development.

Sometimes enterprise data model or conceptual data model is misunderstood to mean something high-level, something nebulous and loosely defined, as in *concept diagram*. That is not the aim, here. Concept diagrams and diagrams in general typically lack well-defined semantics and structure. Therefore, they usually reflect low integrity in order to facilitate "dumbed down" communications, at best, or to cover the lack of established quality processes to produce quality models, at worst.

[1] The ANSI/SPARC (American National Standards Institute Standards Planning and Requirements Committee) distinguishes three schema types: *Conceptual Schema*, the integrated view of data by the enterprise; *External Schema*, a logical view of data by a particular application; and, *Internal Schema*, a physical view of data by database management systems for a particular application. These three schema types make up the *Three-Schema Architecture*. The Three-Schema Architecture arose from the Two-Schema Architecture, which addresses the external schema and the internal schema, only. The conceptual schema was introduced to address the lack of integration inherent in the Two-Schema Architecture.

Background

The lexical conventions for enterprise data modeling presented in this paper aim for consistency with the standard methodology known as *Integration Definition for Information Modeling, Extended* (IDEF1X, pronounced eye-def-one-ex). Moreover, the authoring of definitions by the lexical conventions herein aims for consistency with definitions in relevant public, regulatory, and industry-specific publications [2, 3, 4, 5].

The U.S. Air Force spurred the development of IDEF1X and the complementary methodology known as *Integration Definition for Function Modeling* (IDEF0, pronounced eye-def-zero) nearly thirty years ago. Both were developed to spur program improvement in the aerospace industry, and both proved remarkably successful. The resulting IDEF system of concepts, procedures, and practices was constituted and standardized with a sense of rigor in matters of semantic modeling of systems. Because they are complementary, IDEF1X and IDEF0 may be used separately or together for a particular enterprise modeling effort.

IDEF1X and IDEF0 are complementary in the sense that IDEF1X deals with **what** the enterprise knows, or cares to know, and IDEF0 deals with **how** the enterprise does what it knows. The enterprise's data embody what it knows, or cares to know, and its processes embody how it does what it knows. The two matters are interdependent; one constrained by the other, one necessarily co-exists with the other.

Today, the U.S. Commerce Department, National Institute of Standards and Technology (NIST), promulgates "normative" versions of IDEF1X and IDEF0 in that the original IDEF concepts, procedures, and practices have been relegated to appendices. Still, IDEF1X remains a rigorous yet easy to learn, graphics-based, semantic modeling language. It provides robust syntax rules and semantic rules for modeling real-world enterprise domains in a way that applies *normalization*[6] with minimal need, if any, to directly engage formal normalization theory.

As a result, various industry and commercial organizations have taken on the IDEF methodologies as standards, de facto standards, and emerging standards. They are in the public domain, free. Significantly, the Institute of Electrical and Electronics Engineers (IEEE) has adopted and adapted both IDEF1X and IDEF0 standards.

The IDEF methodologies have been proven in modernization programs, business process improvement programs, and, in general, quality improvement programs for over twenty years. Successful *pencil-and-paper* use of the IDEF methodologies predates their implementations in software. Success occurred by adhering to the simple yet nontrivial and robust IDEF system of concepts, procedures, and practices. The IDEF system of concepts, procedures, and practices is known to foster common, coherent, and consistent language among and between persons in executive management, operations, and Information Technology organizations. Effective use of the IDEF methodologies is also known to aid *System Thinking*[7] and strategic planning.

IDEF1X has been automated to an effective degree in a few software tools; for example, ERwin,[8] SmartER,[9] and most recently ER/Studio[10]. Unfortunately, Information Technology organizations typically acquire these software tools for use restricted to the technician's domain. As a result, the IDEFs get tainted as something "techies" do, so get short-circuited as a common, enterprisewide means of communications.

---

[2] For example, IATA Compendium of International Civil Aviation.
[3] For example, Power Industry Dictionary.
[4] For example, Glossary of Terms, in OSHA Directive, CPL 2-1.18A – *Enforcement of the Electrical Power Generation, Transmission, and Distribution Standard*.
[5] For example, The IEEE Standard Dictionary of Electrical and Electronics Terms, Sixth Edition (IEEE Std 100-1996).
[6] The process of refining and regrouping attributes in entities in pursuit of the general aim "*one fact in one place*."
[7] In the style of The Fifth Discipline, by renowned business management consultant Peter Senge.
[8] Originally a registered trademark of Logic Works, Inc., ERwin today is a registered trademark of Computer Associates.
[9] SmartER is a registered trademark of Knowledge Based Systems, Inc.
[10] ER/Studio is a registered trademark of Embarcadero Technologies, Inc.

Introduction

As with IDEF1X, the lexical conventions for enterprise data definition presented in this paper have a bias for relational methods. However, these conventions have no bias for any particular type of database technology[11] or application development method.[12] This includes the lack of any inherent emphasis on database design and application design.

Instead, there is nontrivial emphasis on understanding and representing the inherent semantics and structure of the fully integrated enterprise as a system of business things (i.e., entities) from business processes. This is what allows the enterprise to know what it knows in support of how it does what it knows. A particular modeling effort may be of the enterprise as-is, should-be, or to-be.

Biases for particular database technology and application development methods must originate elsewhere – typically with *project information models* scoped from the Enterprise Data Model to constrain *external schemas* and *internal schemas*.[13]

This will allow extension of the lexical conventions herein to various project-level information models without compromising the integrity of the Enterprise Data Model for technical convenience and contrivances. This approach also will foster the ability to more easily trace data between their origins and meanings in business and their various usage and representations in applications and other technology, as might be captured in *metadata* (i.e., data about data).

The lexical conventions follow, beginning on the next page. Most are definite and a few are indefinite. The indefinite lexical conventions appear as "Avoid ... doing something." Sometimes it is better to disregard an indefinite lexical convention for the sake of clarity. However, inattention to the definite lexical conventions will invite semantic dissolution, resulting in hidden and misrepresented enterprise semantics. Still, there will be times when the data model author must represent incomplete understanding to be resolved later. The lexical conventions have been crafted to accommodate such times.

Each lexical convention typically includes both an example and a counterexample of its application. This is intended to help persons judge the quality of the application of the lexical convention as well as the quality of the lexical convention itself.

The listed order of the lexical conventions is not significant. This is because the lexical conventions themselves represent a system, each interdependent with the rest.

[11] For example, Relational Database Management System (RDBMS), Hierarchical Database Management System, Network Database Management System, Object-Oriented Database Management System (OODBMS), or any composite of these.

[12] For example, System Development Life Cycle (SDLC), Object-Oriented Analysis (OOA), Object-Oriented Application Design & Development, Joint Application Development (JAD), Web Design, etc.

[13] See, again, footnote 1, page 3.

Lexical Conventions for Entities

E1 Understand an entity as a single instance of a ***thing***. A specific entity instance might be a person, a place, an object, a process, an event, a document, a regulation, a concept, a specification, etc., or a pair of these. A group of entity instances of one type share common and applicable attributes, and is referred to simply as ***entity***.

E2 Avoid thinking of entities as *tables* or *object classes*. Entities embody business concepts, whereas tables and object classes embody subsequent technical designs and implementations in a variety of technologies.

E3 Write entity names using uppercase letters.

Example: CAPACITOR BANK

Instead of: CapacitorBank, or
Capacitor_Bank

E4 Use the entity name that is a simple, singular noun or noun phrase the businessperson perceives to be significant to the enterprise.

Example: PARCEL TYPE

Instead of: PARCEL TYPE CODE

E5 Use no entity name that is or embeds the name of a specific entity instance (i.e., the name of an organization, a standard, a publication, a form, a report, or any other specific entity instance.)

Example: ORGANIZATION FACILITY

Instead of: ACME ELECTRIC FACILITY

E6 Use no entity name that is or embeds a preposition or conjunction.

Example: PROJECT CONTACT

Instead of: CONTACT FOR PROJECT

E7 Use the hyphen to separate the singular entity names within a compound entity name. (Compound entity names often result from the process of resolving *nonspecific* relationships; e.g., many-to-many relationships).

Example: PARTY-ADDRESS

Instead of: PARTY ADDRESS

E8 Replace the compound entity name with a simpler, singular entity name, when doing so will improve the businessperson's understanding. Omit the hyphen from the singular entity name.

Example: CUSTOMER INTEREST CLASSIFICATION

Instead of: CUSTOMER-MARKET

E9 Use no punctuation in entity names other than the hyphen.

Example: SOURCE-DESTINATION

Instead of: SOURCE/DESTINATION

E10 When writing an entity name in business facts, rules, and narratives, form the plural of the entity name simply by adding the lowercase letter "s" to the end of the entity name. Forgo the natural language spelling.

Example: AGREEMENT ADDENDUMs

Instead of: AGREEMENT ADDENDA

E11 Define ***what*** an entity is within one or two simple sentences[14]. Omit computer and database jargon. Though useful, protracted, explanatory narrative is not a definition. Instead, write notes[15] to amplify the significance of the entity to the enterprise (e.g., how the entity is identified, used, its role, what it contains, what it represents, etc.).

Example: PARTY ADDRESS USAGE: Information about the use of an address by a party.

Instead of: PARTY-ADDR-LOOKUP-TBL: An associative table cross-reference between parties and addresses.

E12 Make entity names unique, enterprisewide.

E13 Use industry-standard entities and definitions, when known and applicable.

Example: AIRLINE: A legal entity, such as an association, company, corporation, organization or partnership, holding an operating certificate issued by the Civil Aviation Authority of the State of Registry, to engage in the business of providing scheduled and/or non-scheduled air transport services for the carriage of passengers, baggage, and/or cargo, and/or mail. Synonymous with AIR CARRIER, CARRIER, and OPERATOR. [IATA Compendium of International Civil Aviation]

Example: TRANSMISSION LINE: A set of conductors, insulators, supporting structures, and associated equipment used to move large quantities of power at high voltage. [Power Industry Dictionary]

E14 Define an entity without circular reference.

Circular: CIRCUIT: One or more CIRCUIT SEGMENTs.

CIRCUIT SEGMENT: Any one segment of a CIRCUIT.

E15 Define an entity in terms of a single occurrence of the entity and not in terms of itself.

Example: CIRCUIT: A conductor or system of conductors through which an electric current is intended to flow. [IEEE Std 100-1996]

Instead of: CIRCUIT: Electrical components that form a circuit.

E16 Avoid using abbreviations − that is, acronyms, initialisms, contractions, and shortenings − in entity names.

E17 Represent the structural relationship between entities of the same type with a separate entity.

Example: DOCUMENT STRUCTURE ELEMENT

Instead of: DOCUMENT related to itself

E18 Give no entity a name that is also the name of an attribute of an entity.

[14] When needed (e.g., lack of access to appropriate business expertise), a usually effective start is to turn to an internal glossary of terms, an industry-specific dictionary, or a common dictionary. For a common dictionary, The American Heritage Dictionary of the English Language, Fifth Edition, is recommended. The Third Edition is available in some popular software products. Microsoft Encarta is alternatively recommended.

[15] IDEF specifies recording *notes*, where perhaps other methods specify recording *comments*. The terms are essentially equivalent, in this context.

Lexical Conventions for Entity Relationships

R1 Understand a relationship as a named association between two different entities of the same or different type. A named relationship will suggest, to some degree, an enterprise process or activity such that an essential business fact or rule can be expressed in a basic, plain-English form.

Business Fact form:

<SUBJECT ENTITY NAME><relationship name><OBJECT ENTITY NAME>[s]

Business Rule form:

A[n] <SUBJECT ENTITY NAME><optionality><relationship name><cardinality[16]><OBJECT ENTITY NAME>[s]

R2 Avoid thinking of relationships as *pointers*, *links*, or *object associations*. Relationships embody business concepts. Pointers, links, and object associations embody subsequent technical designs and implementations in a variety of technologies.

R3 Understand the word "many" means "zero, one, or more" when used to denote cardinality in business rules, as shown in examples in this section.

R4 Use the relationship name that is a singular verb or verb phrase the businessperson perceives to be significant to the business. The relationship name necessarily will suggest, to some degree, something the enterprise does or cares to do.

Examples: owns, sells, operates, transmits, constructs, holds membership in

R5 Avoid passive relationship names by using simple verbs and verb phrases in the active voice as much as possible.

Example: A SERVICE carries many SERVICE PRICEs.

Instead of: A SERVICE is priced at many SERVICE PRICEs.

R6 Give the *nonspecific relationship* (i.e., the many-to-many or optional relationship) a name in both directions. Optionally, give the *specific relationship* (e.g., the one-to-many relationship) a name in both directions. Separate the relationship names with the forward slash (/) to show that the relationship must be read in both directions.

Example: ORGANIZATION provides/is provided by COMPETING SERVICEs.

Forward: ORGANIZATION provides COMPETING SERVICEs.

Backward: COMPETING SERVICE is provided by ORGANIZATIONs.

R7 Use no punctuation, other than the forward slash, in relationship names.

Example: incorporates/modifies

Instead of: incorporates_modifies

R8 Omit optionality and cardinality information from relationship names.

Example: A MANUFACTURER produces one or more PRODUCTs.

Instead of: A MANUFACTURER may make many one or more PRODUCTs.

[16] An indication of quantity expressed as a word, phrase, symbol, or number.

R9 Make relationship names unique when two or more relationships exist between entities of different type or the same type. Otherwise, relationship names may be identical.

Example: A PROJECT embodies control of many PROJECT STRUCTURE ELEMENTs, and

A PROJECT particularizes zero or one PROJECT STRUCTURE ELEMENT.

R10 Avoid using nouns as verbs as well as verbs that are also nouns within relationship names.

Example: ORGANIZATION provides SERVICEs.

Instead of: ORGANIZATION provision SERVICEs.

R11 Use only verbs and verb phrases as relationship names. Never use nouns and noun phrases or prepositions and prepositional phrases, except those that function as part of a verb.

Example: An ORGANIZATION holds ownership in many ORGANIZATIONs.

Instead of: An ORGANIZATION membership many ORGANIZATIONs, or

An ORGANIZATION through one or more ORGANIZATIONs.

R12 Avoid using abbreviations − whether acronyms, initialisms, contractions, or shortenings − in a relationship names.

Lexical Conventions for Attributes of Entities

A1 Understand an attribute as a specific characteristic of an entity. The characteristic must be both common to and applicable to all instances of the entity.

A2 Avoid thinking of attributes of entities as *columns*, *fields*, and *object properties*. Attributes embody business concepts; columns, fields, and object properties embody subsequent technical designs and implementations in a variety of technologies.

A3 Form attribute names using an object word, optional modifier words, and a class word.

Object Word: A noun within an attribute name that identifies or suggests the existence of an entity to which the attribute belongs.

Modifier Word: One of possibly several consecutive words within an attribute name that serves to clarify the meaning or the orientation of the attribute.

Class Word: The noun within an attribute name that characterizes the specific usage of all values of the attribute. See Attachment A: Class Words.

Example: Project (object word), status (modifier word), and code (class word) become project-status-code.

A4 Use the attribute name the businessperson perceives to be significant to the enterprise.

Example: state-abbreviated-name

Instead of: state-value

A5 Use the class word ***identifier*** or ***code*** in an attribute name if and only if each value of the attribute serves to uniquely identify an instance of the entity to which the attribute belongs.

A6 Write attribute names using lowercase letters, only. Separate the words with the hyphen and no other character.

Example: work-order-status-code

Instead of: WorkOrderStatusCode, or
Work Order Status Code

A7 Order words within attribute names to aid a natural language reading style.

Example: product-position-code

Instead of: code-position-product

A8 Avoid using the word *of* within attribute names.

Example: revenue-money-amount

Instead of: revenue-amount-of-money

A9 Make attribute names unique, enterprisewide.

A10 Use industry-standard attributes and definitions, when known and applicable.

Example: synthetic-rope-elasticity-ratio: The ratio of ["A" minus "B"] to "B", where 1) "A" equals the elongated length of the rope when fully supporting the load, 2) "B" equals the stretched full length of the rope before supporting any of the load, 3) the rope length is measured from the anchor connection to the safety belt or harness connection, and 4) the load is the combined tool and body weight of the climber. [OSHA, Directive No. CPL 2-1.18A]

A11 Use no attribute name that is or embeds the name of a specific attribute instance.

Example: organization-identity-code

Instead of: RuralElectrificationActCode

A12 Capture the logical ***what*** instead of the physical *where*, *who*, *when*, and *how much* in attribute names.

Example: maintenance-plan-status
maintenance-plan-date

Instead of: january-maintenance-plan-status,
february-maintenance-plan-status,
march-maintenance-plan-status,
etc.

A13 Define ***what*** an attribute is within one or two simple sentences. Though useful, protracted, explanatory narrative is not a definition. Instead, write notes to amplify the significance of the attribute to the enterprise. If the attribute carries a list of allowable values, define each value in the list, likewise.

A14 Define an attribute in terms of other relevant concepts, not in terms of itself.

Example: maintenance-request-description: The narration of the perceived work needed to keep an item of equipment or other asset in proper working order.

Instead of: maintenance-request-description: A meaningful description of the maintenance requested.

A15 Associate an attribute name with one and only one attribute definition (i.e., no homonyms).

A16 Associate an attribute definition with one and only one attribute name (i.e., no synonyms).

A17 Make attributes single-purpose by making their data values atomic in structure and meaning (e.g., no polymorphism).

Example: organization-division-name
organization-department-name
organization-class-name

Instead of: organization-division-department-class

A18 Introduce attribute *role names*, in the form <role-name>.<attribute-name>, for the attribute inherited through one or more relationships between entities, when needed to clarify the purpose of the inherited attribute or to cause or prevent ***unification***[17].

Example: controlling-project-number.project-number, and
supporting-project-number.project-number

introduces a unique role name for each of two different uses of the attribute project-number to prevent unification

[17] The merging of two or more inherited attributes into one attribute based upon the business rule that the original attribute values are the same in every instance of the inheriting entity.

A19 Avoid introducing *group attributes*. However, if needed, compose the group attribute from relevant atomic attributes through unification. Thereafter, treat the group attribute as if it were atomic and single-purpose.

Example: trouble-ticket-number.party-identifier,
trouble-ticket-number.address-identifier, and
trouble-ticket-number.ticket-sequencing-number

introduces the same role name for three different attributes to cause unification

A20 Avoid using abbreviations − whether acronyms, initialisms, contractions, or shortenings − in attribute names.

A21 Give no attribute a name that is also the name of an entity.

Example: CONTACT TYPE (entity name)
contact-type-description (attribute name)

Instead of: CONTACT TYPE (entity name)
contact-type (attribute name)

Attachment A: Class Words

Class Word	Abbreviation.[18]	Definition	Generic Data Type
amount	amt	A total of something, typically indicating a value in monetary units.	numeric
code	cd	A symbol or cipher that gives brevity to other information.	character
count	ct	The number of instances of something.	numeric
date	dt	A particular day on the calendar at which something happened or is expected to happen.	numeric
description	dsc	A phrase intending to quickly convey an idea, impression, or sense about something. It is less than a complete syntactical construction in a natural language.	character
flag	flg	A value, one of two, that either affirms or denies veracity.	boolean
identifier	id	A characteristic that uniquely distinguishes a thing from amongst its kind.	character
name	nm	A designation that does not necessarily distinguish a thing from amongst its kind. Often a proper name, such as for vendors, customers, places, etc.	character
number	nr	A composition of digits used to reference (not necessarily identify) something. Not for use as arithmetic operands.	character
percent	pct	A count of something per one hundred.	numeric
percentage	pctg	A ratio with 100 understood as the denominator. See Ratio.	numeric
quantity	qty	An aspect in which a thing is measured.	numeric
rate	rt	A fraction indicating a proportion between two dissimilar things.	numeric
ratio	rto	A fraction indicating a proportion between two similar things.	numeric
text	txt	A composition of words, in a natural language form and syntax, intending to give complete information about or a complete explanation of something.	character[19]
time	tm	The point at which something happened or is expected to happen as measured by the clock.	numeric
timestamp	ts	A recording of the day on the calendar and the time on the clock at which something happened.	numeric

[18] The abbreviated form of class word may be used only in physical-oriented models for data base management systems and application systems.
[19] Maximum length of text is strictly a function of the physical environment.

Attachment B: Readers' Notes

The reader of this document is invited to record notes, below, as well as to annotate the document wherever the reader deems convenient.