AN ANALYSIS OF BUSINESS OBJECT MODELING

INTENDED AS AN ADDENDUM TO DAVID C. HAY'S OCTOBER 1999 PAPER "A COMPARISON OF DATA MODELING TECHNIQUES"

(found at www.essentialstrategies.com)

by
Bruce Abbott
Data Administrator
Minnesota Dept. of Natural Resources

27 March, 2000

BUSINESS OBJECT MODELING from ADVANCED STRATEGIES, INC.

Advanced Strategies, Inc. of Atlanta, GA (www.advancedstrategiesinc.com) developed the Business Object Modeling approach to user-centric conceptual data modeling. This approach is clearly focused on business users, or Subject Matter Experts (SMEs). Data modelers help SMEs to represent real-world facts about important entities and the relationships among them, for a specific business domain / project scope / universe of discourse. SMEs are pre-selected to be business, not data, experts, and are asked to describe the real world of their business, without filtering it through any knowledge of existing legacy data structures. The modeler guides the SMEs through a set of exercises to produce a Business Object Model, which consists of an entity/relationship diagram (ERD) with accompanying (off-diagram) textual descriptions. This model lays out, in a highly readable format, essential facts about their business objects and the associations among them. Facts related to business policies (which change more quickly and unpredictably than essential facts), and facts of implementation related to technology constraints (which change even more rapidly!) are kept out of the Business Object Model, and saved for later modeling steps. The result is a very clear representation of the essential nature of a business area, easily readable (and therefore verifiable) by the business experts who built it. Because this model is based on real-world facts, it remains stable over a fairly long time frame (as long as the essential nature of that area of the business remains stable).

Note that the term "Business Object Modeling" refers to business objects, as understood by business experts. It has nothing to with UML or other object-oriented modeling approaches.

This approach uses a modification of Mr. Chen's notation. Entities are shown as rectangular boxes, and relationships are shown as diamonds (rhombuses with the shorter axis vertical). Attributes, which can be associated with both entities and relationships, generally are documented only in accompanying text. (Advanced Strategies stresses that all business models consist of both diagram and text.) Figure A shows the Party - Purchase Order example, using this notation. PARTY *places order with* PARTY. This relationship is commonly known as a PURCHASE ORDER. A PURCHASE ORDER *refers to* a PRODUCT or a SERVICE. Each such reference relationship is commonly known as a LINE ITEM.

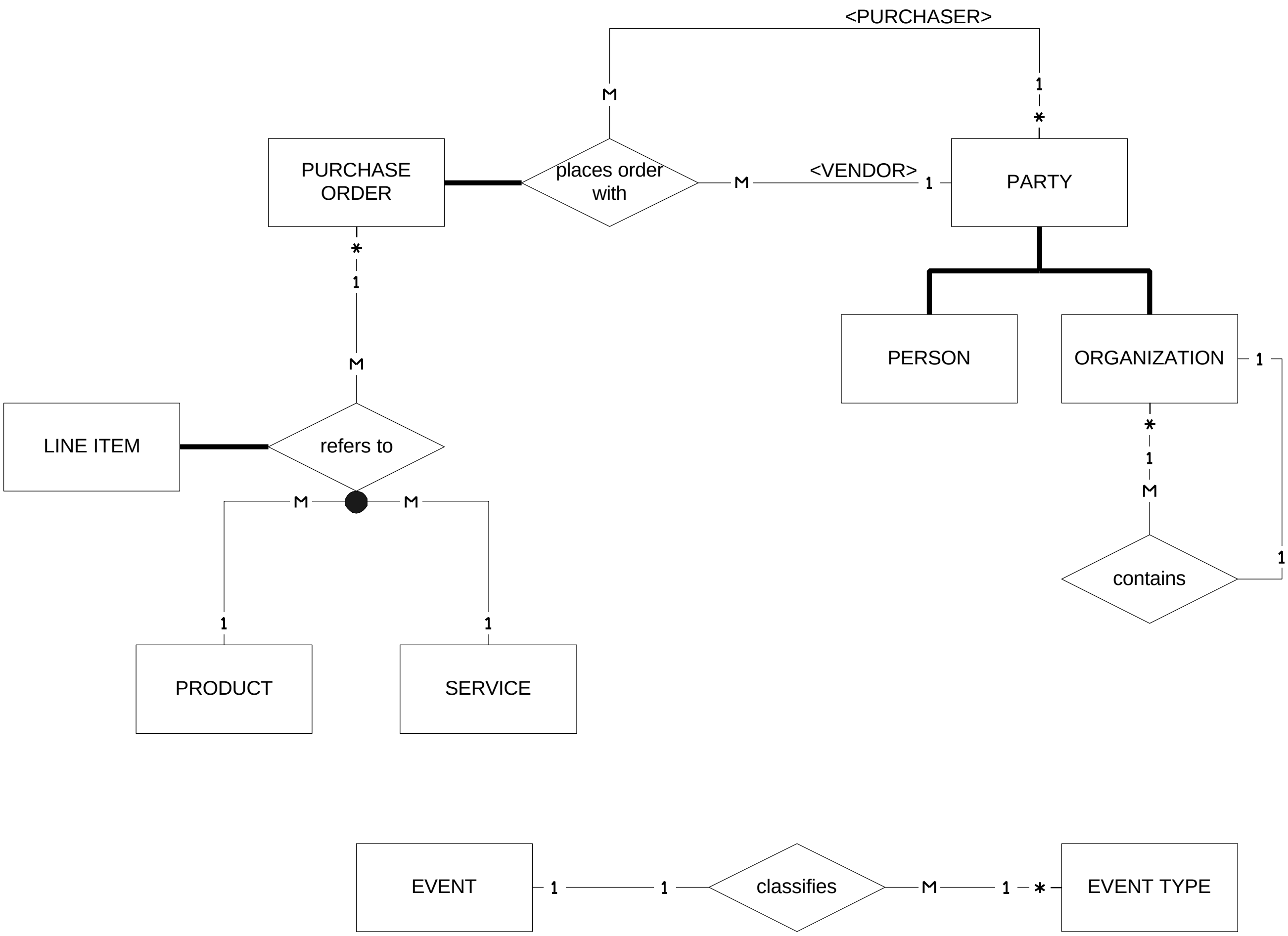

Figure A: Business Object Model

Entities and attributes

Entities are represented by square-cornered rectangles. Entities in this approach are always independent – by definition, they may exist with or without the existence of relationship associations with other entities. ***Attributes*** are generally documented in accompanying text, not shown on the ERD, unless their presence is needed to clarify the ERD for the business experts. If attributes are shown on the ERD, they are displayed as a bulleted list next to (outside) the entity rectangle to which they refer. Either on the ERD or in the accompanying text, attribute names may have one of three capital letters appended to them, inside parentheses: (U) for unique identifier, (D) for derived attribute (calculable from other attributes), or (R) for repeating attribute.

Names of entities and attributes are common business terms, suggested and approved by the SMEs. Entity names are always expressed as singular nouns. Entity names are all uppercase; attribute names are all in initial capitals. Multi-word names are separated by spaces.

Relationships

Relationships are displayed in a manner similar to Mr. Chen's notation. They are represented by a two-dimensional symbol: a rhombus (usually referred to in this notation as a diamond), with the short axis vertical. Relationships may have attributes. If so, relationship attributes are treated the same way as entity attributes. Relationship instances are dependent for their existence on the existence of the associated entity instances. The lines connecting relationship symbols with entity symbols are called relationship links, and are simply unnamed one-dimensional connectors.

From the Subject Matter Expert perspective (absolutely the primary perspective in Business Object Modeling), a relationship represents a way that two or more independent entities may be associated. SMEs perceive relationships as being conceptually different from entities – because they represent an association among things, rather than a single thing. SMEs may also be interested in additional properties of that relationship. So giving relationships a unique two-dimensional symbol, one that is allowed to carry attributes, clarifies the conceptual understanding, definition, and interdependencies of those modeling constructs, and will result in more robust tables that better represent the real world of the SME. Take as an example the relationship "PERSON *belongs to* ORGANIZATION", shown in Figure B.

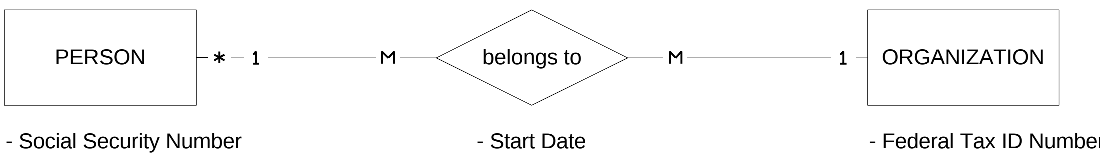

Figure B: Entities and Relationships, with Attributes

"PERSON *belongs to* ORGANIZATION" is a plain-English, straightforward statement of an association between two objects. Now, if this fact were offered by a business expert to a data modeler who was using another modeling notation, it might be represented initially as Person (entity box) belongs to (relationship line) Organization (entity box). However, this representation would not allow for the attribute Start Date, which clearly describes the relationship, not either one of the entities. So a modeler might need to coax another (associative) entity, such as Membership, out of the SME. This results in a conceptually more complicated representation of a real-world fact: Person has Membership, and Membership is for Organization. With the Business Object Modeling notation shown above, the translation from spoken business fact to diagrammed representation is more straightforward, and the SME will more likely retain a sense of ownership of the model.

From a professional modeler's perspective, all relationships in Business Object Modeling can be thought of as associative entities, which will, in later modeling stages, turn into dependent entities and then tables (assuming a relational database implementation). The columns in these tables may be limited to foreign keys from each of the associated entities, or there may be additional columns representing other attributes.

Relationships can be binary or complex – they can associate two or more entities.

There are times during business object modeling sessions when SMEs suggest a new entity that turns out to be a role played by another, more general entity when it engages in a particular relationship. When this happens, or for other reasons of clarity (e.g., in some recursive relationships), that role is noted as an uppercase label enclosed in angle brackets, outside the entity box and parallel to the relationship link. (See the PURCHASER and VENDOR roles played by PARTY in Figure A.) This convention helps to prevent the same object in different roles from appearing as multiple entities on the diagram.

Cardinality

Cardinality is represented in this notation by one of two straightforward symbols at each end of a relationship link: "1" for one, and "M" for many. These are easily translated by SMEs into common English, unlike some of the graphic cardinality symbols in other notations. The fact that relationships are represented as two-dimensional symbols helps to resolve what might otherwise be many-to-many-relationships into multiple one-to-many relationships. In Figure B above, a many-to-many relationship between PERSON and ORGANIZATION resolves to a one-to-many relationship between PERSON and *belongs to*, and a one-to-many relationship between ORGANIZATION and *belongs to*.

Optionality

The concept of optionality is built-in to the Business Object Modeling notation: All entities are, by definition, independent, and all relationships are dependent. An entity instance can exist without the presence of other entity or relationship instances. But a relationship instance requires the presence of an instance of each of the associated entities. Because this optionality is inherent in the concepts of entities and relationships, no additional notation is needed. Forcing the SMEs to designate a concept as independent or dependent right out of the chute really helps to bring clarity to the model for the SMEs, and results in a more robust physical data model.

The concept of associative entity mentioned earlier is sometimes made explicit in Business Object Modeling. Relationship links can only connect relationship diamonds to entity boxes – a relationship cannot be directly connected to another relationship. But sometimes, a relationship itself may participate in other relationships. In this case, it is designated as an associative entity by redrawing it as an entity box, and renaming it as a noun. The associative entity is drawn right next to the relationship diamond from which it sprang, and is connected to it by a short, heavy line. See Figure C.

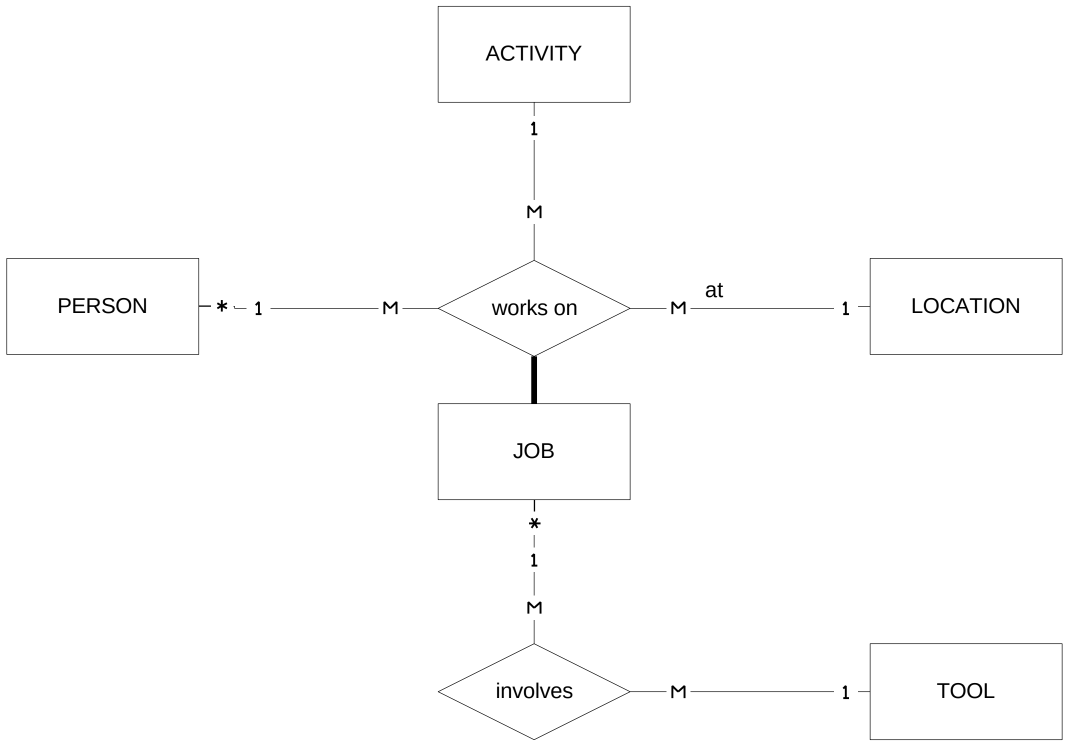

Figure C: Associative Entity "JOB"

While at first blush this may seem like the inclusion of an unnecessary extra symbol on the diagram, the associative entity does have a useful purpose, related to the optionality issue. Remember that a relationship instance <u>must</u> include an instance of each of the entities to which it is connected. However, when that same relationship behaves as an associative entity, an instance of it <u>may</u> participate in another relationship to which it is connected. In Figure C above, PERSON *works on* ACTIVITY at LOCATION. This relationship sometimes involves a tool, but not always. So creating an associative entity JOB allows us to say, "JOB <u>may</u> *involve* TOOL".

There are readability issues here, too. First, the sentence describing a relationship makes reference to <u>all</u> the symbols connected to the relationship diamond by relationship links. If a relationship diamond were simply <u>replaced</u> by its associative entity symbol, then it would be difficult to determine which relationship links were part of the original relationship, and which ones were parts of additional relationships in which the associative entity may participate. Second, if the original verb/preposition is replaced by a noun when the associative entity is added, you lose the original phrasing that made sense to the SMEs.

Associative entities may also be placed on a Business Object Model if relationships require sub-typing. And finally, they may be used if relationships are commonly thought of as "things" by SMEs, and have commonly-used names. Keep in mind that the primary goal of the Business Object Model is to clearly portray the business experts' view of the subject area. This is why, in Figure A above, the associative entity LINE ITEM, which does not participate in any relationships, is included. The term was introduced by SMEs, and was deemed important to them. In the same Figure A, the associative entity PURCHASE ORDER has a commonly used name, <u>and</u> itself participates in another relationship – two reasons to add an associative entity symbol to the diagram.

Names

Relationships are named in one direction only, with a verb and sometimes a preposition. The choice of direction naturally affects the form that the name takes, but not the meaning of the relationship. The direction is based on, once again, what sounds best to the SMEs. If they have no preference, the active form of the verb is preferred to the passive. Relationship verbs are always singular, to agree in number with the associated entities. A relationship taken together with its associated entities reads like a natural-language sentence, and is intended to be a clear statement of a true and relevant business fact. This feature makes the model easily readable and therefore verifiable by the SMEs.

The subject or starting entity in a sentence about a relationship is designated by an asterisk outside the rectangle, on or near the relationship link. This asterisk is called an anchor, and quickly helps the reader determine how to read such a sentence (without having to first check out all the entity nouns and the relationship verb).

Complex relationships, in order to be read as sentences, sometimes have additional prepositions written outside the relationship diamond, parallel to the relationship links that connect to the associated entities. For example, the relationship in Figure D is read as "ORGANIZATION *employs* PERSON <u>in</u> POSITION."

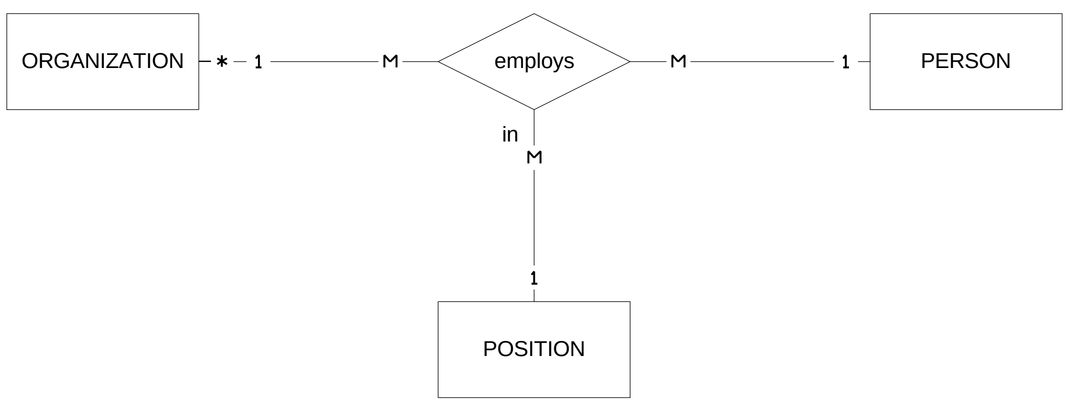

Figure D: Complex Relationship

Note: An instance of the complex relationship in Figure D is defined as an association among an instance of <u>each</u> of the three entities. There are no less-than-*n*-ary relationships implied by an *n*-ary relationship. If an organization creates a position, for instance, but it hasn't yet been filled, then that is a different relationship, not shown or implied above.

Unique Identifiers

If an attribute (or combination of attributes) of an entity or relationship is such that its value domain creates unique identifiers for instances of that entity or relationship, then the attribute is annotated with

"(U)" after its name. However, unique identifiers are not sought out in this phase of modeling. If one comes up during a discussion of attributes, then it is noted. Clearly, this could be useful information later in the formation of a primary key. But, since Business Object Modeling is absolutely conceptual and absolutely focused on real-world business facts contributed by SMEs, no physical constructs such as primary or foreign keys are developed in this model.

This brings up another interesting point. In a Business Object Model, an n-ary relationship with m attributes will lead to a logical data model with $m + n$ attributes. These extra n attributes are the foreign keys of each of the n entities participating in the relationship. The combination of these n foreign keys will very likely form part of (or all of) the primary key of the relationship table.   But once again, these foreign keys have no place in the conceptual Business Object Model.

Sub-types

In Business Object Modeling notation, sub-types are shown as boxes outside super-type boxes. All the sub-types are connected to their super-type with a set of heavy straight lines arranged like the tines of a rake, or the connectors in an organization chart. See Figure E.

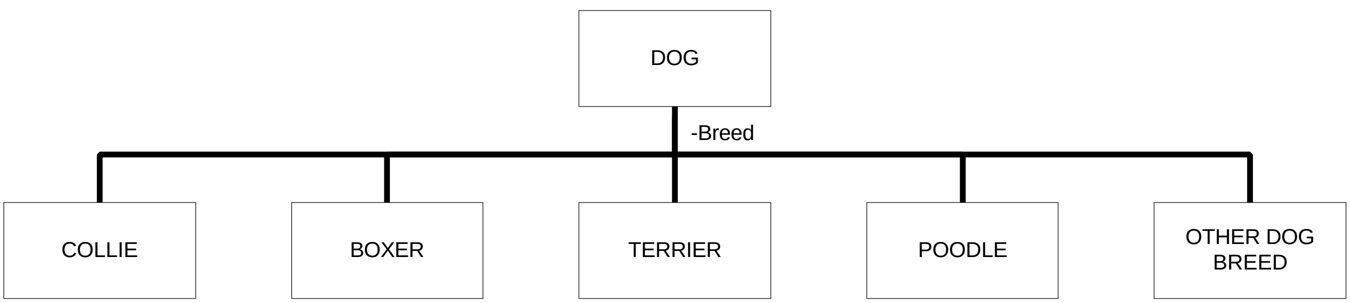

Figure E: Entity Hierarchy

This structure is referred to as an entity hierarchy. Each heavy line connected to a sub-type implies an "isa" fact linking that sub-type to its super-type: A COLLIE *isa* a DOG, a BOXER *isa* DOG, etc. In practice, the "isa" fact is used mainly as one test to insure the correctness of the hierarchy. The structure is usually interpreted verbally as, "There are several breeds of dog – collie, boxer, terrier, poodle, and others."

Sub-typing is always based on an attribute or set of attributes. Indicating the partitioning attribute name(s) on the diagram allows the reader to see the basis of the hierarchy.

Sub-types in Business Object Models are both mutually exclusive and exhaustive (although exhaustiveness is sometimes guaranteed with an "OTHER…" entity). Because sub-types are represented as external boxes, outside the super-type, the notation does support a sub-type having multiple super-types. But in practice this is not common, because it may cause conceptual/redundancy problems in the resulting data structure. A super-type may have multiple sets of orthogonal sub-types.

Constraints between relationships

As mentioned above, an entity hierarchy implies mutually exclusive facts – in a given hierarchy, an instance of a super-type is also an instance of one and only one of its sub-types.

A second type of "exclusive or" constraint is also evidenced in another notational feature of Business Object Modeling: the multi-member link. This is a large dot at the vertex of a relationship diamond, to which multiple relationship links are connected. Each of these links connects to a separate entity. The multi-member link symbol is read as "or". (See "PURCHASE ORDER *refers to* PRODUCT or SERVICE" in Figure A.) This is actually a convenient notational shortcut, not an indication of constraint. The relationship containing a multi-member link attached to *n* entities is nothing more than an abbreviated way to draw *n* separate and distinct relationships, which would take up much more room on the diagram. Using this shortcut reduces clutter on the Business Object Model diagram, and allows the reader to easily see the commonality among certain relationships. When the Business Object Model is later transformed into a logical model, the single multi-member-linked relationship is replaced by *n* separate relationships.

Comments

The primary goal of Business Object Modeling is to clearly portray the Subject Matter Experts' view of the real-world facts about the business area under study. All of the notational conventions used in Business Object Modeling are intended to support that clarity and that perspective. In fact, Business Object Modeling sessions often result in a clearer understanding among SMEs of the essential nature of their business. The fact that the notation is fairly simple and intuitive allows SMEs to build the model themselves, in real time, with the assistance of the data modeler. This gives the SMEs ownership of the model. Some non-IT-oriented SMEs have been known to hang Business Object Model entity/relationship diagrams in their offices, not because the models represent innovative IT methodology, but because they clearly show "what the business is all about".

One of the most visible differences between Business Object Modeling and most of the other techniques concerns the representation of relationships. As with Mr. Chen's notation, Business Object Modeling notation shows relationships as separate two-dimensional symbols (diamonds), which can carry attributes. These relationships are named with verbs and sometimes prepositions, akin to the naming convention for relationship lines in Mr. Martin's version of Information Engineering and in IDEF1X. This naming convention allows for intuitive natural-language verbalization of the facts depicted by the Business Object Model. In addition, it is conceptually simpler to describe a set of facts as a single relationship between two entities, rather than as two binary relationships among three entities, one of which is associative. (Refer to Figure B above: "Person belongs to Organization" is more succinct and intuitive than "Person holds Membership, and Membership is for Organization".)

Business Object Models normally describe attributes in separate text, rather than on the diagram, as does Mr. Finkelstein's version of Information Engineering. This helps to keep the diagram uncluttered. When the presence of an attribute would add clarity for the business expert reading the diagram, the Business Object Model makes use of a bulleted list next to the entity or relationship shape. This is less distracting that the use of circles for attributes in Mr. Chen's notation. Showing the (occasional) attribute outside the box or diamond also makes it easier for the reader to interpret the simple declarative sentences represented by those shapes.

The readability of Business Object Models is greatly enhanced by the rigorous naming conventions applied to both entities and relationships on the diagrams. Entities are named with singular nouns, relationship names include singular verb forms and sometimes prepositions, and the lines connecting relationship diamonds to entity boxes in complex relationships are annotated with prepositions if needed. All of these conventions taken together help the reader to interpret as easily readable sentences the real-world facts shown by the model. The care given to the selection of names, and the preference for names commonly used by SMEs, also add to the diagram's readability. (Also, the use of singular nouns and verbs underscores the fact that one instance of a relationship involves the association of one instance of each of the associated entities.)

The inherent independence of entities, and dependence of relationships, allows for a visually more straightforward representation of optionality in Business Object Modeling. It is the two-dimensional shapes themselves, rather than the continuity of a connecting line (Barker) and/or a pattern of symbols on that line (Information Engineering, IDEF1X, ORM, UML), which portray optionality. The Business Object Modeling convention of using all uppercase letters to name entities, and all lowercase letters to name relationships, subtly underscores this distinction between these two symbols.

Separating cardinality from optionality also lends to the readability of the Business Object Model. The interaction among symbols indicating these two constructs in IDEF1X is not at all intuitive. Having to look for either a zero or a one as the first member of a pair of cardinality symbols (UML) is also somewhat confusing. The Business Object Modeling convention of using simply a "1" or an "M" symbol to indicate cardinality alone is much more straightforward. Finally, "1" and "M" are more easily translated into the verbalizations "one" and "many" than is the presence or absence of crow's feet (Information Engineering, Barker).

Business Object Models depict super-type / sub-type relationships as hierarchical structures, using separate entity boxes. This may take up somewhat more room on the diagram than the nested boxes used in Mr. Barker's notation. However, being outside the box does not have to mean being scattered all over the page: The sub-types in Business Object Modeling are represented as neat, compact rows of boxes, connected to the next level up in the hierarchy by short, heavy lines. The interpretation of these heavy connecting lines as representing "isa" facts is arguably no less intuitive that the interpretation of nested boxes to mean the same thing. Also, the hierarchical structure can easily show many levels of sub-typing in one view. This is a common need, and most business people are comfortable interpreting a hierarchical graphic notation.

Business Object Modeling does take positional conventions into account. In this approach there are no hard and fast positional rules. However, a deliberate and distinct step after discovering the facts is determining the best layout to convey those facts. Several heuristics (such as stretching boxes) and typical patterns exist, but are beyond the scope of this paper. In each model, the positional goal, like the overall goal, is to best convey the facts to the specific intended audience.

"Just the facts, Ma'am…" This statement characterizes the rigorous search for model elements that describe, succinctly and in plain English, the real world of the business area being modeled, in terms provided by the SMEs themselves. Modelers in Business Object Modeling sessions repeatedly address two simple questions to the SMEs, regarding a specific fact under discussion. First, is it true? Second, do we care? In other words, the model is validated by making sure that it only depicts statements about the subject area that are both true and relevant (given the focus of the project). This rigor, coupled with the full participation of the SMEs in building the models in real time, results in models that reflect the essential truths about the business area under analysis.

The user-centric nature of Business Object Modeling benefits everyone – the business experts, the business unit under study, the modelers and downstream data analysts. The SMEs often come away from Business Object Modeling sessions with a better understanding of their own business, because of the care with which they defined their terms and characterized the associations among the important elements of their business. They nearly always come away from these sessions with the sense that they have been heard, that the model is <u>their</u> model. They are also confident that the data modelers who helped them build the model have a clear picture of their business. And when the modelers and downstream data analysts, via a well-established set of techniques beyond the scope of this paper, convert this model of their world into a physical database, the business unit is left with an elegant, robust database structure. This structure is based on a complete understanding of the business area under study, not just the current data requirements. As a result, the database will be able to answer standard user queries, as well as questions that the SMEs didn't know they wanted to ask at the time the model was being developed. This really extends the useful life of data structures built from Business Object Models.

Method	*Entities & Attributes*	*Relationships*	*Unique Identifiers*	*Sub-types*	*Constraints Between Relationships*
Business Object Modeling	***Entities*** shown by square-cornered rectangles; ***attributes*** not normally shown on ERD	Rhombus (diamond) symbol ; ***optionality*** is "built-in": an instance of a relationship requires the presence of an instance of each of the participating entities; an instance of an entity is by definition independent of the existence of any other entity or relationship; ***cardinality*** from "1" or "M" symbols next to entity and relationship; ***named*** one way only with verb and preposition; relationship and entity names taken together form sentence, which describes a relevant fact about the business area; no foreign keys; need not be binary	Optional; shown by "(U)" symbol on attribute (display of attribute on ERD is optional); no primary keys	Shown as ***external rectangles*** related by heavy line symbol that denotes "isa" fact; mutually exclusive and exhaustive (complete); sub-type may be in more than one super-type	Sub-types are mutually exclusive by definition; otherwise no diagram notation – any constraints documented in the text

Addendum to Mr. Hay's Table 2: Comparison of the Syntactic Conventions