

# UNIVERSAL DATA MODELS FOR MANUFACTURING

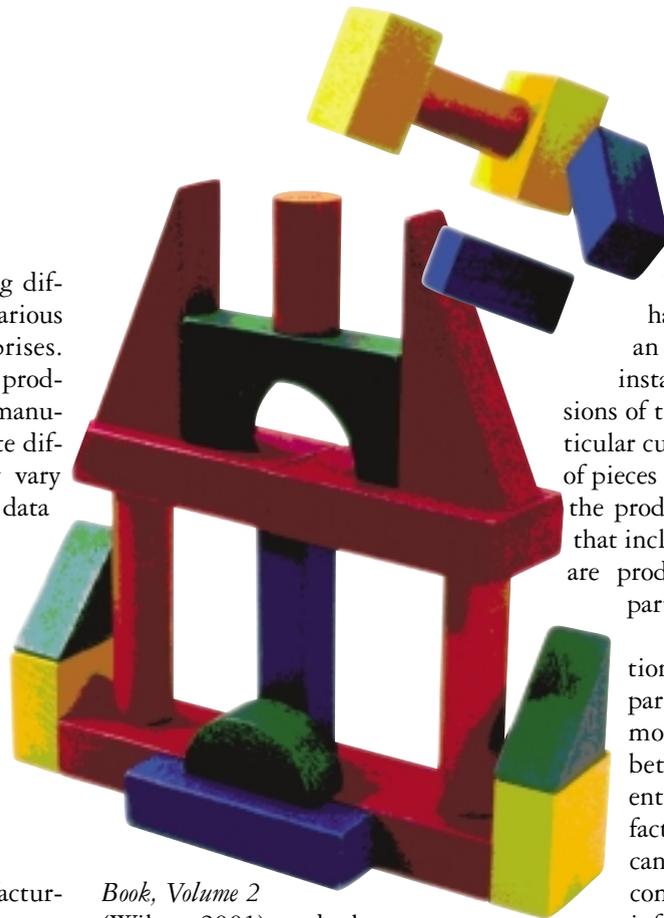
By Len Silverston

There is usually quite a big difference in the nature of various manufacturing enterprises. Each manufacturing enterprise's products may be very unique, their manufacturing processes are often quite different and their strategies may vary greatly. However, the types of data



that manufacturing enterprises capture are actually very similar; thus, there are great benefits that can be derived by taking advantage of common, reusable data structures or "universal data models."

While there are several universal data models available for many aspects of manufacturing such as design engineering, manufacturing orders, process plans and production runs, this article focuses on one of the most critical data areas required in manufacturing, namely: products, parts, (inventory) items and the deployment of manufactured goods at customer locations. Please note that the examples used in this article are based upon some of the models in the manufacturing chapter of *The Data Model Resource*



*Book, Volume 2* (Wiley, 2001) and that only partial models were excerpted. Also note that the models in this article are more geared toward discrete manufacturers versus process manufacturers, although some of the concepts will apply even for process manufacturers.

## Parts are Parts! Getting Clearer Definitions

The nature of product data is complex; and there is often much confusion around the definition of product, parts and their associated data. For instance, the term "bill of materials" may be used by various parts of the enterprise such as engineering, manufacturing, service and sales, although they may all have different

definitions of what the bill of materials actually means. Even the basic terms "product" and "parts" may have several definitions within an enterprise. Do product instances include customized versions of the product offering for a particular customer? If parts are the types of pieces that make up a product, does the product have one high-level part that includes all the subassemblies, or are products made up of multiple parts?

Clear definitions, distinctions and models for products, parts and inventory can facilitate more clear communication between various areas of an enterprise. An effective manufacturing product data model can lead to more integrated and comprehensive product data and information by clearly showing the information requirements of the products, parts and items throughout their life cycle from engineering to manufacturing to the deployment of products at customer locations.

Figure 1 (see note explaining the diagramming notation used) provides an initial data model regarding products and their associated parts. A product could be defined as a good, service or combination of goods and/or services that were, are or will be offered for sale by the enterprise.

However, there are many variations of this definition, and enterprises need to customize these universal data models for their environments. For example, should supplier and competitor products be included as part of

the product definition? If the enterprise often buys and resells the same items from their suppliers or competitors, it may make sense to enhance the definition to include products offered by the enterprise, supplier and/or competitor, rather than having additional entities for each of these types of products.

Figure 1 shows that a PRODUCT may be either a SERVICE or a GOOD. A GOOD is defined as a product that is more tangible in nature and generally created in advance for sale. A SERVICE is defined as a product that involves the use of parties' time and is less tangible in nature than goods.

A PART is defined as a component that is used to make up a product. Each PART may be a RAW MATERIAL, SUBASSEMBLY or a FINISHED GOOD. A RAW MATERIAL is a component used in making a product that has not had any work performed on it by the enterprise. It is the lowest level component that makes up the product. A SUBASSEMBLY is a part that is made up of other parts and which represents a component within a finished good. A FINISHED GOOD is a part that is finished, ready to be shipped and represents the highest level bill of materials

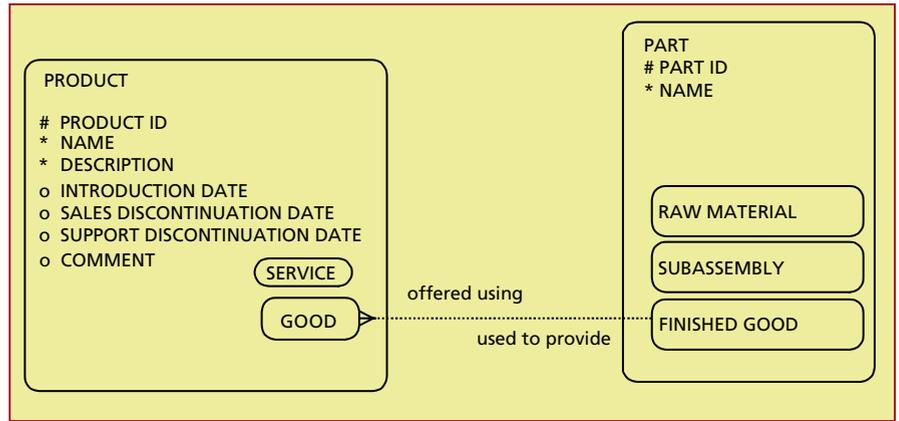


Figure 1: Products and Parts

component that directly corresponds to a PRODUCT.

The PART data model may vary based upon the needs of the enterprise. For example, if the manufacturing organization sells parts that are also used in subassemblies, should these parts exist in more than one of these subtypes (e.g., a part could be both a RAW MATERIAL as well as a FINISHED GOOD)? If this occurs, one may consider modeling the PART subtypes as inclusive subtypes. One can portray mutually inclusive subtypes by showing an additional box without text around each of the subtype boxes (in information engineering notation, this would be equivalent to each sub-

type having its own subtype symbol). For example, if the enterprise sells circuit boards, which they purchase from a supplier, and also uses the circuit boards in another assembly of their computer, the "circuit board" PART may be considered to be a FINISHED GOOD as well as a RAW MATERIAL.

One may initially conclude that the relationship from PRODUCT to PART is that a PRODUCT may be made up of many PARTs. However, the data model in Figure 1 shows the opposite relationship, namely that a PART (and more specifically a FINISHED GOOD) may be used to provide one or more PRODUCTs (and more specifically a GOOD).

This illustrates a subtle, yet important distinction that can be made between PRODUCTs and PARTs. The PRODUCT entity in Figure 1 represents a type of marketing item that is offered for sale, while the PART (and specifically the FINISHED GOOD subtype) represents the type of actual item that physically exists. Many enterprises market the same part as multiple product offerings depending on circumstances or marketing spins. For example, a specific part may be marketed in various countries differently and thus the same part may be sold and marketed as different product offerings with different names, different pricing structures and different marketing strategies. Another scenario is that there may be different product offerings for the same part depending on the intended user of the product. For example, telephone

## Note About the Data Modeling Notation in Figures

The notation in this article was developed by Richard Barker and is fully described in his book *CASE Method: Entity Relationship Modeling* (Adison Wesley, 1989). To briefly explain it:

- A crow's foot (three prongs at the end of the relationship line) indicates that there are many occurrences of the entity near the crow's foot for each entity that is not near the crow's foot. For example, each FINISHED GOOD may be used to provide one and only one GOOD.
- The dotted line indicates optionality (as opposed to mandatory) for each side of the relationship. Each side of the relationship is designed to be read as a complete sentence both ways. For example, Figure 1 shows that each FINISHED GOOD **may be** (since this is a dotted part of the line) *used to provide* one and only one GOOD. If the relationship line is solid, then the relationship indicates a mandatory relationship. For example, Figure 3 shows that each ITEM **must be** the physical occurrence of one and only one PART.
- A "#" in front of an attribute indicates that this attribute is a key.  
A "\*" indicates that the attribute is a mandatory attribute.  
An "o" before an attribute indicates that the attribute is optional.
- Boxes within boxes indicate subtypes or, in other words, subentities.

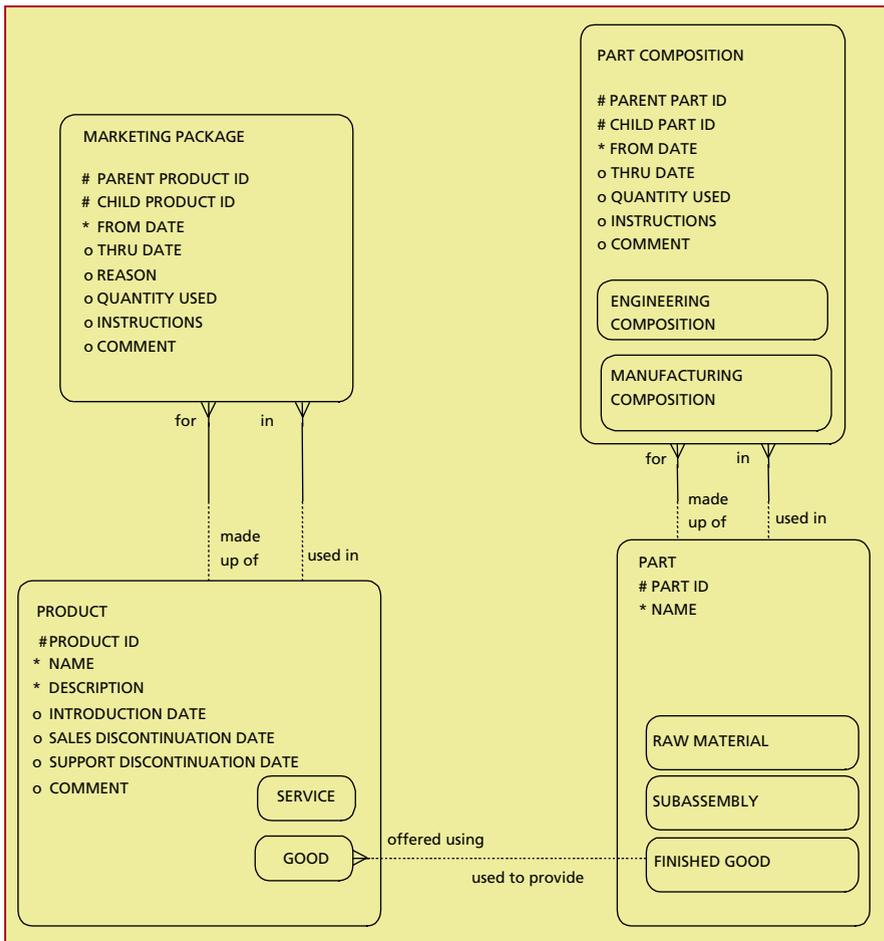
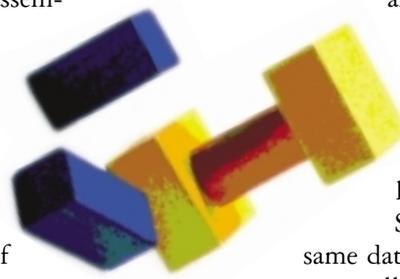


Figure 2: Bill of Materials

PART COMPOSITION's primary key is composed of two attributes: the part ID for the parent part and the part ID for the child part, thus maintaining each parent/child relationship. There are numerous attributes in the part bill of materials that describe how the child part fits into the parent part. For example, the quantity-used attribute shows how many of the child parts are used in the parent part (e.g., there could be a quantity used of "12" to maintain how many screws are in a casing).

An area that often causes confusion in manufacturing enterprises is that the term "bill of materials" can have several meanings for various parts of the organization. The engineering department may define the bill of materials as the precise components and specifications required to build any assembly. The manufacturing department may define the bill of materials as the components that should be used in the manufacturing process based upon the cost, quality and maintainability of these various components. While manufacturing needs to follow the engineering bill of materials and specifications, the manufacturing bill of materials may differ from the engineering bill of materials and it may reference specific parts that are more cost-efficient. The marketing department may have a version of a bill of materials that bundles additional products, T-shirts, gifts or other creative packaging ideas that may be included in the product offering. Finally, the service department may view the bill of materials as the parts that are actually installed in a particular deployment of a product as it changes at the customer's location.

Should one use the same data model construct to represent all these requirements? In most cases, each of these bill of materials structures represents different types of relationships. Figure 2 shows that there are two subtypes of PART COMPOSITION to represent each ENGINEERING COMPOSITION



companies frequently offer two products, a business line and a residential line, even though it is same actual part (or more specifically, the same finished good).

Some enterprises may define the relationship from product to part differently. Another way to define this relationship is that a PRODUCT may not always need to have one and only one associated PART. Instead, a PRODUCT could be made up of many PARTs, which could be considered to be the building blocks in creating products. Therefore, the model shown in Figure 1 could be altered by adding an associative entity between PRODUCT and PART, thus allowing parts to be combined in various ways to create product offerings. Neither one of these models is superior to the other and the decision of which model to use (including other variations) depends on how the enterprise defines products and parts in their environment.

### Bill of Materials and Marketing Packages

Manufacturers frequently use the term "bill of materials" to refer to how various parts are assembled into other parts and eventually into products. Thus the bill of materials structure for a computer may be broken down into a motherboard, a CPU, memory boards, hard-disk drives, a casing, a monitor and a keyboard; and each of these parts (or subassemblies) may further be broken-down into their constituent parts.

On the surface, the underlying data structure to accommodate a bill of materials seems pretty straightforward. Parts are made up of parts; therefore, a many-to-many recursive entity, PART COMPOSITION, shows which parts are made up of other parts and which parts are used within which parts. The

TION and each MANUFACTURING COMPOSITION. Each of these part compositions may maintain different combinations of parts: one breakdown for engineering purposes and another breakdown upon manufacturing needs. The marketing bill of materials is typically referred to as a “marketing package” or a “bundle” which is represented in Figure 2 as a recursive many-to-many relationship between PRODUCTS. Therefore, a PRODUCT may be made up of many other PRODUCTS, each of which may have an associated PART which may, in turn, be made up of other PARTS. The needs of the service department will be discussed within the Inventory and Deployment section.

Again, there could be minor variations to the data model based upon the enterprise’s needs. An enterprise may have a need to store parts used in a MARKETING PACKAGE that are never sold as individual products (e.g., the T-shirts mentioned previously). In that case, the modeler may want to add a relationship from MARKETING PACKAGE to PART. A MARKETING PACKAGE may also include other elements that are not a PART or a PRODUCT (e.g., a vacation); therefore, there may be additional relationships to MARKETING PACKAGE depending on the needs of the enterprise.

### Inventory and Deployment

Thus far, the models have represented types of things: types of marketing offerings and types of parts that make up the marketing offerings. Of course, there are actual physical items that may exist within the enterprise’s inventory or that may be deployed at customer locations.

Figure 3 shows a model portraying the relationships from product to parts to items to deployments. Each ITEM represents the physical occurrences of a PART at a FACILITY such as a WAREHOUSE or PLANT. ITEMS may be either maintained directly at the FACILITY or at a detailed inventory storage location within the facility. These detailed locations are referred to in the model as a CONTAINER, which could rep-

resent a specific “bin,” “barrel” or “shelf.” Each ITEM may be a SERIALIZED ITEM, which is an item that is tracked individually (e.g., a computer server) or a NON-SERIALIZED ITEM, which is an item that is maintained as a group (e.g., screws in a bin). Therefore, the quantity on hand is maintained for the SERIALIZED ITEMS, and a serial number is maintained for non-serialized items.

An ITEM may be owned by a PARTY such as a particular subsidiary of the modeled enterprise or it may be owned by another party, such as a cus-

move from one customer’s facility to another facility. While they won’t always know this information, it can be instrumental in tracking where products are being used and the associated demographics of their users. For instance, a computer manufacturer may be very interested in the facts that their “Extreme 5 PC” serial #3984d98e was purchased by ABC Company, Inc. on April 23, 2000, was then taken as a trade-in back into their inventory and then deployed again at XYZ Company on June 13, 2002.

The model shows a DEPLOY-

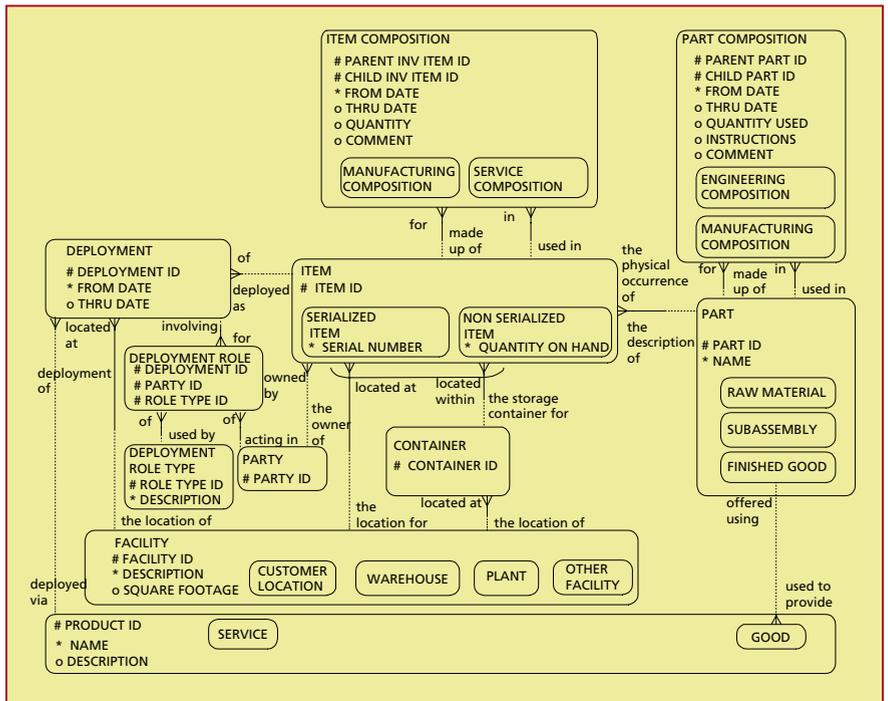


Figure 3: Inventory Items and Deployment

tom who owns the inventory on consignment. Each ITEM may be made up of other ITEMS as represented by the ITEM COMPOSITION, thus providing the capability to store the actual components of an inventory item and how they are configured. This structure is often used to represent the “bill of materials” structure needs of the service department which often wants to know the configuration of the physical items that exist, whether at a customer location or in inventory.

A DEPLOYMENT represents the setting up of a product at a customer site using the associated item(s). Some manufacturers would like to know information about the deployment of each of the physical items as they

MENT ROLE entity allowing any number of PARTYs to be associated to the DEPLOYMENT in any DEPLOYMENT ROLE TYPE. For instance, there could be many parties associated with the deployment in various roles such as the person who installed the deployment, the technical contact and the billing contact. Of course, depending on the needs of the enterprise, this relationship may be different. For example, if the only requirement were to maintain a single “user” for each DEPLOYED PRODUCT, the data model would show a one-to-many relationship from PARTY to DEPLOYMENT.

Is a DEPLOYMENT necessary or is it sufficient to record a many-to-many

relationship between each ITEM and the various FACILITYs where the ITEM is located? If that were the case, there would be an associative entity between ITEM and FACILITY, which could very well be called a DEPLOYMENT. The only difference with the model shown in Figure 3 is that the key to the deployment is the deployment ID because it may be possible that the inventory item is not known. For instance, a service representative could find that there was a deployed product but not know the exact item(s) associated with that deployment.

The models in this article provide template data structures for modeling a portion of “core” subject data areas within manufacturing, namely: products, parts, items and deployments of the

products at customer locations. It is important to realize that these universal data models are intended to offer insights and possibilities for modeling data in order to save time and provide data constructs that have been implemented successfully, and they are not the end-all answer to data modeling issues. These models are designed to be used as either the starting-point for modeling efforts or as a point-of-reference for developing customized data models that are tailored to an enterprise’s needs. Enterprises may define products, parts, items and deployments differently or use different terminology. However, many of the concepts and issues discussed in this article will be very applicable for most manufacturing enterprises. 

*A special thanks to Ed Landale who reviewed and offered additional insights and perspectives on the universal data models presented in this article.*

---

*Len Silverston is a data management consultant with more than 20 years of experience in helping enterprises integrate data. He is the author of the best-selling **The Data Model Resource Book** series (Wiley, 2001), which describes more than 230 integrated, reusable generic and industry-specific data models. Silverston has developed extensive software versions of these data models, some of which are now licensed worldwide by Microsoft and others that are available for licensing directly. Silverston's company, Universal Data Models, provides consulting, training and software to jump-start data modeling and data warehouse design efforts while increasing design quality and facilitating data integration. Silverston can be reached at [lsilverston@univdata.com](mailto:lsilverston@univdata.com).*