TITLE: "PROBLEMS IN DATA DEFINITION"

by Tim Bryce
Managing Director
M. Bryce & Associates (MBA)
P.O. Box 1637
Palm Harbor, FL 34682-1637
United States
Tel: 727/786-4567
E-Mail: timb001@attglobal.net
WWW: http://www.phmainstreet.com/mba/
Since 1971: *"Software for the finest computer - the Mind"*

Within the "PRIDE" methodologies there is a strong delineation between Data and Information, they are not synonymous. Data by itself is meaningless. It is simply the representation of a fact or event. However, it is the raw material which is used to produce information. Information, of the other hand, is the intelligence or insight required to fulfill the actions and/or decisions of the business. As such, both data and information have distinctly different attributes which must be defined carefully. This essay will focus on issues related to data definition, not information (which we will do a later data).

Data is a reusable resource that can serve many purposes. It is used to identify, describe and quantify "objects" of importance to the business, such as products, orders, customers, vendors, parts, billings, payments, shipments, etc. It is also used to perform calculations.

Historically, analysts and programmers have done a superficial job of defining data. More often than not, data definitions consist of nothing more than program labels and picture values as maintained in obscure program source code. If management cannot see what constitutes an element of data or how it is derived, its validity, currency and accuracy will be highly suspect. Ultimately, users will not be able to trust it. They will not be able to truly tell if they can base decisions and actions on information created by using data they do not understand or know its source. If it is not visible, it is not reusable. Because of this, there are actually two aspects to be considered when defining data: its logical meaning and its physical implementation.

In terms of logical definition, the data element should be identified in the same manner as a component part of a product (we recommend a unique control number, not just a name). This allows us to inventory the resource. Each data element should then have a unique Webster or Oxford-like dictionary description which accurately defines the meaning of the element. This should be written in plain business terms.

Each data element serves a unique purpose for the organization which can be categorized into three classes: indicative, descriptive, and quantitative. Indicative data is used to identify the various objects of the enterprise, e.g., "Customer Number," "Order Number," "Territory Code," "Employee Number," "Billing Number," etc. Descriptive data is used to describe the various objects, such as names, addresses, dates, text, etc. Quantitative data is associated with calculations, such as "Net Pay," "Gross Pay," "Quantity Ordered," "Unit Price," "Percent Quota Achieved," "Elapsed Time," etc.

Quite often descriptive data is confused with indicative data. This commonly occurs when numbers and codes are considered, such as "Zip/Postal Code." What distinguishes the two is the fact that indicative data is used to identify an object, in part or in full. If it is not used for this purpose, then it is descriptive. To many companies, a Zip/Postal Territory is of little concern or importance; it is not an object they must deal with in the operation of their organization. "Zip/Postal Code" in this situation is descriptive. However, to a post office or shipping company, a Zip/Postal Territory may be a very important object. In this case, "Zip/Postal Code" is indicative. The definition of data, therefore, relates to the objects affected by the enterprise.

The final logical specification of a data element is its "source" which defines where it comes from. From this perspective, there are two types of data elements: primary and generated. Primary data originates from a user area, outside of a system. For example, "Hours Worked" may originate from the individual employee, and "Pay Rate" from personnel. Generated data is calculated internally within a system and is based on other data elements. For example, "Gross Pay" may be computed by "Hours Worked" multiplied by "Pay Rate." The calculation of generated values can be extensive, requiring several data elements and extensive mathematics. Regardless, in order for generated data elements such as "Gross Pay" to be valid, we must be able to trace it back to the sources of ALL of the primary items.

Another form of generated data is a "group" data element that is based on the assignment of data elements according to a defined algorithm. For example, the typical "Bank Card Number" is usually assembled from four data elements: "Financial Institution Number," "Bank Region Number," "Branch Number," and "Card Holder Number." There are several other familiar examples of "group" type data elements, such as checking account numbers, utility account numbers, manufacturing control codes, etc.

(continued on page 2)

(continued from page 1)

The physical definition of data includes such things as:

* Length - defines the maximum number of characters which may be assigned to a data element.

* Class - defines the type of characters used to express a data element, e.g, alphabetic, numeric, alphanumeric, signed numeric, etc.

* Justification - defines the alignment of data within a field when the number of characters is less than the length of the receiving field, e.g., left, right, around the decimal point.

* Fill Character - defines the character to be used to complete a field when the data item to be placed in the field is shorter than the maximum length, e.g., blank, zero, X, etc.

* Void Character - defines the character to be used to be entered when a data item's value is unknown or non-existent, e.g., blank, zero, X, etc.

* Unit of Measure - defines the representation of numeric data, e.g., area, volume, weight, length, time, energy rate, money, etc.

* Precision - defines for numeric data the number of significant digits in a number.

* Scale - defines for numeric data the placement of the decimal point.

* Base - defines for numeric data the radix to be used for representing the number in programming, e.g., decimal, binary, octal, hexadecimal, etc.

* Mode - defines the format (and type) of a data element for programming, i.e., fixed point integer, floating point, double precision floating point, complex, binary, packed decimal, polar coordinates, etc.

* Picture - defines how the data element is expressed for programming. It is typically based on length, class, precision and scale.

* Program Label - defines the proper name of the data element as it will be referred to in a programming language, such as C, Java, COBOL, FORTRAN, PL/1, Assembler, Basic, etc.

* Validation Rules - defines specific values which the data element may assume. For example, Yes/No, specific codes or numbers to be used, editing rules, etc.

This brings up an important point about data definition: a data element can have only one logical definition but can have one or more physical implementations. If a data element is an expression of a single fact or an event, it is important that it be explicitly defined so it will not be confused with another. If there is a genuine difference in interpretation of the meaning of data between users, more than one data element is involved.

Although standardization of data's physical characteristics is an objective, there can be multiple physical representations of data. For example, there can be several legitimate ways to store and display "Ship Date":

20041211
December 11, 2004
12/11/2004
2004/12/11
11-DEC-04

In this example, the data element has a singular logical definition, "The date when a product was shipped to a customer." All that differs is how they are physically represented. What this points out is the physical characteristics of data may vary from one application to another.

Obviously there are significant differences between how a data element is logically and physically defined. The software engineer and DBA deals with the physical definition, the systems engineer, data engineer, and end user deals with the logical. Data Resource Management must govern the assignment of both.

Data Taxonomy

The management of any resource requires the development of a classification system. Financial resources are typically arranged according to a chart of accounts (based on debits and credits); material and human resources are categorized by type. In science, everything from chemical elements to the animal kingdom are organized according to a class structure. There obviously is a purpose to uniquely identify common elements; to provide for the ability to distinguish one from another, and eliminate redundancy. In all instances, classification is based on the inherent characteristics of the resource.

A Data Taxonomy is a hierarchical class structure (decision tree) that separates data into specific classes of data

(continued on page 3)

(continued from page 2)

based on common characteristics. The taxonomy represents a convenient way to classify data to prove that it is unique and without redundancy. This includes both primary and generated data elements.

The lowest level in the classification hierarchy represents what is commonly referred to as the "domain" of a collection of data elements, one or more, with common characteristics. For example, "text" related data elements would be in one domain, "weights" in another, "percentages" in another, "monetary values" in another, etc.

The domain also defines the standard physical characteristics and values the data may assume. For example, we could establish that all "location" values are alphanumeric, left justified, with blank fill and void characters. In other words, data elements such as "Address," "City," and "State" should assume these physical characteristics for consistency.

If a data element does not have the standard logical and physical characteristics, it must belong to another "domain." In the situation where a data element may have only one logical definition, but multiple physical definitions, its primary physical definition must first conform to the Domain standards before it can be deviated from in an application record. In other words, the primary physical representation of "Unit Cost" is expressed as an eight character numeric to conform to the "Currency" domain. However, in one application, a user desires the data element be expressed as a ten character numeric. It is the same logical data element with just another form of physical expression.

With a classification system in place, data elements can then be uniquely and consistently defined. When this is done, we then have a basis for checking data redundancy. Also, when a data element has been properly specified in this manner, it becomes rather simple to locate it again for use in other applications.

Classifying data helps to fulfill one of the the major objectives of Data Resource Management: to eliminate redundancy and promote the re-use of resources in applications.

Problems

Defining a unique data element can be tricky at times. The analyst must always be cognizant of the "object" being defined. Let's look at some common problem areas:

"Check Number" - valid or invalid?

In most cases, it is invalid for it is bound too tightly to the physical output and not the proper object, "Account" (consider this, is a "Check" an object we really care about or is the "Account" truly what we are after?) Perhaps it would be better named "Transaction Number" as it represents a debit to an account. Let us not forget that a Check is but one physical way to transfer funds from an account. Now, with electronic banking, a physical "Check" might not even exist. A "Check Number" might be legitimate as an "Alternate ID" for the transaction, but still, why bother with "Check Number"? Why not, "Alternate ID" instead?

"Mother's Maiden Name" - valid or invalid?

Does this truly represent an important fact or event to be recorded or does it express how something like a "Password" is assigned? Probably the latter and, as such, is probably the true data element.

"Quantity" - valid or invalid"

This may be fine for a "Domain" name, but it is much too vague for a data definition. Even if we were to bind it to an indicative data element, such as "Shipment Number" or "Order Number", it still lacks any true meaning; e.g., "Shipment Quantity" - does it represent the number of shipments, containers within a shipment, weight, etc.?

"Social Security Number" (used as indicative data) - valid or invalid?

This is frequently misused as "indicative" data. For most companies and organizations (aside from the Social Security Administration) it is "descriptive" data. However, many companies have an "Employee Number" which is based on "Social Security Number." In this situation, "Employee Number" is the valid indicative data element (not "Social Security Number"), The same argument can be applied for "Zip/Postal Code" as mentioned earlier.

IRM Repository

Both the logical and physical definitions of data should be recorded in an IRM Repository and not relegated to program source code. This allows the definition to be uniquely identified and re-used. In addition to maintaining the data's characteristics, an IRM Repository should be able to link it to other information resources (such as other data elements, records, files, information

(continued on page 4)

(continued from page 3)

requirements, etc.) so we can track "where used" for maintenance purposes. It must also be able to store the logical definition of a data element, its standard physical definition, and any physical deviations.

An IRM Repository can be implemented manually using forms, but automation can greatly expedite its implementation and promote the sharing and re-usability of resources.

For more information on this subject, see:

"Establishing an IRM Repository"
http://www.phmainstreet.com/mba/pride/spir.htm

"PRIDE" Forms
http://www.phmainstreet.com/mba/pride/forms.htm

Specifically, see the "Data Description Worksheet" at:
http://www.phmainstreet.com/mba/pride/iw017.jpg

"Data as a Resource" (from "PRIDE"-DBEM)
http://www.phmainstreet.com/mba/pride/dbmeth.htm#resource

END

"PRIDE" Special Subject Bulletins can be found at the "PRIDE Methodologies for IRM Discussion Group" at:

http://groups.yahoo.com/group/mbapride/