By DAVID McGOVERAN

Valuing Data, Part 6

The valuation of data is a non-trivial task, but worth the exercise. Once the math is worked out, it can be implemented in an analytical tool and modified as necessary. Our road to developing a data valuation methodology and cost/benefit model began six issues ago. We've identified the various uses of data that must be considered in establishing a utilitarian model of data value, and we've discussed the various direct and indirect costs associated with data acquisition, retention, use, and even divestiture. Having considered the effects of degradation, depreciation, and appreciation, we are now ready to combine these components to develop a cost/benefit model of data valuation.

Our approach is to compute cumulative and expected costs, and present and future value. The value is its cumulative utility over remaining useful life. Cost and value often differ by potential user group, usage type and data type, so we may have to compute and then sum these contributions. Our computation defines the point-in-time relationships. To build a dynamic model, factor in all the time dependencies by determining (1) which factors can influence change and (2) which constants in the model are merely coincidental. These are converted to time-dependent variables. The steps in the cost/benefit data valuation model are:

1. List the data types of interest, from simple data elements to complex business objects. Treat data in terms of abstract data types or classes: potentially complexly related component data elements, with relationships themselves captured as data.
2. Compute the instantaneous minimum value of each, which is the accumulated cost of data (both direct and indirect — see Parts 2 and 3) minus depreciation (including estimated loss of demand; i.e., future utility) and degradation. For each data class and utility state, compute the cumulative contribution to costs (past and present). Acquisition state and inventory state costs can be actual (summed from historical information), forecast from trends, or simply estimated.
3. For each data class, compute the contribution to potential utility for present and future operational, historical, forecasting, and divestiture states as discussed in Parts 3 and 4, as an allocation of the value of goods and services produced, cost savings, loss avoidance, etc., through data usage. Sum these contributions to find combined potential utility. Factor in data quality degradation as a fractional multiplier of the combined potential utility. Include expected contributions to utility from the operational state, and multiply by a reusage factor: Instances of a class may be either "consumed" or "reused." For historical state data, assume maximum value based on the greater of replacement costs, worst case (i.e., greatest) potential cost avoidance, maximum resell value, etc., then "age" this value by an exponential multiplier ("decay" with time, starting from the date the data last used). Make the simplifying assumption that data contributes historical state utility only once, no matter how many times it is in that state during its life. If data is sold, estimate the number of times to be sold and its value at each sale, and sum this divestiture contribution.
4. Finally, the total cumulative value for any class is the greater of instantaneous minimum value and remaining potential utility, multiplied by an inflation factor. Given this value, and the associated cumulative and anticipated costs, you can perform a standard cost/benefit analysis for any class of data.

More detail can be added, multipliers changed, and the simple allocation rules given above embellished — following the principles outlined in this series — as desired for more sophisticated models. We've focused on analyzing "class value" (value of an abstract data type) rather than instance value. You'll need instance allocation rules for value and cost. Rules may allocate disproportionately, based on data value ranges, relationships, most frequently used, etc. Instance value is positively and perhaps fractionally proportional to class value, based on an equi-partitioning allocation rule. The contribution value of each data element is its value plus the sum of its contributions to all other abstract data types.

In a future column I'll address some issues not addressed in this series. For now, the problem of evaluating the total expected value of a data collection remains, and differs from determining the value of some specified part of it and whether that's worth keeping. To obtain the expected value of an enterprise's data, partition the data and relationships into mutually independent, non-redundant classes, then sum their contribution values. Don't simply add contributions from all classes — interdependencies will ruin the integrity, invalidating the enterprise-level answer. BIJ

David McGoveran is president of Alternative Technologies, Inc. He has more than 20 years of experience with mission-critical applications and has authored numerous technical articles on application integration. e-Mail: mcgoveran@bijonline.com; Website: www.alternativetech.com.